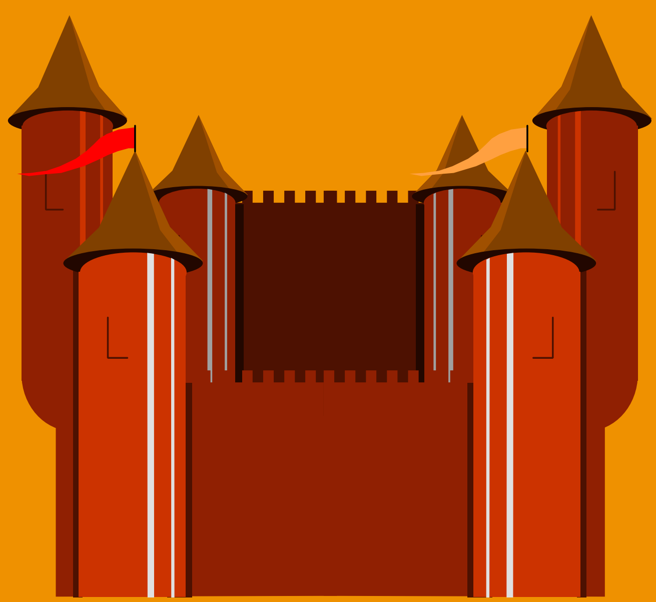
Finding the Best Policy: The Curious Case of Approximate Dynamic Programming

Optimization Days
May 2, 2011

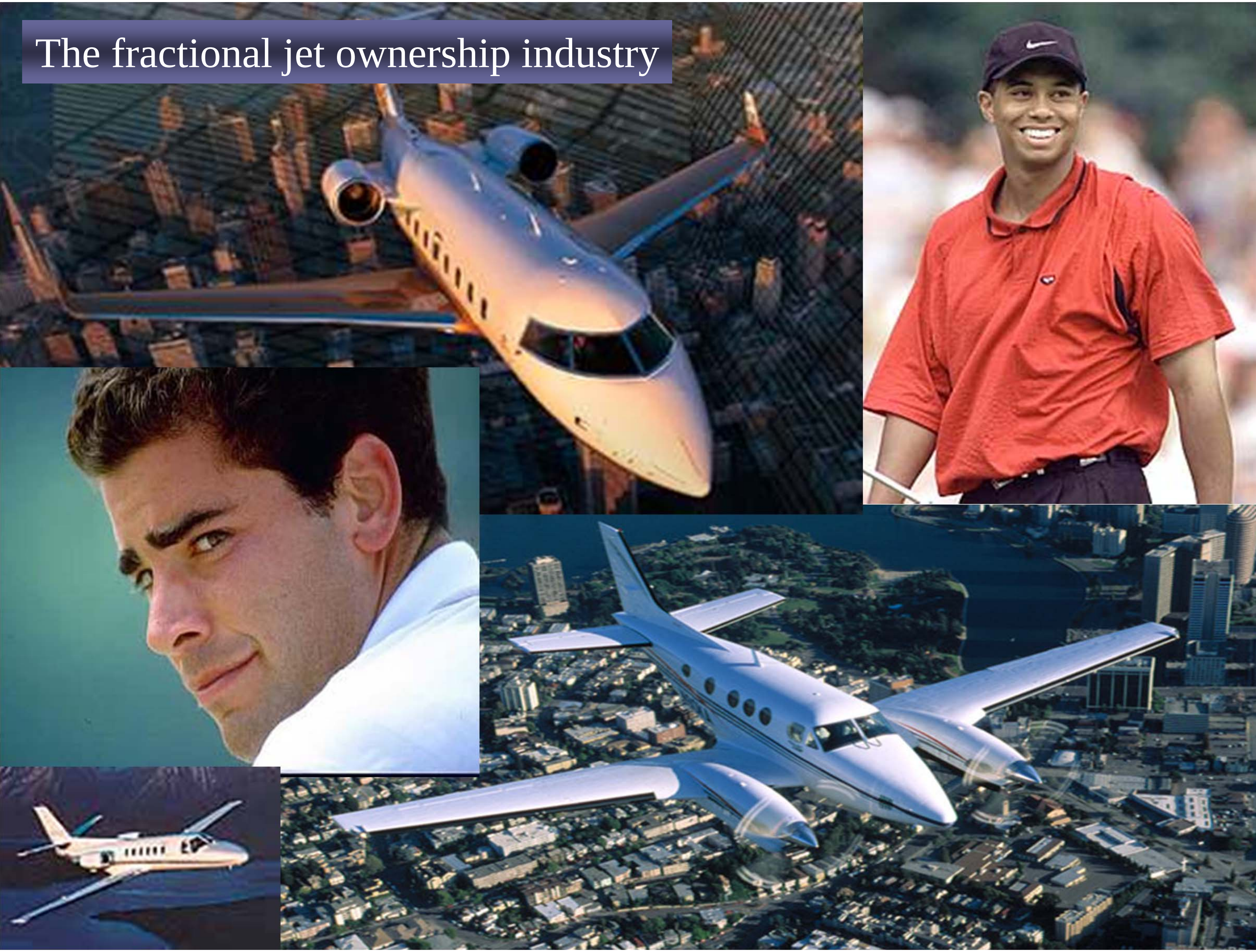


Warren Powell
CASTLE Laboratory
Princeton University
http://www.castlelab.princeton.edu


9533
9533


GRAIN OPERATIONS
LOADED GRAIN TRAINS POWERED
BNSF Coal Business Unit
Powder River & Illinois Coal Train information
Intermodal
On-Time Performance
Industrial Products
Service Summary
By Business Unit

The fractional jet ownership industry

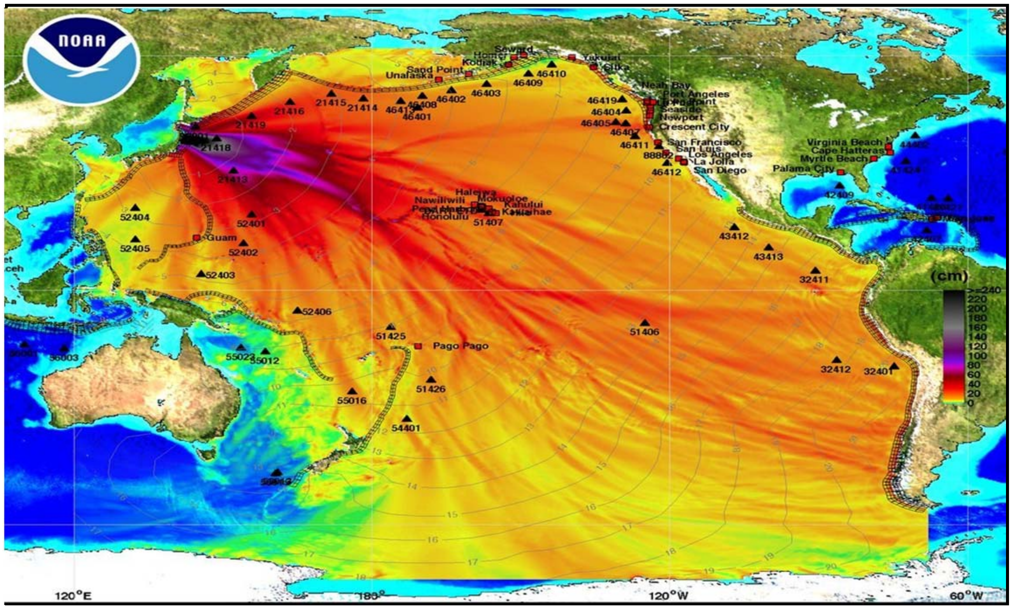


NetJets Inc.

Planning for a risky world

Disaster response

- Robust design of emergency response networks.
- Design of sensor networks and communication systems to manage responses to hurricanes, tsunamis, nuclear disasters and terrorist attacks.



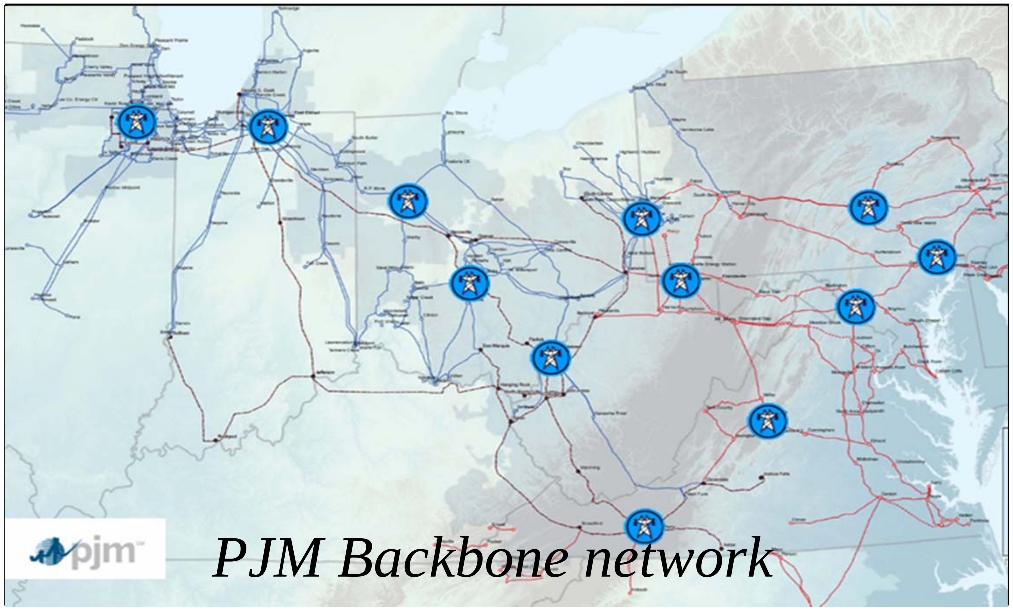
Disease

- Management of medical personnel, equipment and vaccines to respond to a disease outbreak.
- Robust design of supply chains to mitigate the disruption of transportation systems.

Managing the Grid

Electric Power Grid

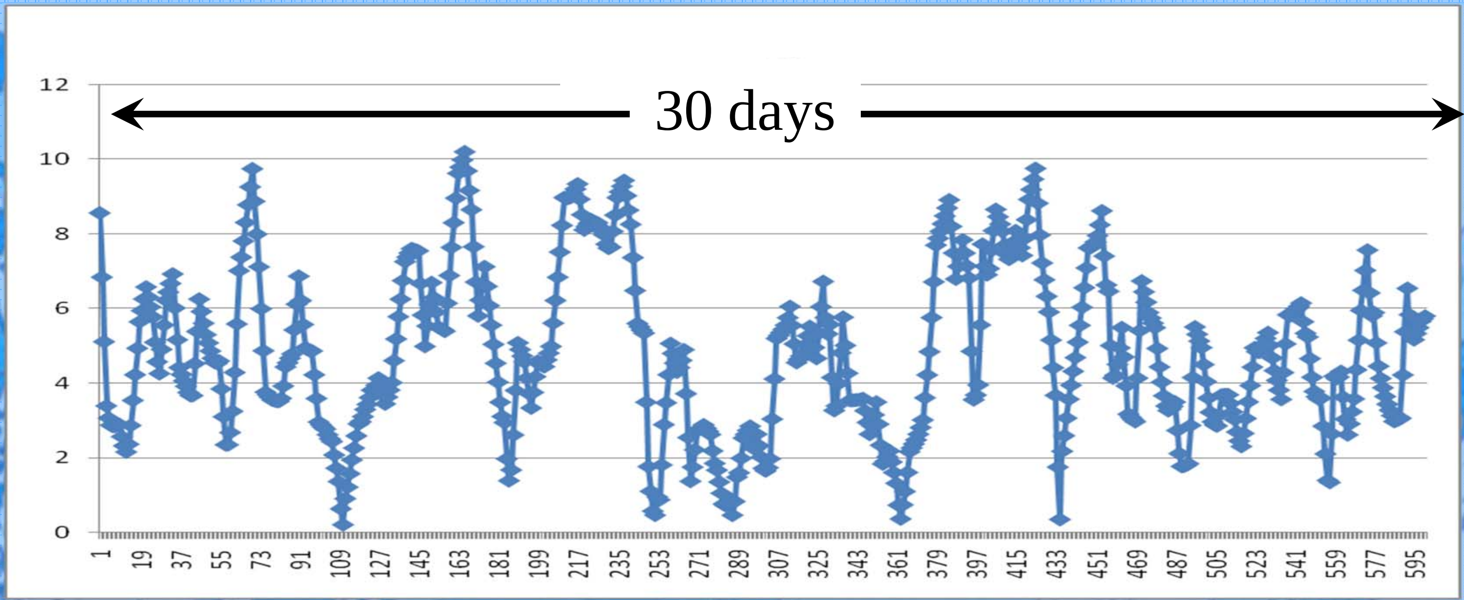
- What is the impact of high percentages of power from intermittent sources such as wind and solar?
- How will energy storage change the stability of the grid?

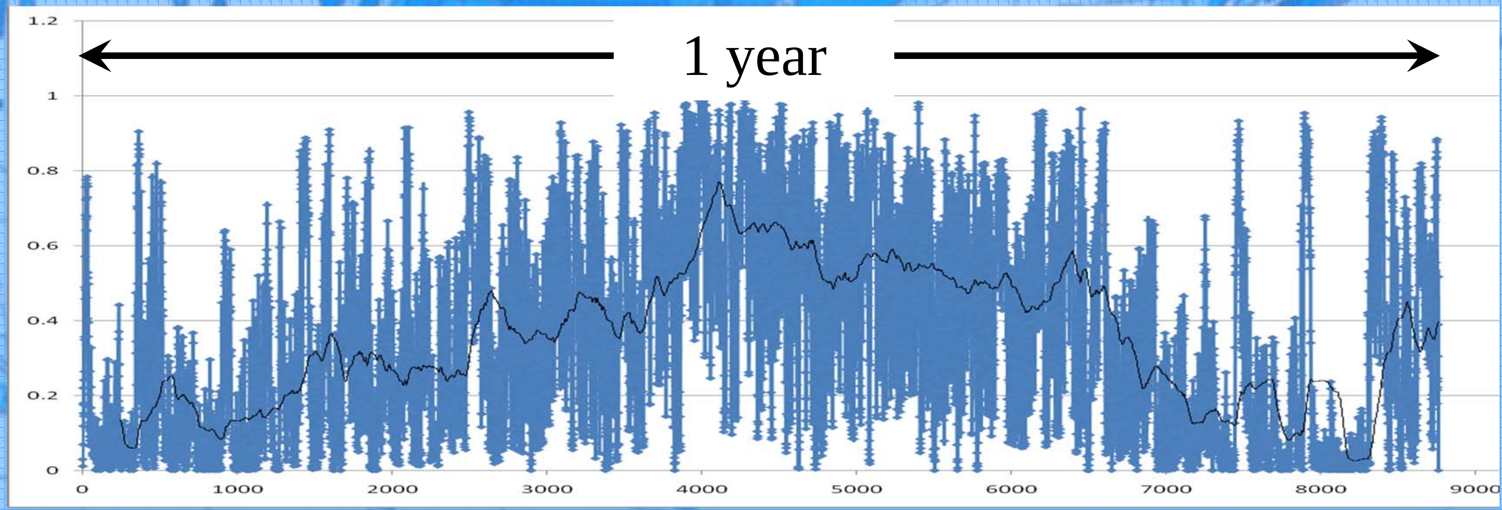


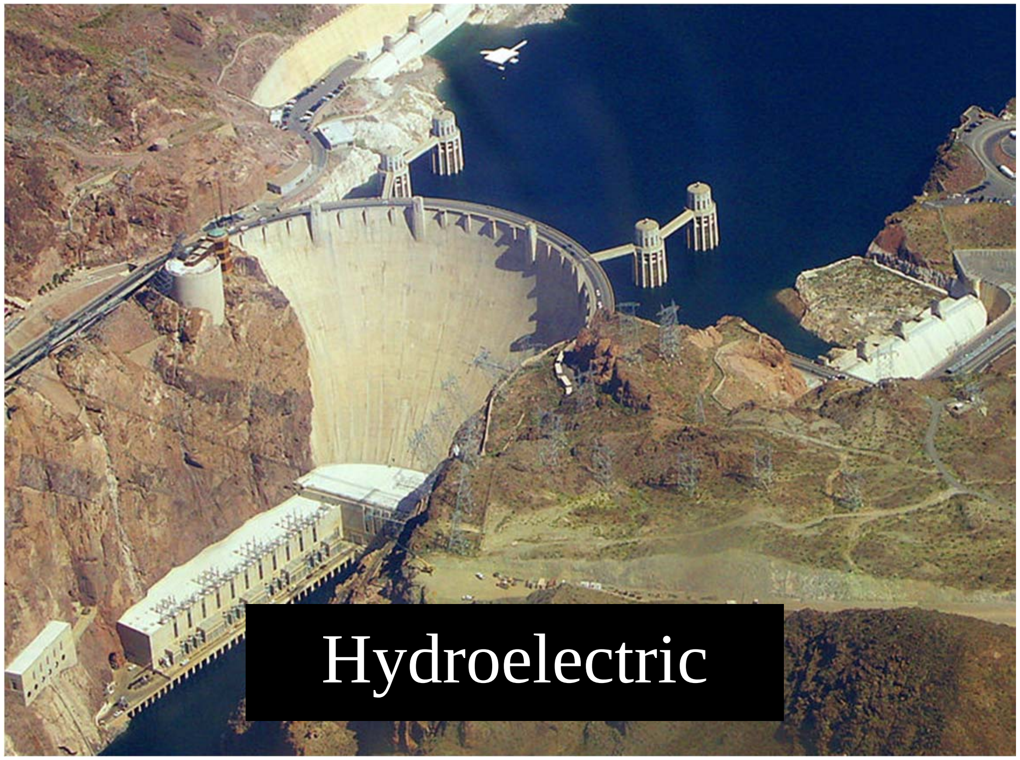
PJM Backbone network

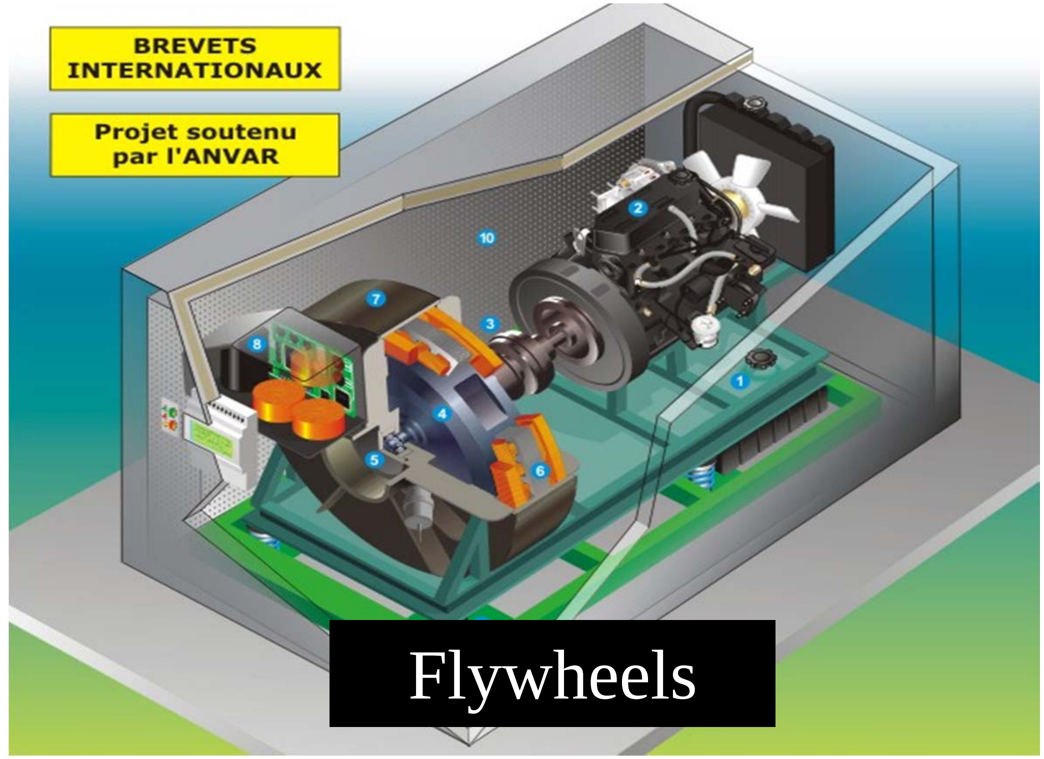
Urban power grids

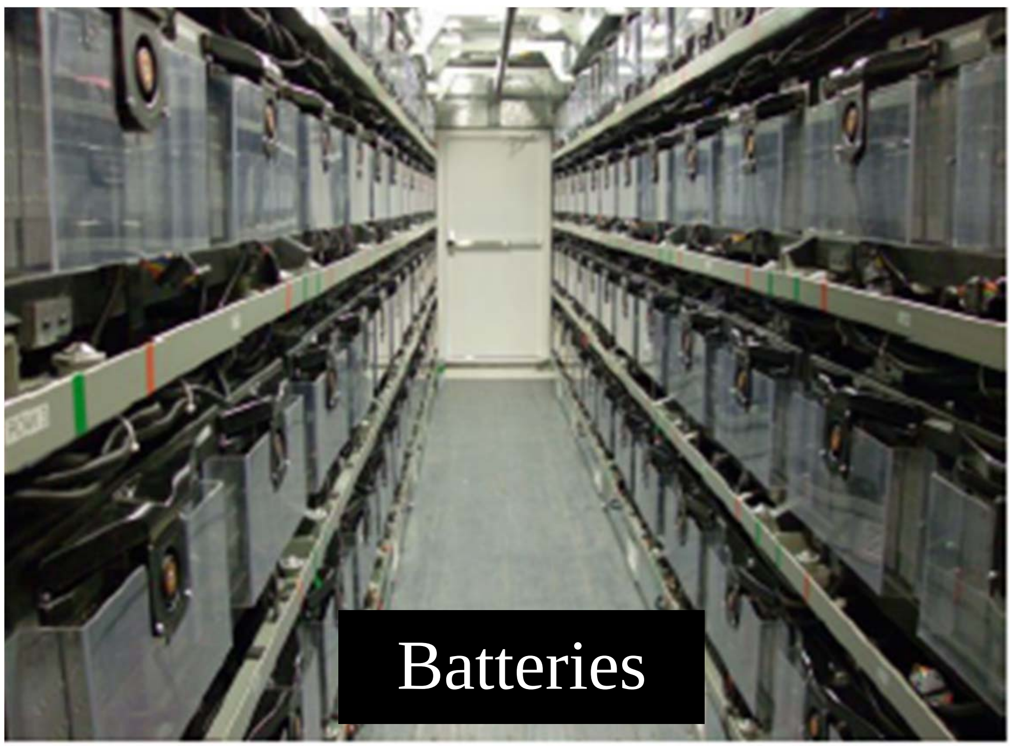
- How should New York City plan load curtailments as electric vehicles push the grid to capacity?
- How should local utilities plan load curtailments when demands cannot be met?

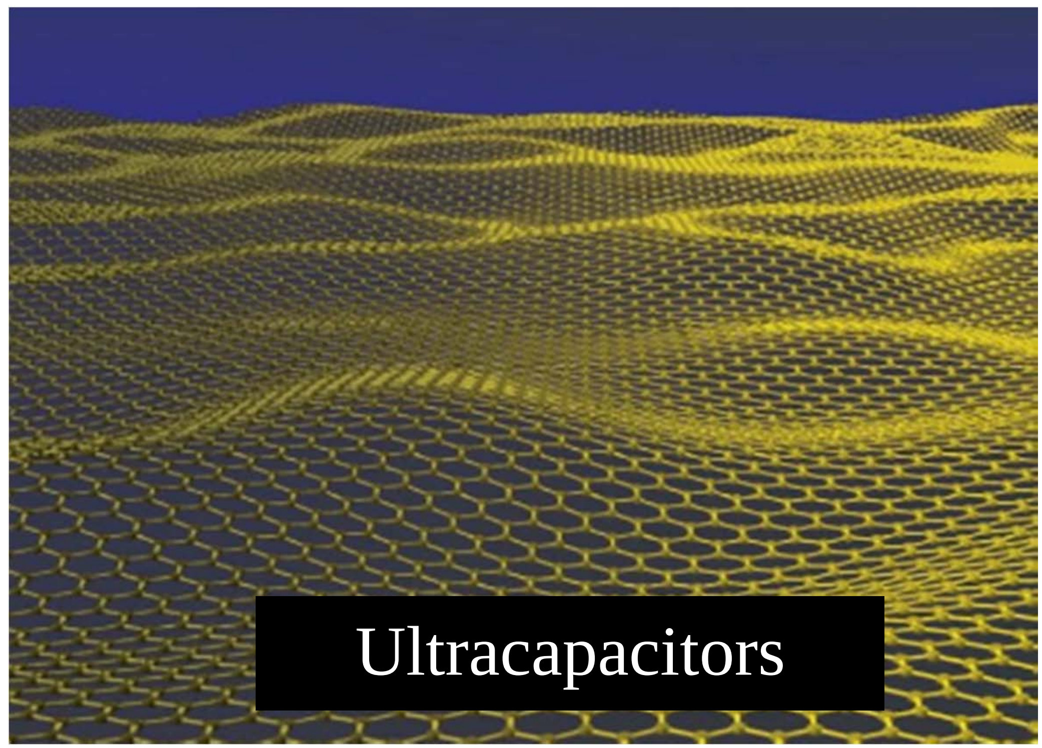

30 days
12
10
8
6
4
2
0
1
19
37
55
73
91
109
127
145
163
181
199
217
235
253
271
289
307
325
343
361
379
397
415
433
451
469
487
505
523
541
559
577
595


1 year
1.2
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

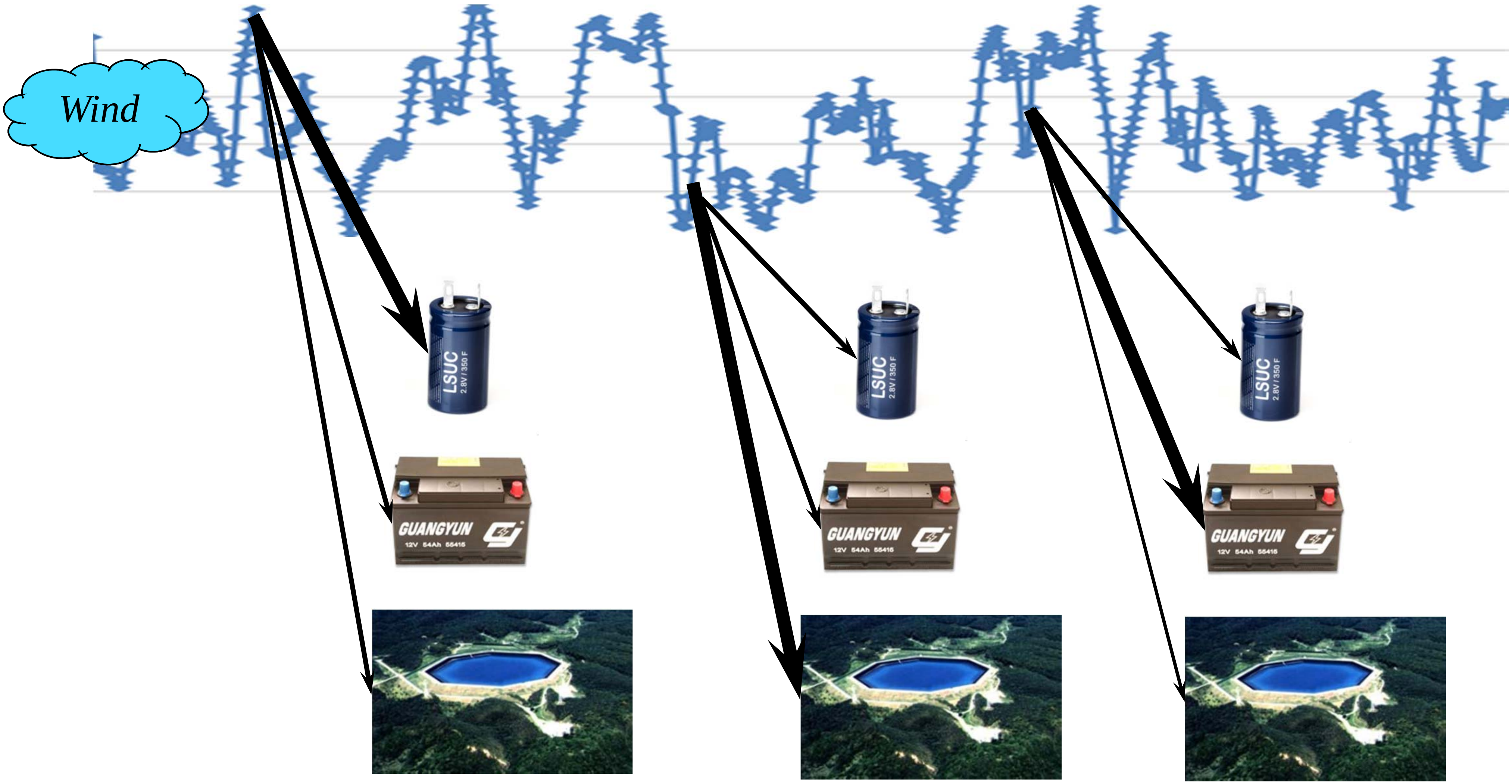

Hydroelectric


BREVETS INTERNATIONAUX
Projet soutenu par l'ANVAR
Flywheels


Batteries


Ultracapacitors

Heterogeneous storage portfolios



Challenges

- Real-time control
  - » Scheduling aircraft, pilots, generators, tankers
  - » Electricity resource allocation
  - » Trading on the spot market
- Near-term tactical planning
  - » Can I accept a customer request?
  - » Should I lease equipment?
  - » How much energy can I commit to with my wind turbines?
- Strategic planning
  - » What is the right equipment mix?
  - » What energy investments should I make?
  - » How do I meet renewable portfolio standards?

Deterministic modeling

- For deterministic problems, we speak the language of mathematical programming
  - » For static problems

$$\begin{aligned} &\min cx \\ &Ax = b \\ &x \geq 0 \end{aligned}$$

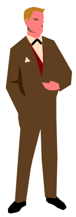
  - » For time-staged problems

$$\begin{aligned} &\min \sum_{t=0}^{T} c_t x_t \\ &A_t x_t - B_{t-1} x_{t-1} = b_t \\ &D_t x_t \leq u_t \\ &x_t \geq 0 \end{aligned}$$

Arguably Dantzig's biggest achievement, more so than the simplex algorithm, was his articulation of optimization problems in a standard format, which has given algorithmic researchers a common language.

Stochastic modeling

- The system state:



$$S_t = \left(R_t, D_t, \rho_t\right) = \text{System state, where:}$$

R_t = Resource state (how much capacity, reserves)

D_t = Market demands

ρ_t = "system parameters"

State of the technology (costs, performance)

Climate, weather (temperature, rainfall, wind)

Government policies (tax rebates on solar panels)

Market prices (oil, coal)

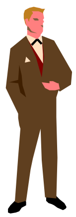
Stochastic modeling

- The decision variable:

$$x_t = \begin{pmatrix} \text{Routing} \\ \text{Assigning people or equipment} \\ \text{Purchase} \\ \text{Repair} \\ \text{Invest} \\ \text{Store} \end{pmatrix}$$

Stochastic modeling

- Exogenous information:



$$W_t = \text{New information} = \left(\hat{R}_t, \hat{D}_t, \hat{\rho}_t\right)$$

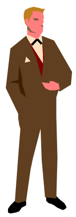
$\hat{R}_t$ = Delays, breakdowns, exogenous purchases

$\hat{D}_t$ = New demands to be served

$\hat{\rho}_t$ = Exogenous changes in parameters.

Stochastic modeling

- The transition function



$$S_{t+1} = S^M(S_t, x_t, W_{t+1})$$

Known as the:

"Transition function"
"Transfer function"
"System model"
"Plant model"
"Model"

Stochastic modeling

- The objective function

$$\min_{\pi} E^{\pi} \left\{ \sum_{t} \gamma^{t} C\left(S_t, X^{\pi}(S_t)\right) \right\}$$

Finding the best policy

Expectation over all random outcomes

State variable

Contribution function

Decision function (policy)

Given a *system model* (transition function)

$$S_{t+1} = S^{M}\left(S_t, x_t, W_{t+1}(\omega)\right)$$

» We have to find the best policy, which is a function that maps states to feasible actions, using only the information available when the decision is made.


Stochastic programming
Stochastic search
Model predictive control
Optimal control
Reinforcement learning
Q – learning
On-policy learning
Off-policy learning
Markov decision processes
Simulation optimization
Policy search

What is a policy?

- 1) Myopic policies
  - » Take the action that maximizes contribution (or minimizes cost) for just the current time period:

$$X^M(S_t) = \arg\min_{x_t} C(S_t, x_t)$$

  - » We can parameterize myopic policies with bonus and penalties to encourage good long-term behavior.

  - » Sometimes there are tunable parameters

$$X^M(S_t \mid \theta) = \arg\min_{x_t} C(S_t, x_t \mid \theta)$$

What is a policy?

- 2) Lookahead policies - Plan over the next T periods, but implement only the action it tells you to do now.
  - » Deterministic forecast

$$X^M(S_t) = \arg\min_{x_t, x_{t+1}, \ldots, x_{t+T}} C(S_t, x_t) + \sum_{t'=t+1}^{T} \gamma^{t'-t} C(S_{t'}, x_{t'})$$

  - » Stochastic programming (e.g. two-stage)

$$X^M(S_t) = \arg\min_{x_t, \left(x_{t+1}, \ldots, x_{t+T}\right)(\omega)} C(S_t, x_t) + \sum_{\omega \in \Omega} p(\omega) \sum_{t'=t+1}^{T} \gamma^{t'-t} C(S_{t'}(\omega), x_{t'}(\omega))$$

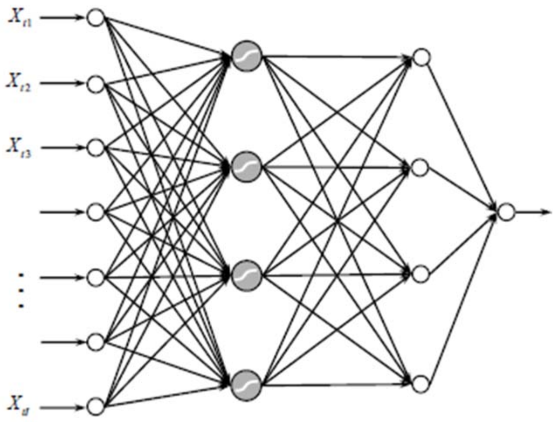
  - » Rolling/receding horizon procedures
  - » Model predictive control
  - » Rollout heuristics
  - » Tree search (decision trees)

What is a policy?

- 3) Policy function approximations
  - » Lookup table
    - When in this state, take this action.
  - » Parameterized functions
    - If the inventory is less than *s*, order up to *S*.
  - » Regression models

$$X^M(S_t \mid \theta) = \theta_0 + \theta_1 S_t + \theta_2 \left(S_t\right)^2$$

  - » Neural networks



What is a policy?

- 4) Policies based on value function approximations
  - » We approximate the value function and solve

$$X^M(S_t) = \arg\min_{x_t} \left(C(S_t, x_t) + \gamma E \bar{V}_{t+1}(S_{t+1}) \right)$$

  - » where

$$S_{t+1} = S^M(S_t, x_t, W_{t+1})$$

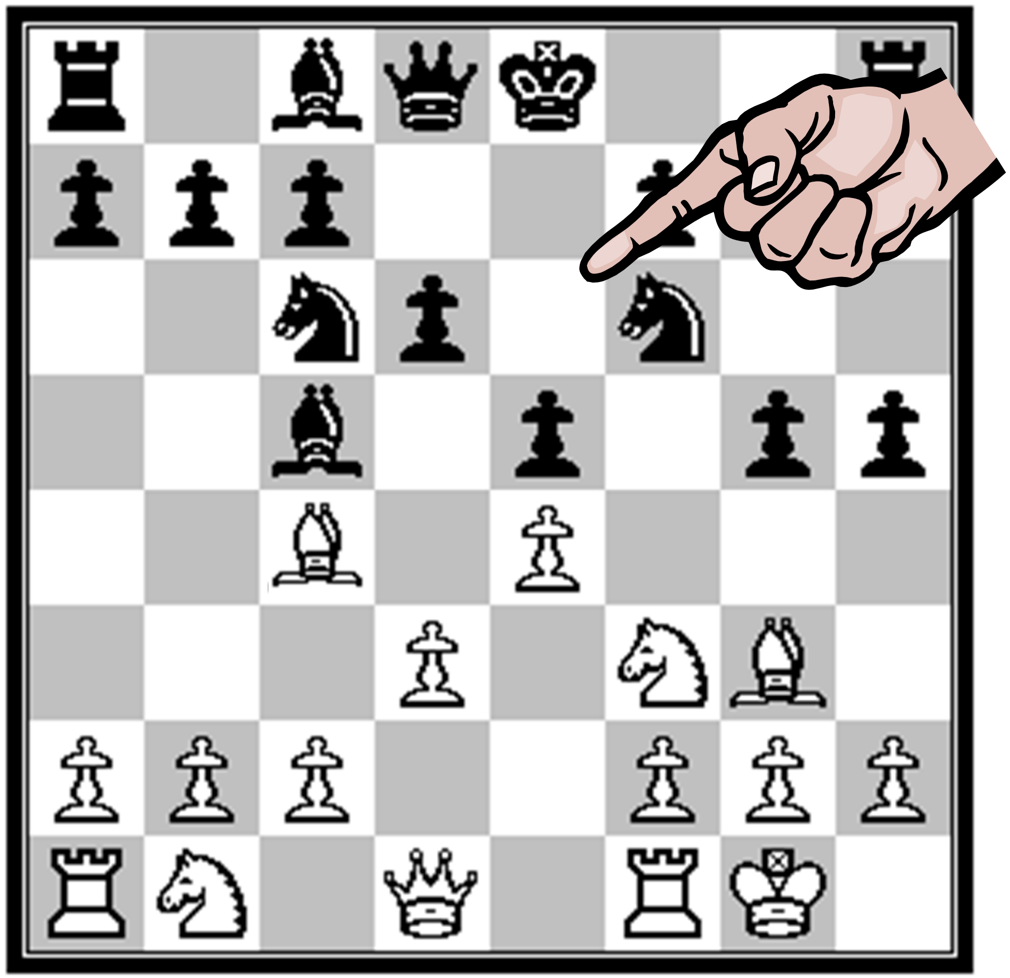
Approximations

- There are three classes of approximation strategies (for policies and value functions):
  - » Lookup table
    - Given a discrete state, return a discrete action or value
  - » Parametric models
    - Linear models (linear in the parameters)
    - Nonlinear models (e.g. an (s,S) inventory policy)
    - Neural networks
  - » Nonparametric models
    - Kernel regression
    - Dirichlet process-based models

Outline

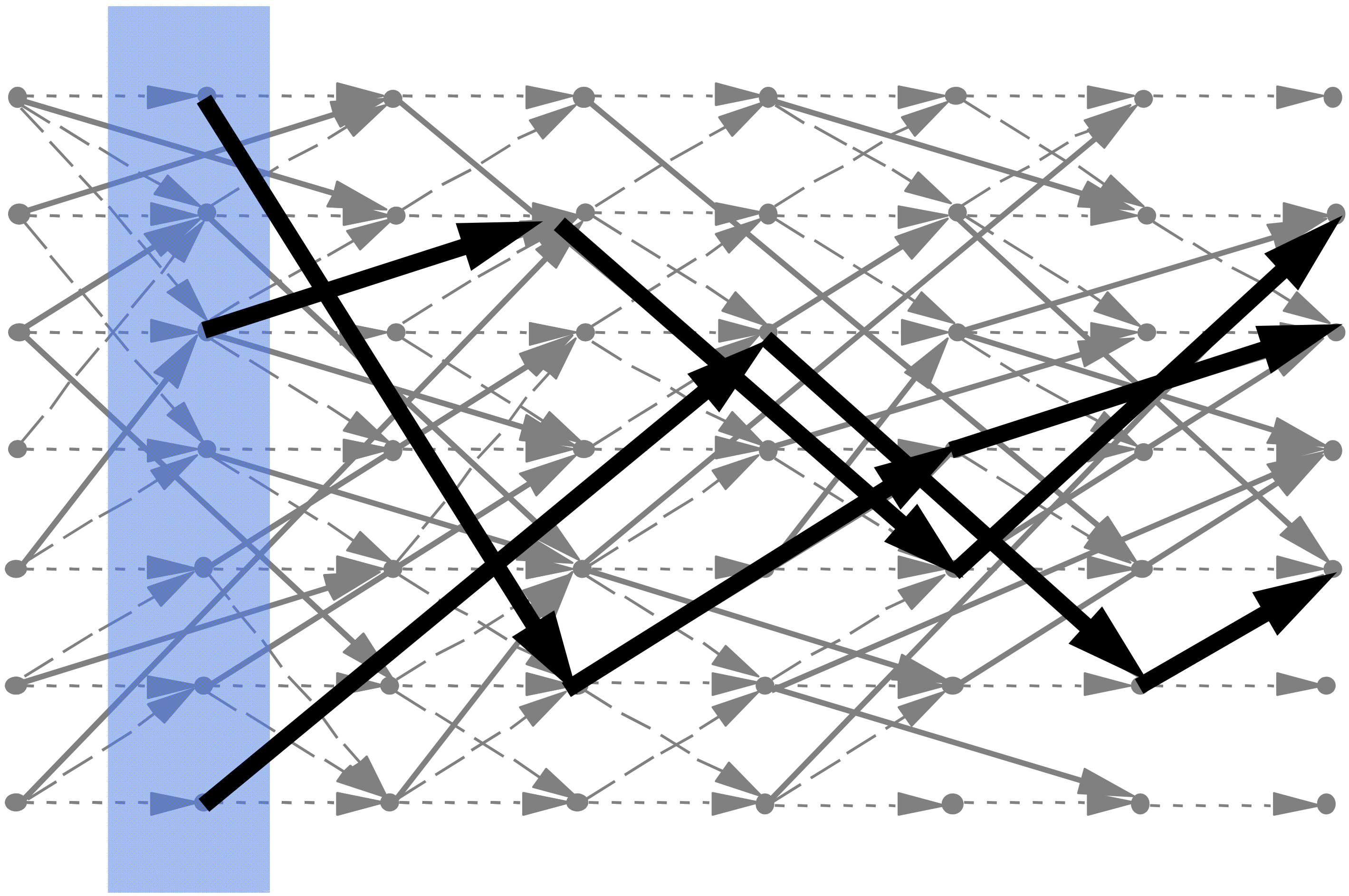
- A brief look at lookahead policies

Lookahead policies



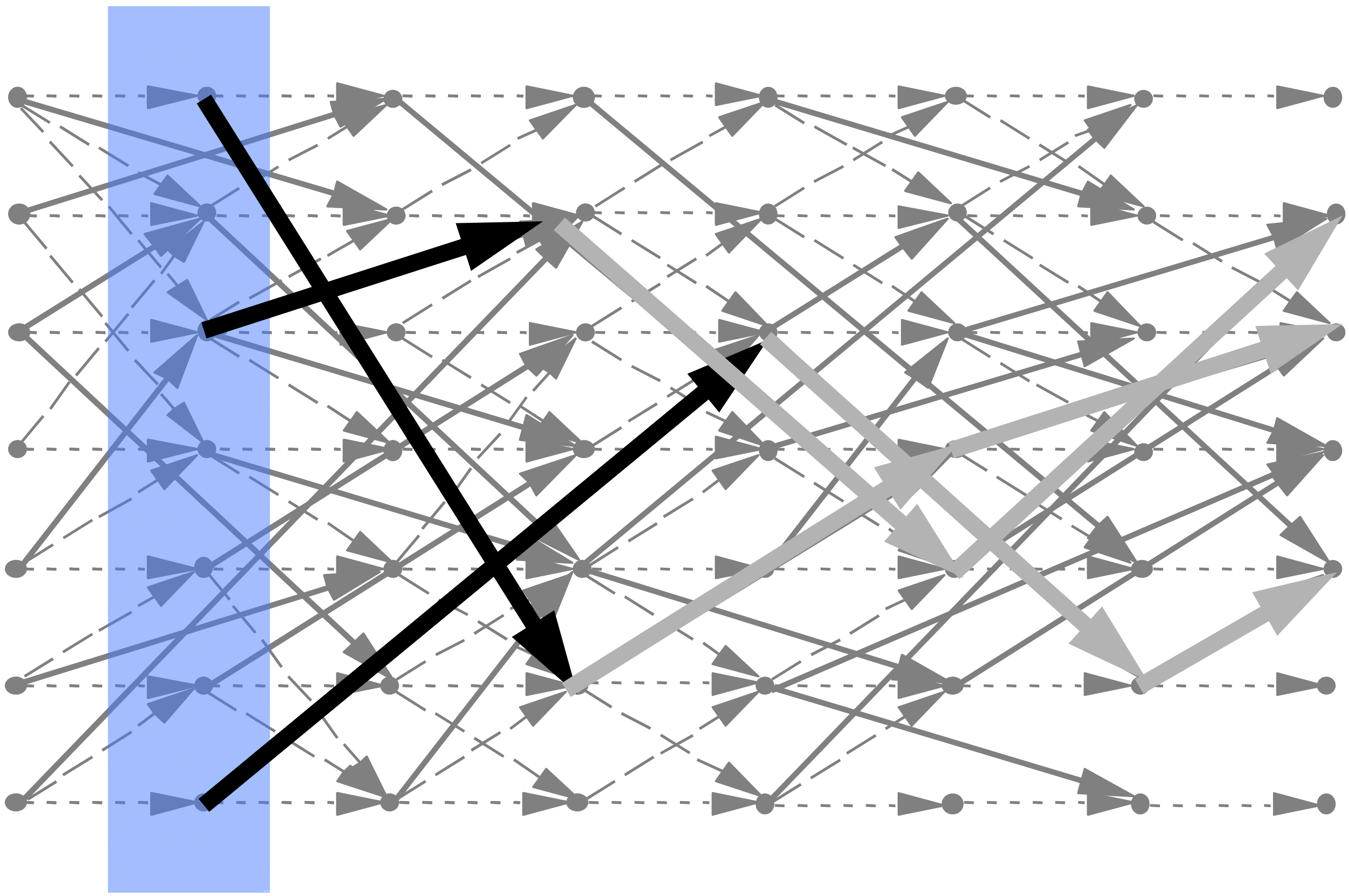
What is a policy?

- Following a lookahead policy



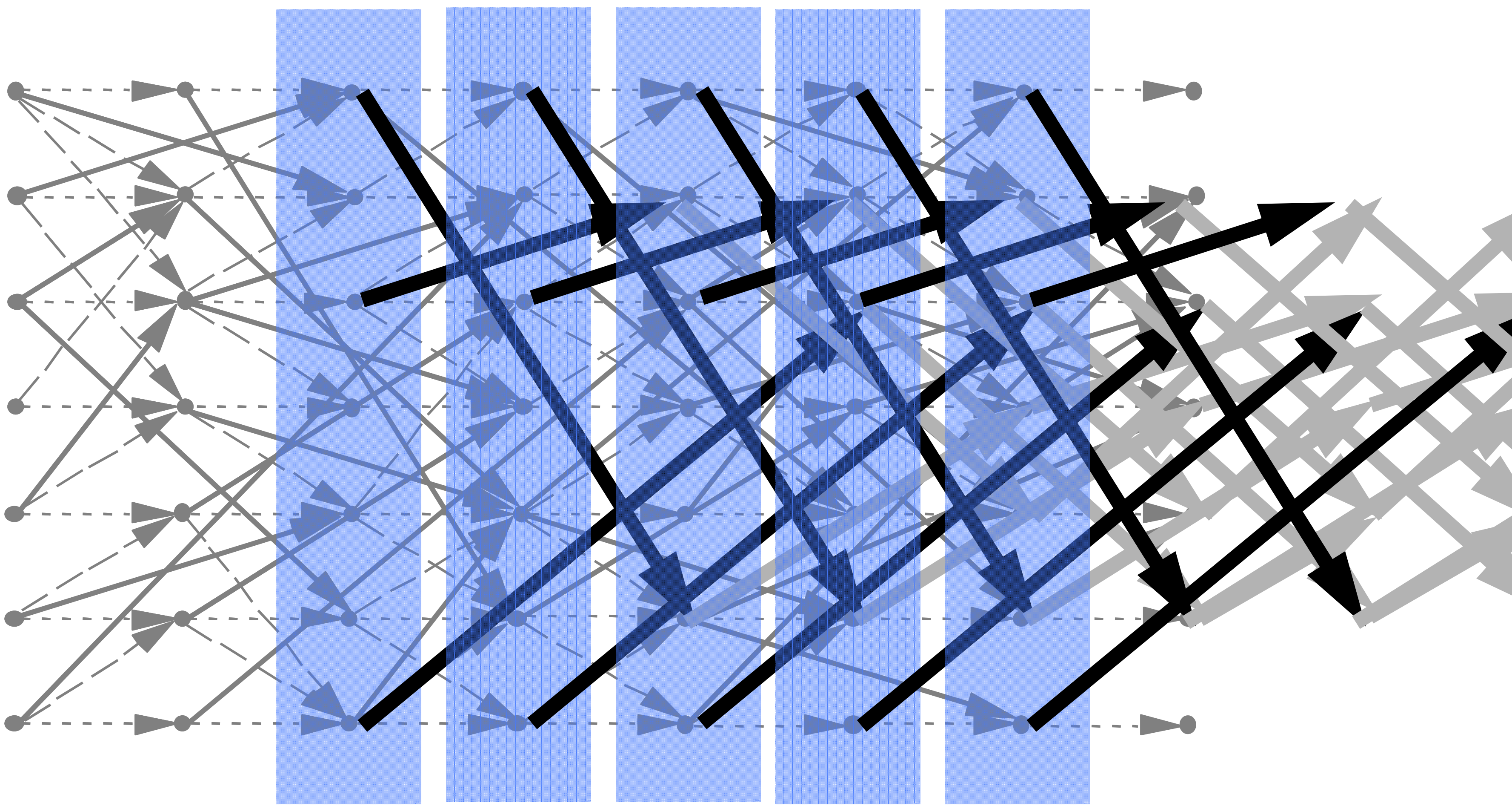
What is a policy?

- Following a lookahead policy



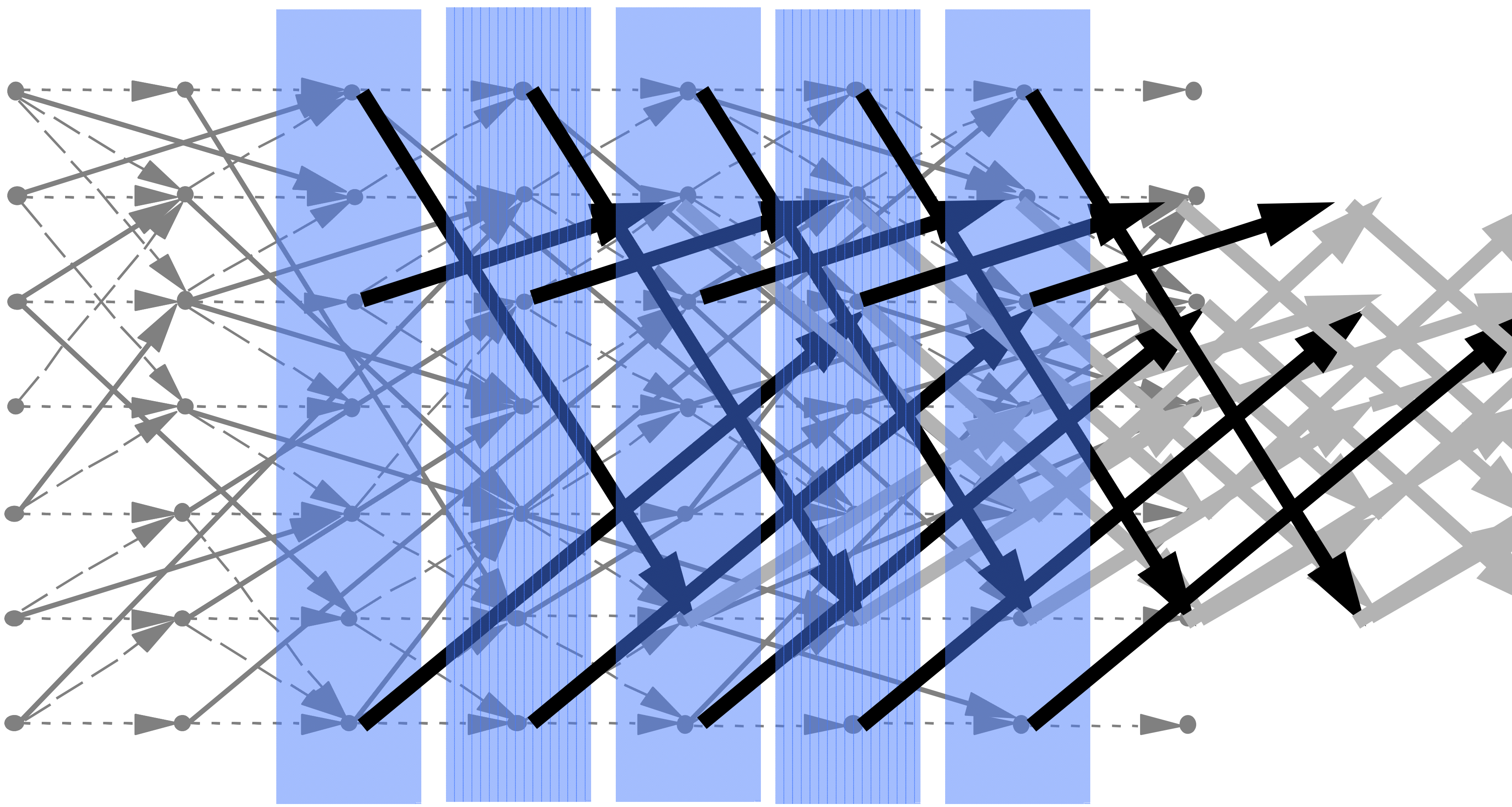
What is a policy?

- Following a lookahead policy



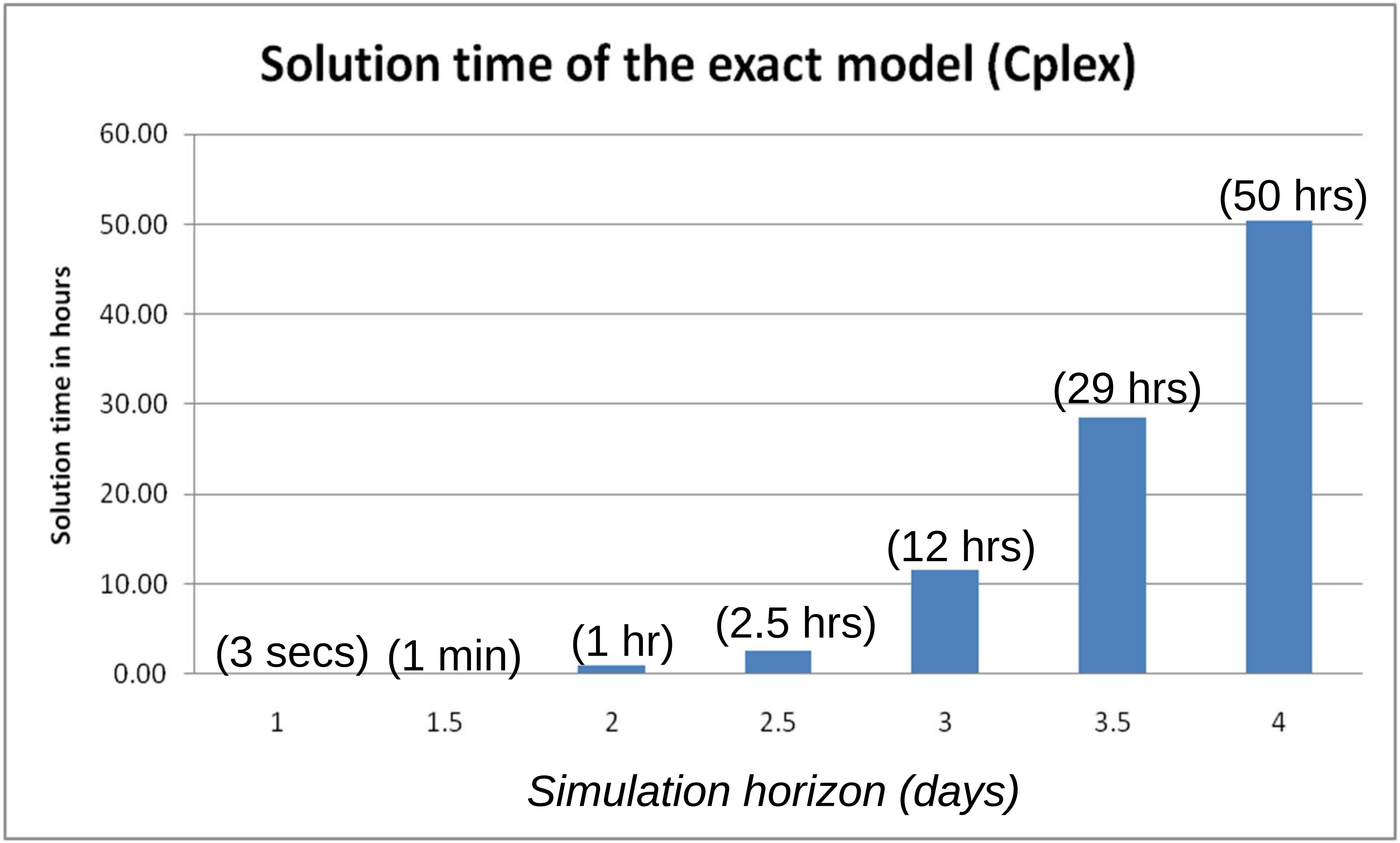
What is a policy?

- Following a lookahead policy



Lookahead policies

- The curse of time horizons



Lookahead policies

- From rolling horizon to stochastic programming to dynamic programming:

$$X^M(S_t) = \arg\min_{x_t, x_{t+1}, \ldots, x_{t+T}} C(S_t, x_t) + \underbrace{\sum_{t'=t+1}^{T} \gamma^{t'-t} C(S_{t'}, x_{t'})}$$

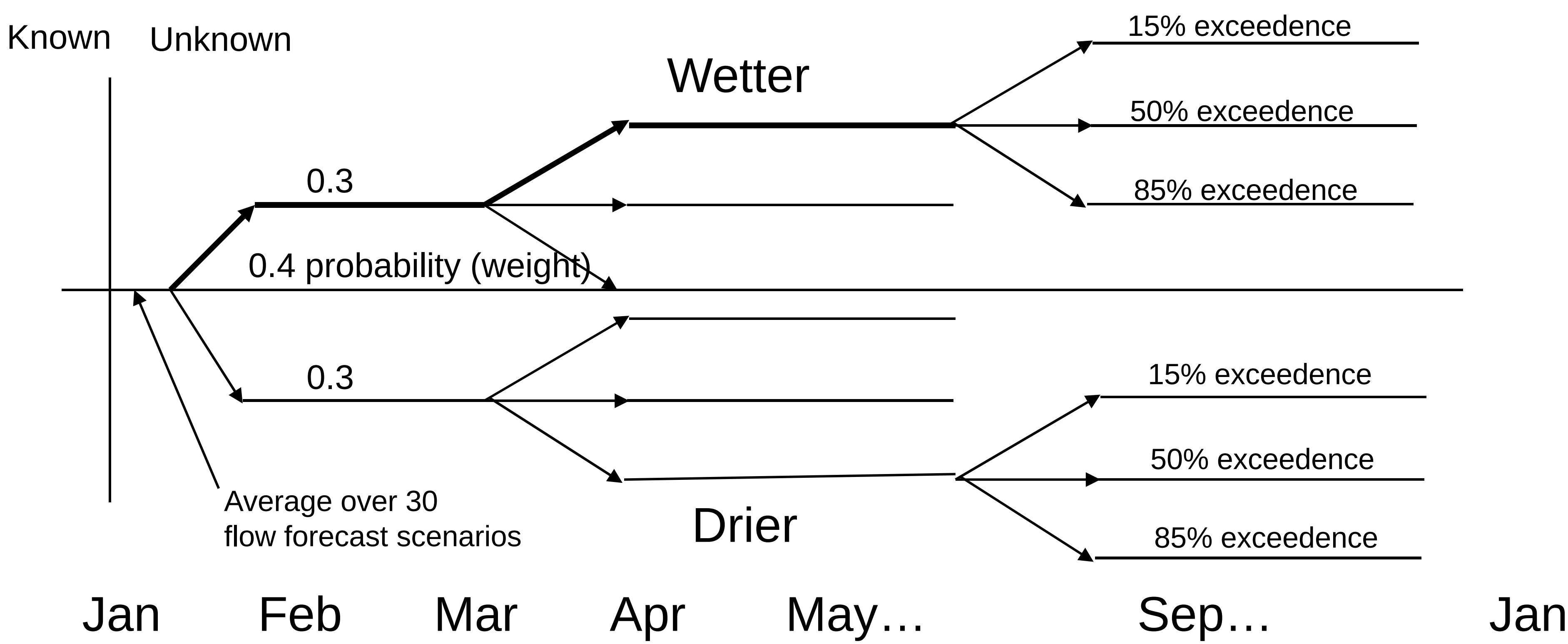
$$X^M(S_t) = \arg\min_{x_t, \left(x_{t+1}, \ldots, x_{t+T}\right)(\omega)} C(S_t, x_t) + \sum_{\omega \in \Omega} p(\omega) \underbrace{\sum_{t'=t+1}^{T} \gamma^{t'-t} C(S_{t'}(\omega), x_{t'}(\omega))}_{V_{t+1}\left(S_{t+1}(S_t, x_t, W_t(\omega))\right)}$$

$$X^M(S_t) = \arg\min_{x_t} \left(C(S_t, x_t) + \gamma E \bar{V}_{t+1}(S_{t+1}) \right)$$

Lookahead policies

- When do you use each policy?
  - » Use a lookahead policy if the behavior of $\sum_{t'=t+1}^{T} \gamma^{t'-t} C(S_{t'}, x_{t'})$ is a complex function of x_t .

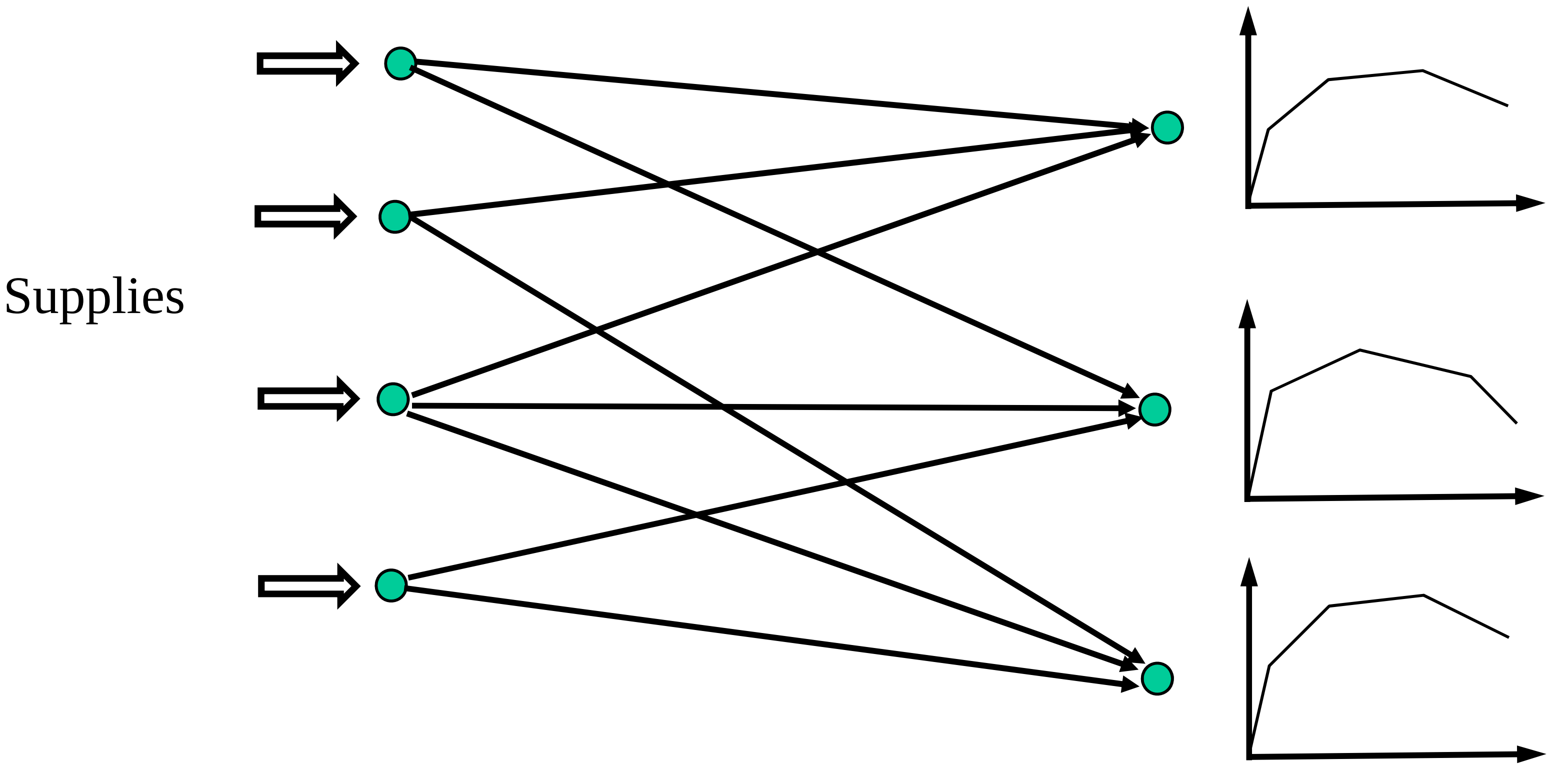
Stochastic programming for hydroelectric power planning (Morton et al)



Lookahead policies

- When do you use each policy?
  - » Use a value function approximation when the relationship is simpler:

$$\min_{x_t}\left(c_t x_t + \gamma E\bar{V}_{t+1}(S_{t+1})\right)$$



Outline

- Value function approximations

Value function approximations

- We can find the best decision by solving Bellman's equation:

$$V_t(S_t) = \min_{x \in \mathcal{X}} C_t(S_t, x_t) + E\left\{V_{t+1}(S_{t+1}) \,|\, S_t\right\}$$

We can find the value of being in state S_t at time *t*.

Given the value of being in state S_{t+1} at time *t+1*….

» We find the value of being in each state by stepping backward through time.

Value function approximations

- The challenge of dynamic programming:

$$V_t(S_t) = \min_{x \in \mathcal{X}} \left(C_t(S_t, x_t) + E\left\{ V_{t+1}(S_{t+1}) \mid S_t \right\} \right)$$

Value function approximations

- The computational challenge:

$$V_t(S_t) = \min_{x \in \mathcal{X}} \left(C_t(S_t, x_t) + E\left\{ V_{t+1}(S_{t+1}) \mid S_t \right\} \right)$$

How do we find $V_{t+1}(S_{t+1})$?

How do we compute the expectation?

How do we find the optimal solution?

Value function approximations

- Classical ADP
  - » Most applications of ADP focus on the challenge of handling multidimensional state variables
  - » Start with

$$V_t(S_t) = \min_{x \in \mathcal{X}} \Big(C_t(S_t, x_t) + E\left\{ V_{t+1}(S_{t+1}) \mid S_t \right\} \Big)$$

  - » Now replace the value function with some sort of approximation

$$V_{t+1}(S_{t+1}) \approx \overline{V}_{t+1}(S_{t+1})$$

Value function approximations

- Approximating the value function:
  - » We might approximate the value function using a simple polynomial

$$\bar{V}_t(S_t \mid \theta) = \theta_0 + \theta_1 S_t + \theta_2 S_t^2$$

  - » .. or a more complicated one:

$$\bar{V}_t(S_t \mid \theta) = \theta_0 + \theta_1 S_t + \theta_2 S_t^2 + \theta_3 \ln\left(S_t\right) + \theta_4 \sin(S_t)$$

  - » Most of the time, we write this in the form of *basis functions:*

$$\bar{V}_t(S_t \mid \theta) = \sum_f \theta_f \phi_f(S_t)$$

Value function approximations

- But we are not out of the woods…
  - » Assume we have an approximate value function.
  - » We still have to solve a problem that looks like

$$V_t(S_t) = \min_{x \in \mathcal{X}} \left(C_t(S_t, x_t) + E \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_{t+1}) \right)$$

  - » This means we still have to deal with an optimization problem (might be a linear, nonlinear or integer program) with an expectation.

The post-decision state

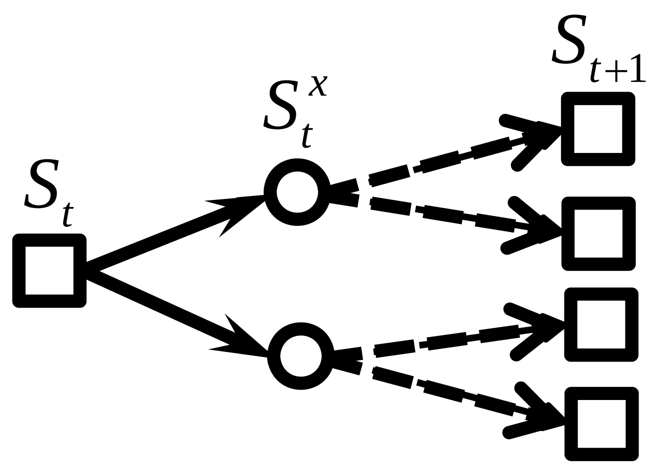
- New concept:
  - » The "pre-decision" state variable:
    - $S_t =$ The information required to make a decision x_t
    - Same as a "decision node" in a decision tree.
  - » The "post-decision" state variable:
    - $S_t^x =$ The state of what we know immediately after we make a decision.
    - Same as an "outcome node" in a decision tree.
    - Also known as:
      - – "Afterstate" variable (Sutton and Barto)
      - – "End-of-period state" (Judd)

The post-decision state

- Representations of the post-decision state:
  - » Decision trees:

$$S_t^x = S^{M,x}\left(S_t, x_t\right)$$

$$S_{t+1} = S^{M,W}\left(S_t^x, W_{t+1}\right)$$



  - » Q-learning:

$$S_t^x = \left(S_t, x_t\right)$$

State-action pair

  - » Transition function with expectation:

$$S_t^x = S^M\left(S_t, x_t, \bar{W}_{t,t+1}\right)$$

$\bar{W}_{t,t+1}$ = Forecast of W_{t+1} at time t.

The post-decision state

- An inventory problem:
  - » Our basic inventory equation:

$$R_{t+1} = \max\left\{0, R_t + x_t - \hat{D}_{t+1}\right\}$$

  - » where

$$R_t = \text{Inventory on hand at time } t$$

$$x_t = \text{Amount ordered}$$

$$\hat{D}_{t+1} = \text{Demand in next time period}$$

  - » Using pre- and post-decision states:

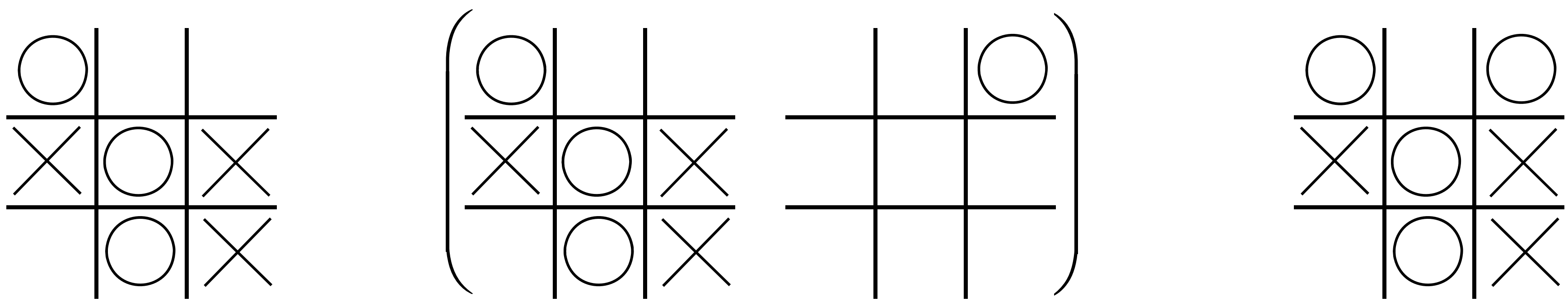
$$R_t^x = R_t + x_t \qquad \text{Pre- to post-}$$

$$R_{t+1} = \max\left\{R_t^x - \hat{D}_{t+1}\right\} \qquad \text{Post- to pre-}$$

The post-decision state

- Pre-decision, state-action, and post-decision

Pre-decision state | State | Action | Post-decision state



3^9 states | $3^9 \times 9$ state-action pairs | 3^9 states

The post-decision state

- Pre- and post-decision attributes for a trucking problem:

$$
\begin{array}{ccccccc}
 & t=40 & t=40 & t=40 & & t=50 \\
 & \text{Pre-decision} & \text{Decision} & \text{Post-decision} & & \text{Pre-decision} \\
\begin{pmatrix} \text{City} \\ \text{ETA} \\ \text{Equip} \end{pmatrix} &
\begin{pmatrix} \text{Dallas} \\ 41.2 \\ \text{Good} \end{pmatrix} &
\begin{pmatrix} \text{Chicago} \\ - \\ - \end{pmatrix} &
\begin{pmatrix} \text{Chicago} \\ 54.7 \\ \text{Good} \end{pmatrix} &
\cdots &
\begin{pmatrix} \text{Chicago} \\ 56.2 \\ \text{Repair} \end{pmatrix}
\end{array}
$$

Approximate value iteration

Step 1: Start with a pre-decision state S_t^n

Step 2: Solve the deterministic optimization using an approximate value function:

$$\hat{v}_t^n = \min_x \left(C_t(S_t^n, x_t) + \bar{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$$

to obtain x_t^n.

Deterministic optimization

Step 3: Update the value function approximation

$$\bar{V}_{t-1}^n(S_{t-1}^{x,n}) = (1-\alpha_{n-1})\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}\hat{v}_t^n$$

Recursive statistics

Step 4: Obtain Monte Carlo sample of $W_t(\omega^n)$ and compute the next pre-decision state:

$$S_{t+1}^n = S^M(S_t^n, x_t^n, W_{t+1}(\omega^n))$$

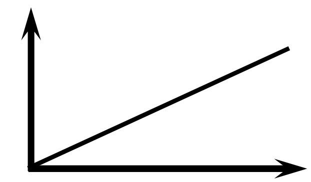
Simulation

Step 5: Return to step 1.

Approximating value functions

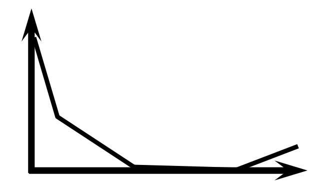
- Approximations for resource allocation problems
  - » Linear (in the resource state):

$$\bar{V}_t(R_t^x) = \sum_{a \in \mathcal{A}} \bar{v}_{ta} \cdot R_{ta}^x$$



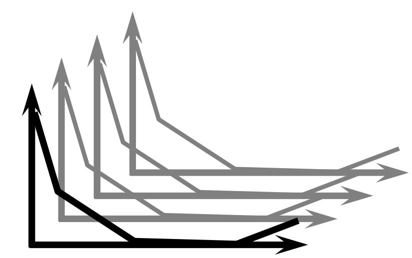
  - » Piecewise linear, separable:

$$\bar{V}_t(R_t^x) = \sum_{a \in \mathcal{A}} \bar{V}_{ta}(R_{ta}^x)$$



  - » Indexed PWL separable:

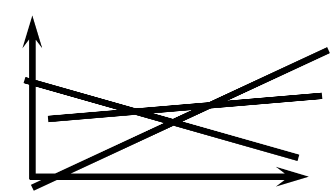
$$\bar{V}_t(R_t^x) = \sum_{a \in \mathcal{A}} \bar{V}_{ta}\left(R_{ta}^x \mid \text{"state of the world"}\right)$$



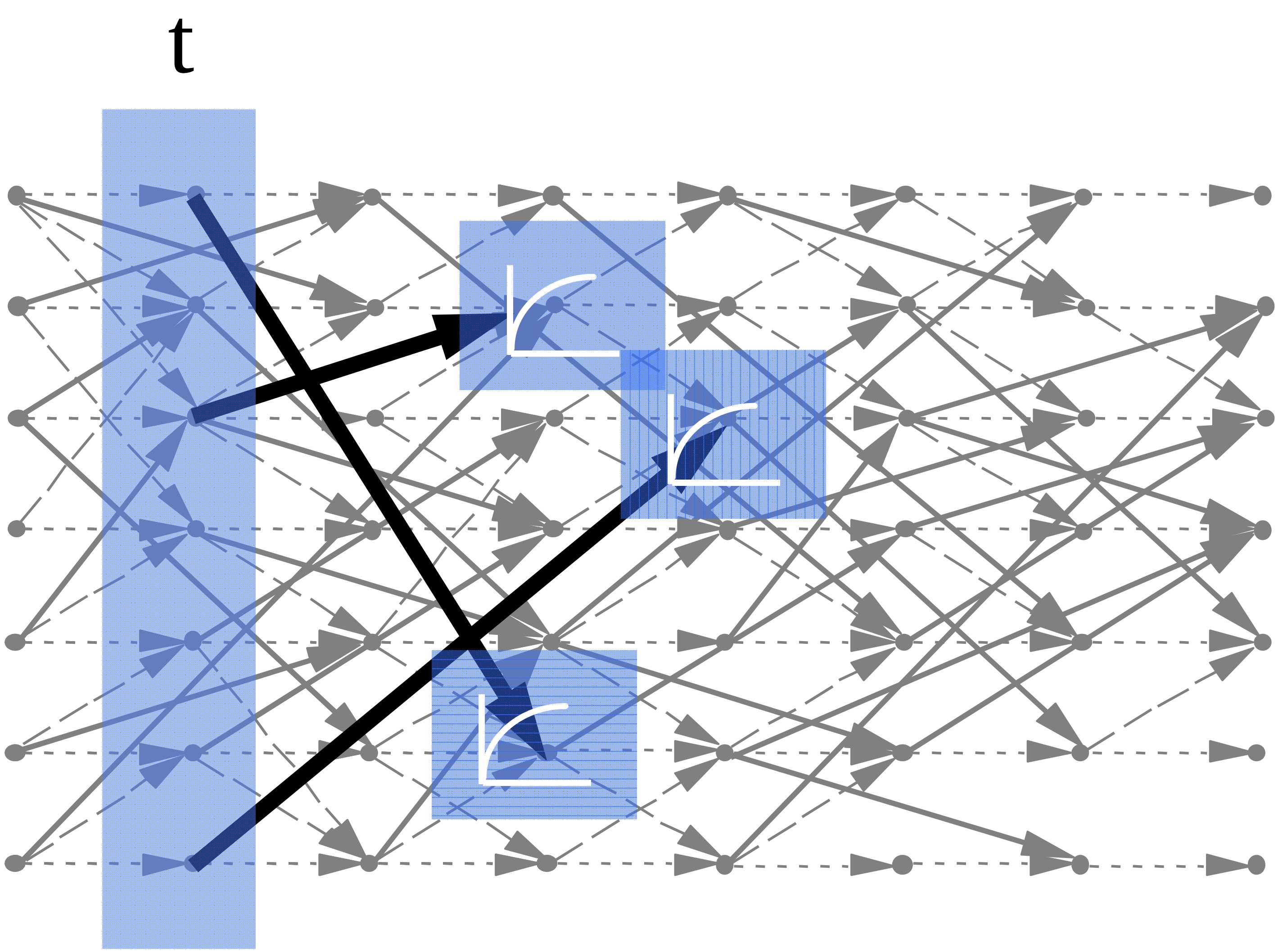
  - » Benders cuts

$$\min cx + z$$

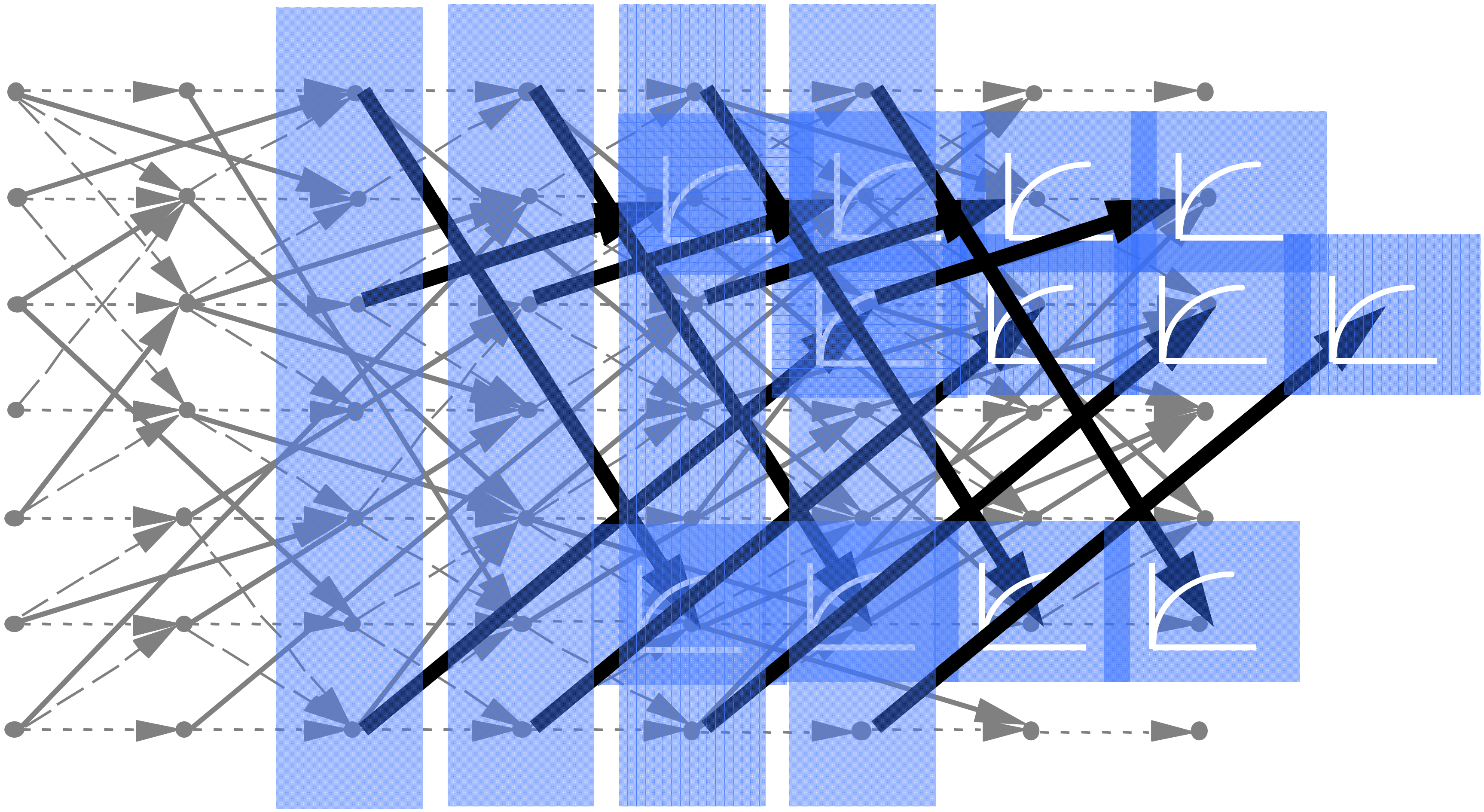
$$z \geq a_i + b_i x$$



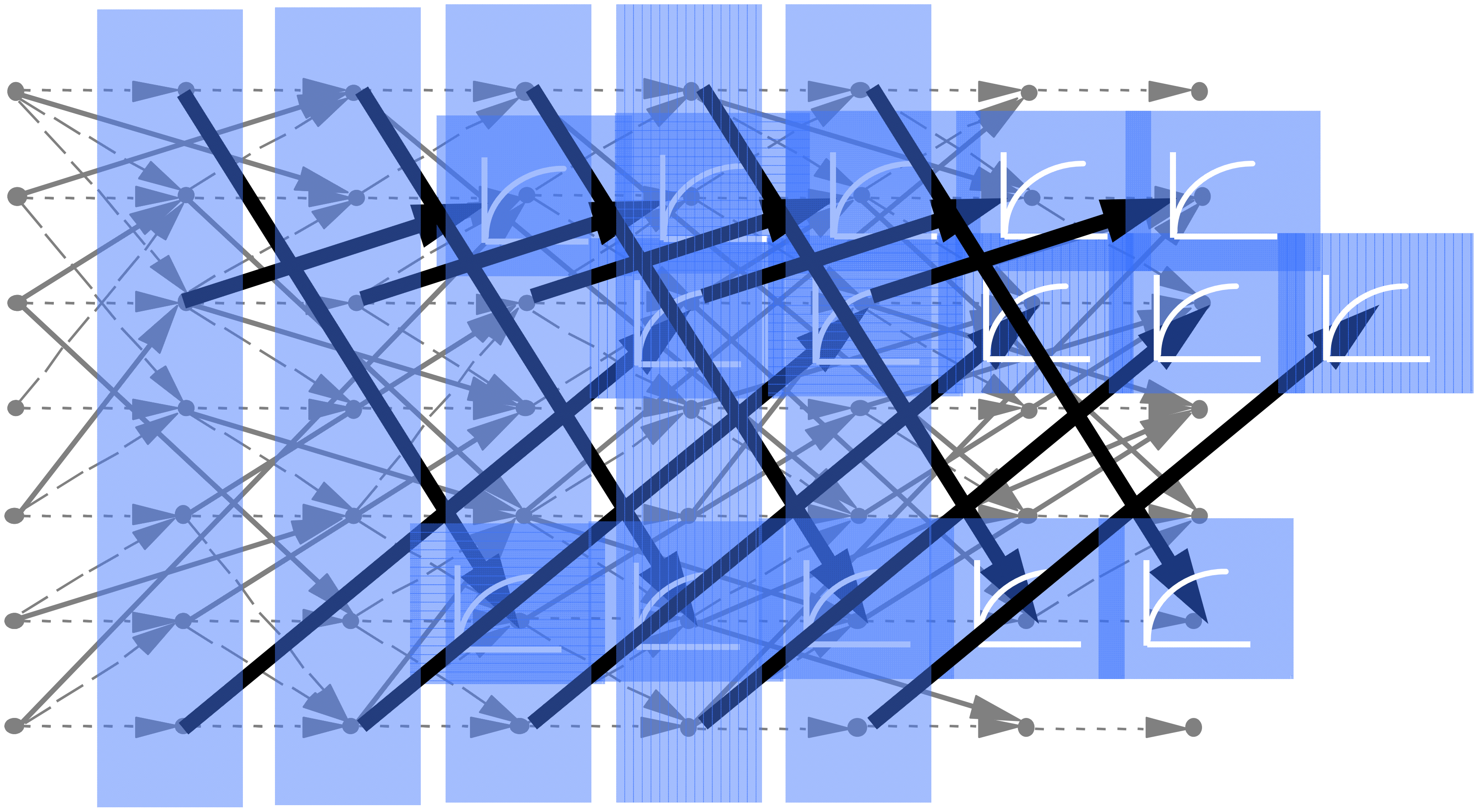
Approximate dynamic programming



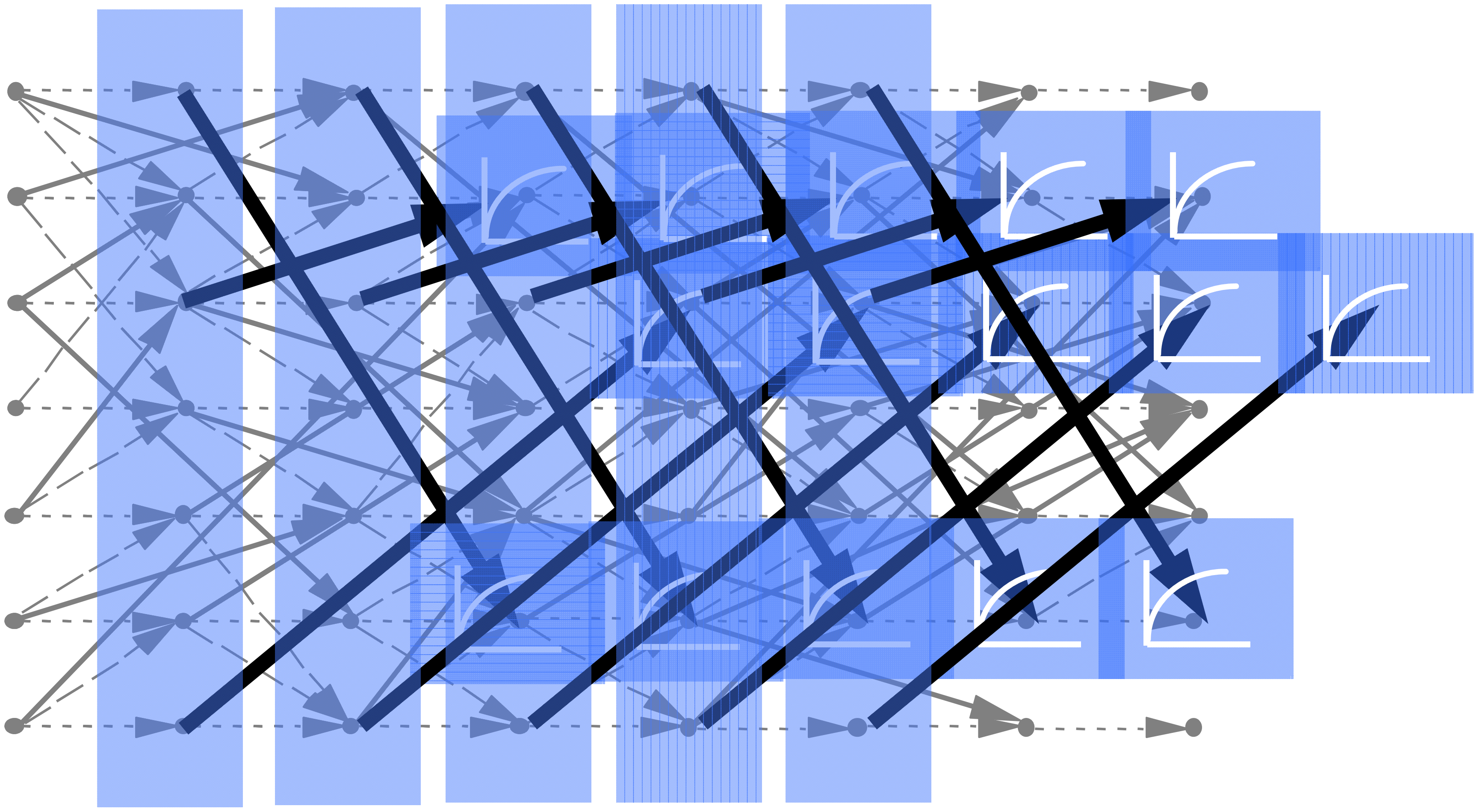
Approximate dynamic programming



Approximate dynamic programming

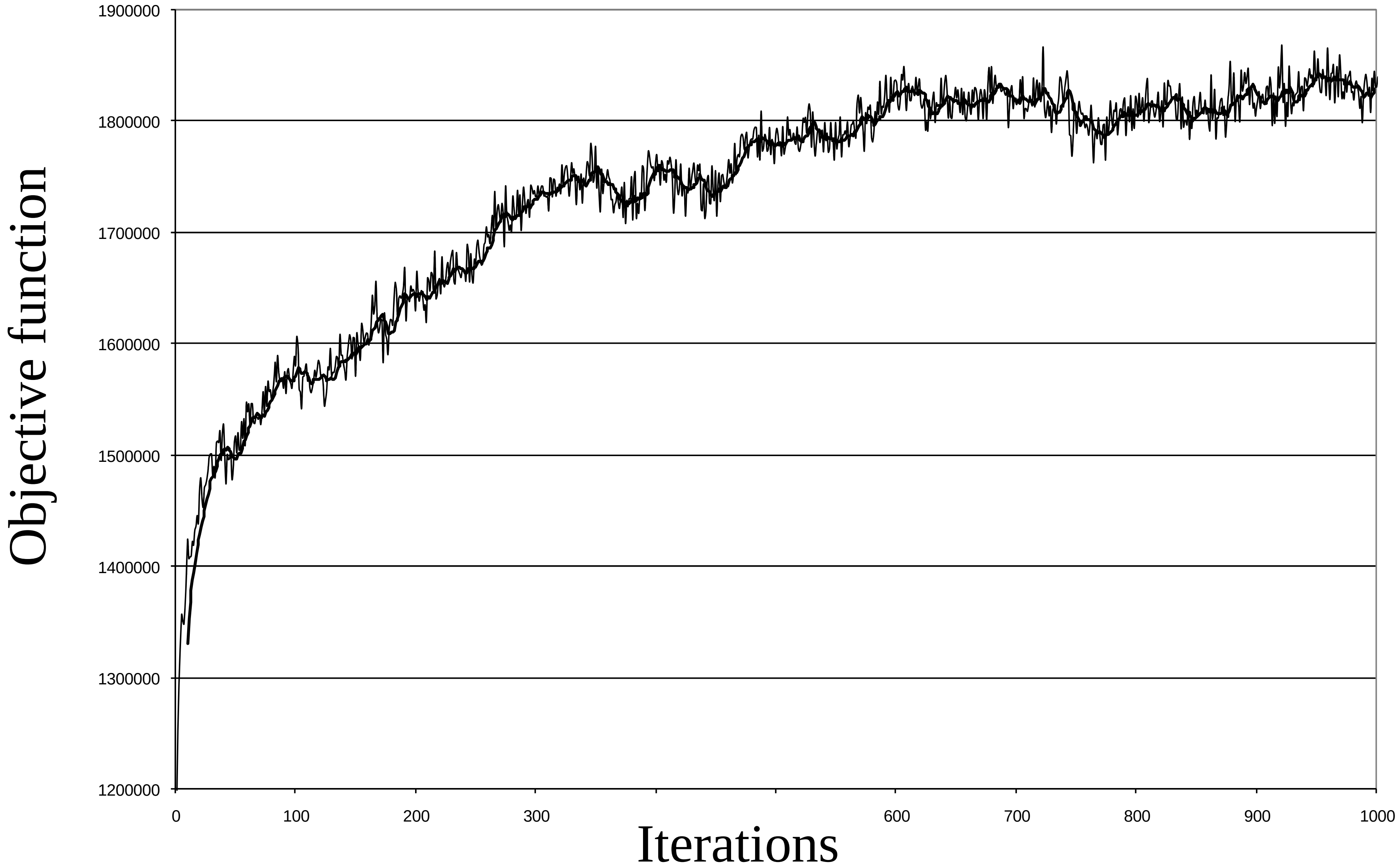


Approximate dynamic programming



Approximate dynamic programming

- With luck, your objective function improves



Approximate value iteration

- Features
  - » Scales to ultra large scale applications (but, “single layer”).
  - » Handles high-dimensional decision vectors and complex state variables.
  - » Fast, stable convergence.
  - » Handles virtually any type of uncertainty, and complex physical processes.
  - » Near-optimal solutions.

Approximate value iteration

Step 1: Start with a pre-decision state S_t^n

Step 2: Solve the deterministic optimization using an approximate value function:

$$\hat{v}_t^n = \min_x \left(C_t(S_t^n, x_t) + \bar{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$$

to obtain x_t^n.

Step 3: Update the value function approximation

$$\bar{V}_{t-1}^n(S_{t-1}^{x,n}) = (1-\alpha_{n-1})\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}\hat{v}_t^n$$

Step 4: Obtain Monte Carlo sample of $W_t(\omega^n)$ and compute the next pre-decision state:

$$S_{t+1}^n = S^M(S_t^n, x_t^n, W_{t+1}(\omega^n))$$

Step 5: Return to step 1.

Deterministic optimization

Recursive statistics

Simulation

“on policy learning”

Approximate value iteration

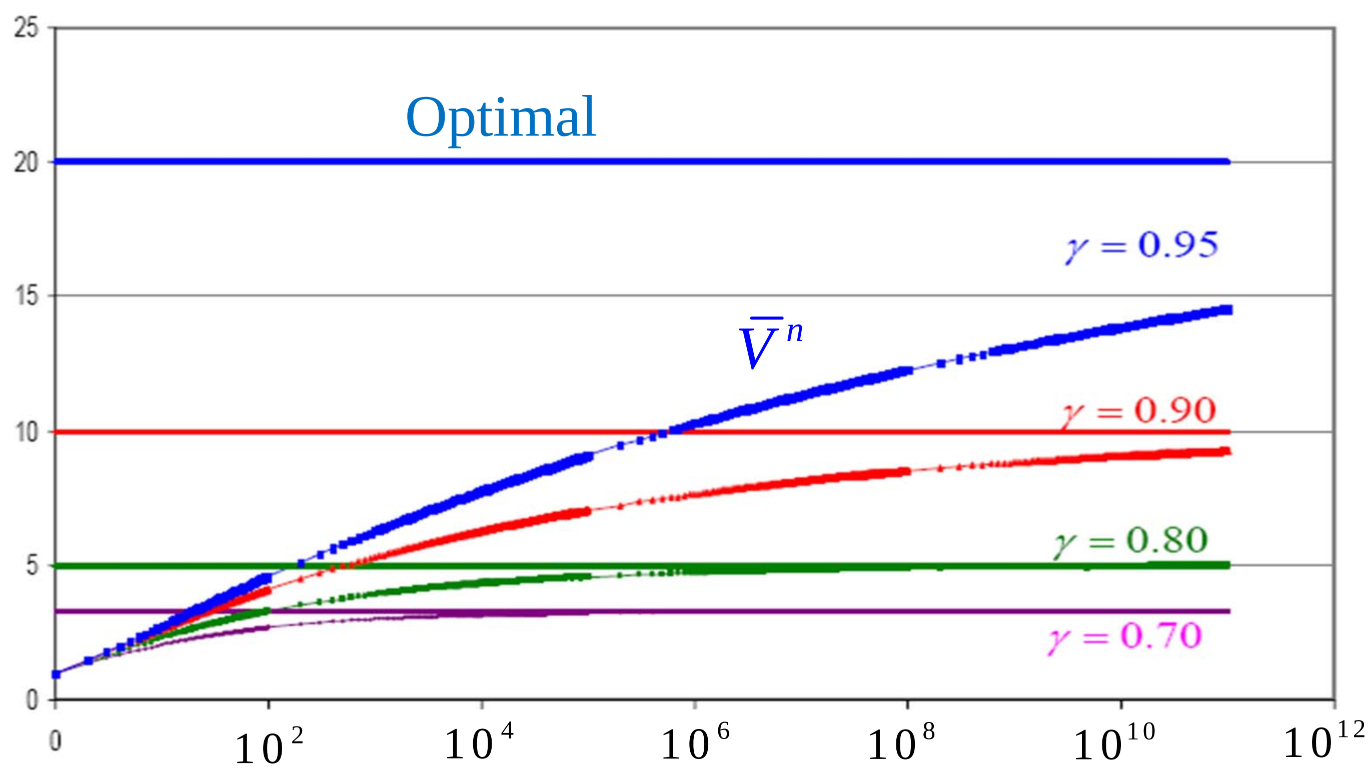
- The bad news:
  - » For one problem, we can *prove* it converges and *prove* that it converge so slowly as to be absolutely useless (e.g. 10^{20} Iterations).
  - » Provably convergent algorithms running on small problems can be shown to *diverge* initially, with extremely slow convergence.
  - » The attraction of "solving" the curse of dimensionality using statistical models ("basis functions") is illusory – the simplicity of approximating a complex value function is replaced with severe convergence issues.

Simple problems can be hard

- Single state, single action
  - » Approximate value iteration

$$\hat{v}^n = \hat{C}^n + \gamma \bar{V}^{n-1}$$ $\hat{C}^n$ sampled or observed

$$\bar{V}^n = (1-\alpha_{n-1})\bar{V}^{n-1} + \alpha_{n-1}\hat{v}^n$$ Stepsize $\alpha_{n-1} = \frac{1}{n}$

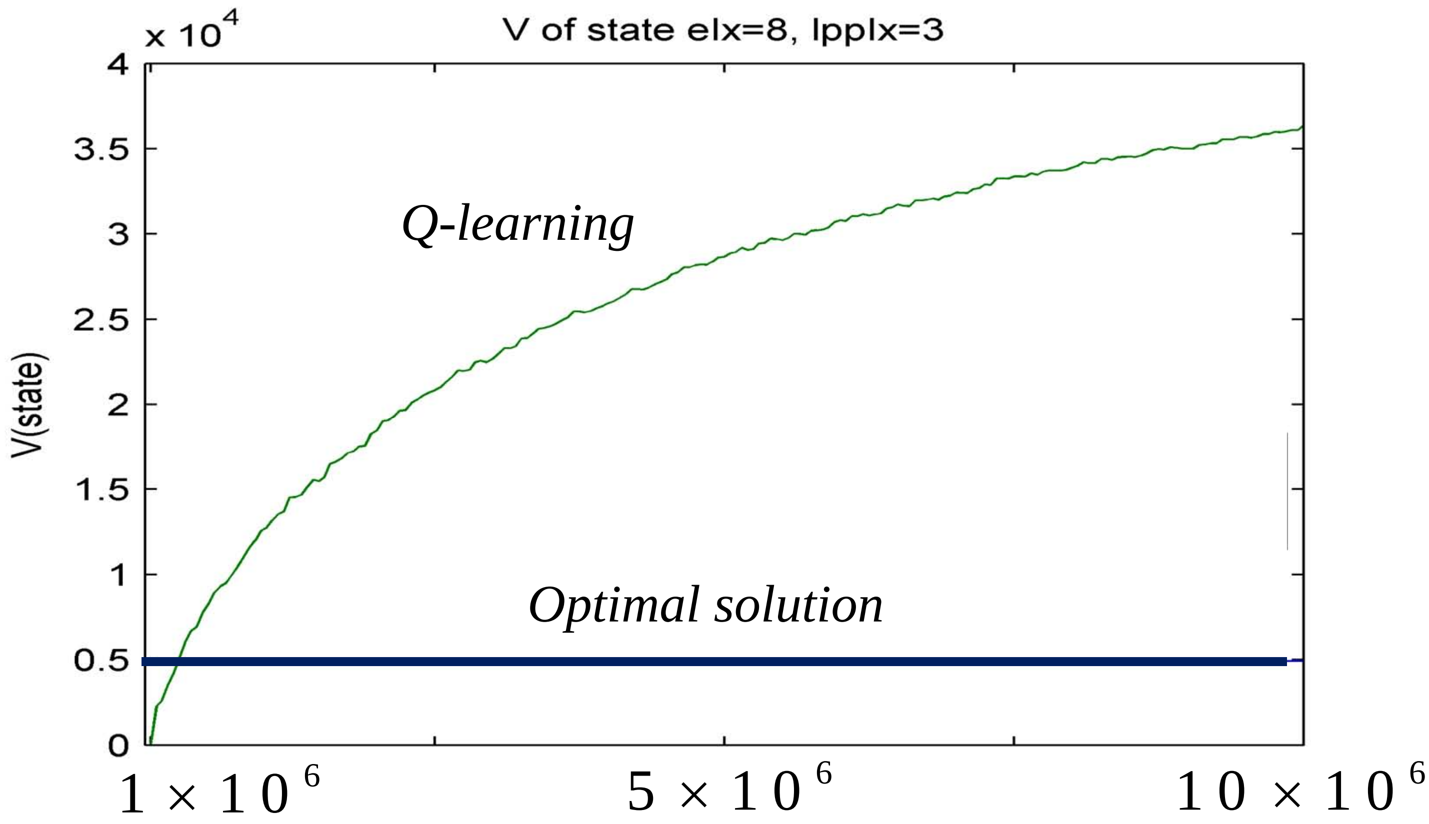


Simple problems can be hard

- Small state-action space, noisy observations
  - » Q-learning, optimized stepsize

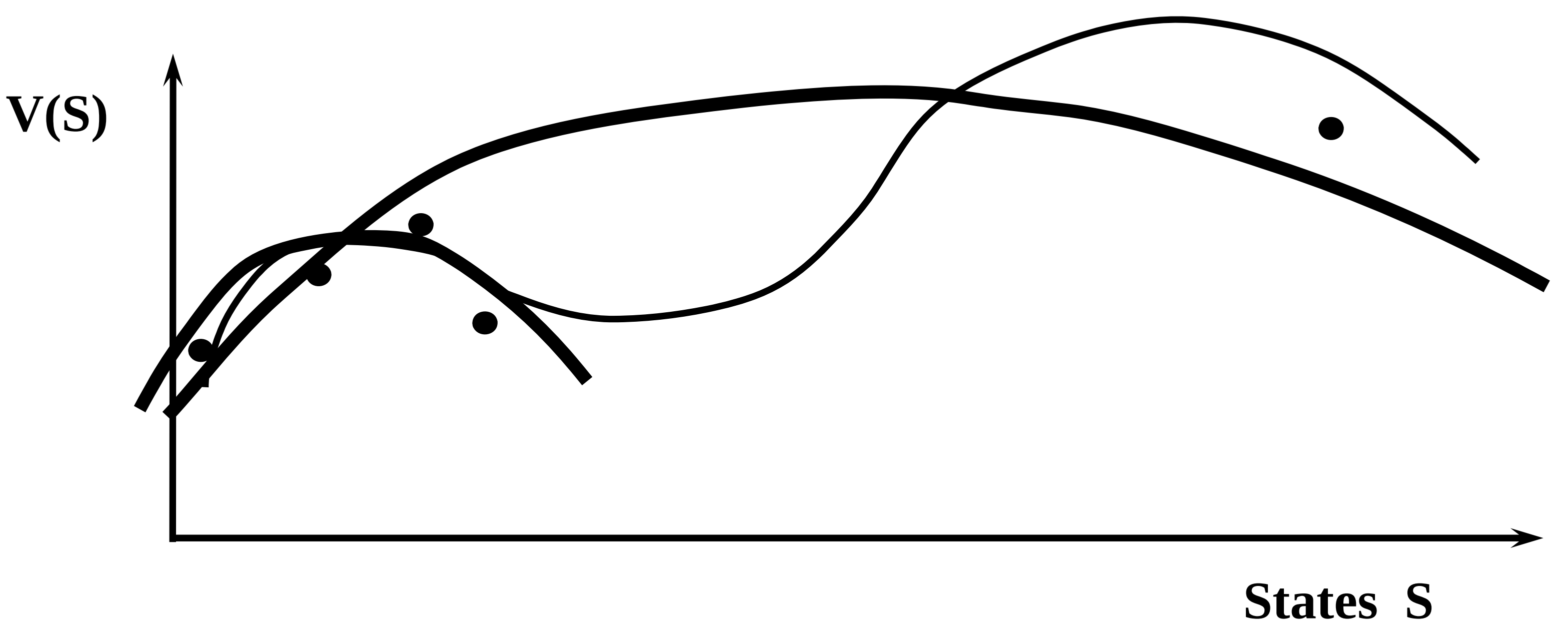
$$\hat{q}^n(s,a) = \hat{C}^n(s,a) + \gamma \max_{a'} \bar{Q}^{n-1}(s',a')$$

$$\bar{Q}^n(s,a) = (1-\alpha_{n-1})\bar{Q}^{n-1}(s,a) + \alpha_{n-1}\hat{q}^n(s,a)$$



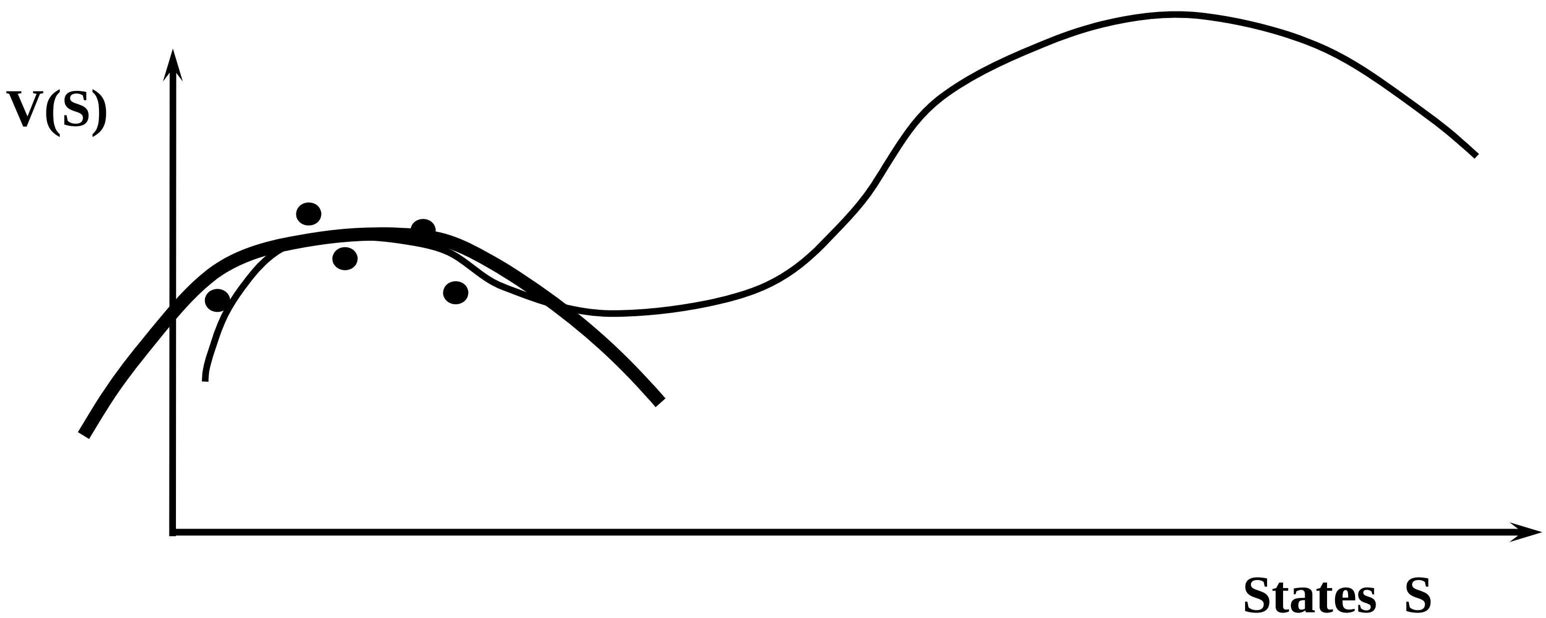
Parametric approximations

- Fitting basis functions
  - » If the basis functions are not perfect, the fit depends on the states we visit. If we visit the wrong states, we may get a terrible fit.



Parametric approximations

- Fitting basis functions
  - » If the basis functions are not perfect, the fit depends on the states we visit. If we visit the wrong states, we may get a terrible fit.



Approximate value iteration

Step 1: Start with a pre-decision state S_t^n

Step 2: Solve the deterministic optimization using an approximate value function:

$$\hat{v}_t^n = \min_x \left(C_t(S_t^n, x_t) + \bar{V}_t^{n-1}(S^{M,x}(S_t^n, x_t)) \right)$$

to obtain x_t^n.

Deterministic optimization

Step 3: Update the value function approximation

$$\bar{V}_{t-1}^n(S_{t-1}^{x,n}) = (1-\alpha_{n-1})\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}\hat{v}_t^n$$

Recursive statistics

Step 4: Obtain Monte Carlo sample of $W_t(\omega^n)$ and compute the next pre-decision state:

$$S_{t+1}^n = S^M(S_t^n, x_t^n, W_{t+1}(\omega^n))$$

Simulation

Step 5: Return to step 1.

Approximate policy iteration

Step 1: Start with a pre-decision state S_t^n

Step 2: Inner loop: Do for m=1,…,M:

Step 2a: Solve the deterministic optimization using an approximate value function:

$$\hat{v}^m = \min_x \left(C(S^m, x) + \bar{V}^{n-1}(S^{M,x}(S^m, x)) \right)$$

to obtain x^m.

Step 2b: Update the value function approximation

$$\bar{V}^{n-1,m}(S^{x,m}) = (1-\alpha_{m-1})\bar{V}^{n-1,m-1}(S^{x,m}) + \alpha_{m-1}\hat{v}^m$$

Step 2c: Obtain Monte Carlo sample of $W(\omega^m)$ and compute the next pre-decision state:

$$S^{m+1} = S^M(S^m, x^m, W(\omega^m))$$

Step 3: Update $\bar{V}^n(S)$ using $\bar{V}^{n-1,M}(S)$ and return to step 1.

Approximate policy iteration

Step 1: Start with a pre-decision state S_t^n

Step 2: Inner loop: Do for m=1,…,M:

Step 2a: Solve the deterministic optimization using an approximate value function:

$$\hat{v}^m = \min_x \left(C(S^m, x) + \sum_f \theta_f^{n-1} \phi_f (S^M(S^m, x)) \right)$$

to obtain x^m.

Step 2b: Update the value function approximation

$$\bar{V}^{n-1,m}(S^{x,m}) = (1-\alpha_{m-1})\bar{V}^{n-1,m-1}(S^{x,m}) + \alpha_{m-1}\hat{v}^m$$

Step 2c: Obtain Monte Carlo sample of $W(\omega^m)$ and compute the next pre-decision state:

$$S^{m+1} = S^M(S^m, x^m, W(\omega^m))$$

Step 3: Update $\bar{V}^n(S)$ using $\bar{V}^{n-1,M}(S)$ and return to step 1.

Algorithms

- Classical approximate dynamic programming
  - » We can estimate the value of being in a state using

$$\hat{v}^n = \min_x \left(C(S_t^n, x) + \gamma \sum_f \theta_f^{n-1} \phi(S_t^x(S_t^n, x)) \right)$$

  - » Use recursive least squares to update θ^{n-1}.
  - » Our policy is then given by

$$X^\pi(S_t \mid \theta^n) = \arg\min_x \left(C(S_t, x) + \gamma \sum_f \theta_f^n \phi(S_t^x(S_t, x)) \right)$$

  - » This is known as ***Bellman error minimization.***
  - » Can scale to problems with thousands or millions of parameters, but can be highly unstable.

Algorithms

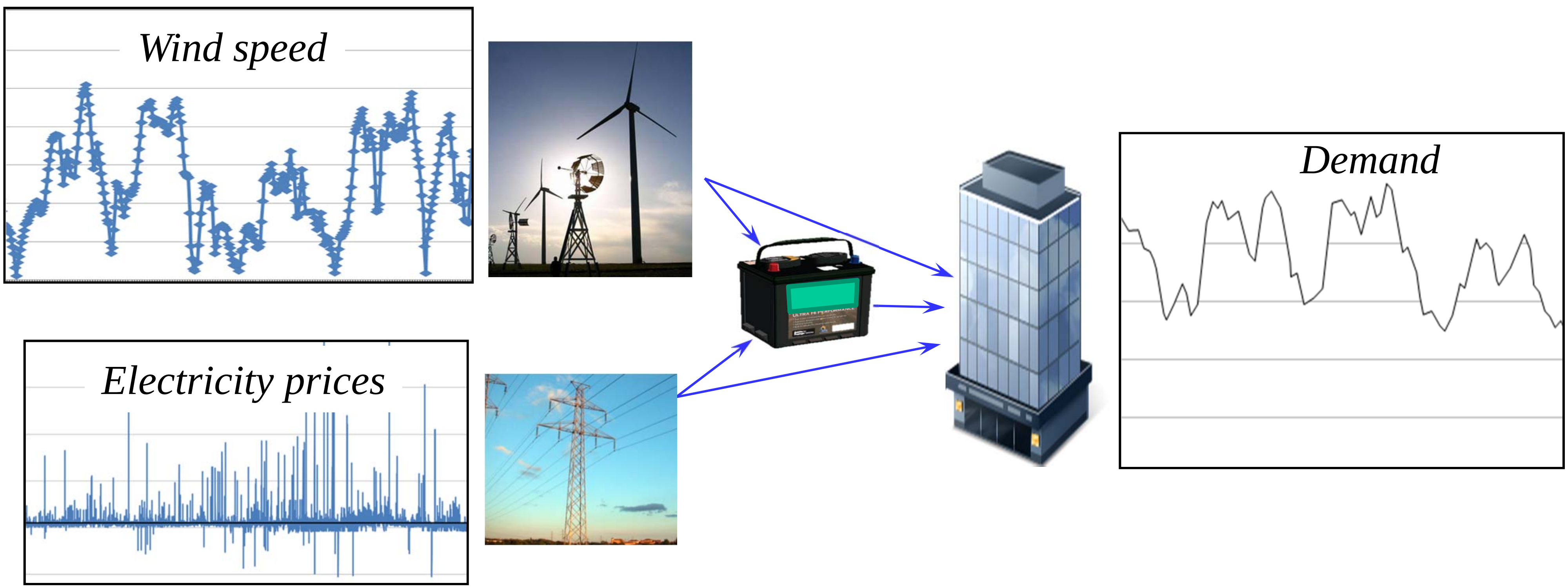
- But what if we simply view θ as a static design parameter?

$$\min_\theta \mathbb{E}F(\theta,W) = \mathbb{E}\sum_{t=0}^{T}\gamma^t C\left(S_t, X^\pi(S_t \mid \theta)\right)$$

  - » This is known as ***policy search***. It builds on classical fields such as
    - Stochastic search
    - Simulation optimization
  - » Very stable, but it is generally limited to problems with a much smaller number of parameters.

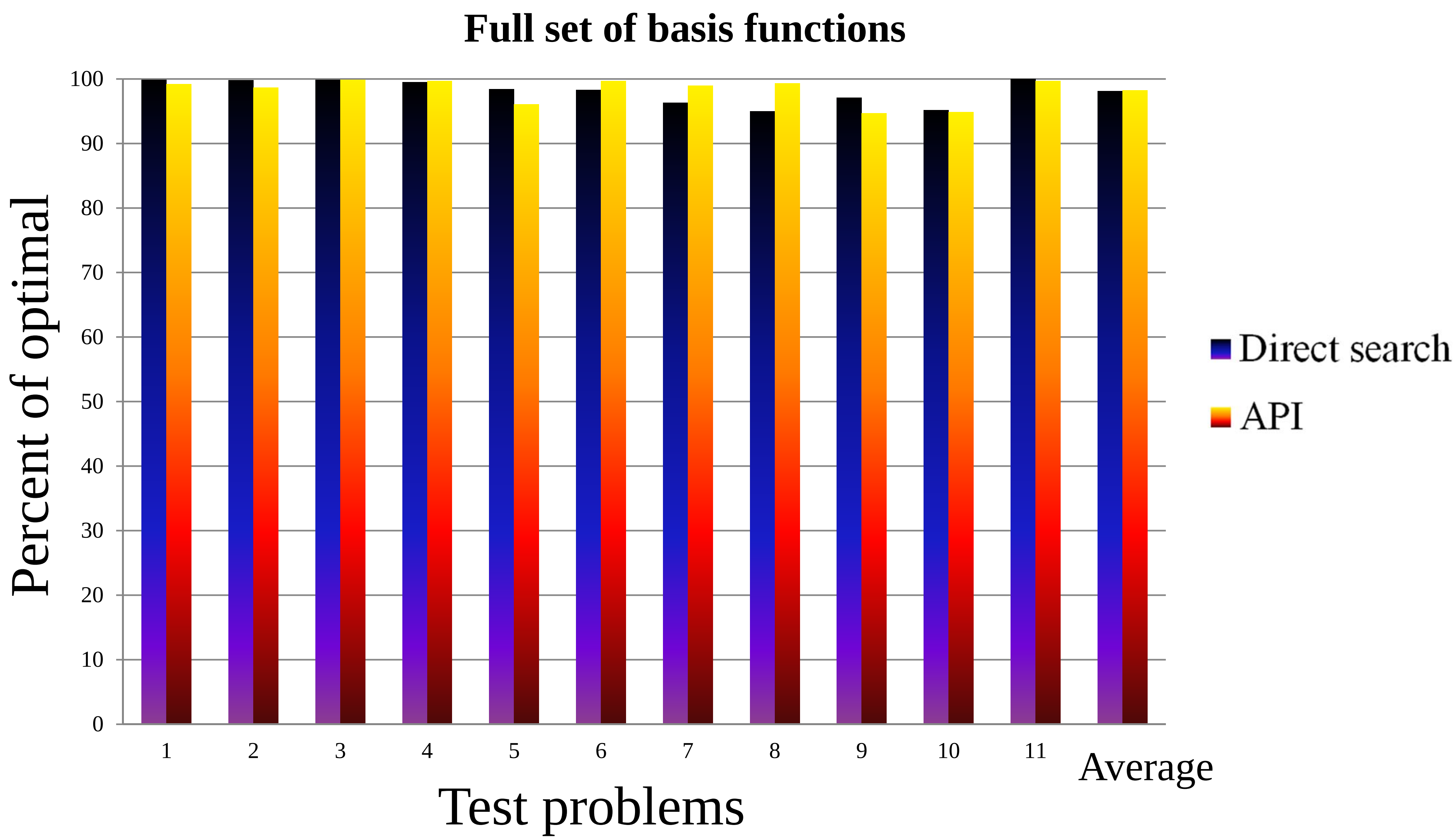
Managing uncertainty

- Wind, prices and loads….



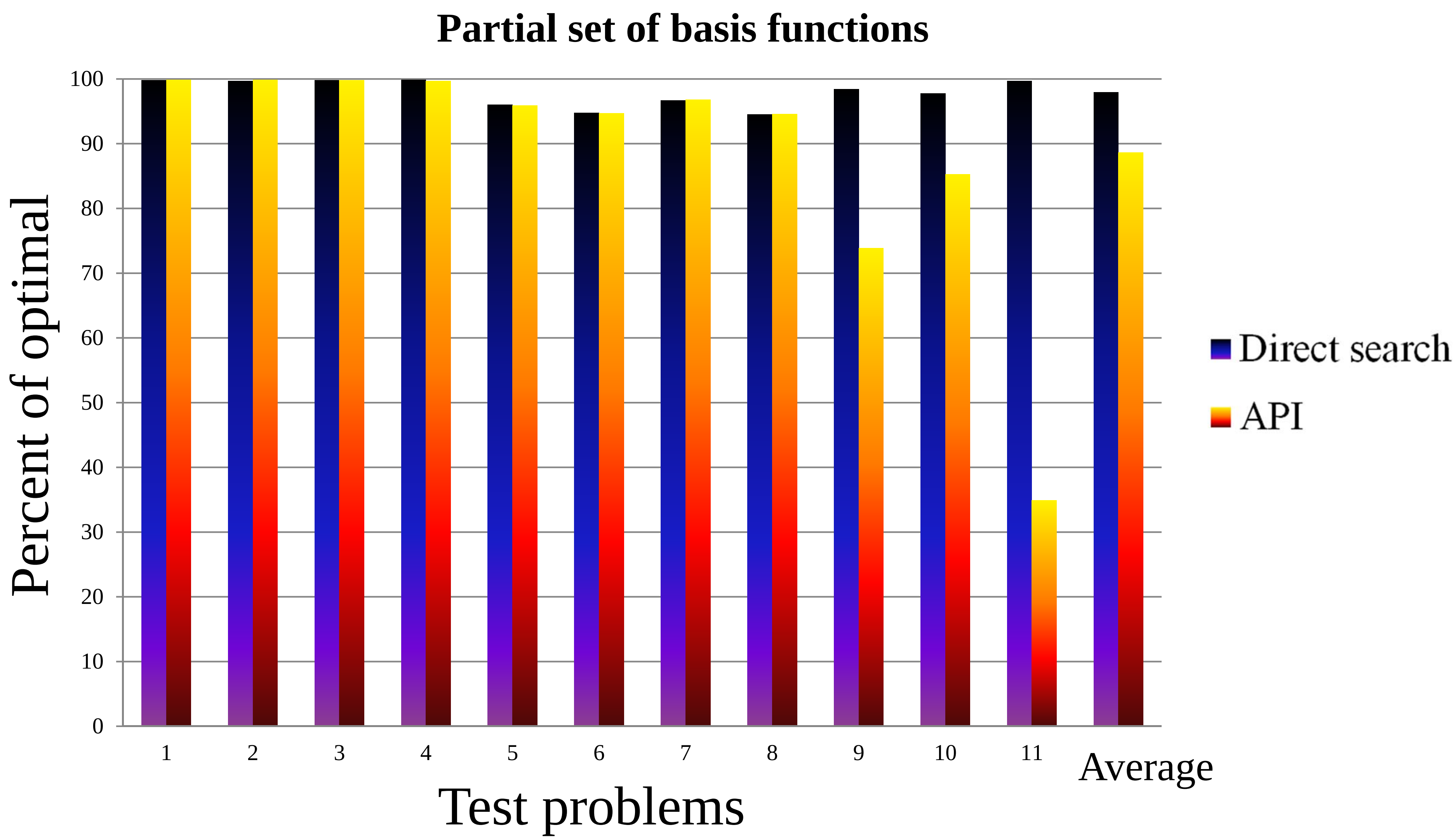
Algorithms

- Approximate policy iteration vs. policy search



Algorithms

- Approximate policy iteration vs. policy search

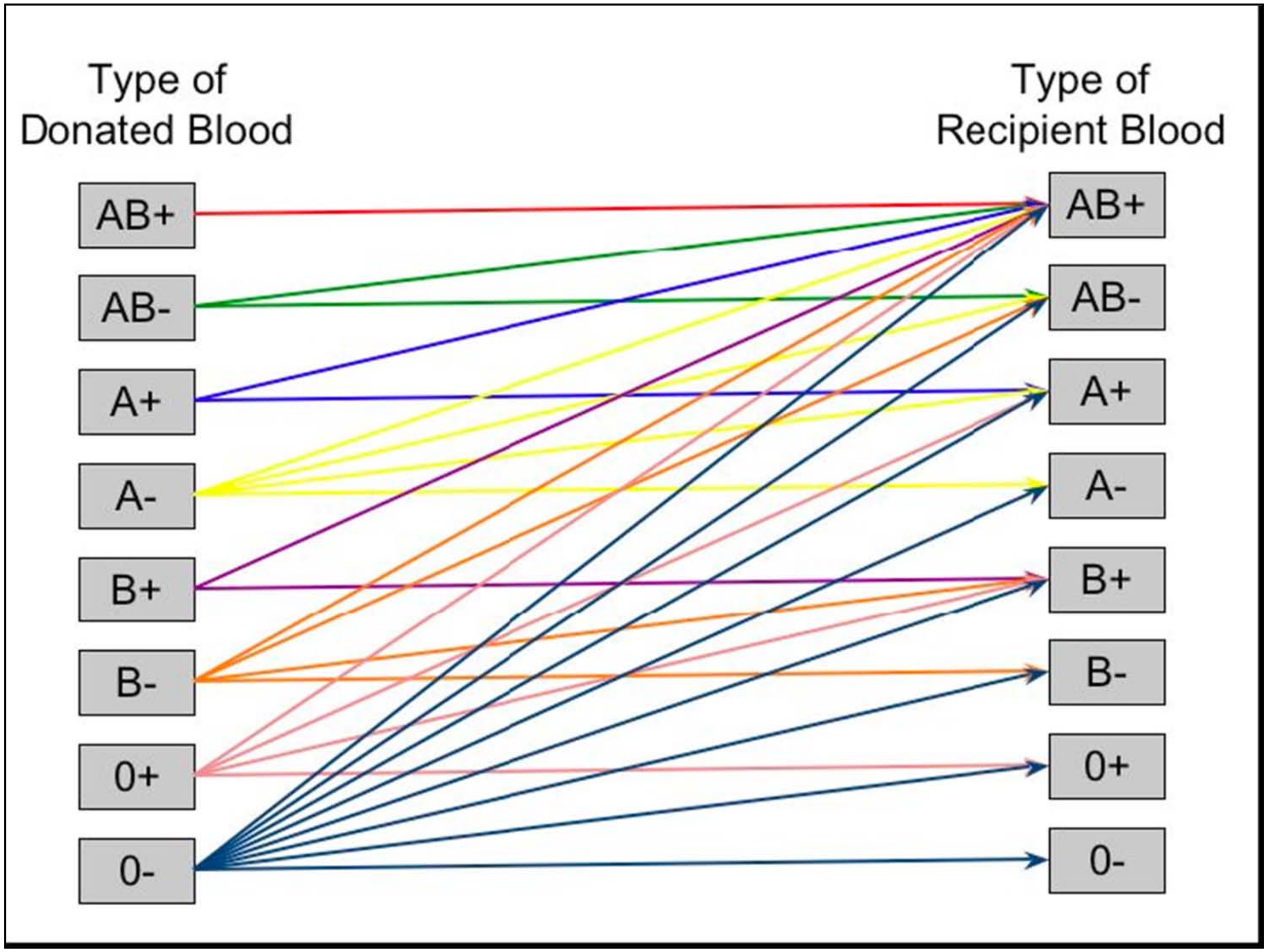


Outline

- A blood management example

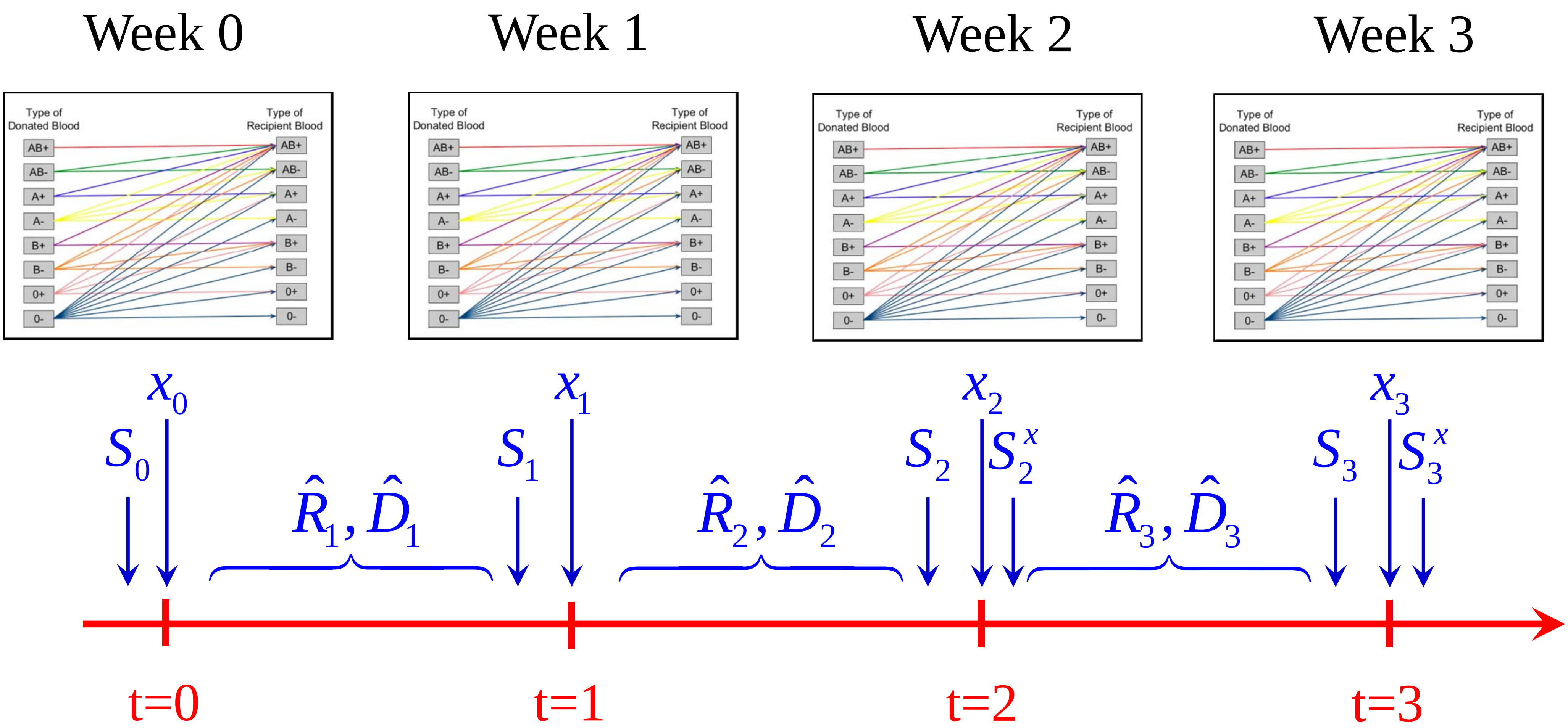
Blood management

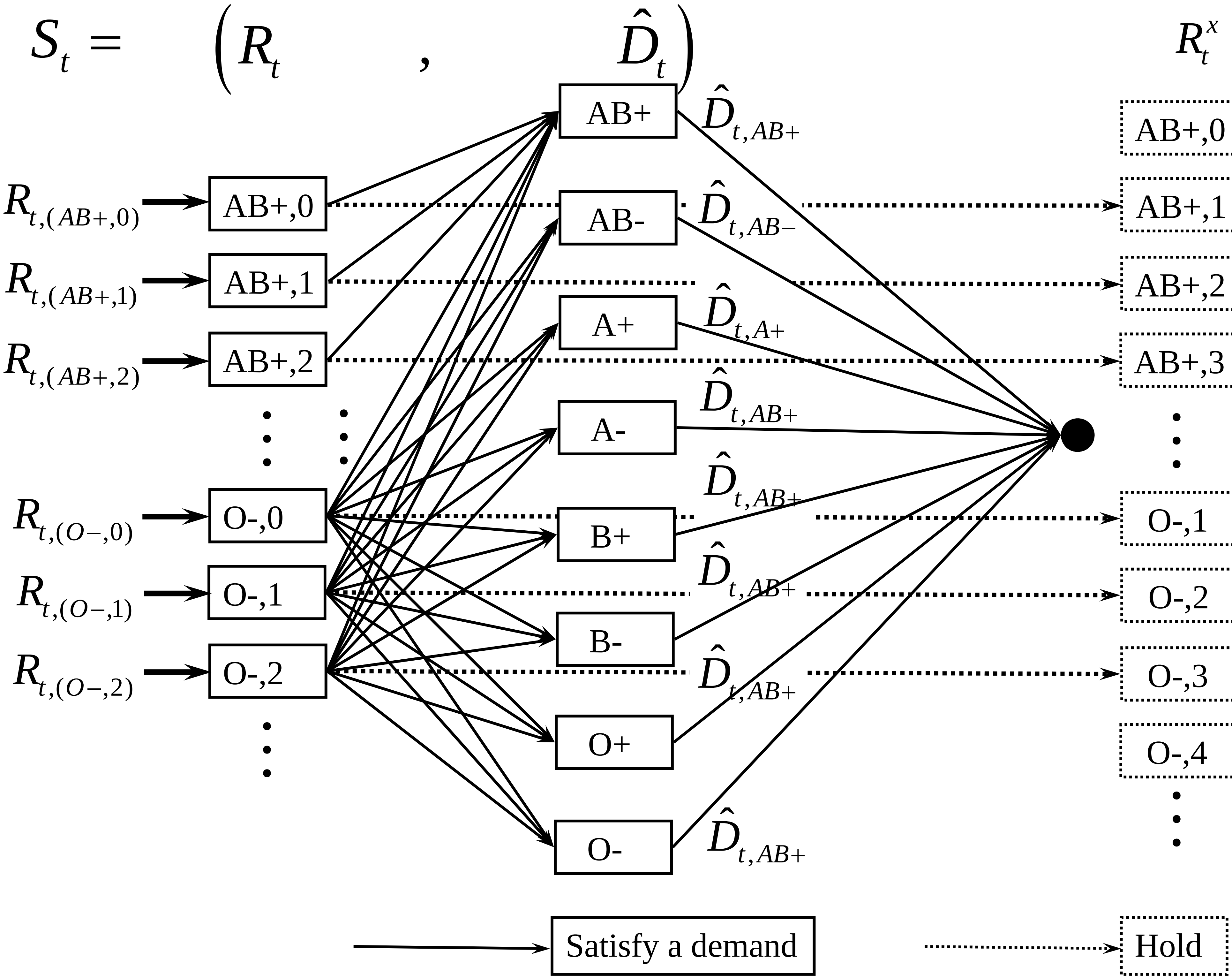
- Managing blood inventories



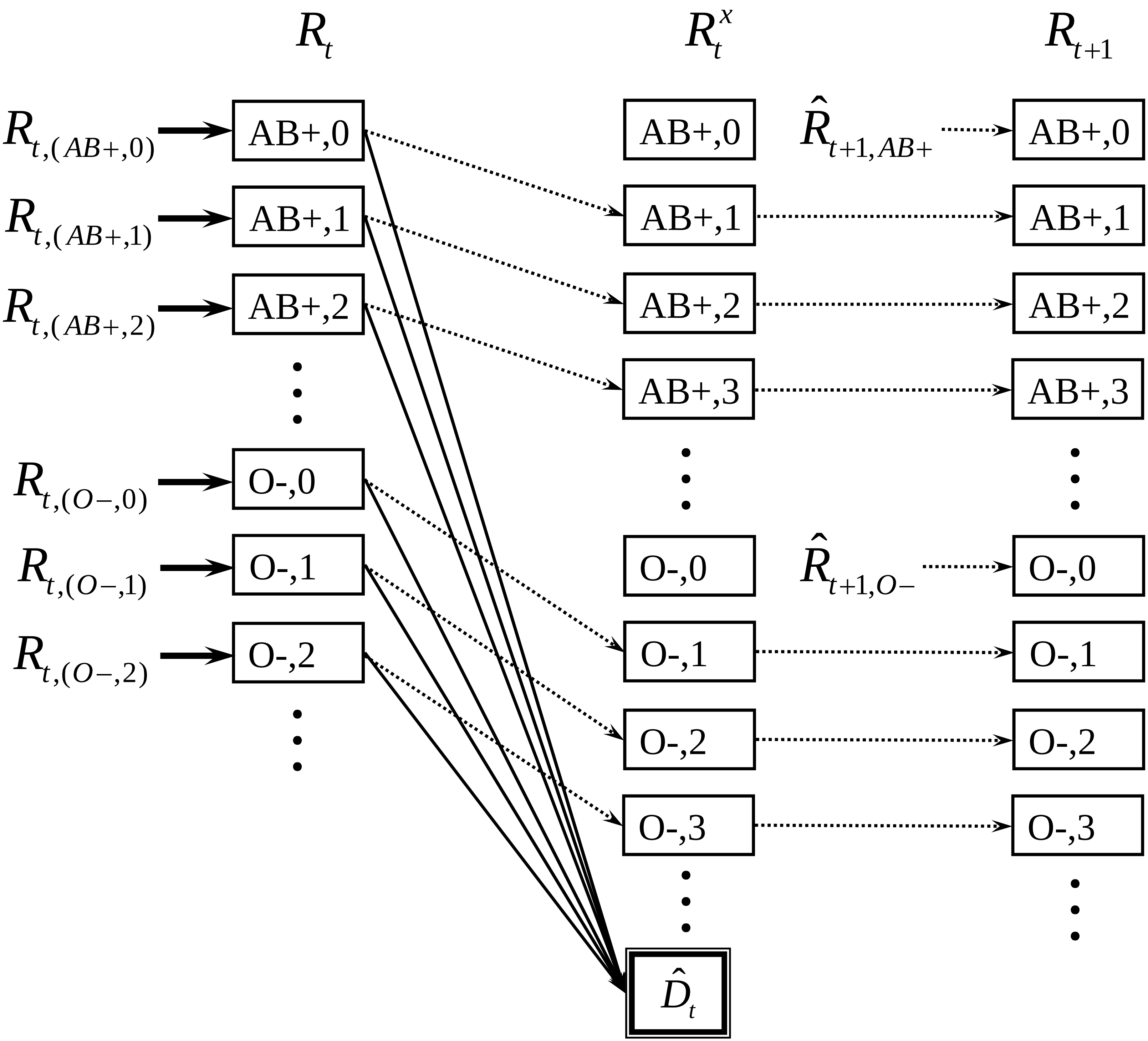
Blood management

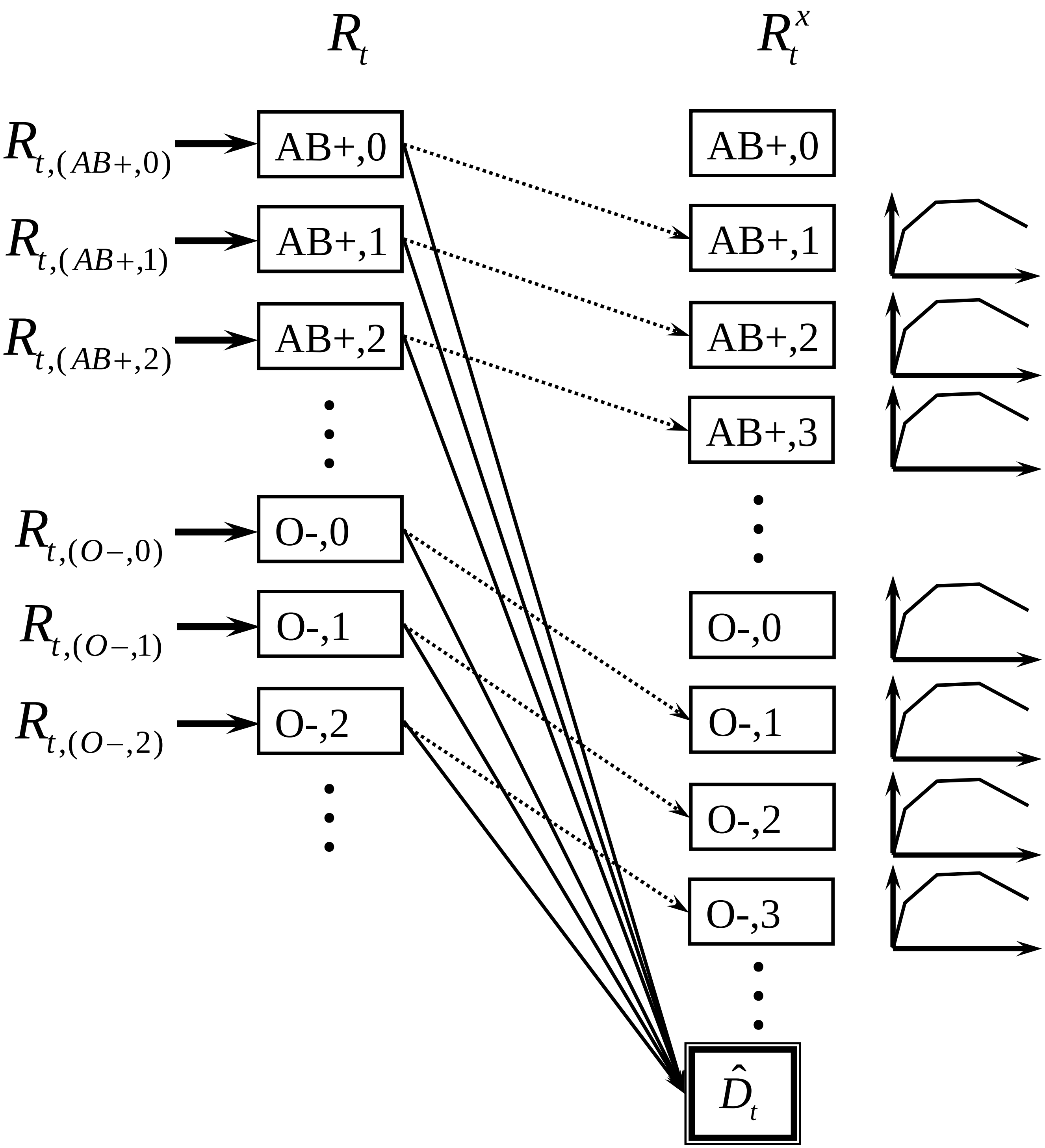
- Managing blood inventories over time



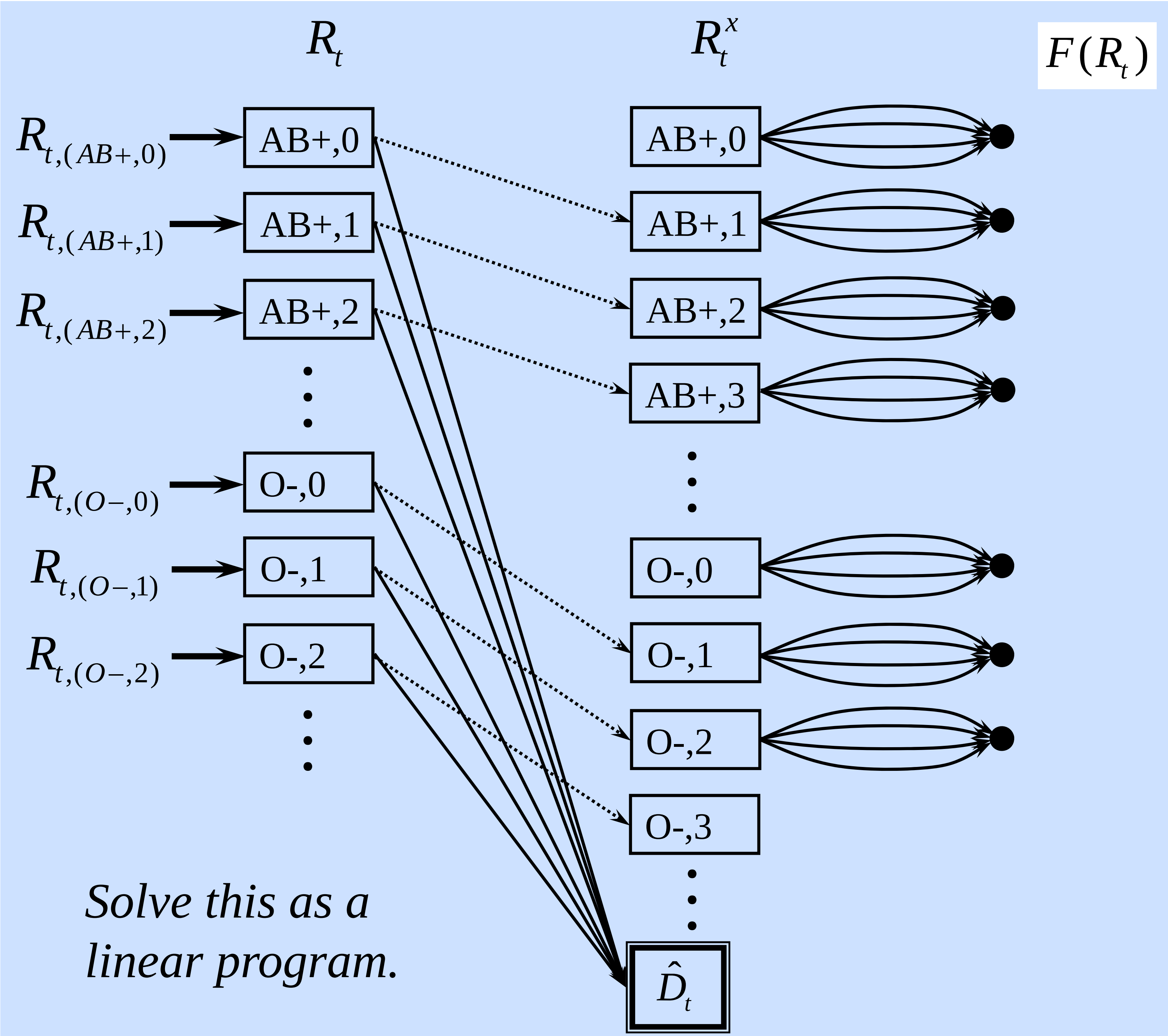


$S_t = \left(R_t \quad , \quad \hat{D}_t \right)$
R_t^x
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(O-,0)}$
$R_{t,(O-,1)}$
$R_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
O-,0
O-,1
O-,2
AB+
AB-
A+
A-
B+
B-
O+
O-
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB-}$
$\hat{D}_{t,A+}$
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB+}$
$\hat{D}_{t,AB+}$
AB+,0
AB+,1
AB+,2
AB+,3
O-,1
O-,2
O-,3
O-,4
Satisfy a demand
Hold

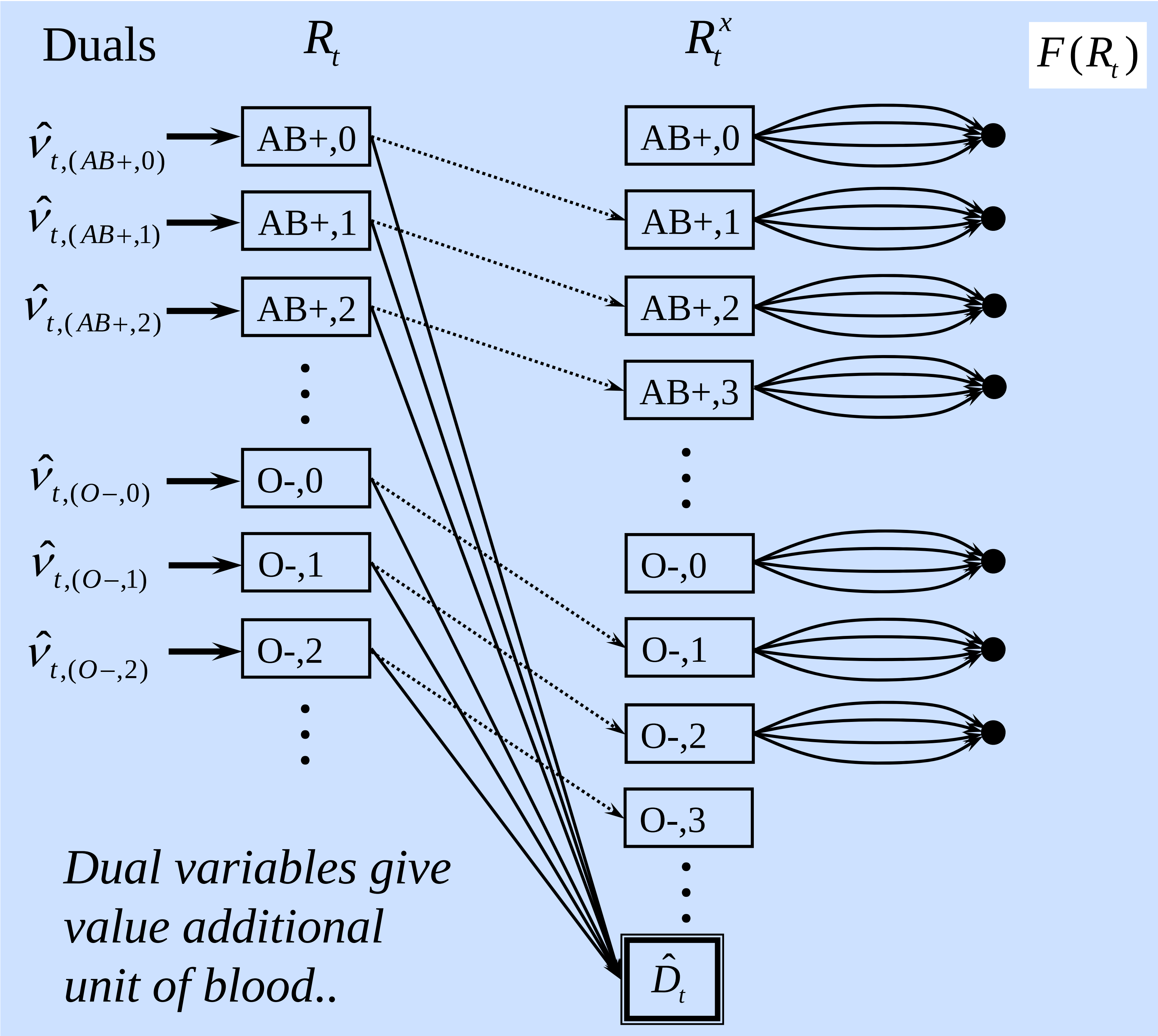

R_t
R_t^x
R_{t+1}
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(O-,0)}$
$R_{t,(O-,1)}$
$R_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
AB+,3
O-,0
O-,1
O-,2
O-,3
$\hat{R}_{t+1,AB+}$
$\hat{R}_{t+1,O-}$
$\hat{D}_t$



R_t
R_t^x
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(O-,0)}$
$R_{t,(O-,1)}$
$R_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
O-,0
O-,1
O-,2
AB+,0
AB+,1
AB+,2
AB+,3
O-,0
O-,1
O-,2
O-,3
$\hat{D}_t$



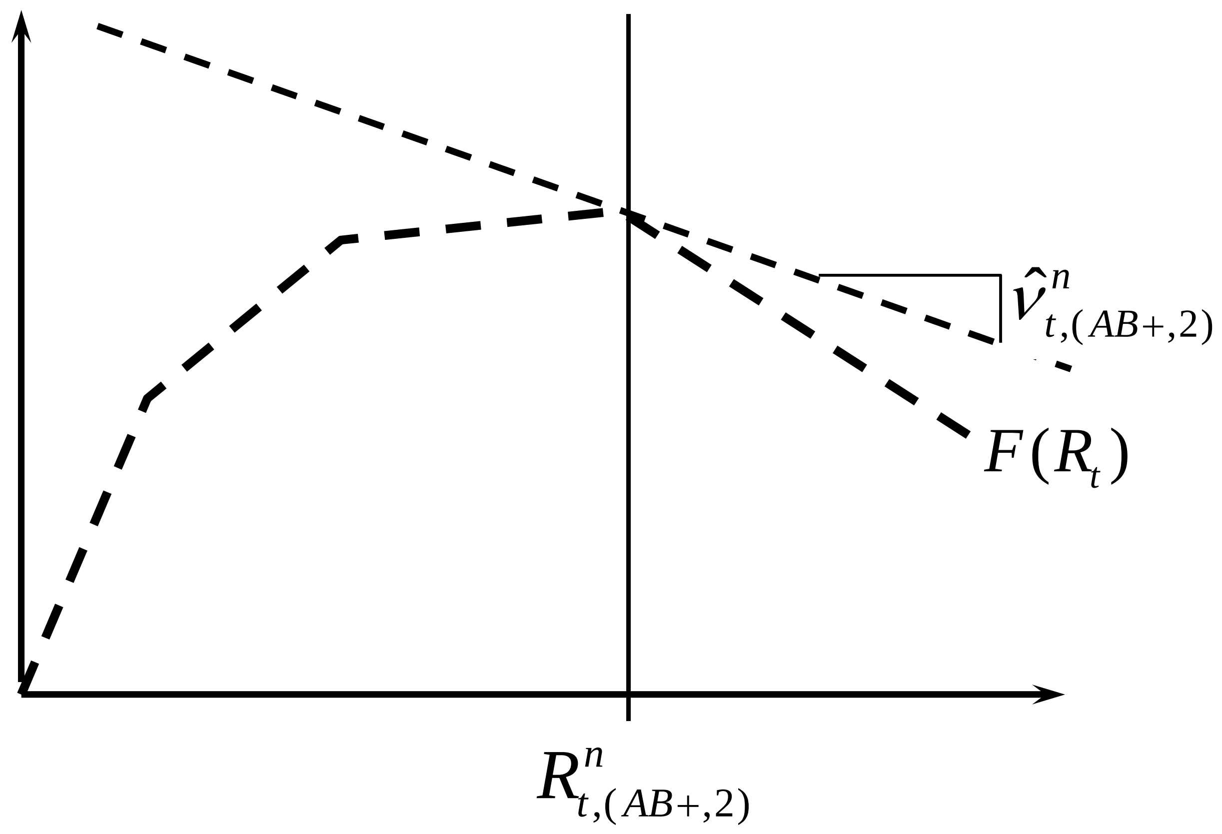
Solve this as a linear program.



Duals
R_t
R_t^x
$F(R_t)$
$\hat{v}_{t,(AB+,0)}$
$\hat{v}_{t,(AB+,1)}$
$\hat{v}_{t,(AB+,2)}$
$\hat{v}_{t,(O-,0)}$
$\hat{v}_{t,(O-,1)}$
$\hat{v}_{t,(O-,2)}$
AB+,0
AB+,1
AB+,2
AB+,3
O-,0
O-,1
O-,2
O-,3
$\hat{D}_t$
Dual variables give value additional unit of blood..

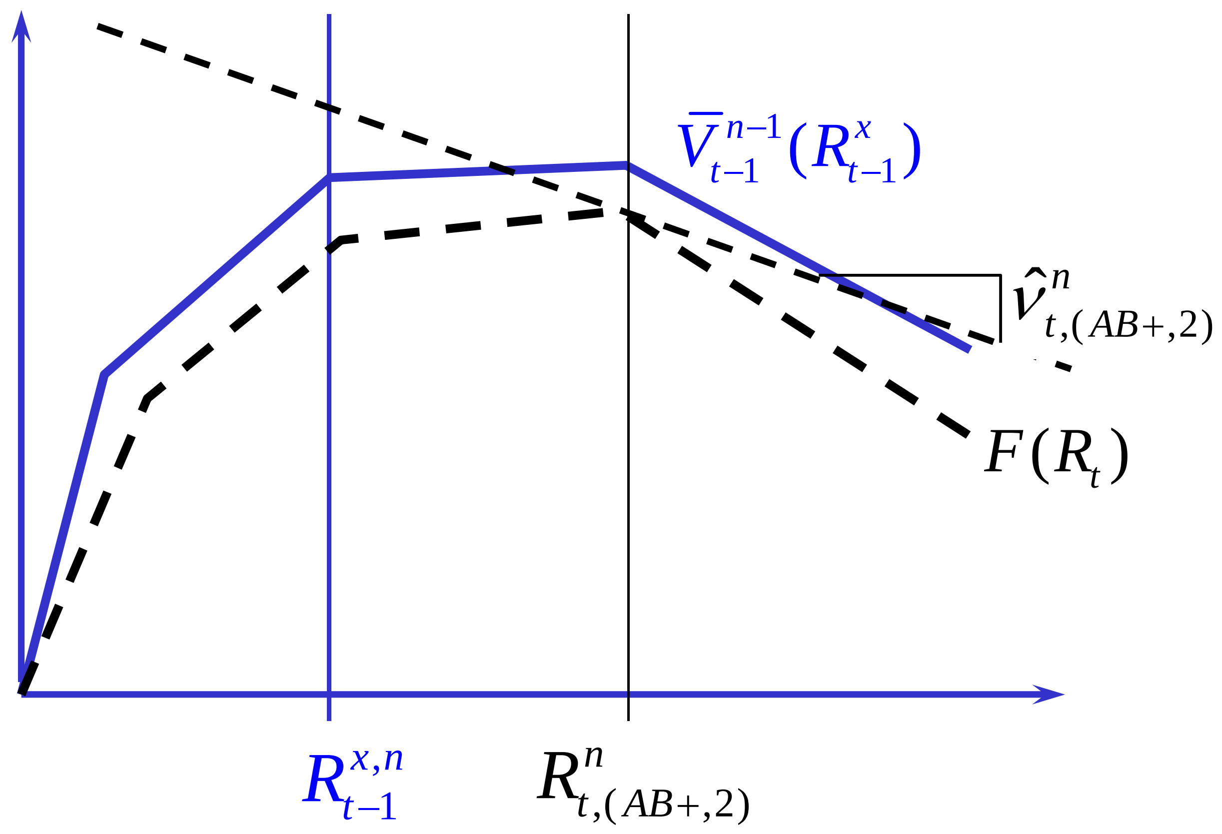
Updating the value function approximation

- Estimate the gradient at R_t^n



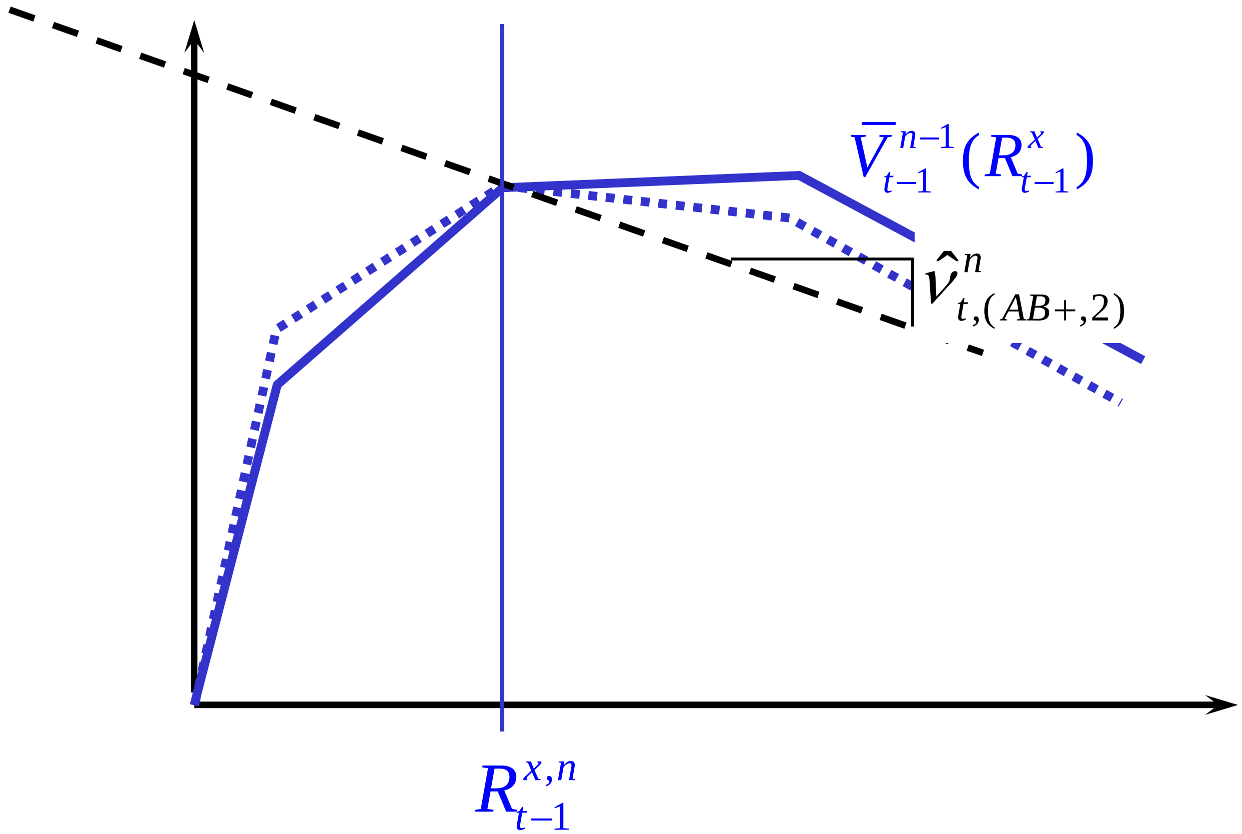
Updating the value function approximation

- Update the value function at $R^{x,n}_{t-1}$



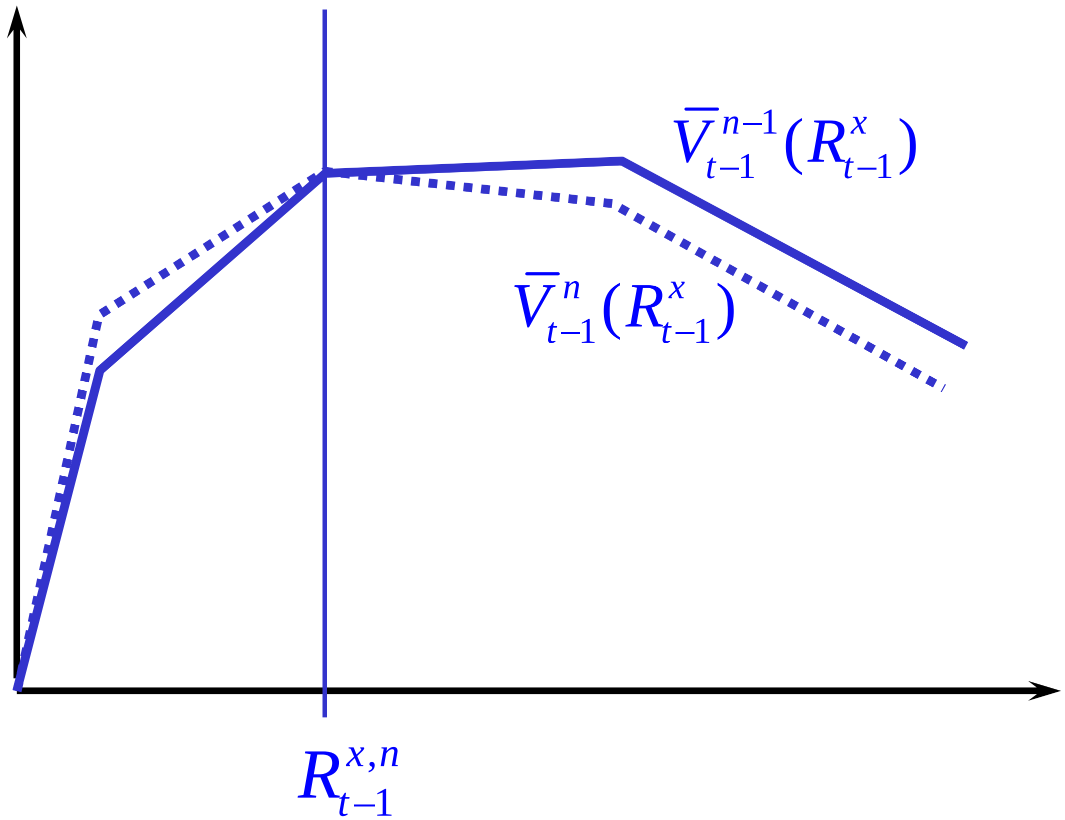
Updating the value function approximation

- Update the value function at $R_{t-1}^{x,n}$



Updating the value function approximation

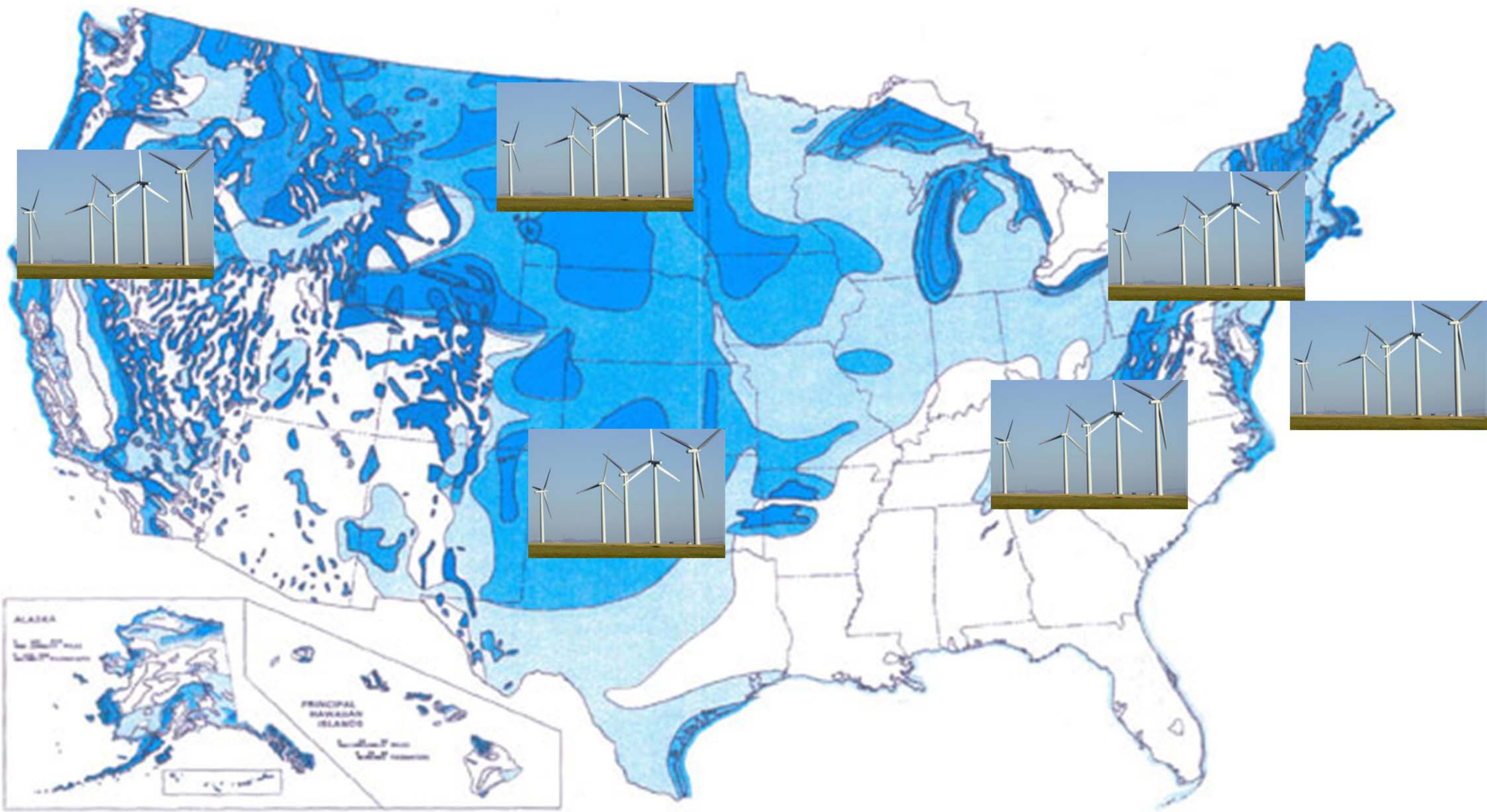
- Update the value function at $R_{t-1}^{x,n}$



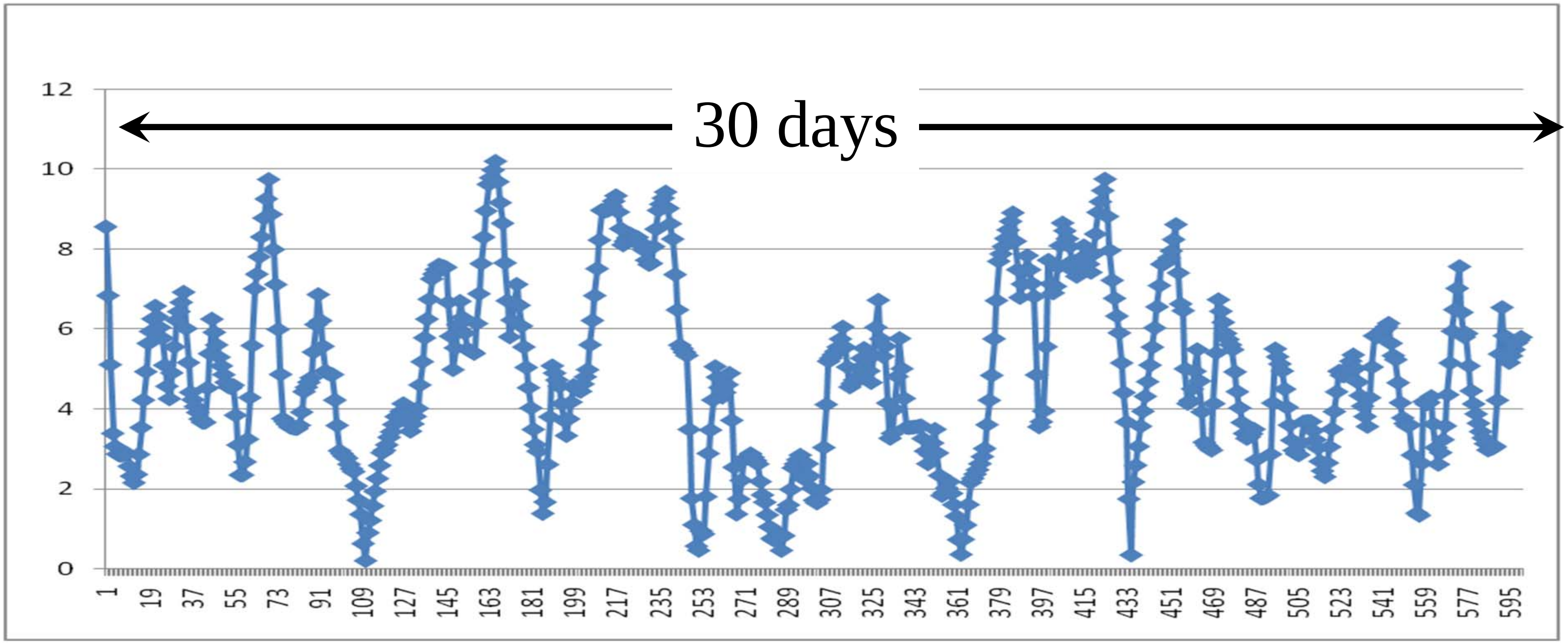
Outline

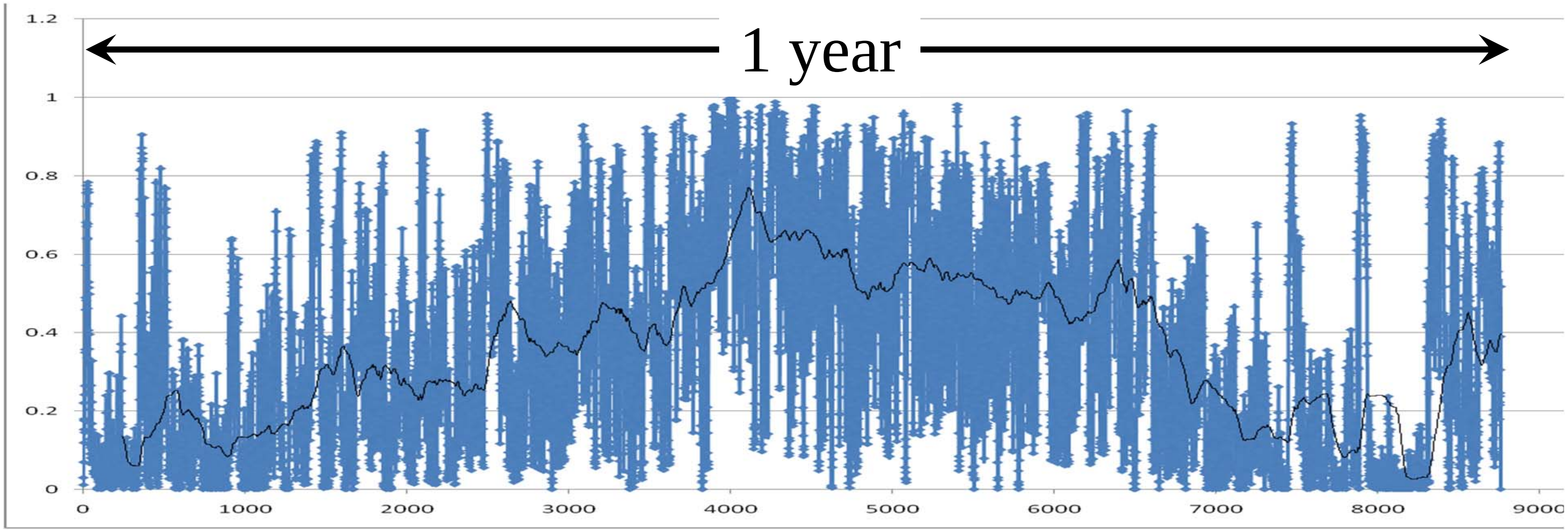
- An energy policy model

Wind

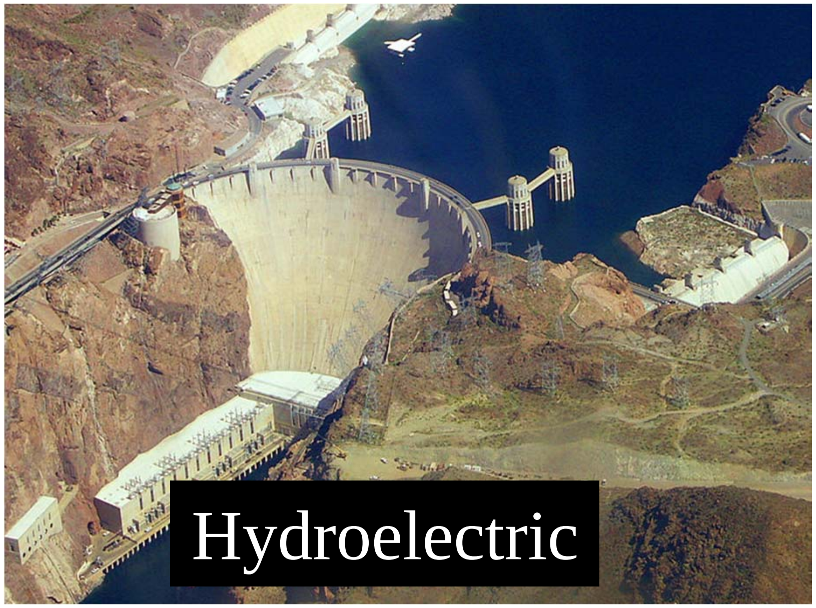


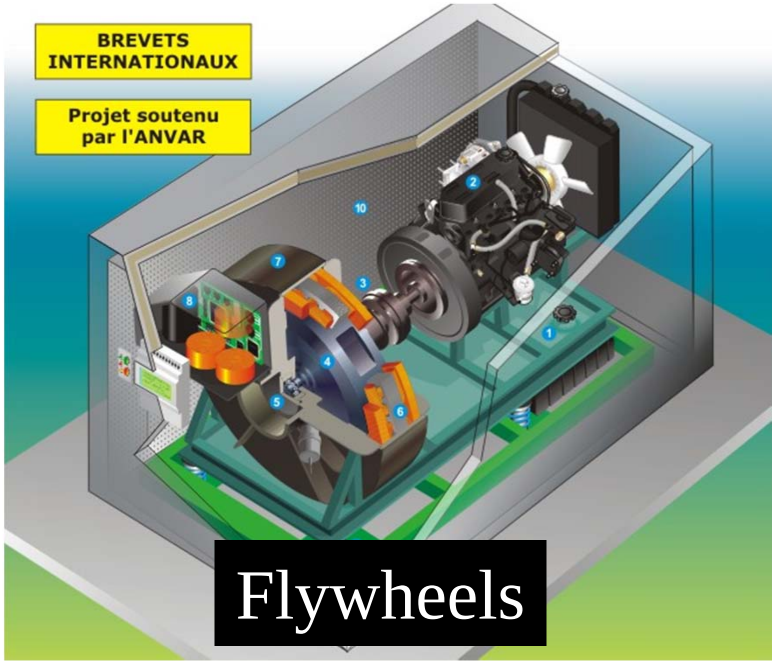
Wind


30 days
12
10
8
6
4
2
0
1 19 37 55 73 91 109 127 145 163 181 199 217 235 253 271 289 307 325 343 361 379 397 415 433 451 469 487 505 523 541 559 577 595

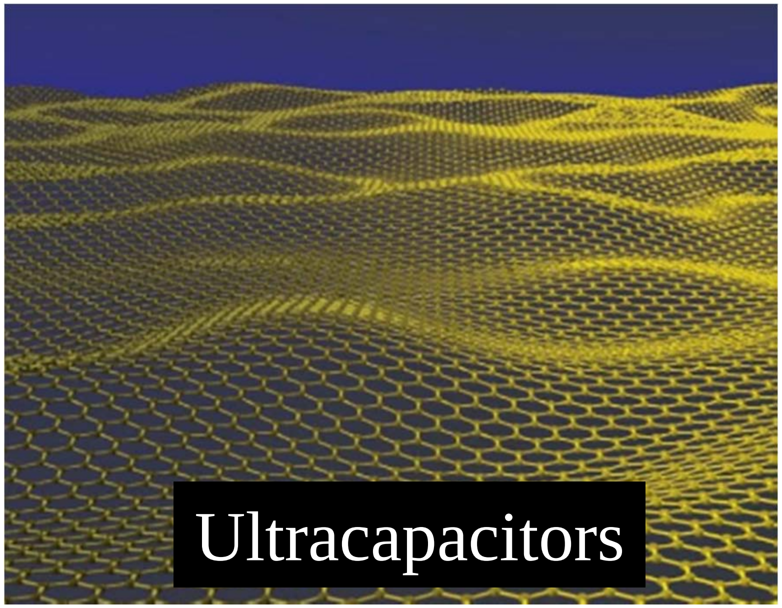

1 year
1.2
1
0.8
0.6
0.4
0.2
0
0 1000 2000 3000 4000 5000 6000 7000 8000 9000

Storage


Hydroelectric

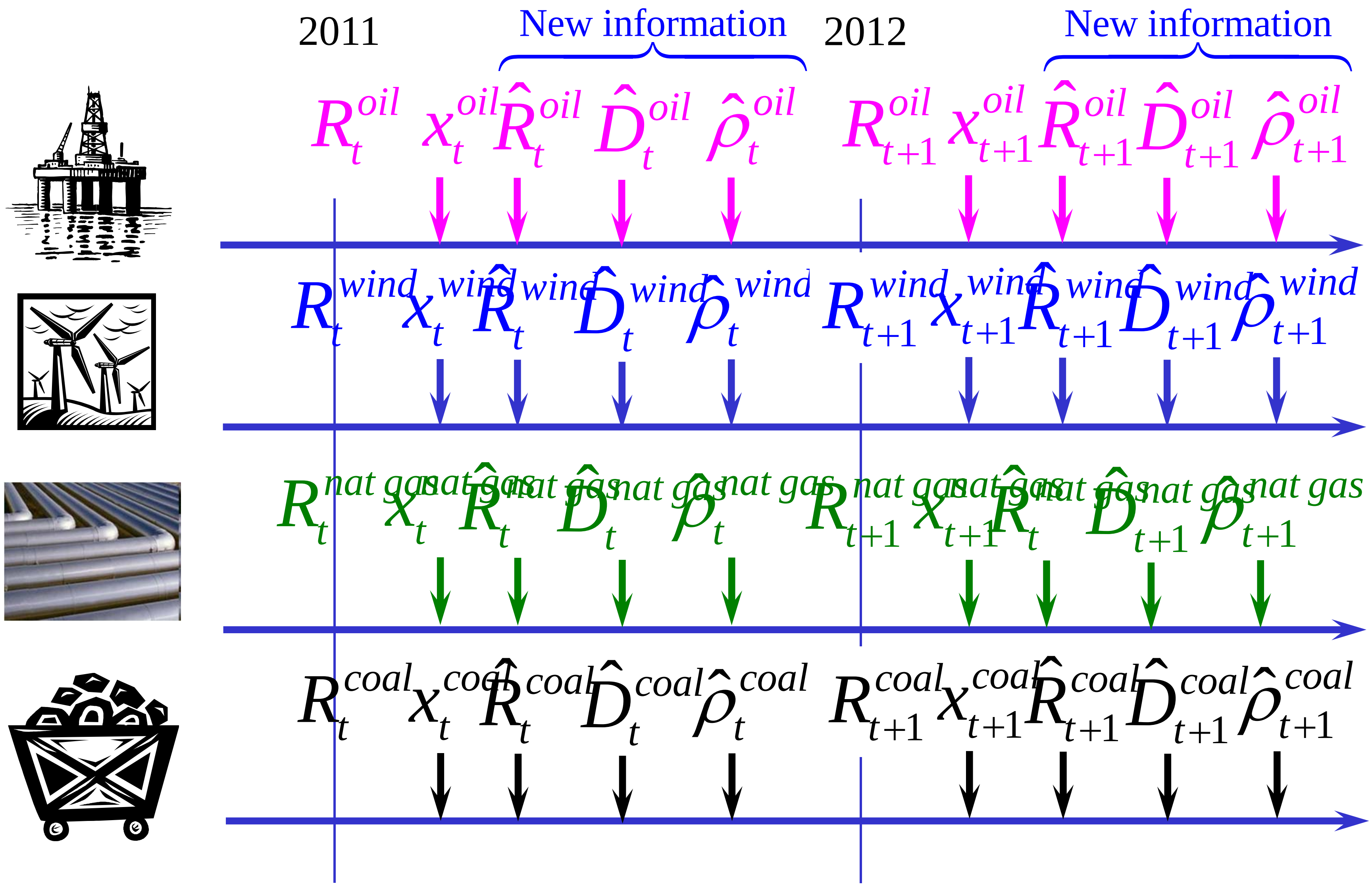

BREVETS INTERNATIONAUX
Projet soutenu par l'ANVAR
Flywheels


Batteries


Ultracapacitors

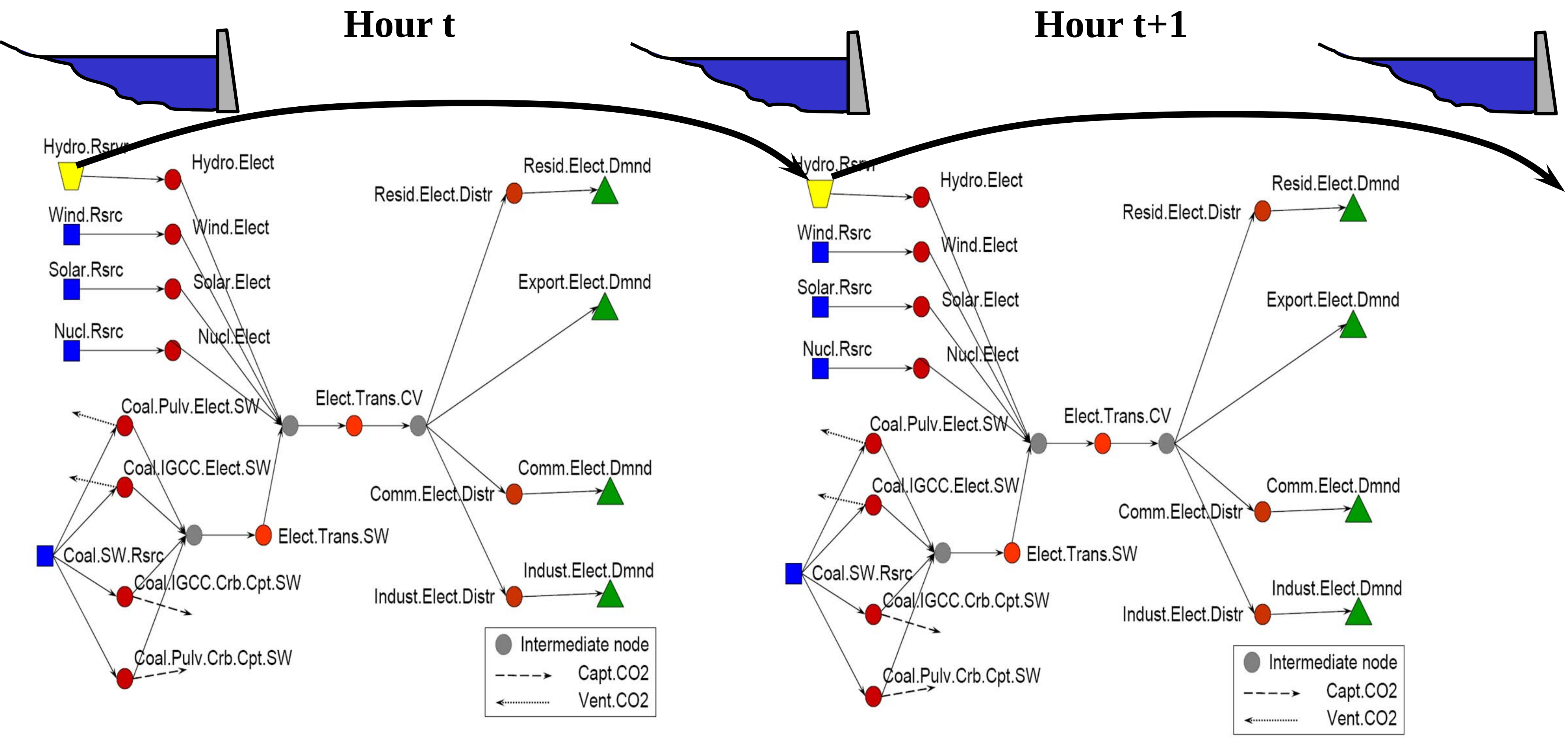
The energy resource planning problem

- The investment problem:



Energy resource modeling

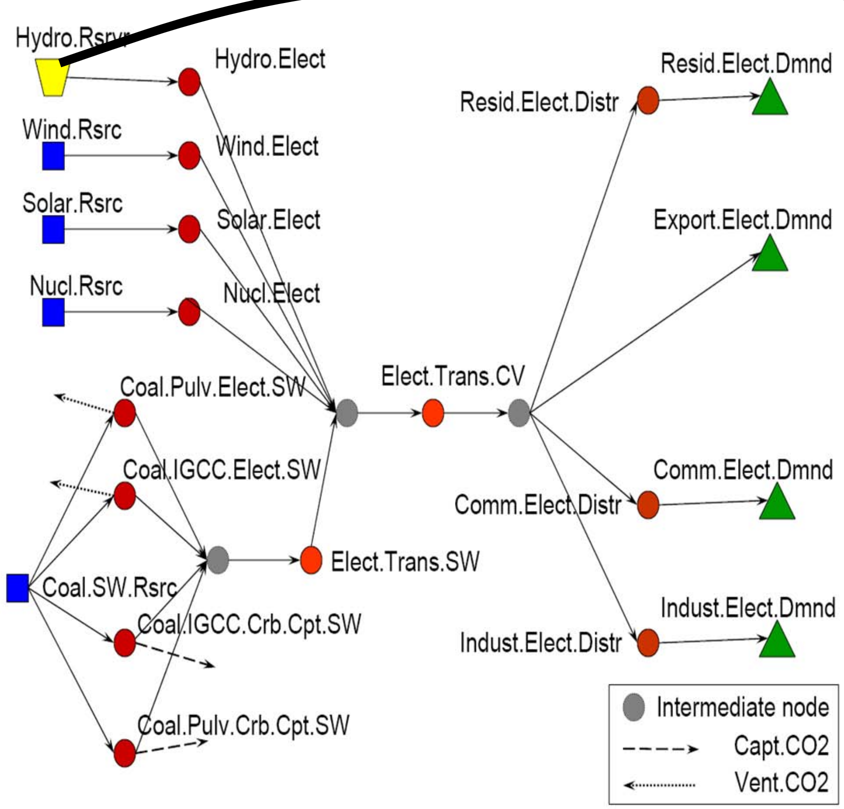
- Hourly electricity dispatch

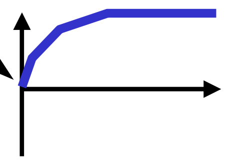


Energy resource modeling

- Hourly electricity dispatch

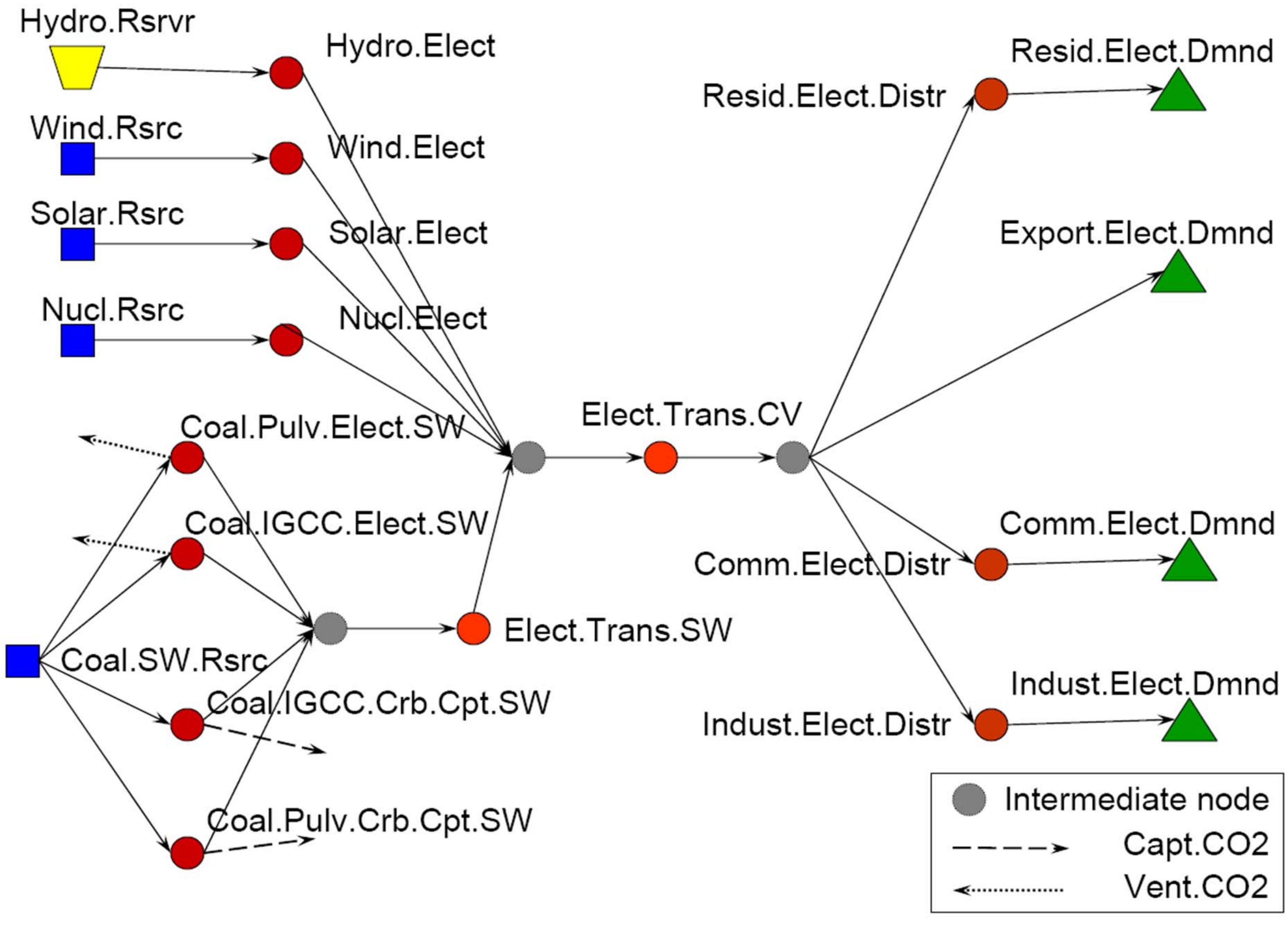
Hour t





Value of holding water in the reservoir for future time periods.

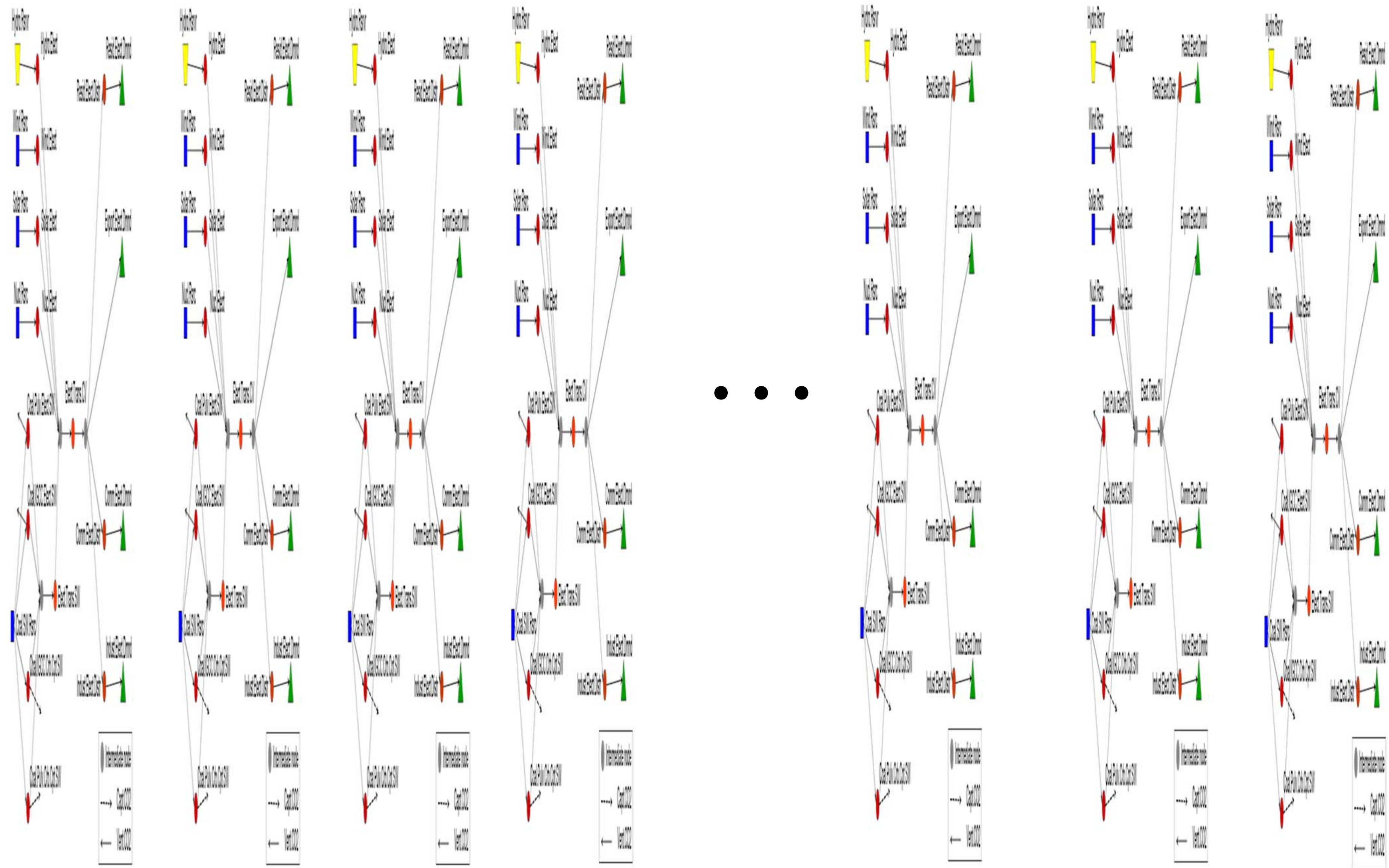
Energy resource modeling



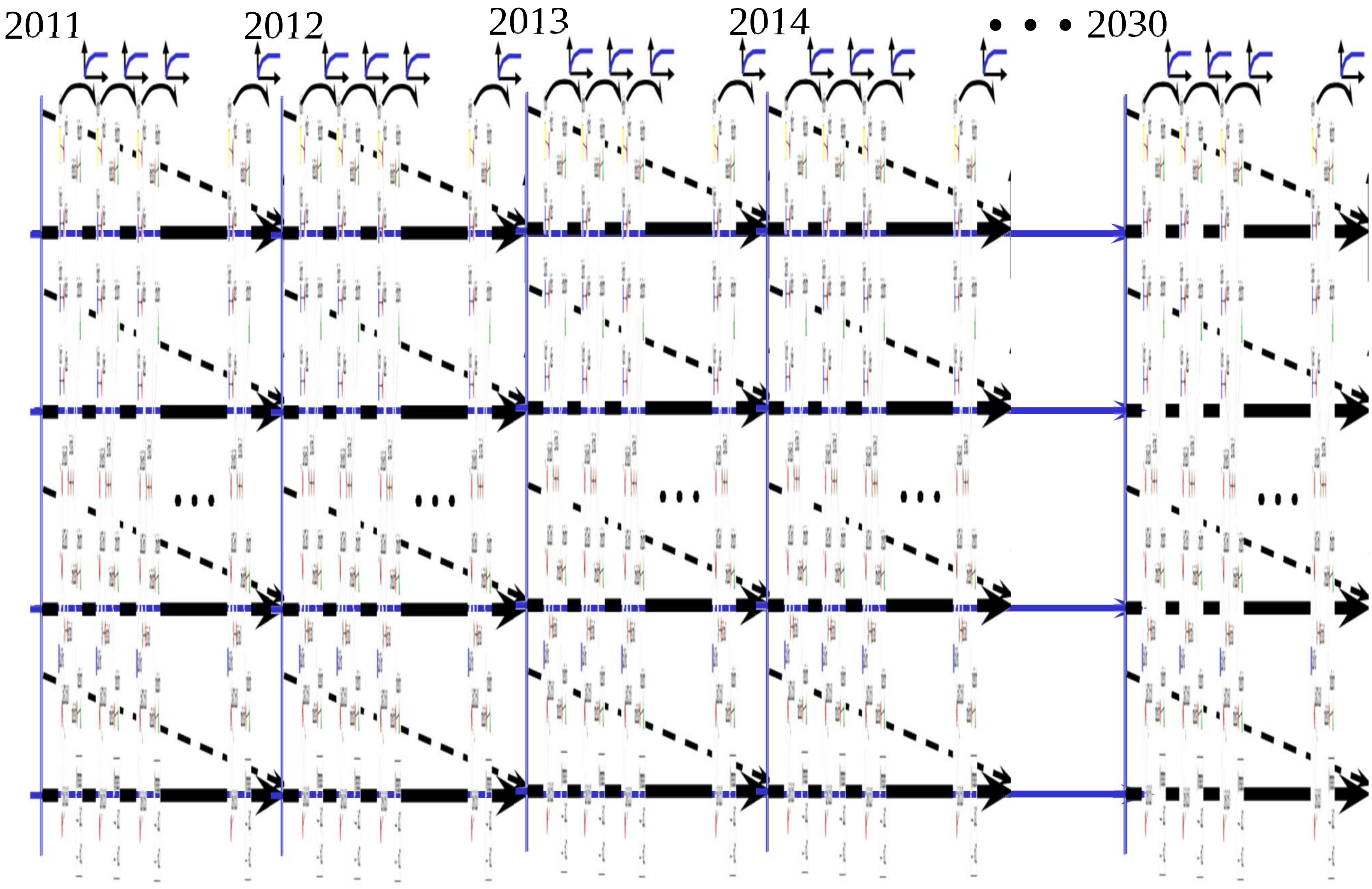
Energy resource modeling

2011 … 2011

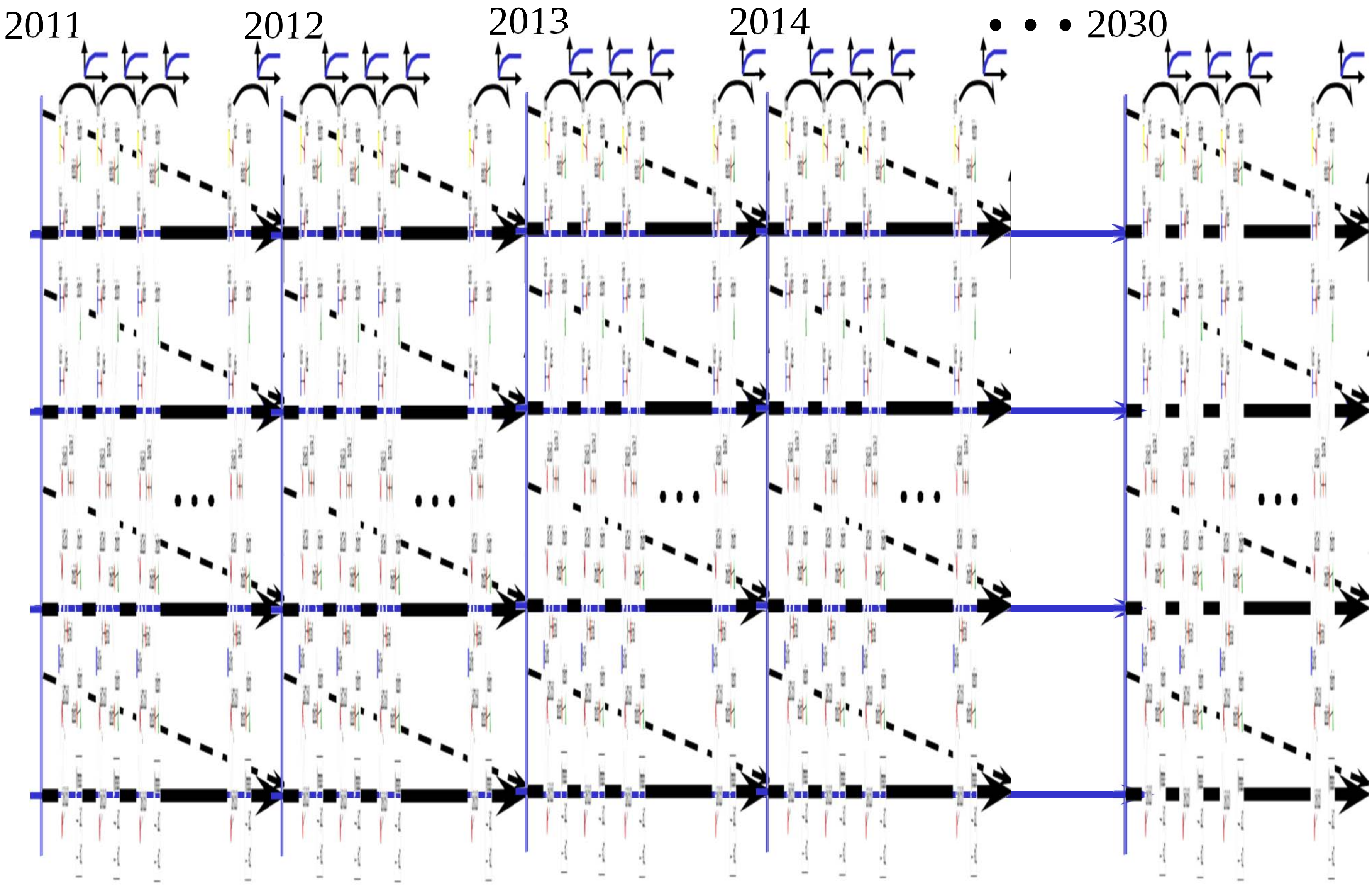
Hour 1 2 3 4 … 8760 1 2



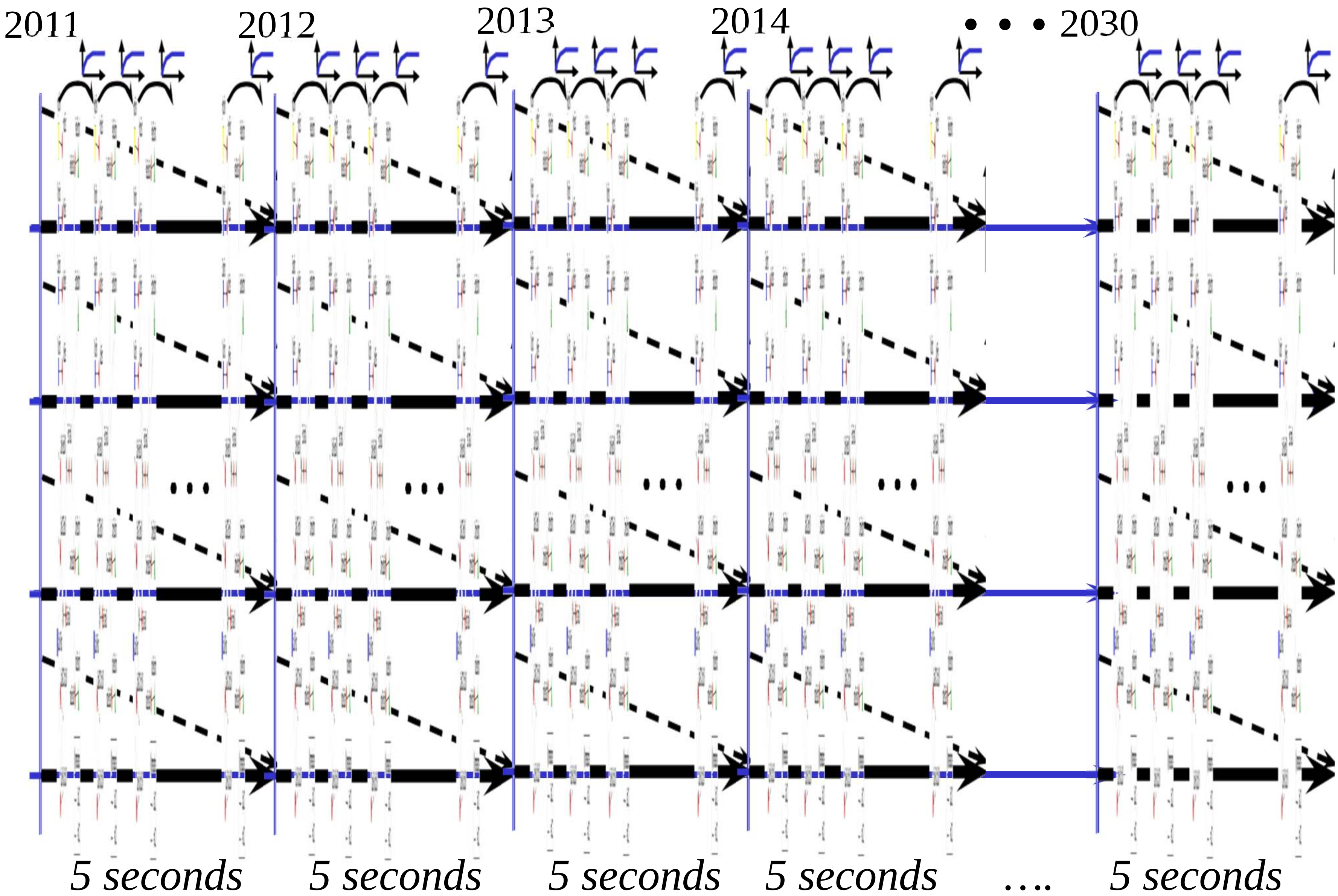
Energy resource modeling



Energy resource modeling

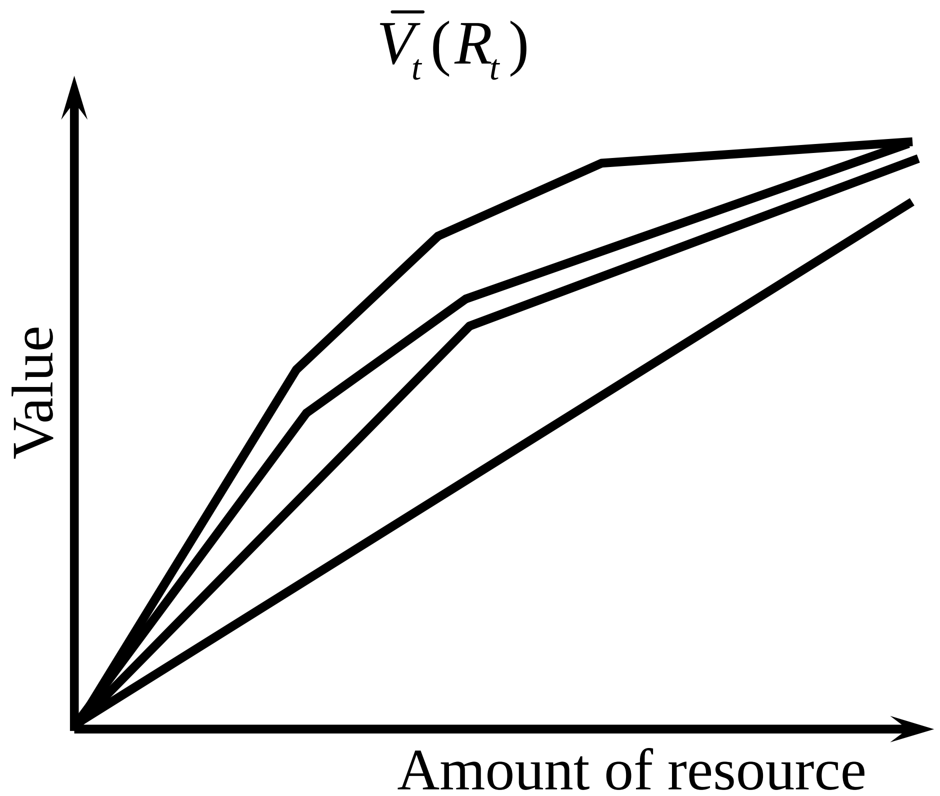


Energy resource modeling



Energy resource modeling

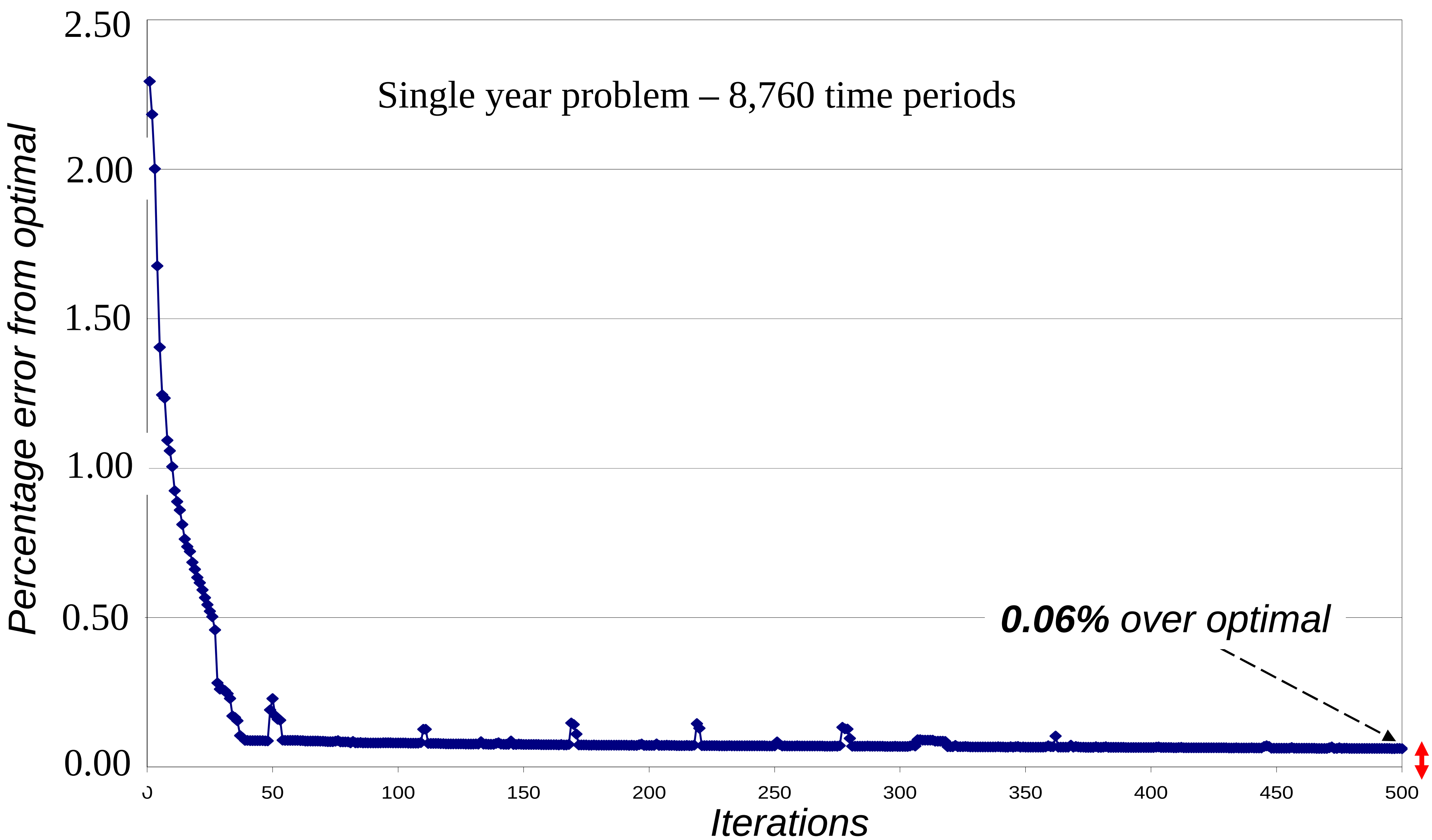
- Use statistical methods to learn the value of resources in the future.
- Resources may be:
  - » Stored energy
    - Hydro
    - Flywheel energy
    - …
  - » Storage capacity
    - Batteries
    - Flywheels
    - Compressed air
  - » Energy transmission capacity
    - Transmission lines
    - Gas lines
    - Shipping capacity
  - » Energy production sources
    - Wind mills
    - Solar panels
    - Nuclear power plants



Unlike our transportation applications, these functions are *continuous*.

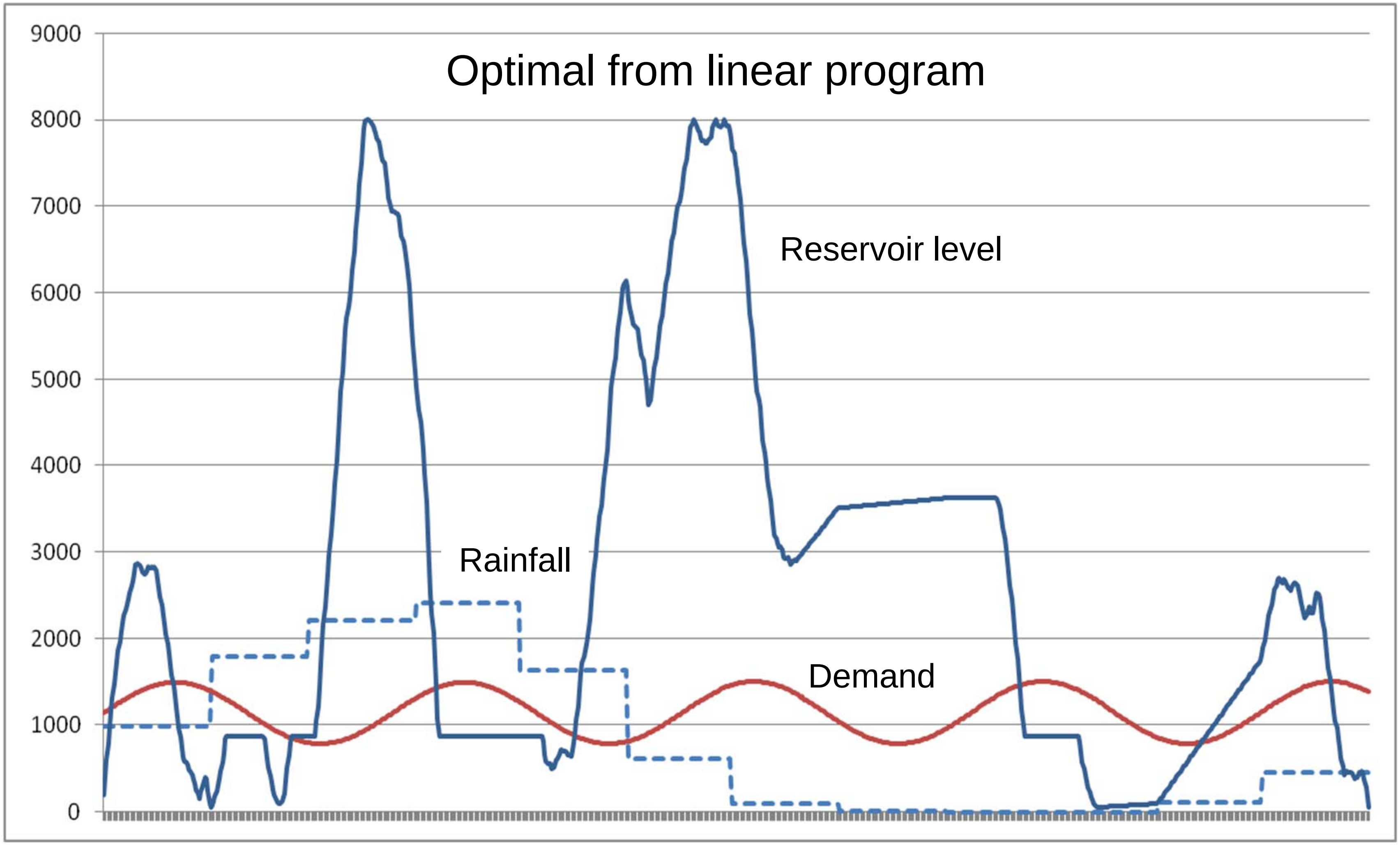
Energy resource modeling

- ADP objective function relative to optimal LP



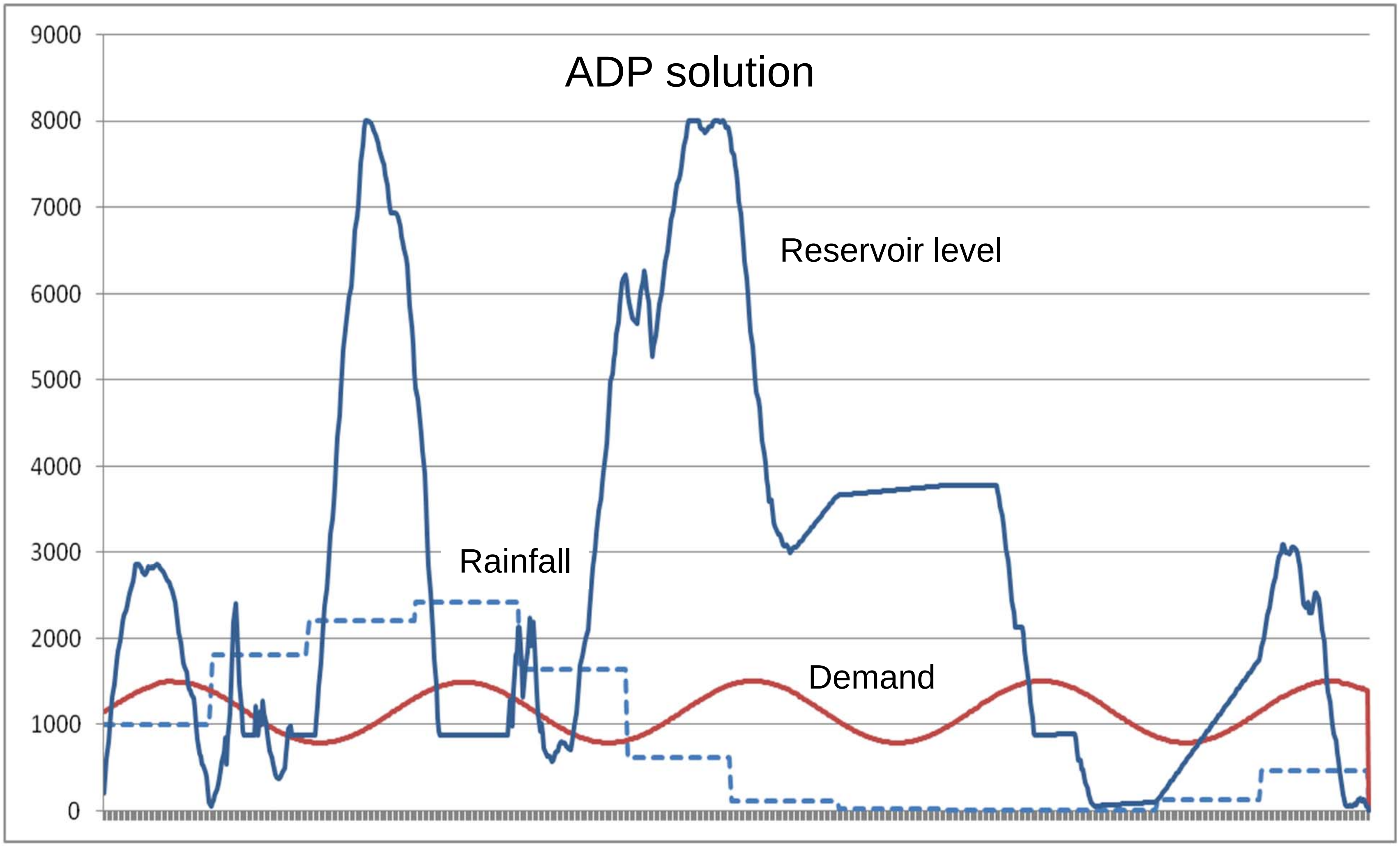
Energy resource modeling

- Optimal from linear program



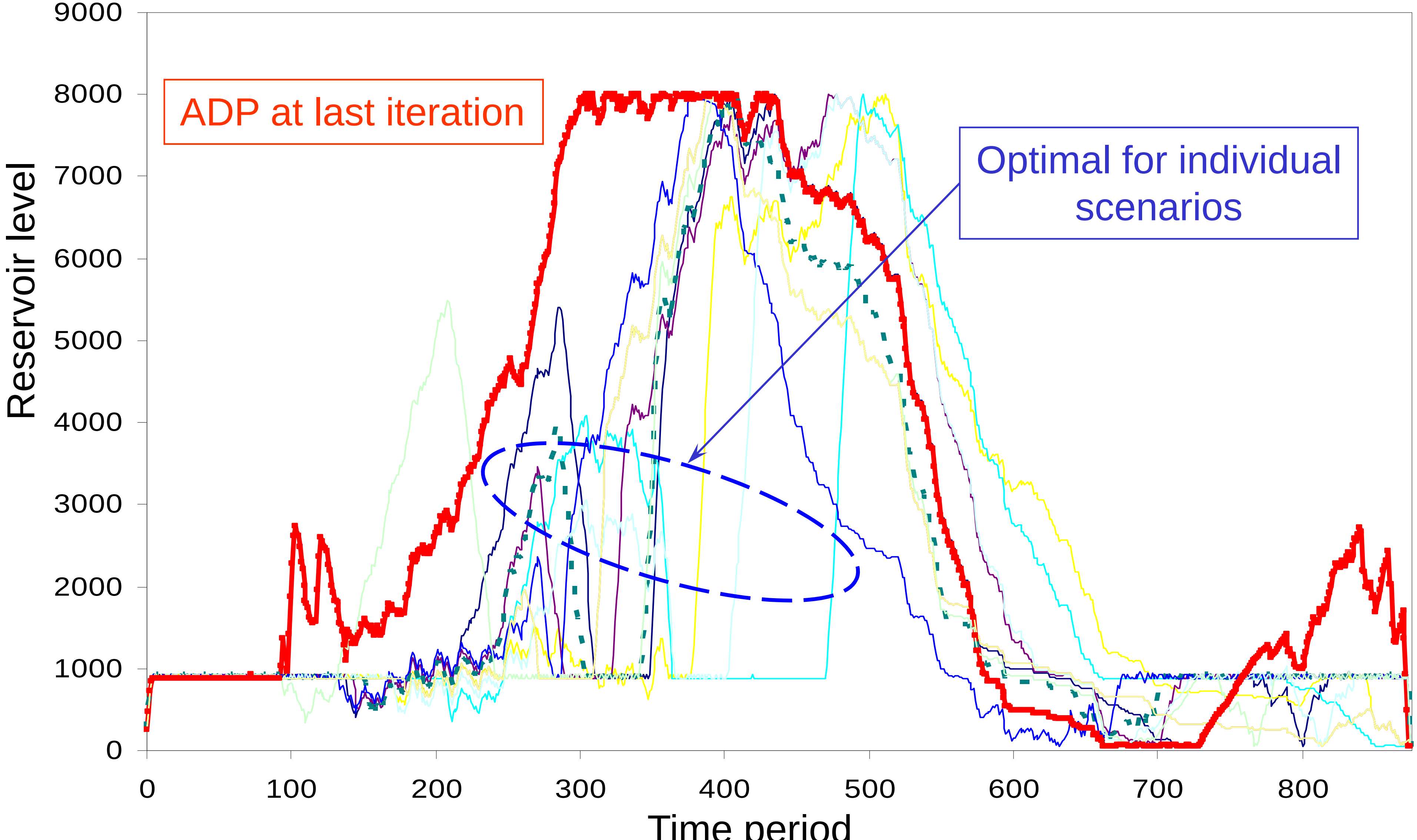
Energy resource modeling

- Approximate dynamic programming



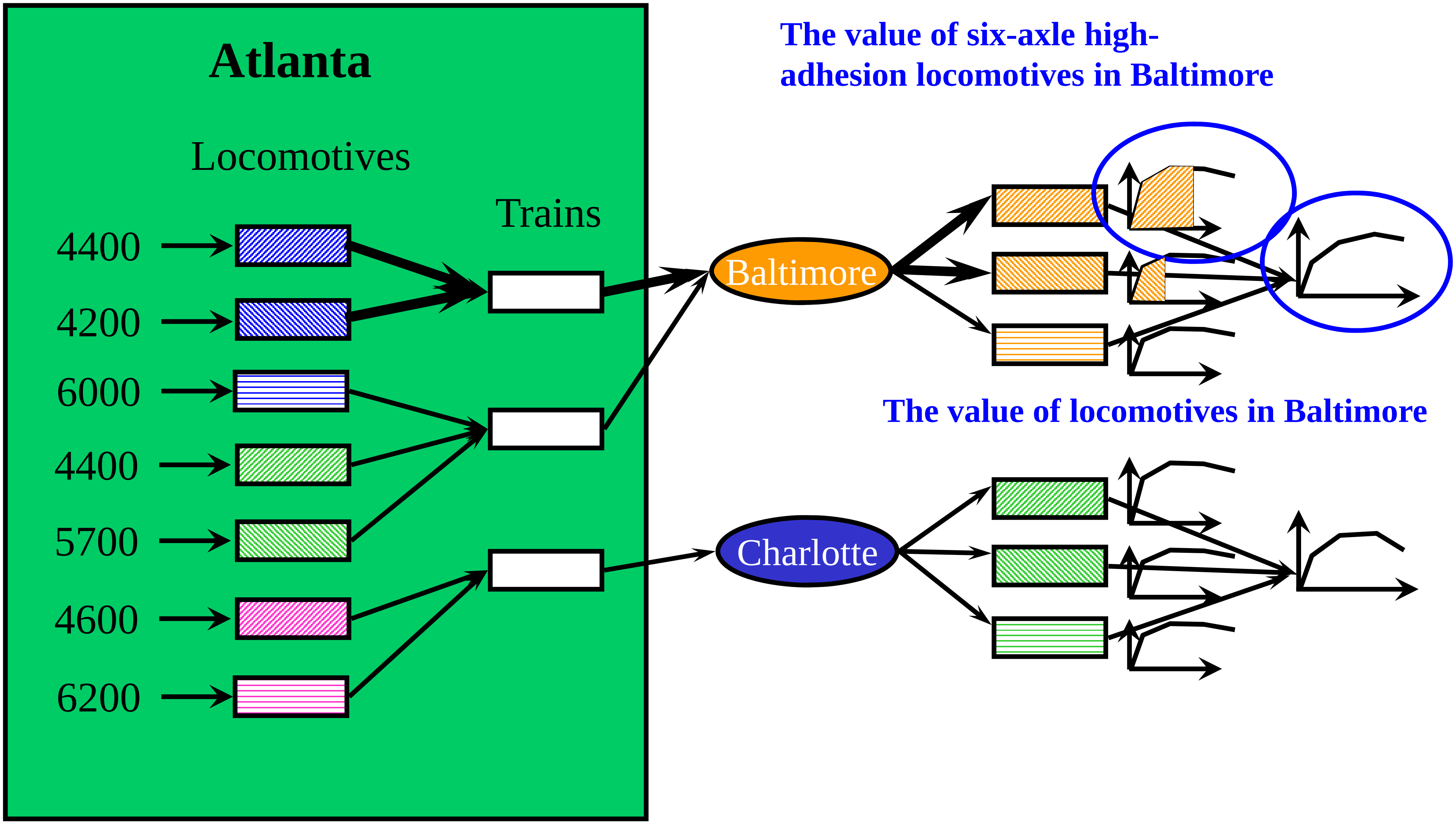
Energy resource modeling

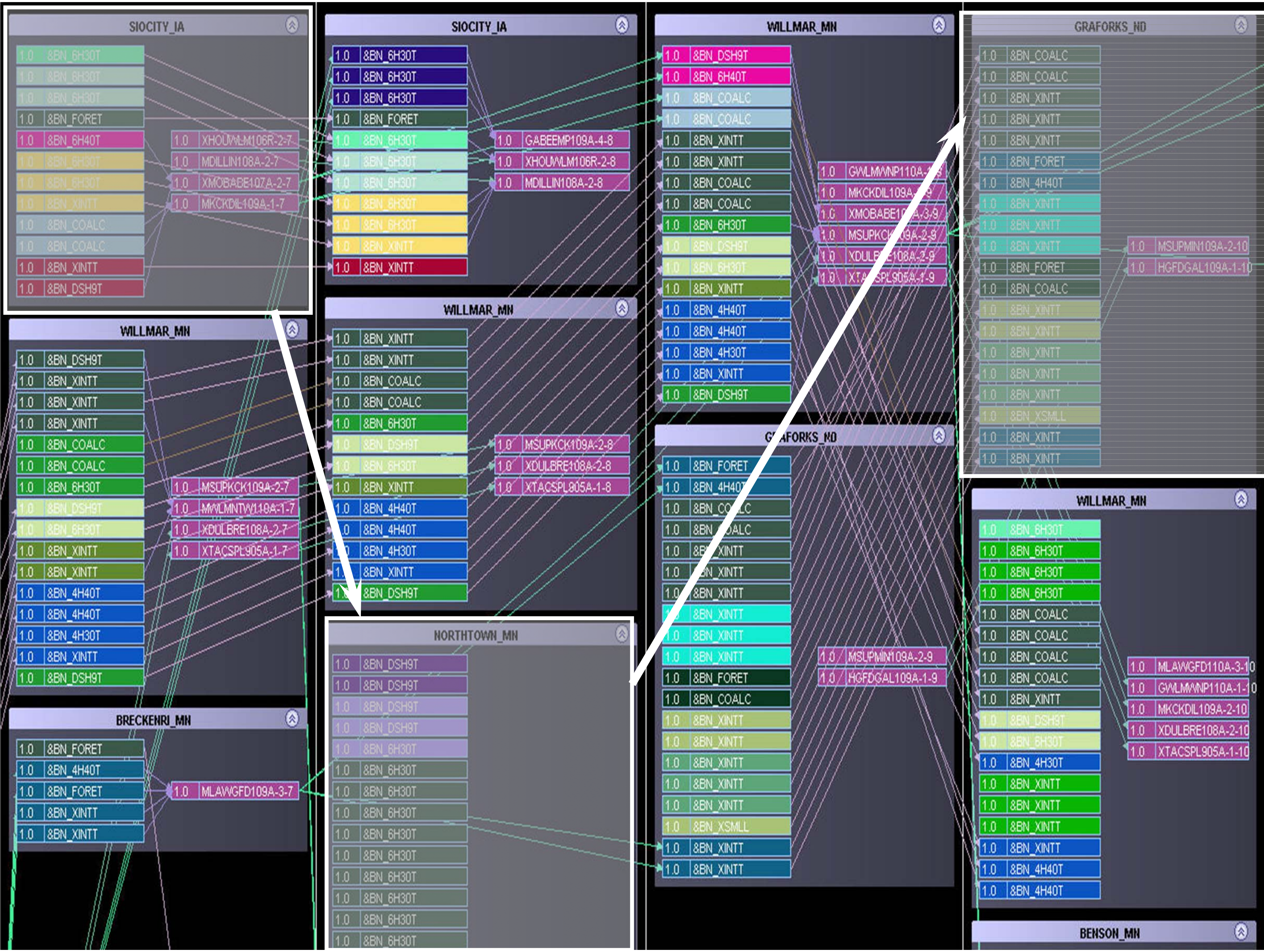
- ADP vs optimal reservoir levels for stochastic rainfall

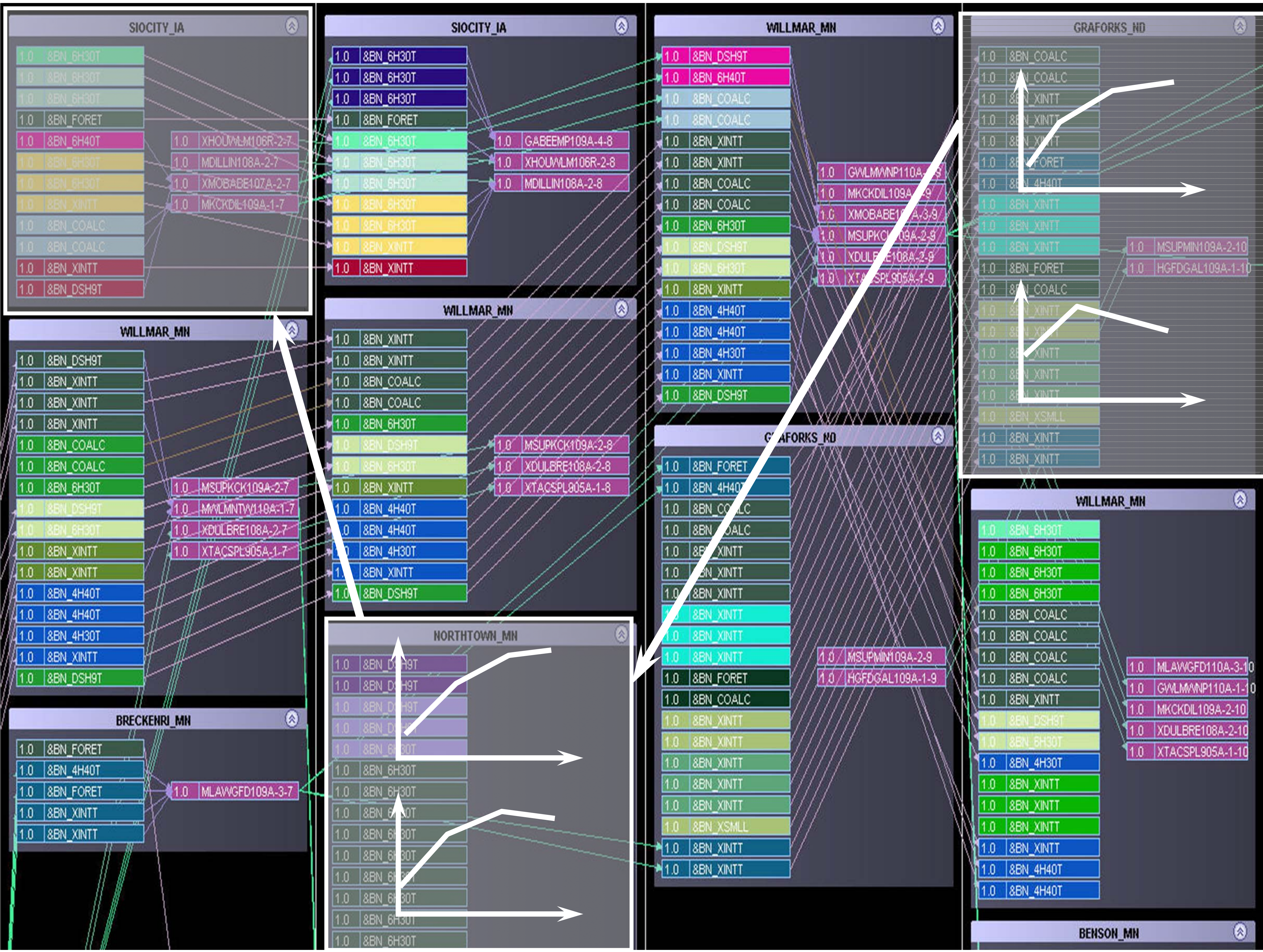



Princeton team:
Warren Powell
Belgacem Bouzaiene-Ayari
NS team:
Clark Cheng
Ricardo Fiorillo
Junxia Chang
Sourav Das
9533
9533

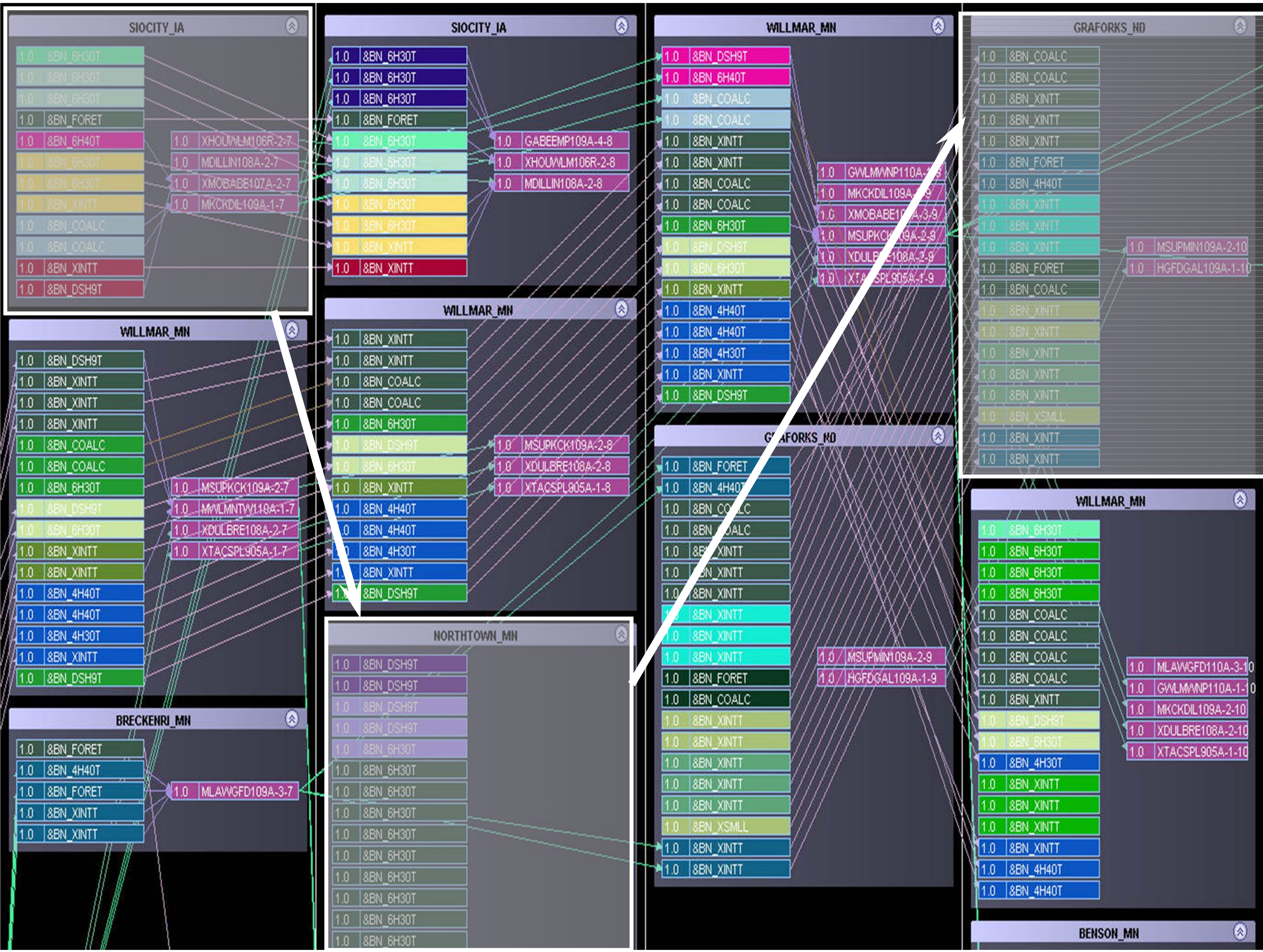
Solving the subproblem



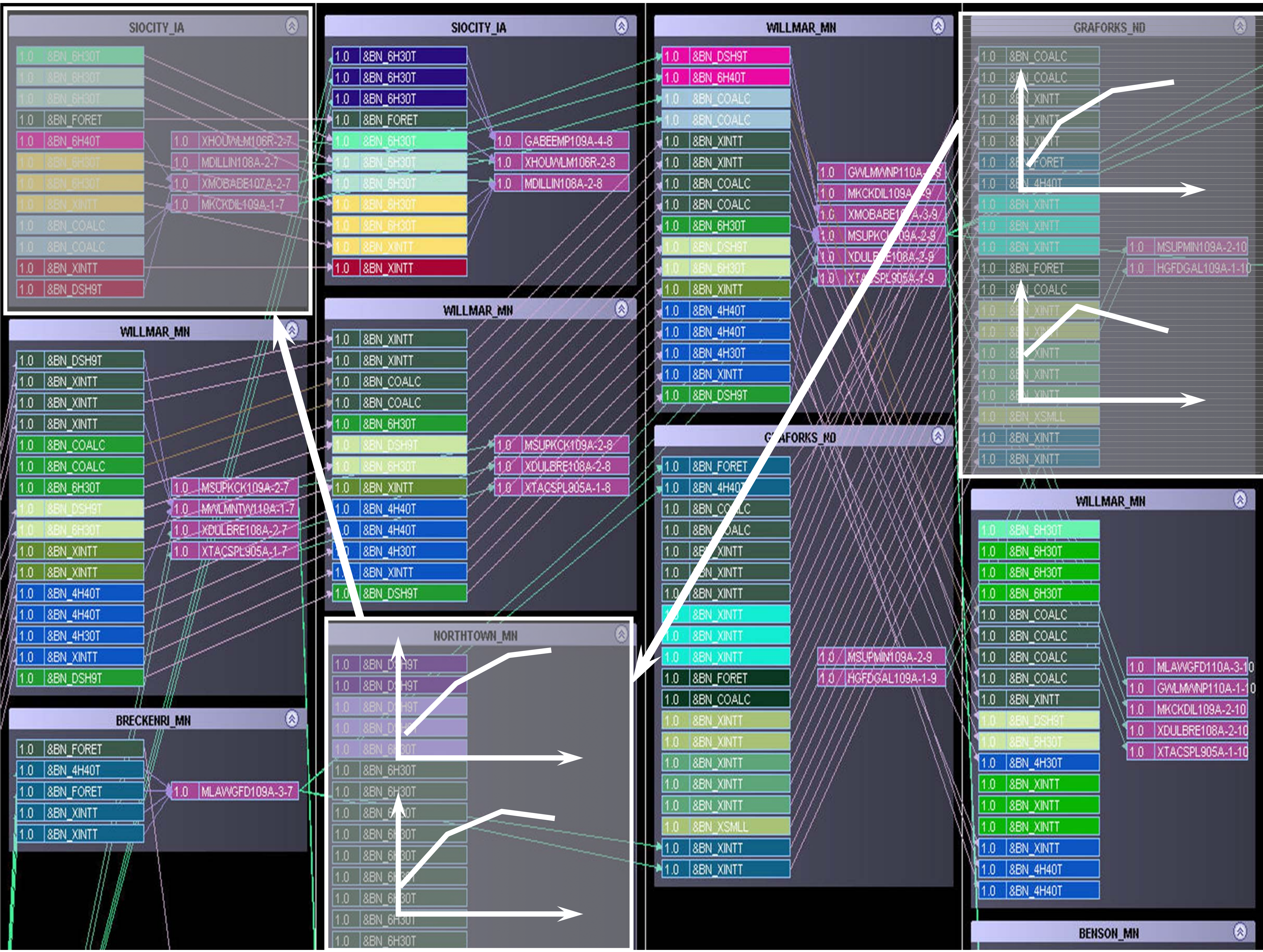

SIOCITY_IA
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_FORET
1.0 8BN_6H40T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 XHOUWLM106R-2-7
1.0 MDILLIN108A-2-7
1.0 XMOBABE107A-2-7
1.0 MKCKDIL109A-1-7
WILLMAR_MN
1.0 8BN_DSH9T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_6H30T
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_4H40T
1.0 8BN_4H40T
1.0 8BN_4H30T
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 MSUPKCK109A-2-7
1.0 MWLMNTW110A-1-7
1.0 XDULBRE108A-2-7
1.0 XTACSPL905A-1-7
BRECKENRI_MN
1.0 8BN_FORET
1.0 8BN_4H40T
1.0 8BN_FORET
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 MLAWGFD109A-3-7
SIOCITY_IA
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_FORET
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 GABEEMP109A-4-8
1.0 XHOUWLM106R-2-8
1.0 MDILLIN108A-2-8
WILLMAR_MN
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_6H30T
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_4H40T
1.0 8BN_4H40T
1.0 8BN_4H30T
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 MSUPKCK109A-2-8
1.0 XDULBRE108A-2-8
1.0 XTACSPL905A-1-8
NORTHTOWN_MN
1.0 8BN_DSH9T
1.0 8BN_DSH9T
1.0 8BN_DSH9T
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
WILLMAR_MN
1.0 8BN_DSH9T
1.0 8BN_6H40T
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_6H30T
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_4H40T
1.0 8BN_4H40T
1.0 8BN_4H30T
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 XDULBRE108A-2-9
GRAFORKS_ND
1.0 8BN_FORET
1.0 8BN_4H40T
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_FORET
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XSMLL
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 MSUPMIN109A-2-9
1.0 HGFDGAL109A-1-9
GRAFORKS_ND
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_FORET
1.0 8BN_4H40T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_FORET
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XSMLL
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 MSUPMIN109A-2-10
1.0 HGFDGAL109A-1-10
WILLMAR_MN
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_4H30T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_4H40T
1.0 8BN_4H40T
1.0 MLAWGFD110A-3-10
1.0 GWLMWNP110A-1-10
1.0 MKCKDIL109A-2-10
1.0 XDULBRE108A-2-10
1.0 XTACSPL905A-1-10
BENSON_MN



SIOCITY_IA
1.0 8BN_6H30T
1.0 8BN_FORET
1.0 8BN_6H40T
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_DSH9T
1.0 XHOUWLM106R-2-7
1.0 MDILLIN108A-2-7
1.0 XMOBABE107A-2-7
1.0 MKCKDIL109A-1-7
SIOCITY_IA
1.0 GABEEMP109A-4-8
1.0 XHOUWLM106R-2-8
1.0 MDILLIN108A-2-8
WILLMAR_MN
1.0 8BN_4H40T
1.0 8BN_4H30T
1.0 MSUPKCK109A-2-7
1.0 MWLMNTW110A-1-7
1.0 XDULBRE108A-2-7
1.0 XTACSPL905A-1-7
BRECKENRI_MN
1.0 MLAWGFD109A-3-7
1.0 MSUPKCK109A-2-8
1.0 XDULBRE108A-2-8
1.0 XTACSPL905A-1-8
NORTHTOWN_MN
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 GWLMWNP110A-3-9
1.0 MKCKDIL109A-1-9
1.0 XMOBABE107A-3-9
1.0 MSUPKCK109A-2-9
1.0 XDULBRE108A-2-9
1.0 XTACSPL905A-1-9
GRAFORKS_ND
1.0 8BN_XSMLL
1.0 MSUPMIN109A-2-9
1.0 HGFDGAL109A-1-9
1.0 MSUPMIN109A-2-10
1.0 HGFDGAL109A-1-10
1.0 MLAWGFD110A-3-10
1.0 GWLMWNP110A-1-10
1.0 MKCKDIL109A-2-10
1.0 XDULBRE108A-2-10
1.0 XTACSPL905A-1-10
BENSON_MN

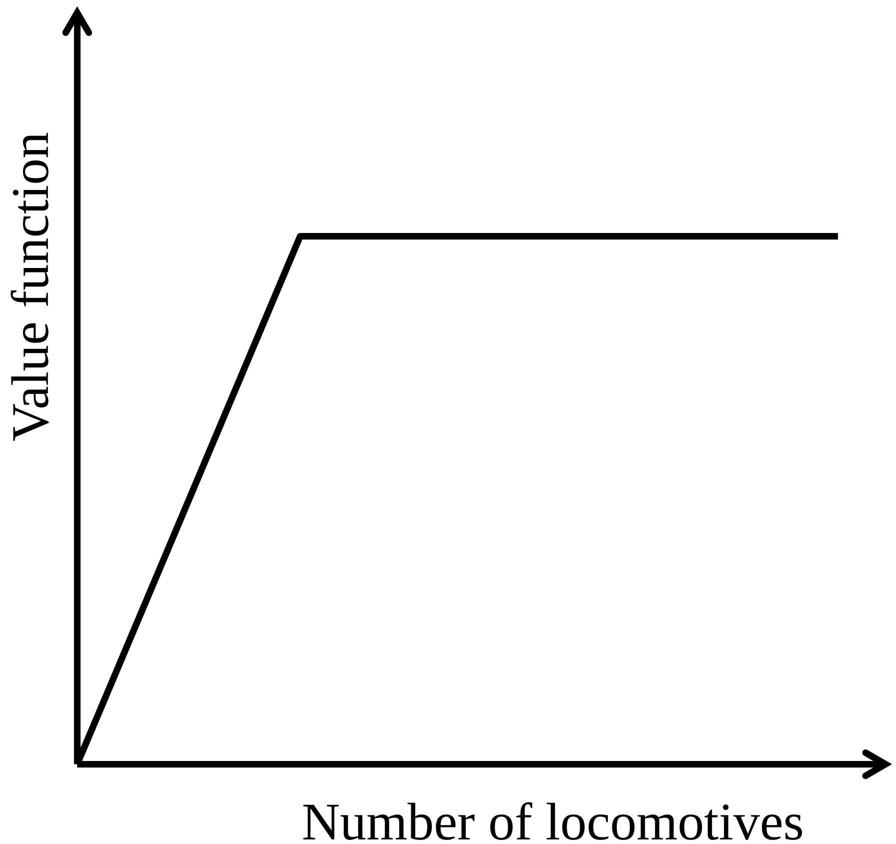


SIOCITY_IA
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_FORET
1.0 8BN_6H40T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 XHOUWLM106R-2-7
1.0 MDILLIN108A-2-7
1.0 XMOBABE107A-2-7
1.0 MKCKDIL109A-1-7
SIOCITY_IA
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_FORET
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 GABEEMP109A-4-8
1.0 XHOUWLM106R-2-8
1.0 MDILLIN108A-2-8
WILLMAR_MN
1.0 8BN_DSH9T
1.0 8BN_6H40T
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_6H30T
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_4H40T
1.0 8BN_4H40T
1.0 8BN_4H30T
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 GWLMWNP110A-3-9
1.0 MKCKDIL109A-1-9
1.0 XMOBABE107A-3-9
1.0 MSUPKCK109A-2-9
1.0 XDULBRE108A-2-9
1.0 XTACSPL905A-1-9
GRAFORKS_ND
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_FORET
1.0 8BN_4H40T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_FORET
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XSMLL
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 MSUPMIN109A-2-10
1.0 HGFDGAL109A-1-10
WILLMAR_MN
1.0 8BN_DSH9T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_6H30T
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_4H40T
1.0 8BN_4H40T
1.0 8BN_4H30T
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 MSUPKCK109A-2-7
1.0 MWLMNTW110A-1-7
1.0 XDULBRE108A-2-7
1.0 XTACSPL905A-1-7
WILLMAR_MN
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_6H30T
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_XINTT
1.0 8BN_4H40T
1.0 8BN_4H40T
1.0 8BN_4H30T
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 MSUPKCK109A-2-8
1.0 XDULBRE108A-2-8
1.0 XTACSPL905A-1-8
GRAFORKS_ND
1.0 8BN_FORET
1.0 8BN_4H40T
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_FORET
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XSMLL
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 MSUPMIN109A-2-9
1.0 HGFDGAL109A-1-9
WILLMAR_MN
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_COALC
1.0 8BN_XINTT
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_4H30T
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 8BN_4H40T
1.0 8BN_4H40T
1.0 MLAWGFD110A-3-10
1.0 GWLMWNP110A-1-10
1.0 MKCKDIL109A-2-10
1.0 XDULBRE108A-2-10
1.0 XTACSPL905A-1-10
BRECKENRI_MN
1.0 8BN_FORET
1.0 8BN_4H40T
1.0 8BN_FORET
1.0 8BN_XINTT
1.0 8BN_XINTT
1.0 MLAWGFD109A-3-7
NORTHTOWN_MN
1.0 8BN_DSH9T
1.0 8BN_DSH9T
1.0 8BN_DSH9T
1.0 8BN_DSH9T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
1.0 8BN_6H30T
BENSON_MN

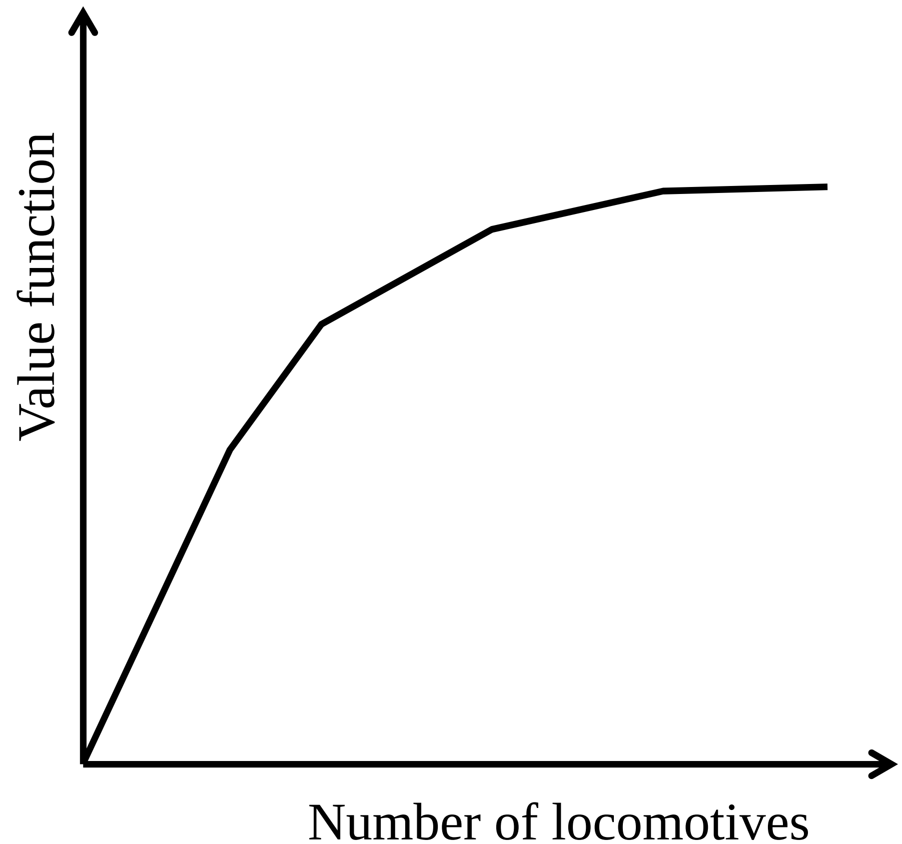

SIOCITY_IA
1.0 8BN_6H30T
1.0 8BN_FORET
1.0 8BN_6H40T
1.0 8BN_XINTT
1.0 8BN_COALC
1.0 8BN_DSH9T
1.0 XHOUWLM106R-2-7
1.0 MDILLIN108A-2-7
1.0 XMOBABE107A-2-7
1.0 MKCKDIL109A-1-7
SIOCITY_IA
1.0 GABEEMP109A-4-8
1.0 XHOUWLM106R-2-8
1.0 MDILLIN108A-2-8
WILLMAR_MN
1.0 8BN_4H40T
1.0 8BN_4H30T
1.0 MSUPKCK109A-2-7
1.0 MWLMNTW110A-1-7
1.0 XDULBRE108A-2-7
1.0 XTACSPL905A-1-7
WILLMAR_MN
1.0 MSUPKCK109A-2-8
1.0 XDULBRE108A-2-8
1.0 XTACSPL905A-1-8
BRECKENRI_MN
1.0 MLAWGFD109A-3-7
NORTHTOWN_MN
WILLMAR_MN
GRAFORKS_ND
1.0 MSUPMIN109A-2-9
1.0 HGFDGAL109A-1-9
GRAFORKS_ND
1.0 8BN_XSMLL
1.0 MSUPMIN109A-2-10
1.0 HGFDGAL109A-1-10
WILLMAR_MN
1.0 MLAWGFD110A-3-10
1.0 GWLMWNP110A-1-10
1.0 MKCKDIL109A-2-10
1.0 XDULBRE108A-2-10
1.0 XTACSPL905A-1-10
BENSON_MN

Stochastic optimization

- Deterministic training

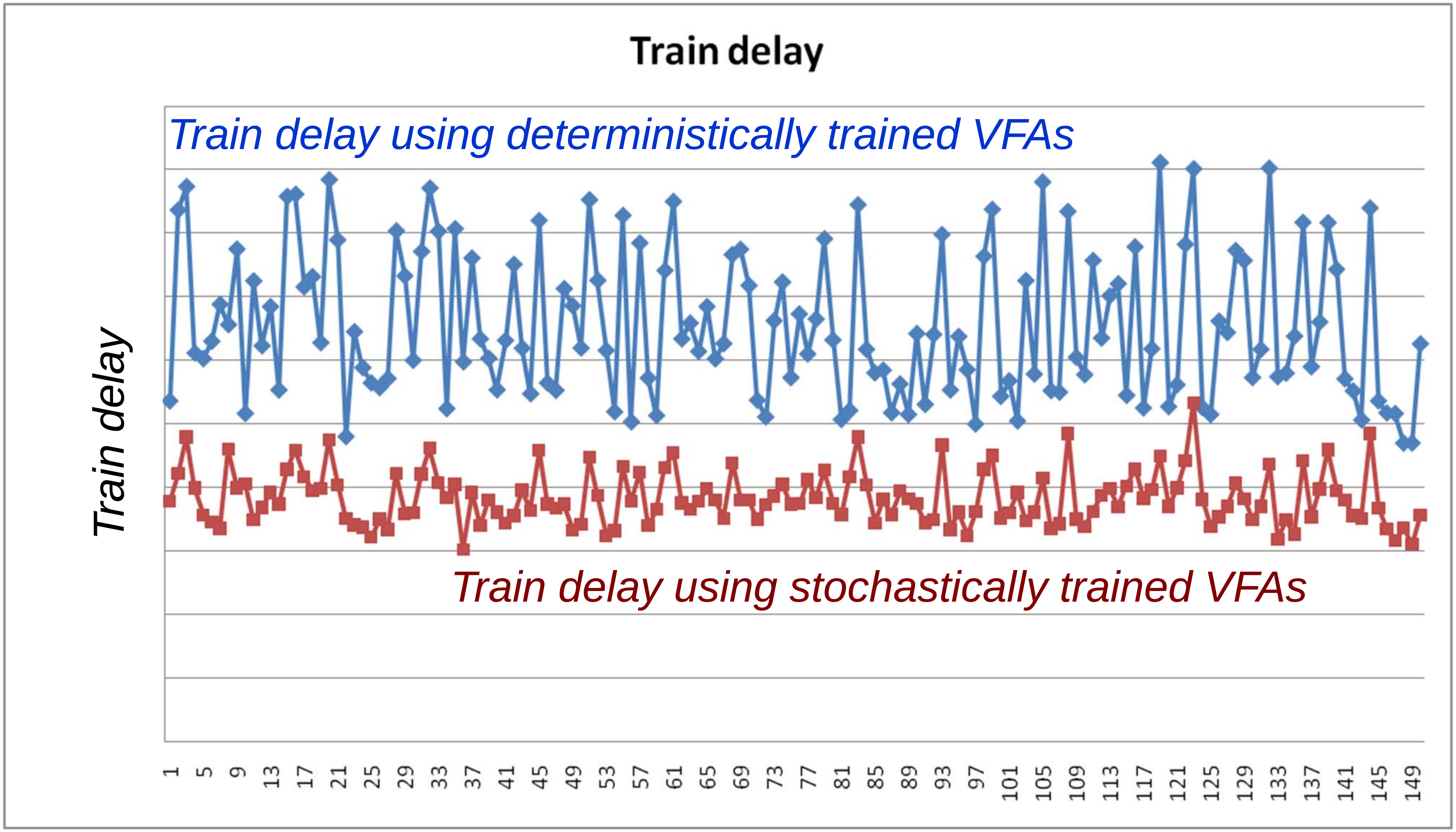


- Stochastic training



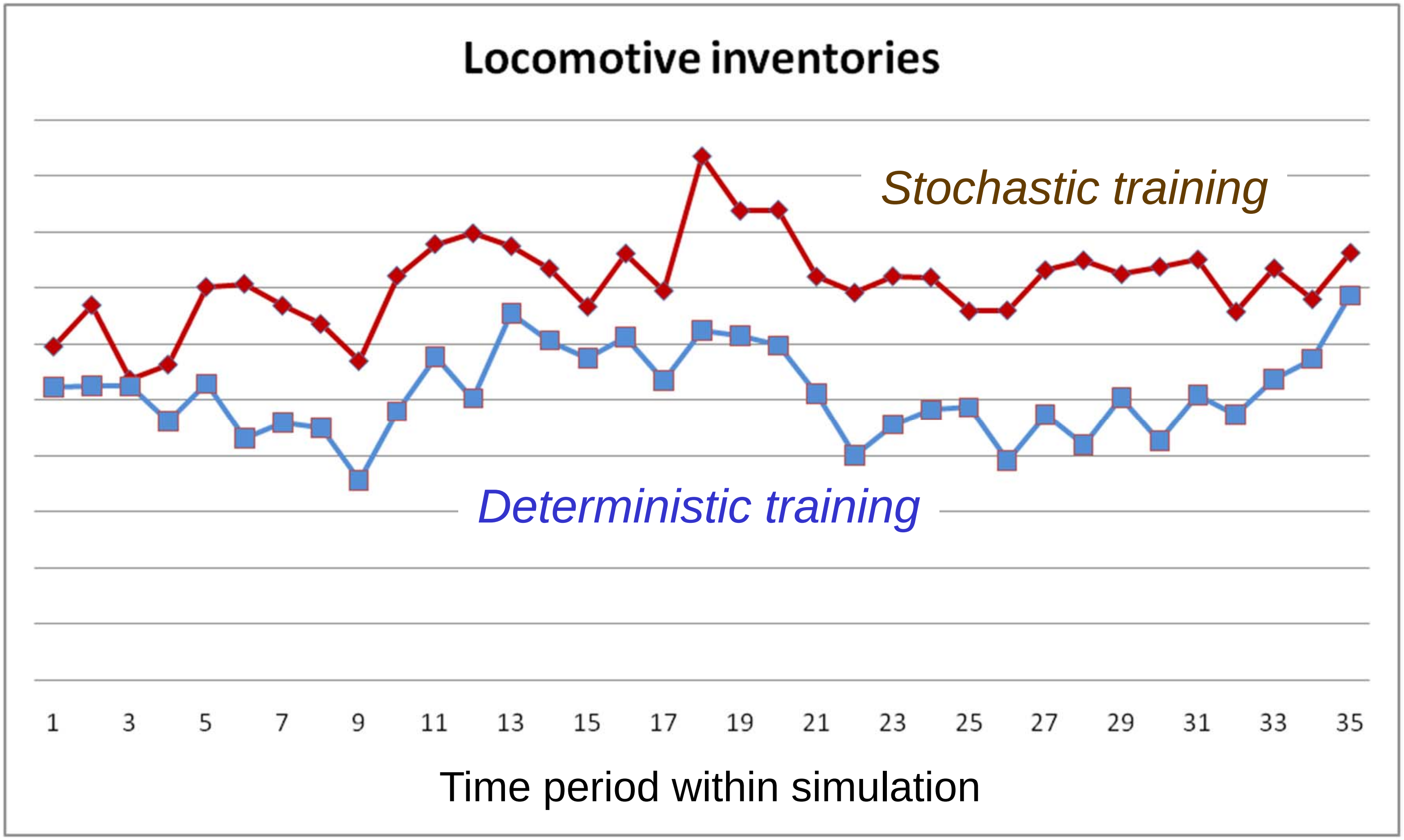
Stochastic optimization

- Train delay with uncertain transit times and yard delays



Laboratory testing

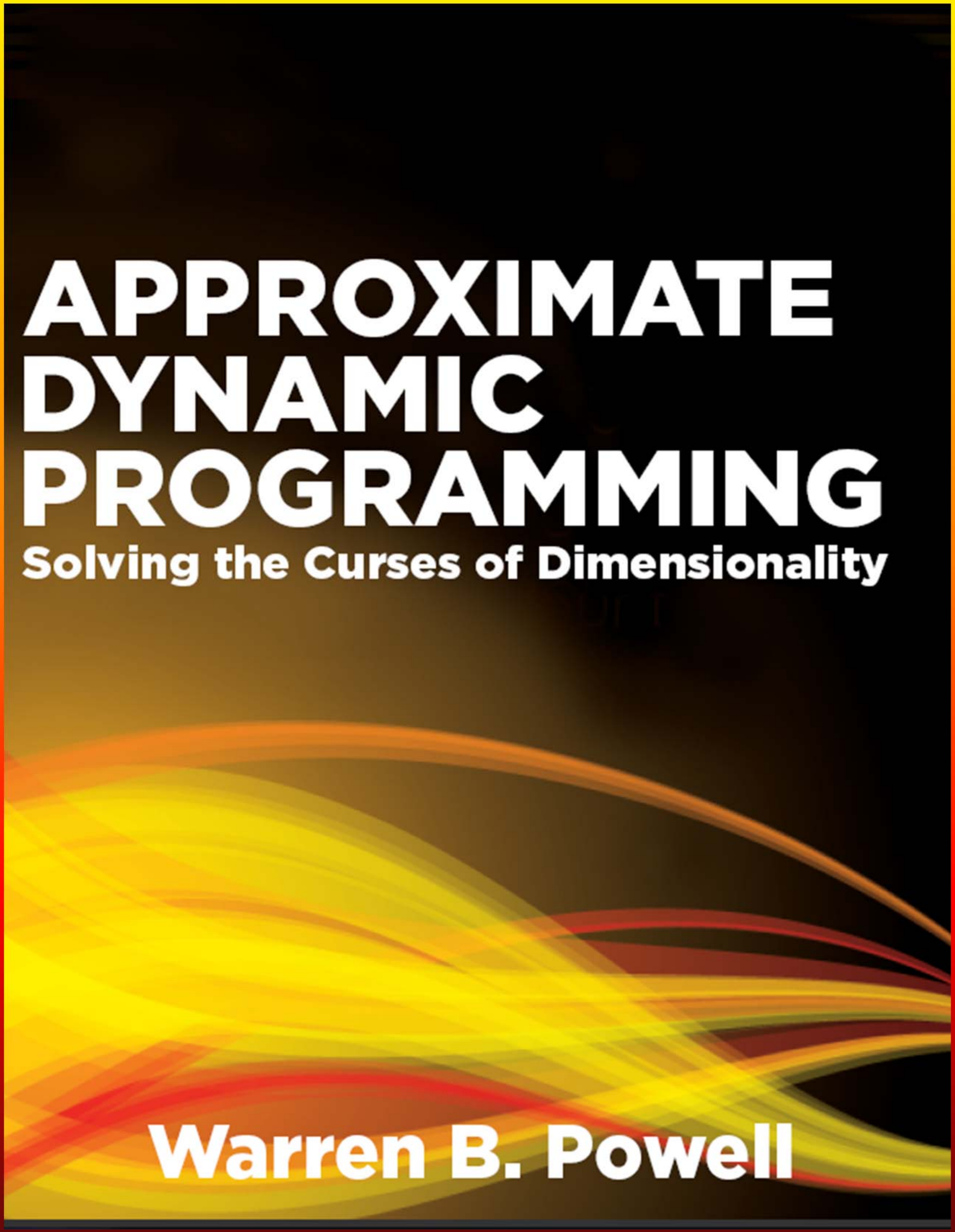
- How do we do it?



» The stochastic model keeps more power in inventory. The challenge is knowing when and where.

Other ADP projects

- Schneider National
  - » Optimizes assignments of 6,000 drivers over 30 days, 50,000 variables per time period. Drivers modeled using 15 dimensional attribute vector. Model closely matches historical performance.
- Embraer
  - » Optimize inventories of 700 high value spare parts over multiple locations, balancing cost and service.
- Netjets
  - » Optimize the fleet mix of 15 types of aircraft over 20 years, capturing daily variations in demand and equipment substitutions.
- Car distribution at Norfolk Southern
  - » Optimize thousands of freight cars over multiweek horizon capturing uncertainty in demands and transit times, while modeling detailed information process.



- Second edition!
  - Due September, 2011.
  - Major revision. Complete restructuring and rewriting of middle third of the book.
  - 300 new/rewritten pages
  - Covers policy search in depth, along with high structured approach to value function approximations.