Optimization of stochastic combinatorial problem: scheduling maintenance and refueling outages of nuclear power plants

C. Pira, R. Wolfler Calvo, V. Jost, A. Rozenknop

AOC, LIPN, Université Paris 13, Villetaneuse, France

LIX, École Polytechnique, Palaiseau, France

October 04, 2013, Palaiseau

Context (1) : the original problem (Roadef 2010 challenge)

- A purely combinatorial part (scheduling) :
  - A set of nuclear power plants $i \in I$.
  - Outages to be scheduled.
  - Resource contraints.
- An optimal control part (supply and demand) :
  - For each $i \in I$, evolution of stock $x(i,t)$ is determined by production $p(i,t)$ and refueling $r(i,k)$.
  - Coupling constraints : additional thermal power plants $j \in J$ to balance energy supply and demand.

$$\sum_{i \in I} p(i,t) + \sum_{j \in J} p(j,t) = Dem_t \qquad \forall t \in [0, T-1]$$

Context (1) : the original problem (Roadef 2010 challenge)

- A purely combinatorial part (scheduling) :
  - A set of nuclear power plants $i \in I$.
  - Outages to be scheduled.
  - Resource contraints.
- An optimal control part (supply and demand) :
  - For each $i \in I$, evolution of stock $x(i,t)$ is determined by production $p(i,t)$ and refueling $r(i,k)$.
  - Coupling constraints : additional thermal power plants $j \in J$ to balance energy supply and demand.

$$\sum_{i \in I} p(i,t) + \sum_{j \in J} p(j,t) = Dem_t \qquad \forall t \in [0, T-1]$$

Context (1) : the original problem (Roadef 2010 challenge)

- A purely combinatorial part (scheduling) :
  - A set of nuclear power plants $i \in I$.
  - Outages to be scheduled.
  - Resource contraints.
- An optimal control part (supply and demand) :
  - For each $i \in I$, evolution of stock $x(i,t)$ is determined by production $p(i,t)$ and refueling $r(i,k)$.
  - Coupling constraints : additional thermal power plants $j \in J$ to balance energy supply and demand.

$$\sum_{i \in I} p(i,t) + \sum_{j \in J} p(j,t) = Dem_t \qquad \forall t \in [0, T-1]$$

Context (1) : the original problem (Roadef 2010 challenge)

- A purely combinatorial part (scheduling) :
  - A set of nuclear power plants $i \in I$.
  - Outages to be scheduled.
  - Resource contraints.
- An optimal control part (supply and demand) :
  - For each $i \in I$, evolution of stock $x(i,t)$ is determined by production $p(i,t)$ and refueling $r(i,k)$.
  - Coupling constraints : additional thermal power plants $j \in J$ to balance energy supply and demand.

$$\sum_{i \in I} p(i,t) + \sum_{j \in J} p(j,t) = Dem_t \qquad \forall t \in [0, T-1]$$

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

  - Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

- Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

- Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

- Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

  - Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

  - Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

  - Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

  - Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

  - Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

  - Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (2) : stability of the schedule

- The problem is reoptimized each month
  - Some variables can be adapted easily (production, refuel), other must be decided in advance (dates).
  - Stability : outages don't change two much over time.

$\Rightarrow$ We initially tought to study multistage robust optimization :

  - Robustness : a schedule can withstand changes in the input

- But stability of the whole reoptimization process $\neq$ robustness of the "one-shot" problem.
  - Instances involve scenarios representing possible evolutions.
  - A robust solution is feasible for all these possible variations.
  - Problem : scenarios are modified each month (some have proven to be false).
  - Robustness was garanteed only for those evolutions.
  - Worst : we calibrate the solution w.r. worst cases (potentially unlikely to occur).

$\Rightarrow$ We focus on stochastic optimization instead

Context (3) : lagrangian relaxation and column generation

- The original model (Roadef 2010) can naturally be handled by a column generation approach :
  - Relaxation of the supply-demand balance constraints to obtain independant subproblems.
- Scenarios represent aleas which affect all the plants together.
- We focus on new aleas specific to each plant, hence handled mostly in the subproblems :
  - Incidents temporarily stop production : binary random variables $\omega_{i,t}$.
  - Outages durations : positive random variables $\psi_{i,k}$.

$\Rightarrow$ Possible adaptations of CG to deal with stochastic subproblems.

Context (3) : lagrangian relaxation and column generation

- The original model (Roadef 2010) can naturally be handled by a column generation approach :
  - Relaxation of the supply-demand balance constraints to obtain independant subproblems.
- Scenarios represent aleas which affect all the plants together.
- We focus on new aleas specific to each plant, hence handled mostly in the subproblems :
  - Incidents temporarily stop production : binary random variables $\omega_{i,t}$.
  - Outages durations : positive random variables $\psi_{i,k}$.

$\Rightarrow$ Possible adaptations of CG to deal with stochastic subproblems.

Context (3) : lagrangian relaxation and column generation

- The original model (Roadef 2010) can naturally be handled by a column generation approach :
  - Relaxation of the supply-demand balance constraints to obtain independant subproblems.
- Scenarios represent aleas which affect all the plants together.
- We focus on new aleas specific to each plant, hence handled mostly in the subproblems :
  - Incidents temporarily stop production : binary random variables $\omega_{i,t}$.
  - Outages durations : positive random variables $\psi_{i,k}$.

$\Rightarrow$ Possible adaptations of CG to deal with stochastic subproblems.

Context (3) : lagrangian relaxation and column generation

- The original model (Roadef 2010) can naturally be handled by a column generation approach :
  - Relaxation of the supply-demand balance constraints to obtain independant subproblems.
- Scenarios represent aleas which affect all the plants together.
- We focus on new aleas specific to each plant, hence handled mostly in the subproblems :
  - Incidents temporarily stop production : binary random variables $\omega_{i,t}$.
  - Outages durations : positive random variables $\psi_{i,k}$.

$\Rightarrow$ Possible adaptations of CG to deal with stochastic subproblems.

Context (3) : lagrangian relaxation and column generation

- The original model (Roadef 2010) can naturally be handled by a column generation approach :
  - Relaxation of the supply-demand balance constraints to obtain independant subproblems.
- Scenarios represent aleas which affect all the plants together.
- We focus on new aleas specific to each plant, hence handled mostly in the subproblems :
  - Incidents temporarily stop production : binary random variables $\omega_{i,t}$.
  - Outages durations : positive random variables $\psi_{i,k}$.

$\Rightarrow$ Possible adaptations of CG to deal with stochastic subproblems.

Context (3) : lagrangian relaxation and column generation

- The original model (Roadef 2010) can naturally be handled by a column generation approach :
  - Relaxation of the supply-demand balance constraints to obtain independant subproblems.
- Scenarios represent aleas which affect all the plants together.
- We focus on new aleas specific to each plant, hence handled mostly in the subproblems :
  - Incidents temporarily stop production : binary random variables $\omega_{i,t}$.
  - Outages durations : positive random variables $\psi_{i,k}$.

$\Rightarrow$ Possible adaptations of CG to deal with stochastic subproblems.

Plan

Relaxation of coupling constraints (1)

- Simplified model : no combinatorial constraints

$$\min \quad \sum_i \left(\sum_k C_{i,k} r(i,k) - C_T x(i,T)\right) + \sum_{j,t} c_{j,t} p(j,t)$$

$$s.t. \quad \boxed{\sum_i p(i,t) + \sum_j p(j,t) = Dem_t}$$

$$x(i,t+1) - x(i,t) = -p(i,t) D^t, \qquad \forall i,k,t \in ec(i,k)$$

$$x(i,t+1) = \frac{Q_{i,k}-1}{Q_{i,k}} (x(i,t) - BO_{i,k-1}) + r(i,k) + BO_{i,k},$$

$$\forall i,k,t = N^t \times ea(i,k)$$

$\ldots$

Relaxation of coupling constraints (1)

- Simplified model : no combinatorial constraints

$$\min \quad \sum_i \left(\sum_k C_{i,k} r(i,k) - C_T x(i,T)\right) + \sum_{j,t} c_{j,t} p(j,t)$$

$$s.t. \quad \boxed{\sum_i p(i,t) + \sum_j p(j,t) = Dem_t}$$

$$x(i,t+1) - x(i,t) = -p(i,t) D^t, \qquad \forall i,k,t \in ec(i,k)$$

$$x(i,t+1) = \frac{Q_{i,k}-1}{Q_{i,k}} (x(i,t) - BO_{i,k-1}) + r(i,k) + BO_{i,k},$$

$$\forall i,k,t = N^t \times ea(i,k)$$

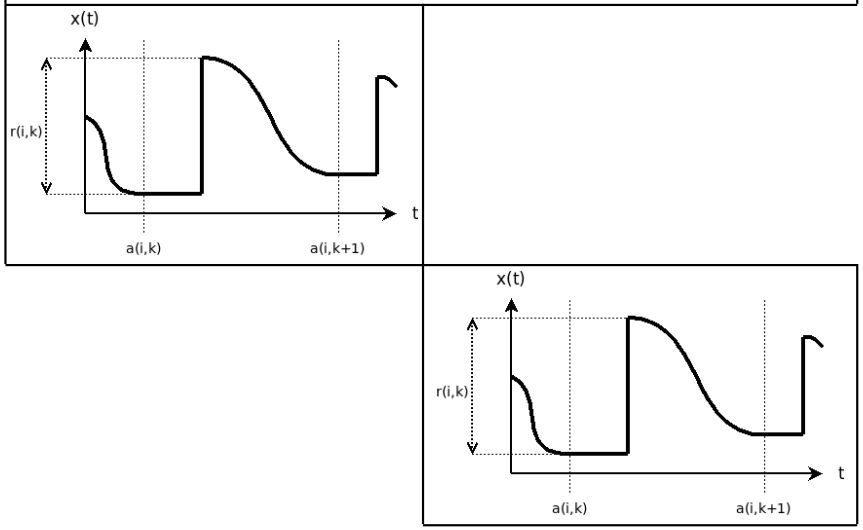
$\ldots$

Relaxation of coupling constraints (1)

- Simplified model : no combinatorial constraints

$$\min \quad \sum_i \left(\sum_k C_{i,k} r(i,k) - C_T x(i,T)\right) + \sum_{j,t} c_{j,t} p(j,t)$$

$$s.t. \quad \sum_i p(i,t) + \sum_j p(j,t) = Dem_t$$

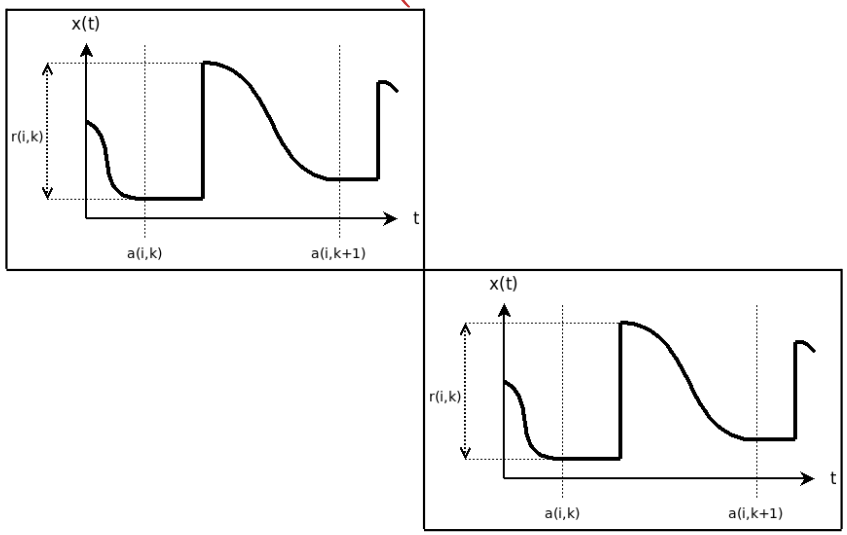


Relaxation of coupling constraints (2)

- Obtaining independant subproblems by relaxing the coupling constraints : price μ_t on each demand constraint.

$$\max_{(\mu_t)} \quad \min \quad \sum_i \left(\sum_k C_{i,k} r(i,k) - C_T x(i,T)\right) + \sum_{j,t} c_{j,t} p(j,t)$$
$$- \sum_t \mu_t \left(\sum_i p(i,t) + \sum_j p(j,t) - Dem_t\right)$$

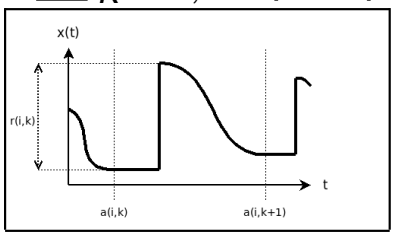
s.t.



Columns generation (1)

- Given prices (μ_t), subproblem i generates production planning :

$$\begin{array}{ll} \min & \sum_k C_{i,k} r(i,k) - C_T x(i,T) - \sum_t \mu_t p(i,t) \\ \text{s.t.} & \end{array}$$



- Plans with negative reduced costs are added in a master problem (new columns)

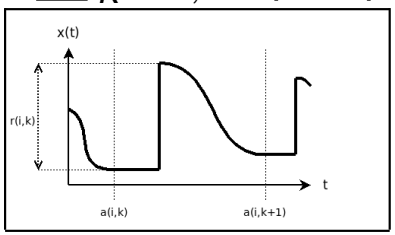
	$\lambda^1_{i_1}$	$\lambda^2_{i_1}$	$\lambda^1_{i_2}$	$\lambda^2_{i_2}$	$p(j,0)$	$\cdots$	$p(j,T-1)$	
μ_1	$p^1_{i_1}(0)$	$p^2_{i_1}(0)$	$p^1_{i_2}(0)$	$p^2_{i_2}(0)$	1	$\cdots$	0	Dem_0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
μ_{T-1}	$p^1_{i_1}(T-1)$	$p^2_{i_1}(T-1)$	$p^1_{i_2}(T-1)$	$p^2_{i_2}(T-1)$	0	$\cdots$	1	Dem_{T-1}
	$c^1_{i_1}$	$c^2_{i_1}$	$c^1_{i_2}$	$c^2_{i_2}$	$c_{j,0}$	$\cdots$	$c_{j,T-1}$	

- Master solves relaxation and proposes new prices.

Columns generation (1)

- Given prices (μ_t), subproblem i generates production planning :

$$\begin{array}{ll} \min & \sum_k C_{i,k} r(i,k) - C_T x(i,T) - \sum_t \mu_t p(i,t) \\ \text{s.t.} & \end{array}$$



- Plans with negative reduced costs are added in a master problem (new columns)

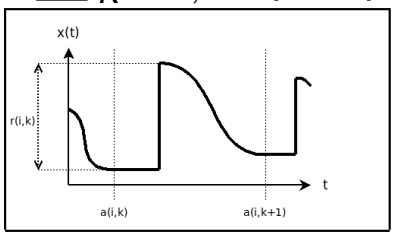
	$\lambda^1_{i_1}$	$\lambda^2_{i_1}$	$\lambda^1_{i_2}$	$\lambda^2_{i_2}$	$p(j,0)$	$\cdots$	$p(j,T-1)$	
μ_1	$p^1_{i_1}(0)$	$p^2_{i_1}(0)$	$p^1_{i_2}(0)$	$p^2_{i_2}(0)$	1	$\cdots$	0	Dem_0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
μ_{T-1}	$p^1_{i_1}(T-1)$	$p^2_{i_1}(T-1)$	$p^1_{i_2}(T-1)$	$p^2_{i_2}(T-1)$	0	$\cdots$	1	Dem_{T-1}
	$c^1_{i_1}$	$c^2_{i_1}$	$c^1_{i_2}$	$c^2_{i_2}$	$c_{j,0}$	$\cdots$	$c_{j,T-1}$	

- Master solves relaxation and proposes new prices.

Columns generation (1)

- Given prices (μ_t), subproblem i generates production planning :

$$\begin{array}{ll} \min & \sum_k C_{i,k} r(i,k) - C_T x(i,T) - \sum_t \mu_t p(i,t) \\ \text{s.t.} & \end{array}$$



- Plans with negative reduced costs are added in a master problem (new columns)

	$\lambda^1_{i_1}$	$\lambda^2_{i_1}$	$\lambda^1_{i_2}$	$\lambda^2_{i_2}$	$p(j,0)$	$\cdots$	$p(j,T-1)$	
μ_1	$p^1_{i_1}(0)$	$p^2_{i_1}(0)$	$p^1_{i_2}(0)$	$p^2_{i_2}(0)$	1	$\cdots$	0	Dem_0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
μ_{T-1}	$p^1_{i_1}(T-1)$	$p^2_{i_1}(T-1)$	$p^1_{i_2}(T-1)$	$p^2_{i_2}(T-1)$	0	$\cdots$	1	Dem_{T-1}
	$c^1_{i_1}$	$c^2_{i_1}$	$c^1_{i_2}$	$c^2_{i_2}$	$c_{j,0}$	$\cdots$	$c_{j,T-1}$	

- Master solves relaxation and proposes new prices.

Columns generation (2)

- Occasionally, the master chooses a combination of columns satisfying demand constraints...

$\lambda_{i_1}^1$	$\lambda_{i_1}^2$	$\lambda_{i_2}^1$	$\lambda_{i_2}^2$	$p(j,0)$	$\cdots$	$p(j,T-1)$	
$p_{i_1}^1(0)$	$p_{i_1}^2(0)$	$p_{i_2}^1(0)$	$p_{i_2}^2(0)$	1	$\cdots$	0	Dem_0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$p_{i_1}^1(T-1)$	$p_{i_1}^2(T-1)$	$p_{i_2}^1(T-1)$	$p_{i_2}^2(T-1)$	0	$\cdots$	1	Dem_{T-1}
$c_{i_1}^1$	$c_{i_1}^2$	$c_{i_2}^1$	$c_{i_2}^2$	$c_{j,0}$	$\cdots$	$c_{j,T-1}$	

- ... with exactly one planning for each plant...

$$\sum_n \lambda_i^n = 1 \quad \forall i$$
$$\lambda_i^n \in \{0,1\} \quad \forall i,n$$

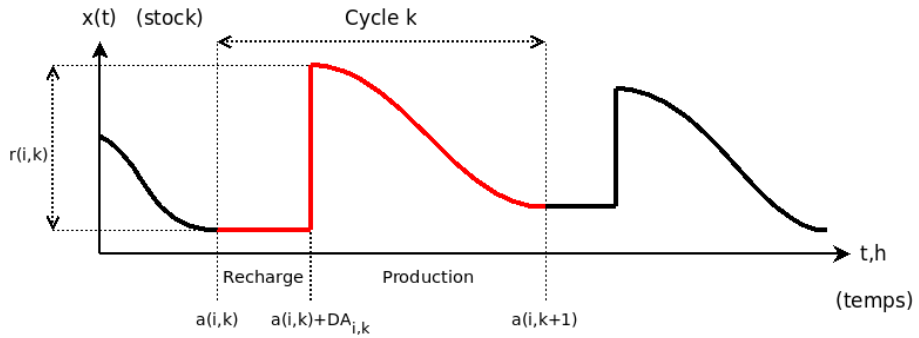
- ... and respecting additional combinatorial constraints.

Subproblem : command of one nuclear plant

- A discret time optimal control problem :
  - Set of equations describing the evolution of stock :

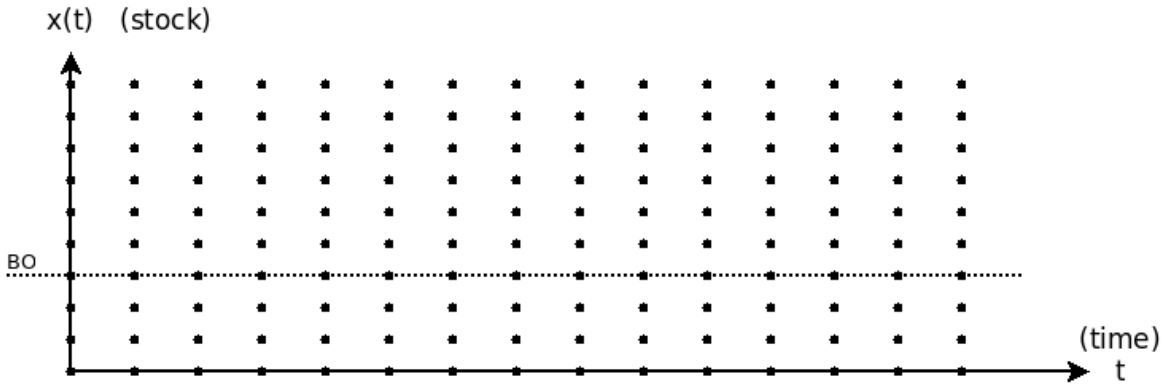
$$x(i,t+1) - x(i,t) = -p(i,t)D^t, \qquad \forall i,k,t \in ec(i,k)$$
$$x(i,t+1) = \frac{Q_{i,k}-1}{Q_{i,k}}(x(i,t) - BO_{i,k-1}) + r(i,k) + BO_{i,k}, \qquad \forall i,k,t = N^t \times ea(i,k)$$
$$RMIN_{i,k} \leq r(i,k) \leq RMAX_{i,k} \qquad \forall i,k$$
$$PMIN_{i,t} \leq p(i,t) \leq PMAX_{i,t} \qquad \forall i,t$$
$$\ldots$$

  - To determine policy (production and refueling) minimizing the cost along the resulting path.



Subproblem : command of one nuclear plant

- Discretisation of stock level and Dynamic Programming

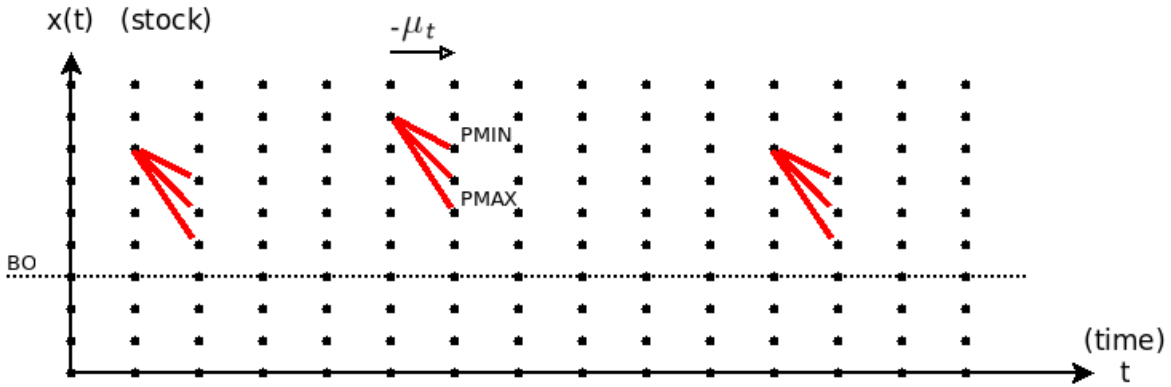


  - Cost of a production transition : $\mu_t \times (x(i, t+1) - x(i, t))$.
  - Under a threshold BO_k, transitions are imposed.
  - On some state, we can perform refuel of cycle k.
  - Cost of a refuel transition : $C_k \times (x(i, t + DA_k) - x(i, t))$

- Forward or backward induction : in $O(E) = O(L^2 \times T)$ where L is the number of levels.

Subproblem : command of one nuclear plant

- Discretisation of stock level and Dynamic Programming

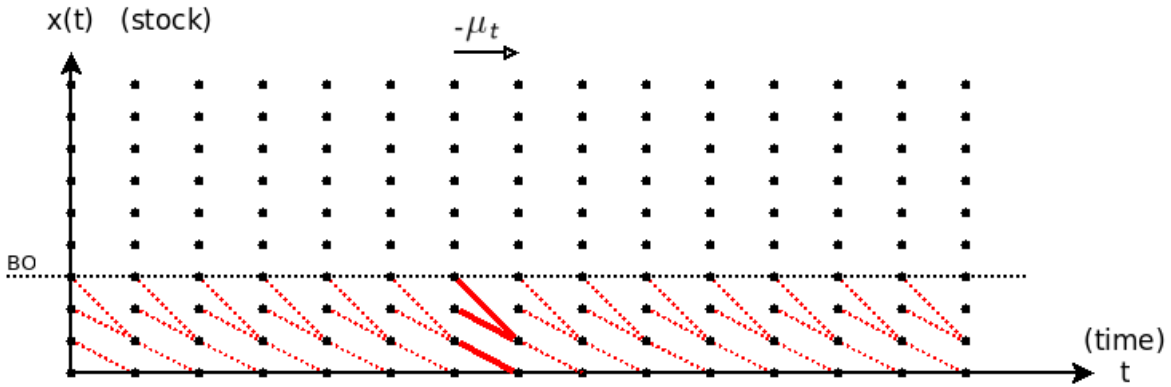


  - Cost of a production transition : $\mu_t \times (x(i, t+1) - x(i, t))$.
  - Under a threshold BO_k, transitions are imposed.
  - On some state, we can perform refuel of cycle k.
  - Cost of a refuel transition : $C_k \times (x(i, t + DA_k) - x(i, t))$

- Forward or backward induction : in $O(E) = O(L^2 \times T)$ where L is the number of levels.

Subproblem : command of one nuclear plant

- Discretisation of stock level and Dynamic Programming

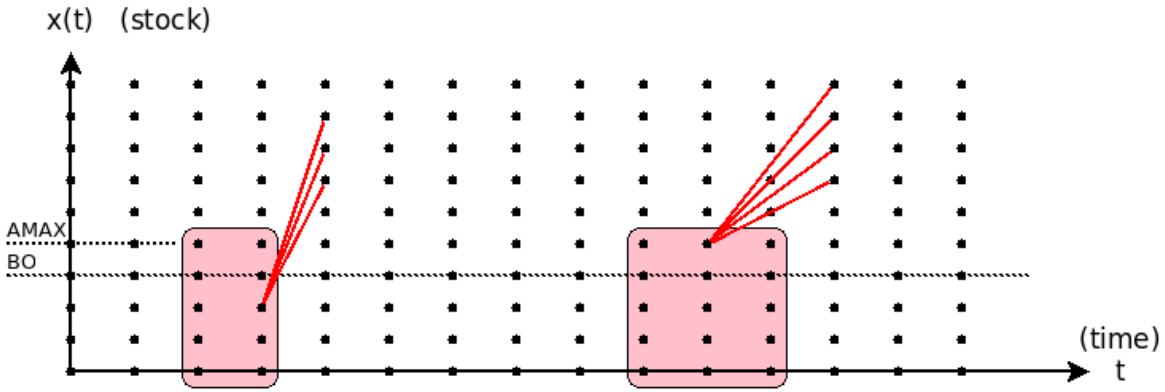


  - Cost of a production transition : $\mu_t \times (x(i, t+1) - x(i, t))$.
  - Under a threshold BO_k, transitions are imposed.
  - On some state, we can perform refuel of cycle k.
  - Cost of a refuel transition : $C_k \times (x(i, t + DA_k) - x(i, t))$

- Forward or backward induction : in $O(E) = O(L^2 \times T)$ where L is the number of levels.

Subproblem : command of one nuclear plant

- Discretisation of stock level and Dynamic Programming

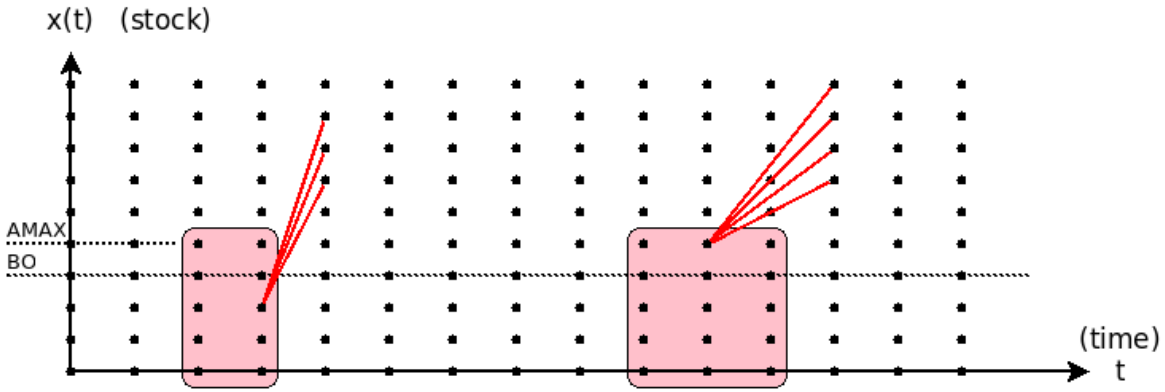


  - Cost of a production transition : $\mu_t \times (x(i, t+1) - x(i, t))$.
  - Under a threshold BO_k, transitions are imposed.
  - On some state, we can perform refuel of cycle k.
  - Cost of a refuel transition : $C_k \times (x(i, t + DA_k) - x(i, t))$

- Forward or backward induction : in $O(E) = O(L^2 \times T)$ where L is the number of levels.

Subproblem : command of one nuclear plant

- Discretisation of stock level and Dynamic Programming

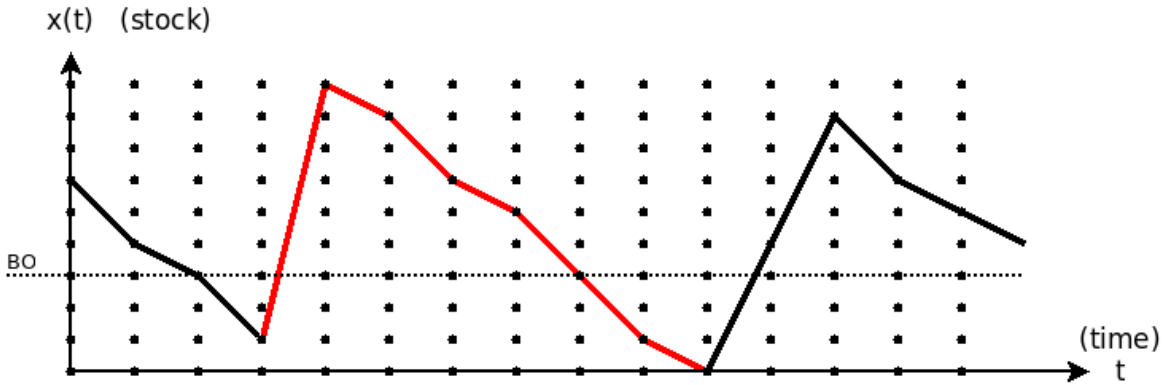


  - Cost of a production transition : $\mu_t \times (x(i, t+1) - x(i, t))$.
  - Under a threshold BO_k, transitions are imposed.
  - On some state, we can perform refuel of cycle k.
  - Cost of a refuel transition : $C_k \times (x(i, t + DA_k) - x(i, t))$

- Forward or backward induction : in $O(E) = O(L^2 \times T)$ where L is the number of levels.

Subproblem : command of one nuclear plant

- Discretisation of stock level and Dynamic Programming



  - Cost of a production transition : $\mu_t \times (x(i, t+1) - x(i, t))$.
  - Under a threshold BO_k, transitions are imposed.
  - On some state, we can perform refuel of cycle k.
  - Cost of a refuel transition : $C_k \times (x(i, t + DA_k) - x(i, t))$

- Forward or backward induction : in $O(E) = O(L^2 \times T)$ where L is the number of levels.

Feasibility of the path : maximum modulation (1)

- Problem : there is a structural path-constraint.
  - Modulation consists in not producing at PMAX.
  - Modulation is cumulated over a cycle and cannot exceed a certain amount :

$$\sum_{t \in ec(i,k) \wedge x_{i,t} \geq BO_{i,k}} (PMAX_i^t - p(i,t)) \leq MMAX_{i,k}$$

$\Rightarrow$ Searching in a reduced graph (Rozenknop, Wolfler *et al*).

$\vee$ Algorithms for shortest path with resources

Feasibility of the path : maximum modulation (1)

- Problem : there is a structural path-constraint.
  - Modulation consists in not producing at PMAX.
  - Modulation is cumulated over a cycle and cannot exceed a certain amount :

$$\sum_{t \in ec(i,k) \wedge x_{i,t} \geq BO_{i,k}} (PMAX_i^t - p(i,t)) \leq MMAX_{i,k}$$

$\Rightarrow$ Searching in a reduced graph (Rozenknop, Wolfler *et al*).

$\vee$ Algorithms for shortest path with resources

Feasibility of the path : maximum modulation (1)

- Problem : there is a structural path-constraint.
  - Modulation consists in not producing at PMAX.
  - Modulation is cumulated over a cycle and cannot exceed a certain amount :

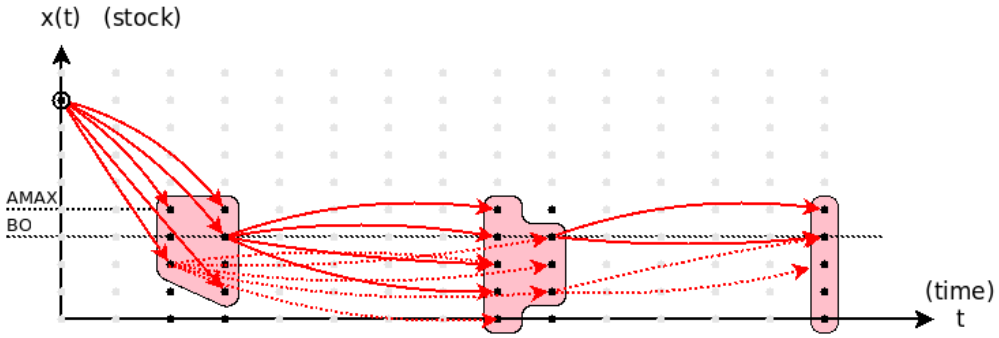
$$\sum_{t \in ec(i,k) \wedge x_{i,t} \geq BO_{i,k}} (PMAX_i^t - p(i,t)) \leq MMAX_{i,k}$$

$\Rightarrow$ Searching in a reduced graph (Rozenknop, Wolfler *et al*).

$\vee$ Algorithms for shortest path with resources

Feasibility of the path : maximum modulation (2)

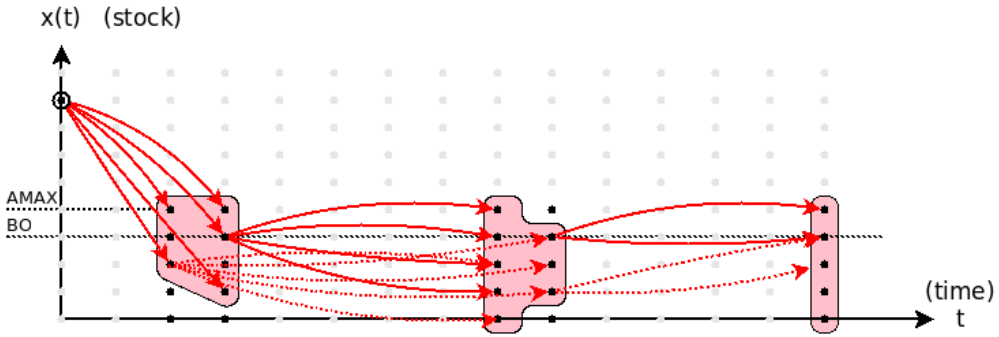
- Working on a reduced graph (Rozenknop, Wolfler *et al*) :



- Optimal policy between nodes (x, t) and (x', t') can be computed greedily.
- Improvement : possibility to valuate all the successors arcs of a node (x, t) at once.
- The use of forward induction allows to construct the graph dynamically.

Feasibility of the path : maximum modulation (2)

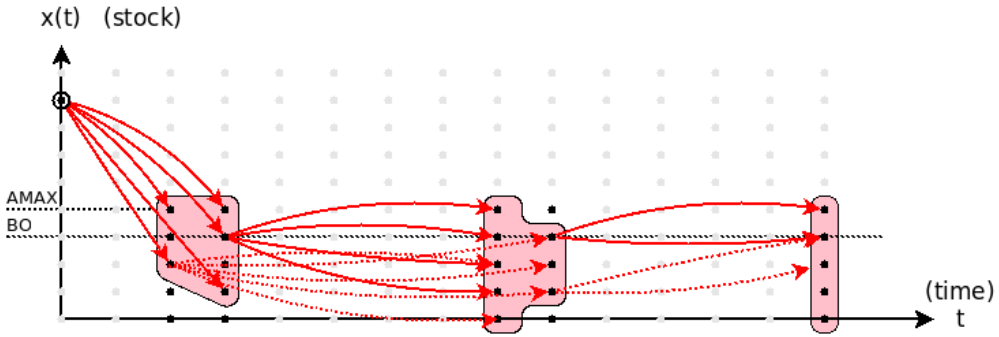
- Working on a reduced graph (Rozenknop, Wolfler *et al*) :



- Optimal policy between nodes (x, t) and (x', t') can be computed greedily.
- Improvement : possibility to valuate all the successors arcs of a node (x, t) at once.
- The use of forward induction allows to construct the graph dynamically.

Feasibility of the path : maximum modulation (2)

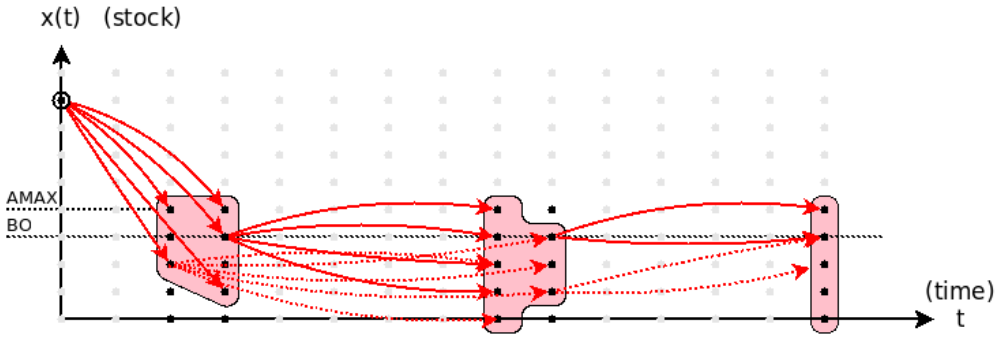
- Working on a reduced graph (Rozenknop, Wolfler *et al*) :



- Optimal policy between nodes (x, t) and (x', t') can be computed greedily.
- Improvement : possibility to valuate all the successors arcs of a node (x, t) at once.
- The use of forward induction allows to construct the graph dynamically.

Feasibility of the path : maximum modulation (2)

- Working on a reduced graph (Rozenknop, Wolfler *et al*) :



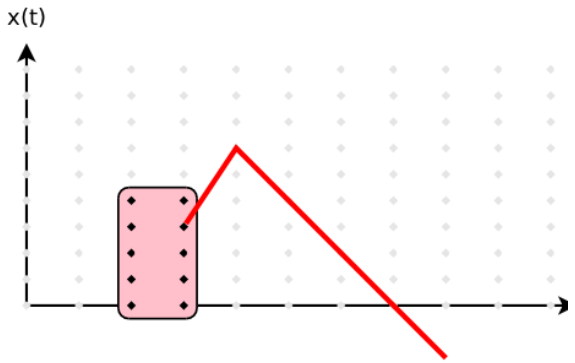
- Optimal policy between nodes (x, t) and (x', t') can be computed greedily.
- Improvement : possibility to valuate all the successors arcs of a node (x, t) at once.
- The use of forward induction allows to construct the graph dynamically.

Feasibility of the path : maximum modulation (3)

- Start from (x, t) with a minimal refuel $RMIN_k$ and maximal production $PMAX^t$.
- Sort the time periods by decreasing reduced costs (hence increasing μ_t) :

$$\mu_{t_1} \leq \cdots \leq \mu_{t_j} \leq \cdots \leq \mu_{t_n}$$

- While modulation $< MMAX_k$ and $\mu_{t_j} < C_k$, modulate at t_j.
- When $\mu_{t_j} \geq C_k$, refueling is more interesting than modulation.
- Refuel until $\min(RMAX_k, SMAX_k - x)$ is reached.
- While modulation $< MMAX_k$, modulate at t_j.

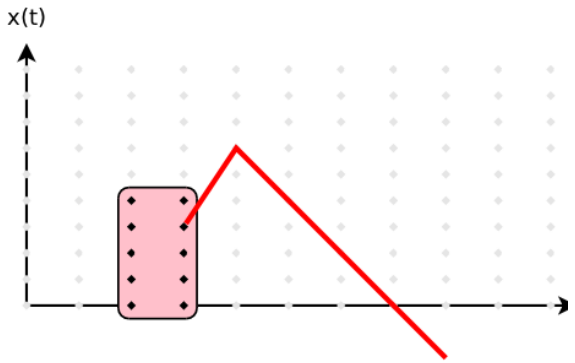


Feasibility of the path : maximum modulation (3)

- Start from (x, t) with a minimal refuel $RMIN_k$ and maximal production $PMAX^t$.
- Sort the time periods by decreasing reduced costs (hence increasing μ_t) :

$$\mu_{t_1} \leq \cdots \leq \mu_{t_j} \leq \cdots \leq \mu_{t_n}$$

- While modulation $< MMAX_k$ and $\mu_{t_j} < C_k$, modulate at t_j.
- When $\mu_{t_j} \geq C_k$, refueling is more interesting than modulation.
- Refuel until $\min(RMAX_k, SMAX_k - x)$ is reached.
- While modulation $< MMAX_k$, modulate at t_j.

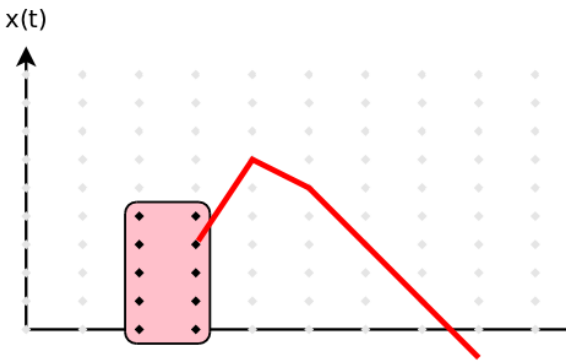


Feasibility of the path : maximum modulation (3)

- Start from (x, t) with a minimal refuel $RMIN_k$ and maximal production $PMAX^t$.
- Sort the time periods by decreasing reduced costs (hence increasing μ_t) :

$$\mu_{t_1} \leq \cdots \leq \mu_{t_j} \leq \cdots \leq \mu_{t_n}$$

- While modulation $< MMAX_k$ and $\mu_{t_j} < C_k$, modulate at t_j.
- When $\mu_{t_j} \geq C_k$, refueling is more interesting than modulation.
- Refuel until $\min(RMAX_k, SMAX_k - x)$ is reached.
- While modulation $< MMAX_k$, modulate at t_j.

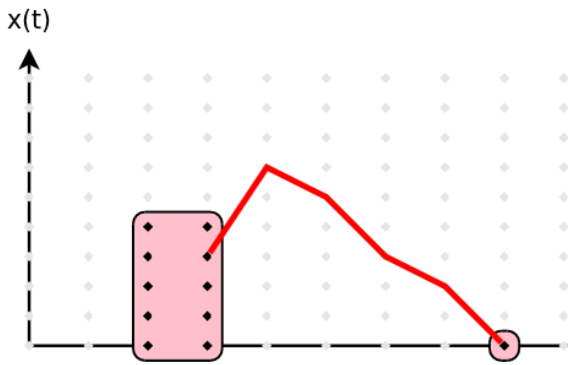


Feasibility of the path : maximum modulation (3)

- Start from (x, t) with a minimal refuel $RMIN_k$ and maximal production $PMAX^t$.
- Sort the time periods by decreasing reduced costs (hence increasing μ_t) :

$$\mu_{t_1} \leq \cdots \leq \mu_{t_j} \leq \cdots \leq \mu_{t_n}$$

- While modulation $< MMAX_k$ and $\mu_{t_j} < C_k$, modulate at t_j.
- When $\mu_{t_j} \geq C_k$, refueling is more interesting than modulation.
- Refuel until $\min(RMAX_k, SMAX_k - x)$ is reached.
- While modulation $< MMAX_k$, modulate at t_j.

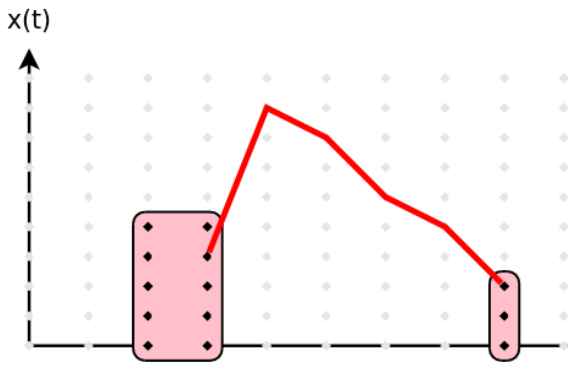


Feasibility of the path : maximum modulation (3)

- Start from (x, t) with a minimal refuel $RMIN_k$ and maximal production $PMAX^t$.
- Sort the time periods by decreasing reduced costs (hence increasing μ_t) :

$$\mu_{t_1} \leq \cdots \leq \mu_{t_j} \leq \cdots \leq \mu_{t_n}$$

- While modulation $< MMAX_k$ and $\mu_{t_j} < C_k$, modulate at t_j.
- When $\mu_{t_j} \geq C_k$, refueling is more interesting than modulation.
- Refuel until $\min(RMAX_k, SMAX_k - x)$ is reached.
- While modulation $< MMAX_k$, modulate at t_j.

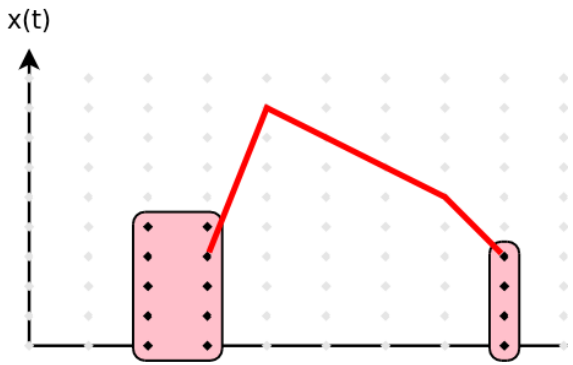


Feasibility of the path : maximum modulation (3)

- Start from (x, t) with a minimal refuel $RMIN_k$ and maximal production $PMAX^t$.
- Sort the time periods by decreasing reduced costs (hence increasing μ_t) :

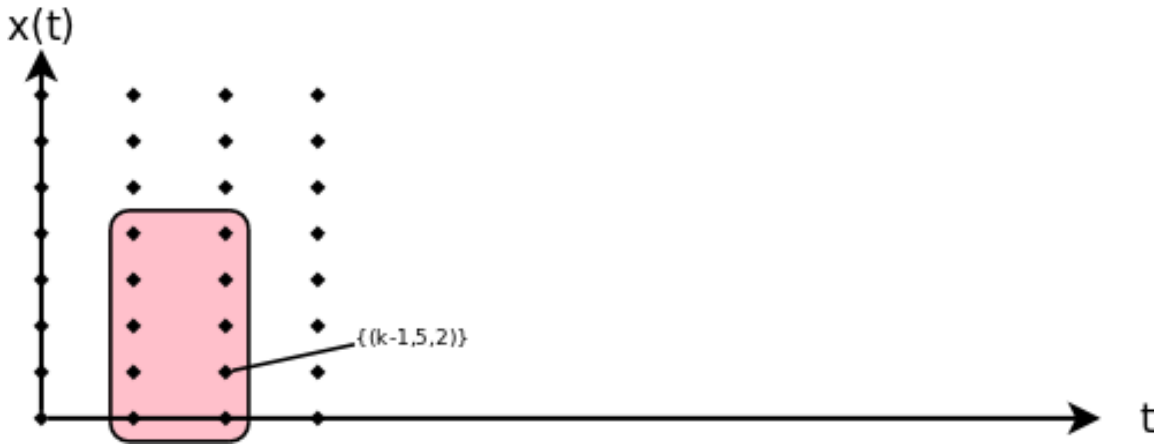
$$\mu_{t_1} \leq \cdots \leq \mu_{t_j} \leq \cdots \leq \mu_{t_n}$$

- While modulation $< MMAX_k$ and $\mu_{t_j} < C_k$, modulate at t_j.
- When $\mu_{t_j} \geq C_k$, refueling is more interesting than modulation.
- Refuel until $\min(RMAX_k, SMAX_k - x)$ is reached.
- While modulation $< MMAX_k$, modulate at t_j.



Feasibility of the path : maximum modulation (4)

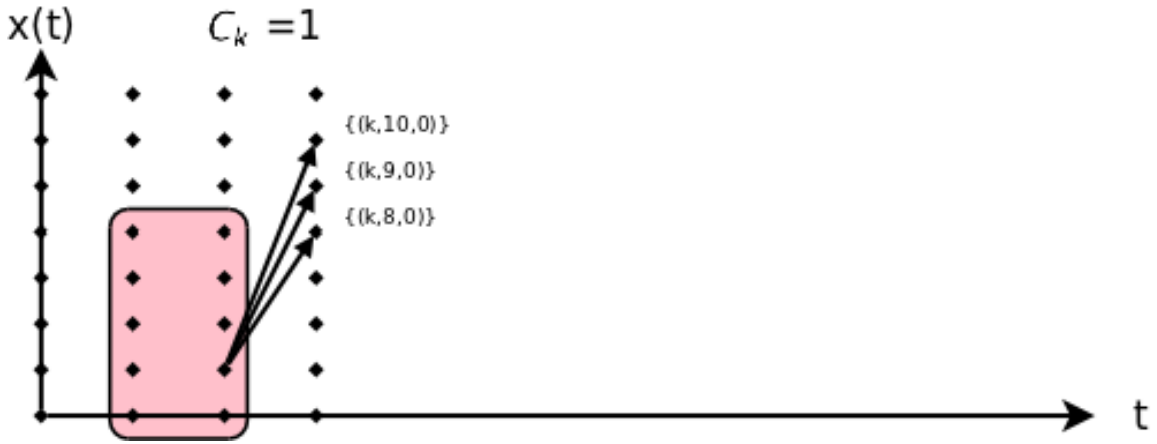
- Shortest path with a resource constraint :
  - Forward induction : propagation of labels (k, c, m)
    - k is the cycle
    - c is the cumulated cost from the source
    - m is the cumulated modulation from the source
  - There are possibly several labels at each node :
    - Labels with a modulation $m > MMAX_{i,k}$ are discarded.
    - We keep only labels in the Pareto-Front
  - Heuristic to tell if we can ignore modulation.



  - Works in backward induction !

Feasibility of the path : maximum modulation (4)

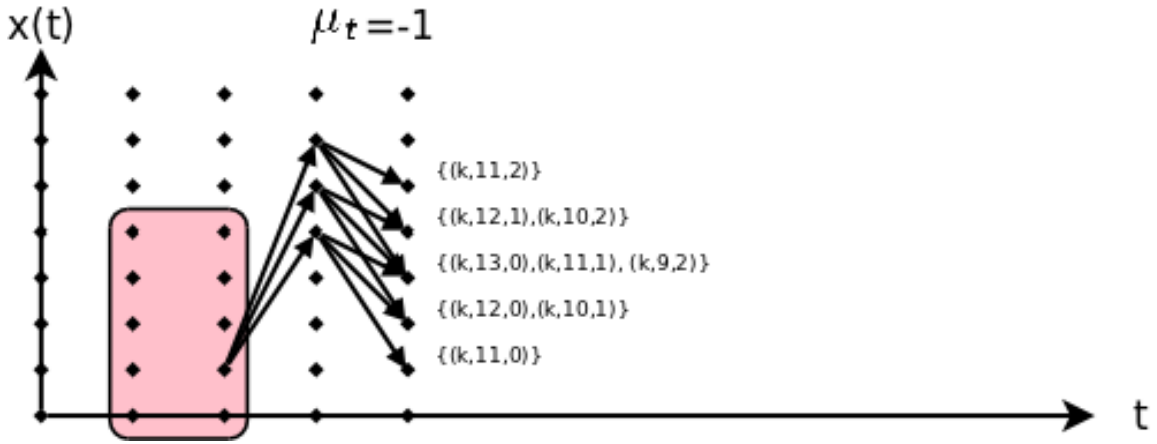
- Shortest path with a resource constraint :
  - Forward induction : propagation of labels (k, c, m)
    - k is the cycle
    - c is the cumulated cost from the source
    - m is the cumulated modulation from the source
  - There are possibly several labels at each node :
    - Labels with a modulation $m > MMAX_{i,k}$ are discarded.
    - We keep only labels in the Pareto-Front
  - Heuristic to tell if we can ignore modulation.



  - Works in backward induction !

Feasibility of the path : maximum modulation (4)

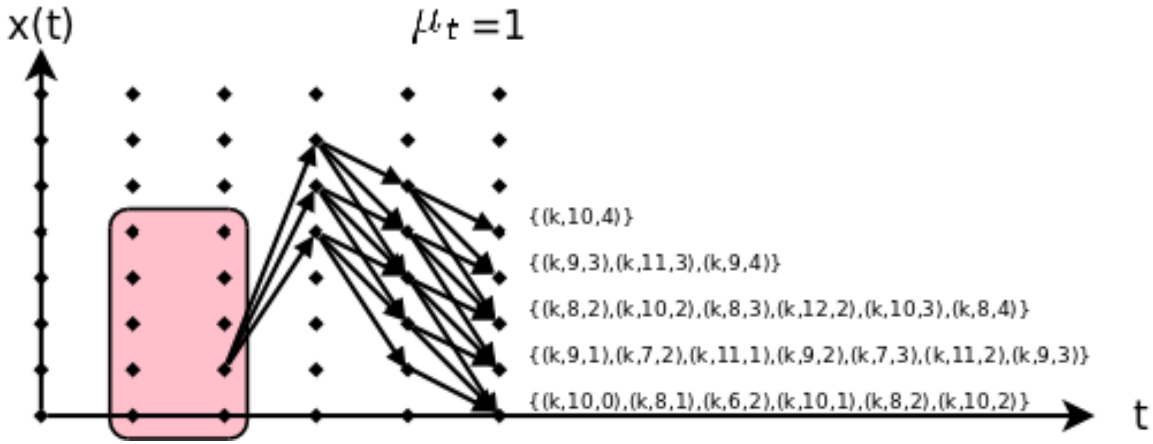
- Shortest path with a resource constraint :
  - Forward induction : propagation of labels (k, c, m)
    - k is the cycle
    - c is the cumulated cost from the source
    - m is the cumulated modulation from the source
  - There are possibly several labels at each node :
    - Labels with a modulation $m > MMAX_{i,k}$ are discarded.
    - We keep only labels in the Pareto-Front
  - Heuristic to tell if we can ignore modulation.



  - Works in backward induction !

Feasibility of the path : maximum modulation (4)

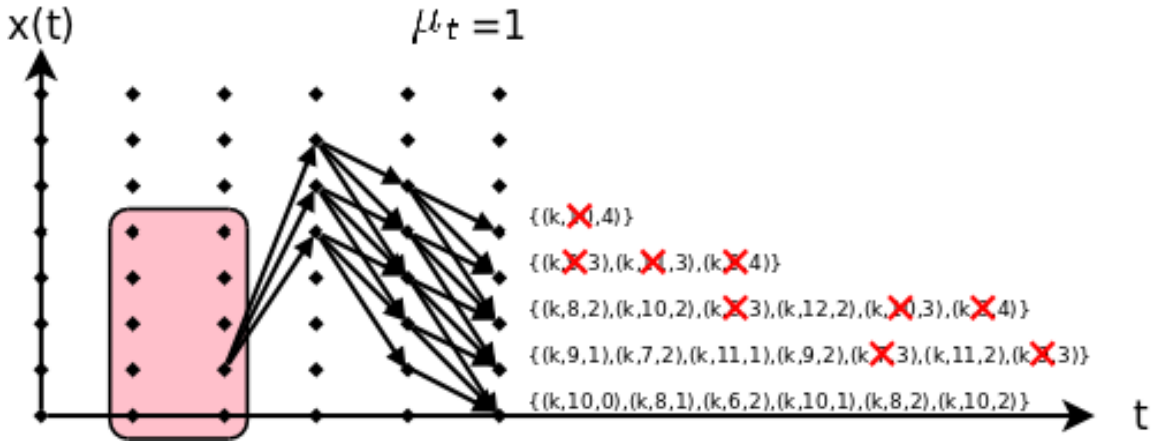
- Shortest path with a resource constraint :
  - Forward induction : propagation of labels (k, c, m)
    - k is the cycle
    - c is the cumulated cost from the source
    - m is the cumulated modulation from the source
  - There are possibly several labels at each node :
    - Labels with a modulation $m > MMAX_{i,k}$ are discarded.
    - We keep only labels in the Pareto-Front
  - Heuristic to tell if we can ignore modulation.



  - Works in backward induction !

Feasibility of the path : maximum modulation (4)

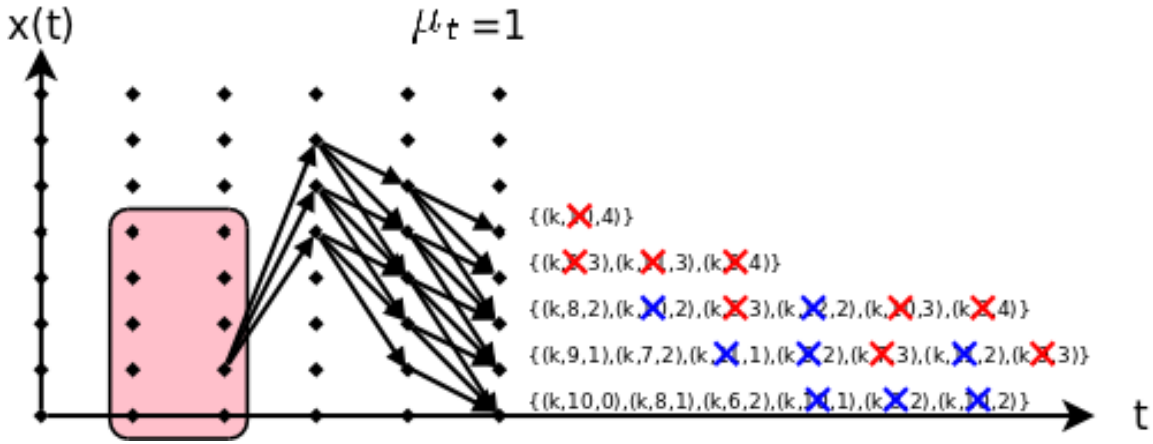
- Shortest path with a resource constraint :
  - Forward induction : propagation of labels (k, c, m)
    - k is the cycle
    - c is the cumulated cost from the source
    - m is the cumulated modulation from the source
  - There are possibly several labels at each node :
    - Labels with a modulation $m > MMAX_{i,k}$ are discarded.
    - We keep only labels in the Pareto-Front
  - Heuristic to tell if we can ignore modulation.



  - Works in backward induction !

Feasibility of the path : maximum modulation (4)

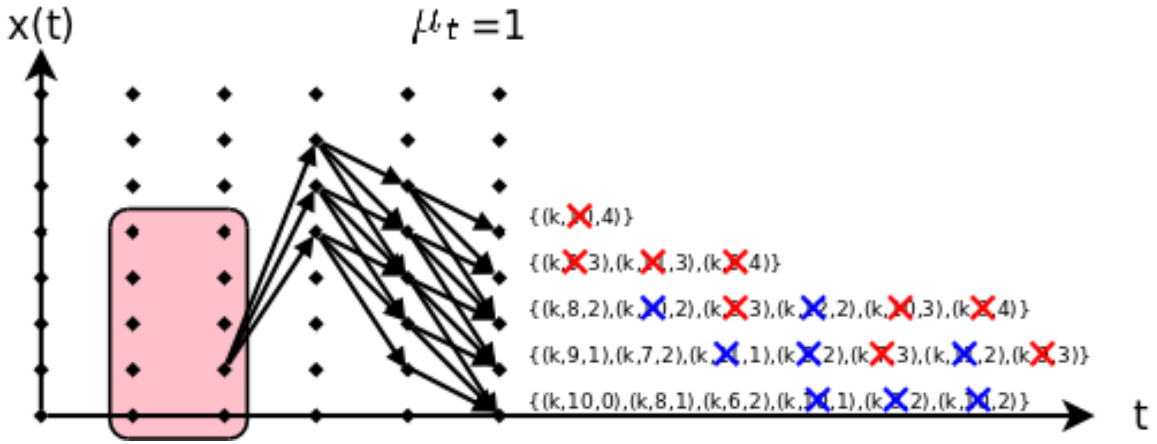
- Shortest path with a resource constraint :
  - Forward induction : propagation of labels (k, c, m)
    - k is the cycle
    - c is the cumulated cost from the source
    - m is the cumulated modulation from the source
  - There are possibly several labels at each node :
    - Labels with a modulation $m > MMAX_{i,k}$ are discarded.
    - We keep only labels in the Pareto-Front
  - Heuristic to tell if we can ignore modulation.



  - Works in backward induction !

Feasibility of the path : maximum modulation (4)

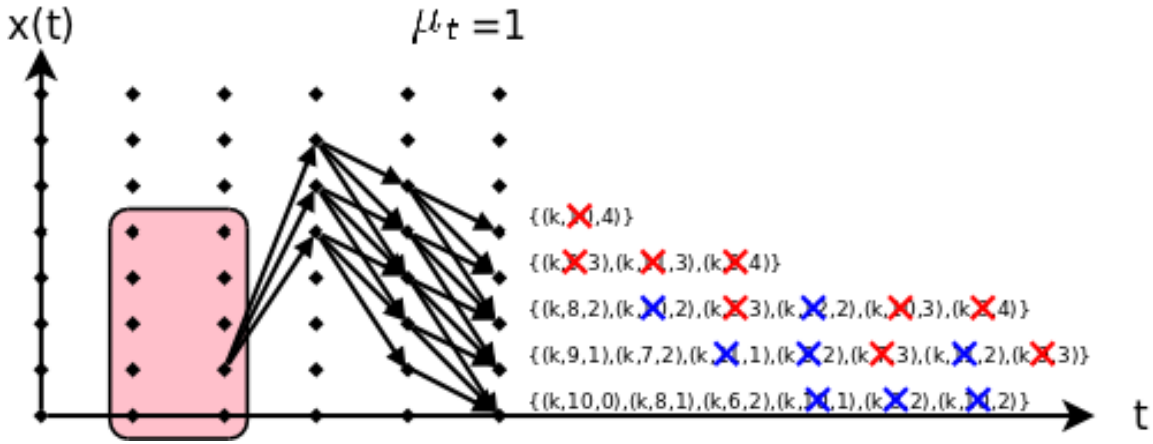
- Shortest path with a resource constraint :
  - Forward induction : propagation of labels (k, c, m)
    - k is the cycle
    - c is the cumulated cost from the source
    - m is the cumulated modulation from the source
  - There are possibly several labels at each node :
    - Labels with a modulation $m > MMAX_{i,k}$ are discarded.
    - We keep only labels in the Pareto-Front
  - Heuristic to tell if we can ignore modulation.



  - Works in backward induction !

Feasibility of the path : maximum modulation (4)

- Shortest path with a resource constraint :
  - Forward induction : propagation of labels (k, c, m)
    - k is the cycle
    - c is the cumulated cost from the source
    - m is the cumulated modulation from the source
  - There are possibly several labels at each node :
    - Labels with a modulation $m > MMAX_{i,k}$ are discarded.
    - We keep only labels in the Pareto-Front
  - Heuristic to tell if we can ignore modulation.

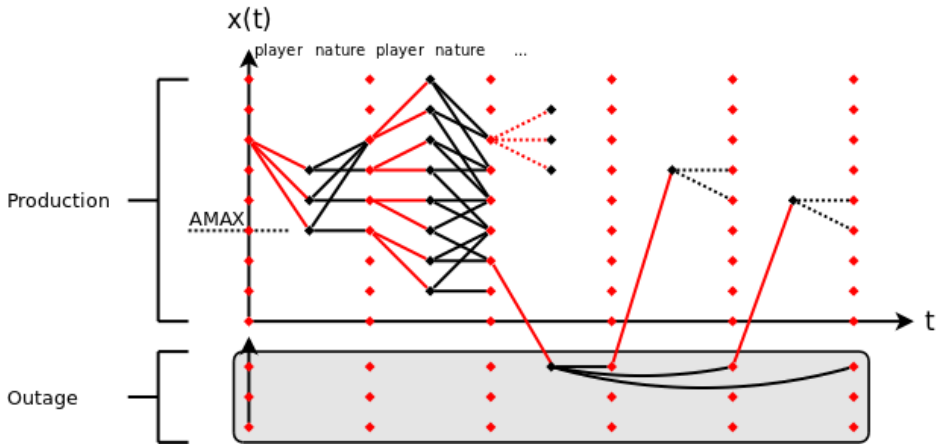


  - Works in backward induction !

Plan

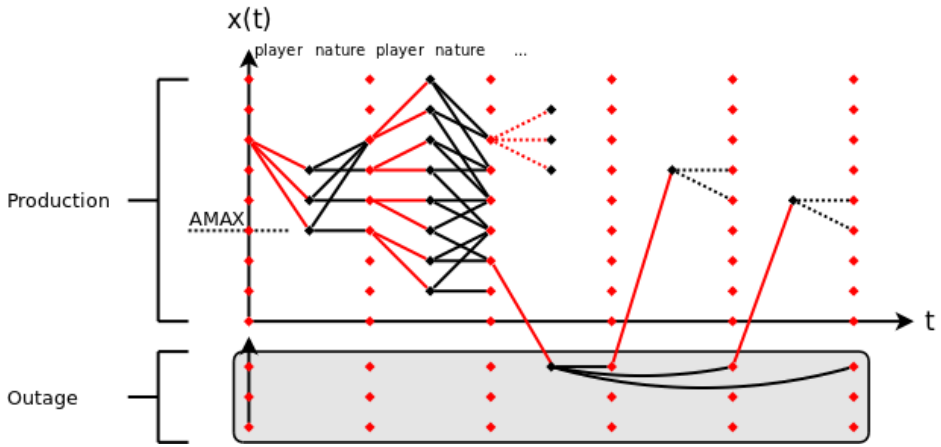
Stochastic subproblem

- Game against nature :
  - During production : player decides production level, nature decides to produce or not (incident)
  - During outage : player decides an outage, nature decides the duration, player decides the refuel and the first production, nature decides to produce or not (incident)
- Duplicate nodes to make explicite the current phase of the cycle (outage or production) :



Stochastic subproblem

- Game against nature :
  - During production : player decides production level, nature decides to produce or not (incident)
  - During outage : player decides an outage, nature decides the duration, player decides the refuel and the first production, nature decides to produce or not (incident)
- Duplicate nodes to make explicite the current phase of the cycle (outage or production) :



Backward induction for stochastic process

- Backward induction immediately generalizes to minimize expected cost :

$$
\begin{array}{lll}
\text{Horizon} & V_T(x) & = -C_T(x) \\
\text{Production state} & V_t(x) & = \min_{\substack{p \geq PMIN \\ p \leq PMAX \\ p \leq x}} \begin{array}{r} (V_{t+1}(x-p) - \mu_t p) \times Pr(\omega_t) \\ + V_{t+1}(x) \times (1 - Pr(\omega_t)) \end{array} \\
\text{Refueling state} & \cdots &
\end{array}
$$

- Does not take maximum modulation into account.

Backward induction for stochastic process

- Backward induction immediately generalizes to minimize expected cost :

$$
\begin{array}{lll}
\text{Horizon} & V_T(x) & = -C_T(x) \\
\text{Production state} & V_t(x) & = \min_{\substack{p \geq PMIN \\ p \leq PMAX \\ p \leq x}} \begin{array}{r} (V_{t+1}(x-p) - \mu_t p) \times Pr(\omega_t) \\ + V_{t+1}(x) \times (1 - Pr(\omega_t)) \end{array} \\
\text{Refueling state} & \cdots &
\end{array}
$$

- Does not take maximum modulation into account.

Backward induction and maximum modulation

- How should we count modulation when there is incident ?
  - Hypothesis : a zero production does not count as modulation
- Backward induction of labels (c, m) :
  - c is the expected cost needed to reach the horizon.
  - m is the maximal modulation needed to reach the next outage or the horizon, whatever the future incidents.
- A solution minimizes expected cost and satisfies maximum modulation in all the cases.
- Whatever the number of incidents, the outages occur in the allowed intervals.
  - Too strong $\Rightarrow$ relax this contraint as a chance contraint.

Backward induction and maximum modulation

- How should we count modulation when there is incident ?
  - Hypothesis : a zero production does not count as modulation
- Backward induction of labels (c, m) :
  - c is the expected cost needed to reach the horizon.
  - m is the maximal modulation needed to reach the next outage or the horizon, whatever the future incidents.
- A solution minimizes expected cost and satisfies maximum modulation in all the cases.
- Whatever the number of incidents, the outages occur in the allowed intervals.
  - Too strong $\Rightarrow$ relax this contraint as a chance contraint.

Backward induction and maximum modulation

- How should we count modulation when there is incident ?
  - Hypothesis : a zero production does not count as modulation
- Backward induction of labels (c, m) :
  - c is the expected cost needed to reach the horizon.
  - m is the maximal modulation needed to reach the next outage or the horizon, whatever the future incidents.
- A solution minimizes expected cost and satisfies maximum modulation in all the cases.
- Whatever the number of incidents, the outages occur in the allowed intervals.
  - Too strong $\Rightarrow$ relax this contraint as a chance contraint.

Backward induction and maximum modulation

- How should we count modulation when there is incident ?
  - Hypothesis : a zero production does not count as modulation
- Backward induction of labels (c, m) :
  - c is the expected cost needed to reach the horizon.
  - m is the maximal modulation needed to reach the next outage or the horizon, whatever the future incidents.
- A solution minimizes expected cost and satisfies maximum modulation in all the cases.
- Whatever the number of incidents, the outages occur in the allowed intervals.
  - Too strong $\Rightarrow$ relax this contraint as a chance contraint.

Backward induction and maximum modulation

- How should we count modulation when there is incident ?
  - Hypothesis : a zero production does not count as modulation
- Backward induction of labels (c, m) :
  - c is the expected cost needed to reach the horizon.
  - m is the maximal modulation needed to reach the next outage or the horizon, whatever the future incidents.
- A solution minimizes expected cost and satisfies maximum modulation in all the cases.
- Whatever the number of incidents, the outages occur in the allowed intervals.
  - Too strong $\Rightarrow$ relax this contraint as a chance contraint.

Stochastic master

- Problem : a planning is now defined as a closed-loop policy :

$$p(i, t, x)$$

- Considering only incidents, can be seen as a policy :

$$p(i, t, (\omega_i^l)^{l=0..t-1})$$

- Hypothesis : thermal plants can adapt instantaneously :

$$p(j, t, (\omega_i^l)_{i \in I}^{l=0..t})$$

- How can we satisfy energy balance exactly ?

$$\sum_{i \mid \omega_i^t = 1} p(i, t, (\omega_i^l)^{l=0..t-1}) + \sum_{j \in J} p(j, t, (\omega_i^l)_{i \in I}^{l=0..t}) = Dem_t, \ \forall (\omega_i^l)_{i \in I}^{l=0..t}$$

Stochastic master

- Problem : a planning is now defined as a closed-loop policy :

$$p(i, t, x)$$

- Considering only incidents, can be seen as a policy :

$$p(i, t, (\omega_i^l)^{l=0..t-1})$$

- Hypothesis : thermal plants can adapt instantaneously :

$$p(j, t, (\omega_i^l)_{i \in I}^{l=0..t})$$

- How can we satisfy energy balance exactly ?

$$\sum_{i \mid \omega_i^t = 1} p(i, t, (\omega_i^l)^{l=0..t-1}) + \sum_{j \in J} p(j, t, (\omega_i^l)_{i \in I}^{l=0..t}) = Dem_t, \ \forall (\omega_i^l)_{i \in I}^{l=0..t}$$

Stochastic master

- Problem : a planning is now defined as a closed-loop policy :

$$p(i, t, x)$$

- Considering only incidents, can be seen as a policy :

$$p(i, t, (\omega_i^l)^{l=0..t-1})$$

- Hypothesis : thermal plants can adapt instantaneously :

$$p(j, t, (\omega_i^l)_{i \in I}^{l=0..t})$$

- How can we satisfy energy balance exactly ?

$$\sum_{i \mid \omega_i^t = 1} p(i, t, (\omega_i^l)^{l=0..t-1}) + \sum_{j \in J} p(j, t, (\omega_i^l)_{i \in I}^{l=0..t}) = Dem_t, \ \forall (\omega_i^l)_{i \in I}^{l=0..t}$$

Stochastic master

- Problem : a planning is now defined as a closed-loop policy :

$$p(i, t, x)$$

- Considering only incidents, can be seen as a policy :

$$p(i, t, (\omega_i^l)^{l=0..t-1})$$

- Hypothesis : thermal plants can adapt instantaneously :

$$p(j, t, (\omega_i^l)_{i \in I}^{l=0..t})$$

- How can we satisfy energy balance exactly ?

$$\sum_{i \mid \omega_i^t = 1} p(i, t, (\omega_i^l)^{l=0..t-1}) + \sum_{j \in J} p(j, t, (\omega_i^l)_{i \in I}^{l=0..t}) = Dem_t, \ \forall (\omega_i^l)_{i \in I}^{l=0..t}$$

Simple stochastic master

- Subproblem gives only the mean production during each time periods : $\overline{p}(i,t)$.
- Master satisfies energy balance only in expectation :

$$\sum_{i \in I} \overline{p}(i,t) + \sum_{j \in J} p(j,t) = Dem_t$$

- Gives a lower bound for the previous master problem.

Simple stochastic master

- Subproblem gives only the mean production during each time periods : $\overline{p}(i,t)$.
- Master satisfies energy balance only in expectation :

$$\sum_{i \in I} \overline{p}(i,t) + \sum_{j \in J} p(j,t) = Dem_t$$

- Gives a lower bound for the previous master problem.

Simple stochastic master

- Subproblem gives only the mean production during each time periods : $\overline{p}(i,t)$.
- Master satisfies energy balance only in expectation :

$$\sum_{i \in I} \overline{p}(i,t) + \sum_{j \in J} p(j,t) = Dem_t$$

- Gives a lower bound for the previous master problem.

Stochastic master

- Subproblem can give the whole distribution $\rho(i,t)$ on $p(i,t)$.
- When the master combines planning for each subproblem, the probability distribution on total nuclear production is given by the convolution $*_i \rho(i,t)$ (since the aleas are independant).
- Problem : this operation is not linear.
  - Mean : linear constraint
  - Variance : quadratic constraint
  - ...

Stochastic master

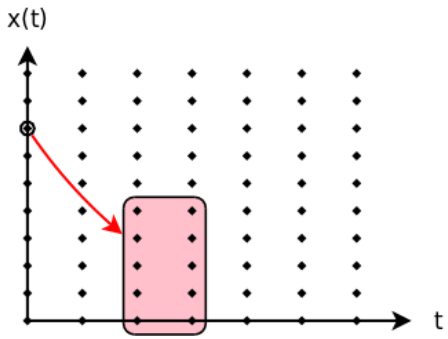
- Subproblem can give the whole distribution $\rho(i,t)$ on $p(i,t)$.
- When the master combines planning for each subproblem, the probability distribution on total nuclear production is given by the convolution $*_i \rho(i,t)$ (since the aleas are independant).
- Problem : this operation is not linear.
  - Mean : linear constraint
  - Variance : quadratic constraint
  - ...

Stochastic master

- Subproblem can give the whole distribution $\rho(i,t)$ on $p(i,t)$.
- When the master combines planning for each subproblem, the probability distribution on total nuclear production is given by the convolution $*_i\rho(i,t)$ (since the aleas are independant).
- Problem : this operation is not linear.
  - Mean : linear constraint
  - Variance : quadratic constraint
  - ...

Forward induction for stochastic process

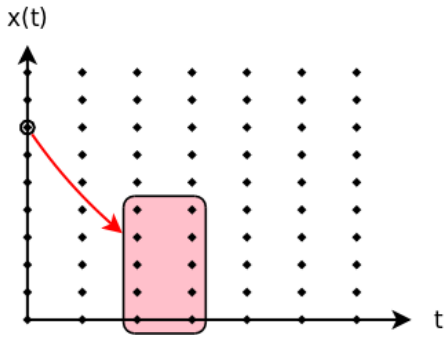
- Forward induction does not generalize easily.
- Nor the reduced graph approach.
- In the stochastic context :
  - We can compute an optimal policy from a node to the horizon.
  - We cannot compute an optimal policy from a node to a specific goal.



- Two values : mean cost and probability to reach the goal.

Forward induction for stochastic process

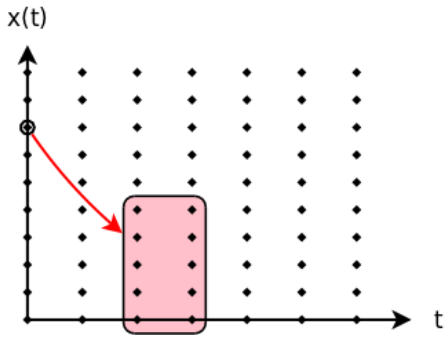
- Forward induction does not generalize easily.
- Nor the reduced graph approach.
- In the stochastic context :
  - We can compute an optimal policy from a node to the horizon.
  - We cannot compute an optimal policy from a node to a specific goal.



- Two values : mean cost and probability to reach the goal.

Forward induction for stochastic process

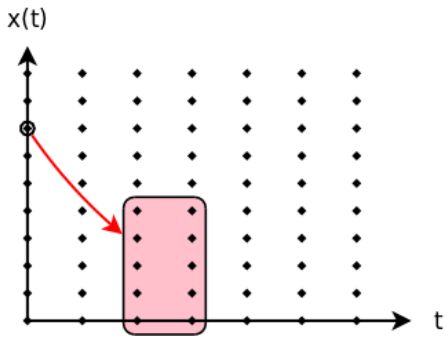
- Forward induction does not generalize easily.
- Nor the reduced graph approach.
- In the stochastic context :
  - We can compute an optimal policy from a node to the horizon.
  - We cannot compute an optimal policy from a node to a specific goal.



- Two values : mean cost and probability to reach the goal.

Forward induction for stochastic process

- Forward induction does not generalize easily.
- Nor the reduced graph approach.
- In the stochastic context :
  - We can compute an optimal policy from a node to the horizon.
  - We cannot compute an optimal policy from a node to a specific goal.



- Two values : mean cost and probability to reach the goal.

Subproblem and chance constraints

- Determine a solution which is a compromise between the classical reduced cost and the success probability of the solution :

$$\min \sum_{i,j} -\mu_t p_t - \gamma \log(Pr(success))$$

- Why log ?
  - In the master, we want to garantee that the solution satisfies a global success probability $Pr(success) \geq \alpha$.
  - Hence, the selected columns must satisfy :

$$\prod_{i,r \mid x_i^r} Pr(success_i^r) \geq \alpha$$

  - Using log, we obtain a linear constraint :

$$\sum_{i} \log(Pr(success_i^r)) \geq \log(\alpha)$$

Subproblem and chance constraints

- Determine a solution which is a compromise between the classical reduced cost and the success probability of the solution :

$$\min \sum_{i,j} -\mu_t p_t - \gamma \log(Pr(success))$$

- Why log ?
  - In the master, we want to garantee that the solution satisfies a global success probability $Pr(success) \geq \alpha$.
  - Hence, the selected columns must satisfy :

$$\prod_{i,r \mid x_i^r} Pr(success_i^r) \geq \alpha$$

  - Using log, we obtain a linear constraint :

$$\sum_i \log(Pr(success_i^r)) \geq \log(\alpha)$$

Subproblem and chance constraints

- Determine a solution which is a compromise between the classical reduced cost and the success probability of the solution :

$$\min \sum_{i,j} -\mu_t p_t - \gamma \log(Pr(success))$$

- Why log ?
  - In the master, we want to garantee that the solution satisfies a global success probability $Pr(success) \geq \alpha$.
  - Hence, the selected columns must satisfy :

$$\prod_{i,r \mid x_i^r} Pr(success_i^r) \geq \alpha$$

  - Using log, we obtain a linear constraint :

$$\sum_i \log(Pr(success_i^r)) \geq \log(\alpha)$$

Subproblem and chance constraints

- Determine a solution which is a compromise between the classical reduced cost and the success probability of the solution :

$$\min \sum_{i,j} -\mu_t p_t - \gamma \log(Pr(success))$$

- Why log ?
  - In the master, we want to garantee that the solution satisfies a global success probability $Pr(success) \geq \alpha$.
  - Hence, the selected columns must satisfy :

$$\prod_{i,r \mid x_i^r} Pr(success_i^r) \geq \alpha$$

  - Using log, we obtain a linear constraint :

$$\sum_i \log(Pr(success_i^r)) \geq \log(\alpha)$$

Subproblem and chance constraints

- Determine a solution which is a compromise between the classical reduced cost and the success probability of the solution :

$$\min \sum_{i,j} -\mu_t p_t - \gamma \log(Pr(success))$$

- Why log ?
  - In the master, we want to garantee that the solution satisfies a global success probability $Pr(success) \geq \alpha$.
  - Hence, the selected columns must satisfy :

$$\prod_{i,r \mid x_i^r} Pr(success_i^r) \geq \alpha$$

  - Using log, we obtain a linear constraint :

$$\sum_i \log(Pr(success_i^r)) \geq \log(\alpha)$$