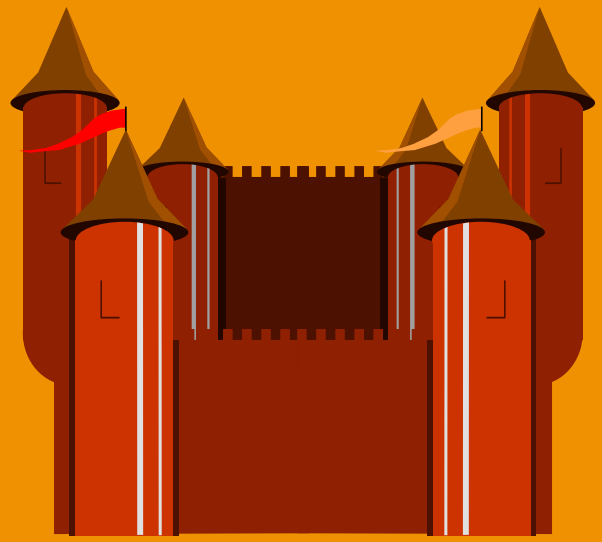
Approximate Dynamic Programming for High-Dimensional Problems

2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning
April, 2007



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

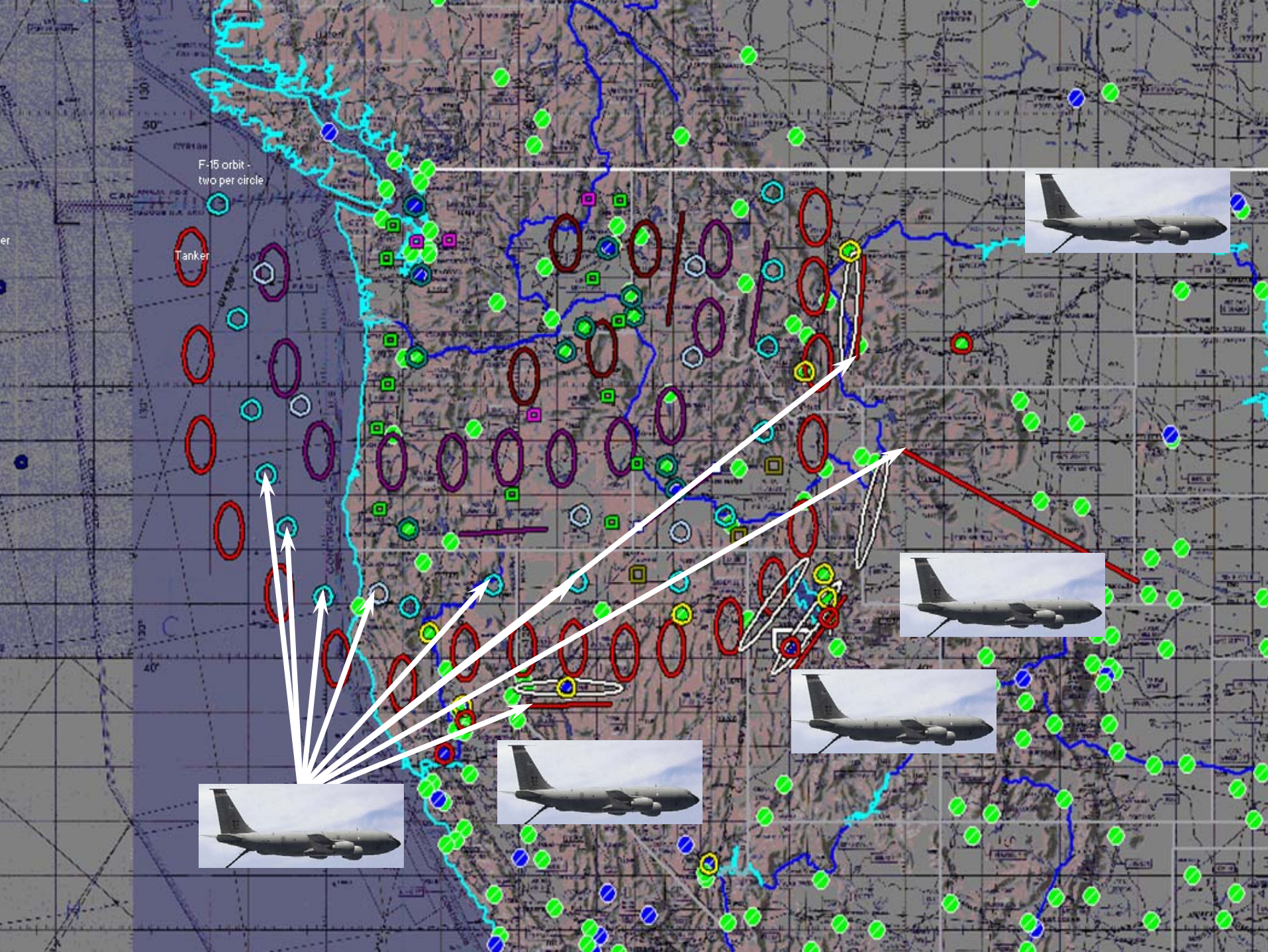

4448
4448
BNSF
BNSF
Burlington Northern
Santa Fe
Railway


GRAIN OPERATIONS
LOADED GRAIN TRAINS POWERED
BNSF Coal Business Unit
Intermodal
On-Time Performance
Industrial Products
Service Summary
By Business Unit

The fractional jet ownership industry


NetJets Inc.

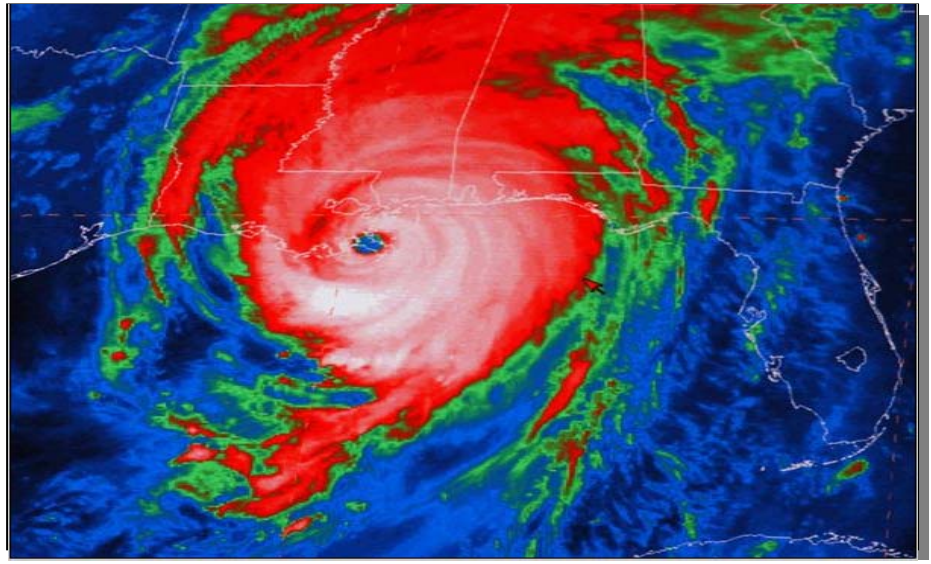

05
22

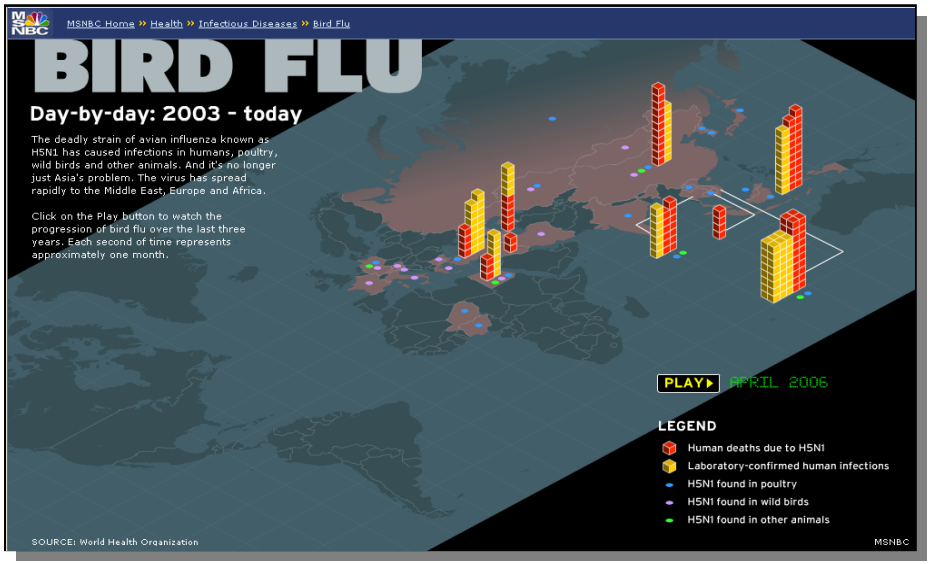

F-15 orbit -
two per circle
Tanker

Planning for a risky world

Weather

- Robust design of emergency response networks.
- Design of financial instruments to hedge against weather emergencies to protect individuals, companies and municipalities.
- Design of sensor networks and communication systems to manage responses to major weather events.

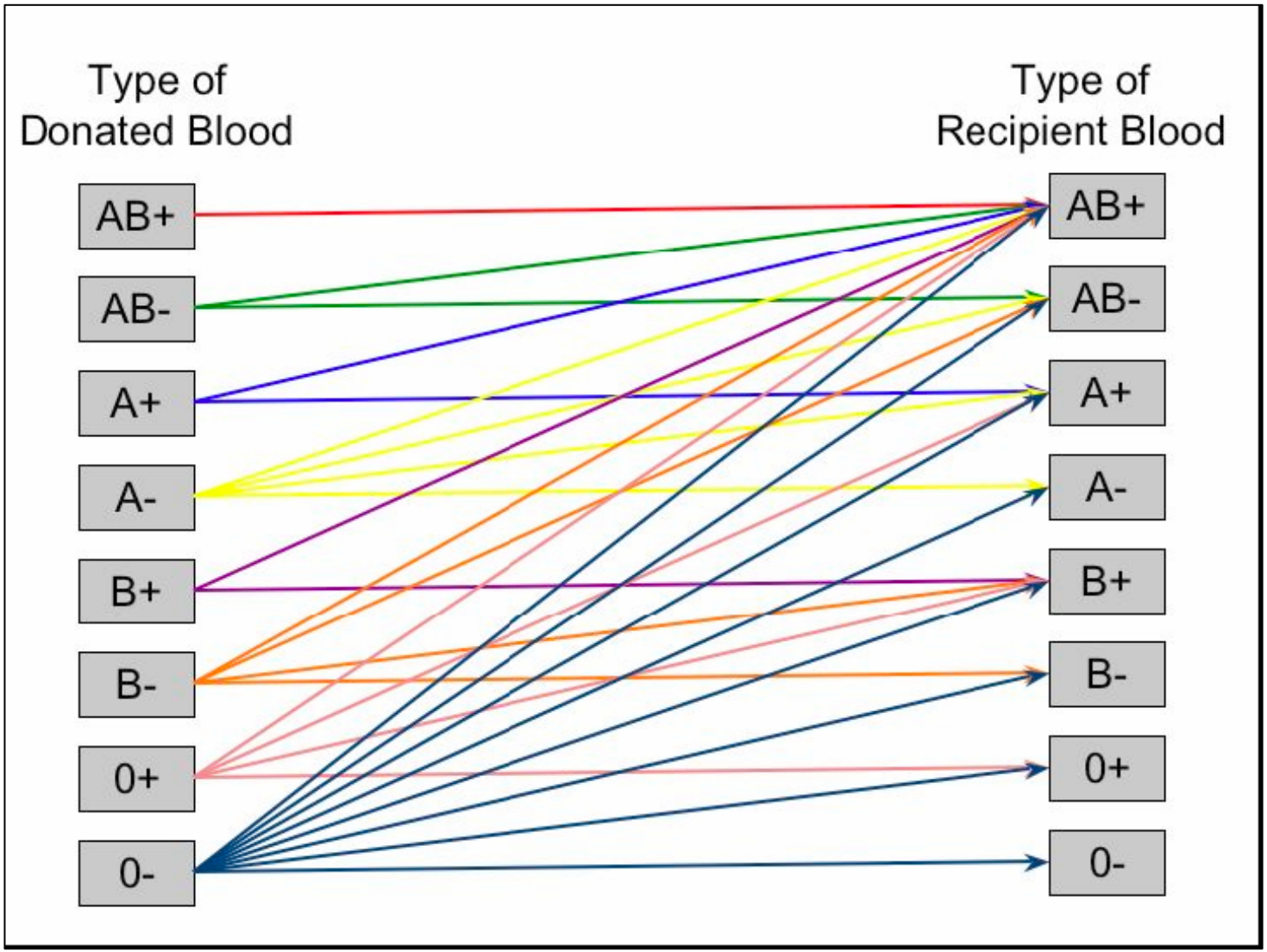




Disease

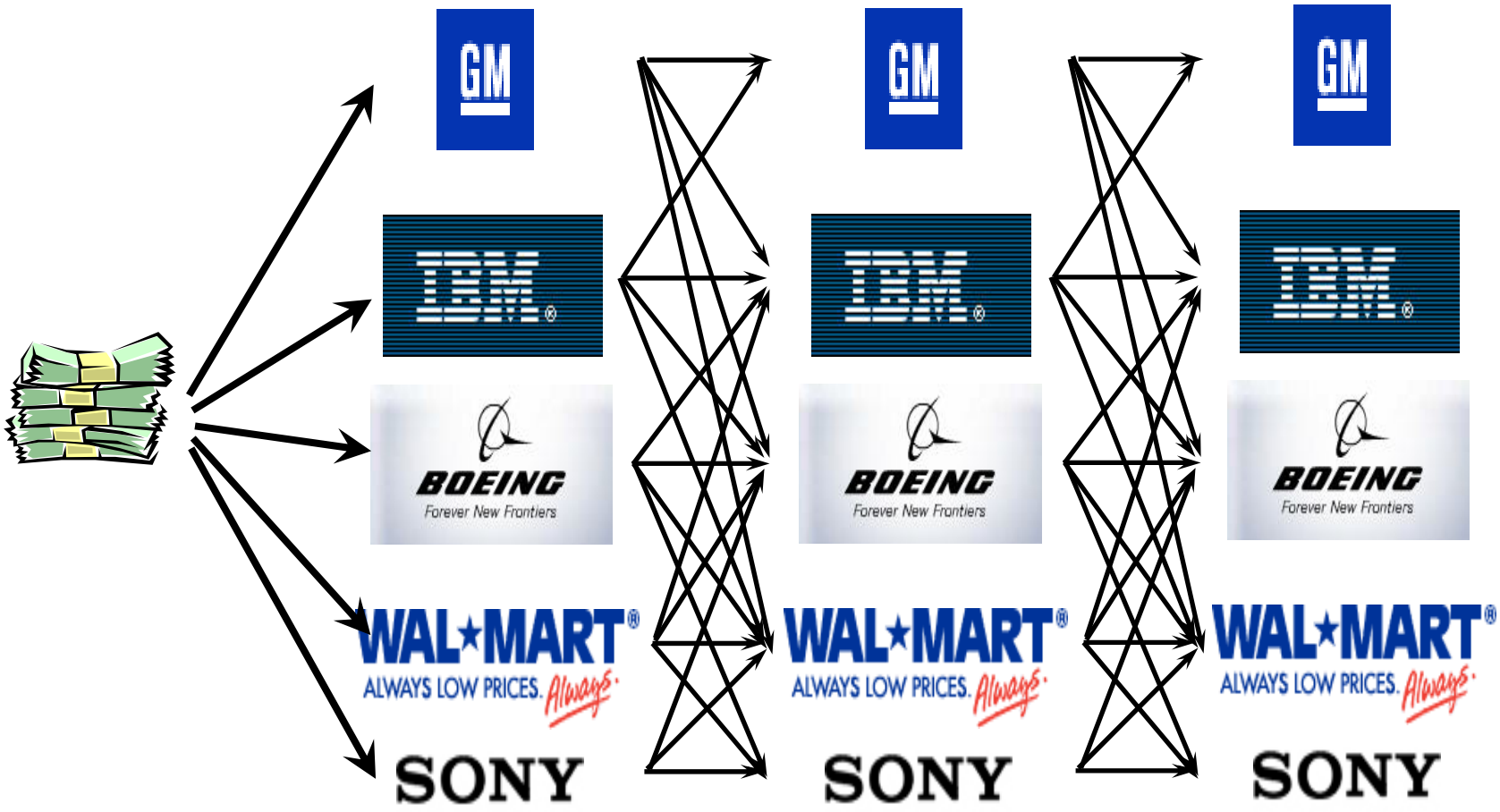
- Models of disease propagation for response planning.
- Management of medical personnel, equipment and vaccines to respond to a disease outbreak.
- Robust design of supply chains to mitigate the disruption of transportation systems.

Blood management



Managing financial portfolios

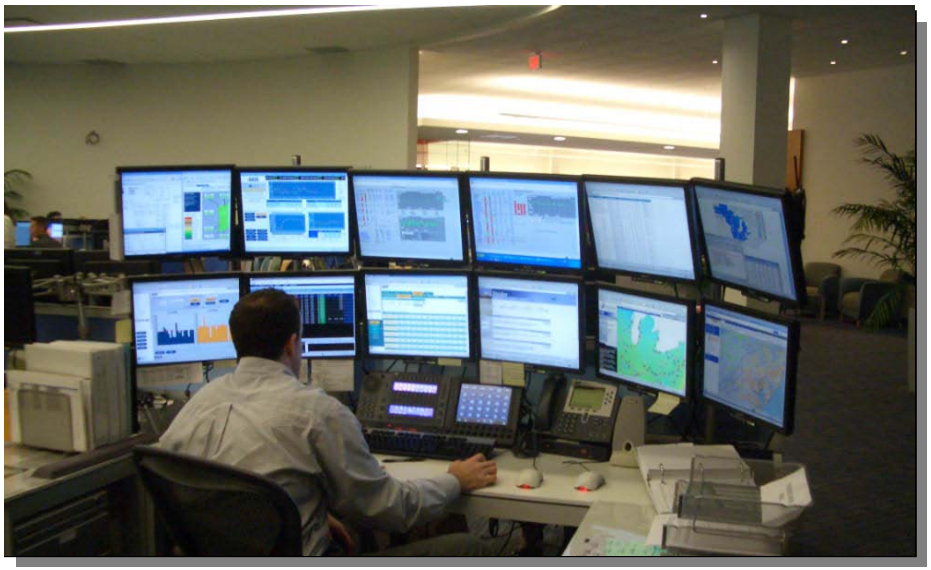
- Money can be invested and then reinvested….



Energy management

Applications

- Jet fuel hedging – Designing strategies to hedge against fluctuations in jet fuel (and other commodities).
- Valuing energy contracts
- Planning the use of future technologies
- R&D portfolio management



Research in ADP

- Convergence proofs
- Rate of convergence research
- Design and evaluation of approximation strategies
- Design of advanced approximation strategies


NRG Energy

Challenges

- Real-time control
  - » Scheduling aircraft, pilots, generators, tankers
  - » Pricing stocks, options
- Near-term tactical planning
  - » Can I accept a customer request?
  - » Should I lease equipment?
  - » Do I have to purchase extra energy?
- Strategic planning
  - » What is the right equipment mix?
  - » What is the value of this contract?
  - » What is the value of more reliable aircraft?

Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

Languages

- The languages of “optimization over time”

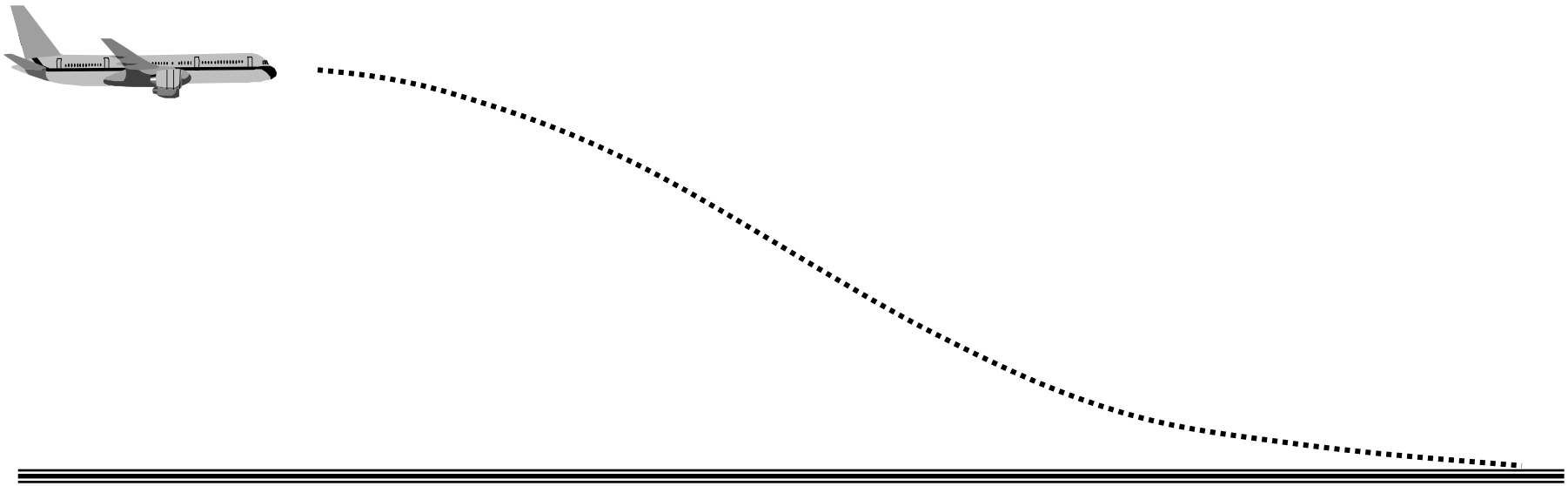
	Engineering	OR/AI/Probability	OR/Math programming
Discipline	Optimal control	Markov decision processes	Stochastic programming
Decision (English)	Control	Action	Decision
Decision (math)	u	a	x
"Value function" (English)	Cost-to-go	Value function	Recourse function
"Value function" (Math)	J	V	Q
State variable	x	S	Huh? (oh, "tenders")
Optimality equations	Hamilton-Jacobi	Bellman	Huh?

Languages

- "Approximate dynamic programming" has been discovered independently by different communities under different names:
  - Neuro-dynamic programming
  - Reinforcement learning
  - Forward dynamic programming
  - Adaptive dynamic programming
  - Heuristic dynamic programming
  - Iterative dynamic programming

Languages

- How to land a plane:



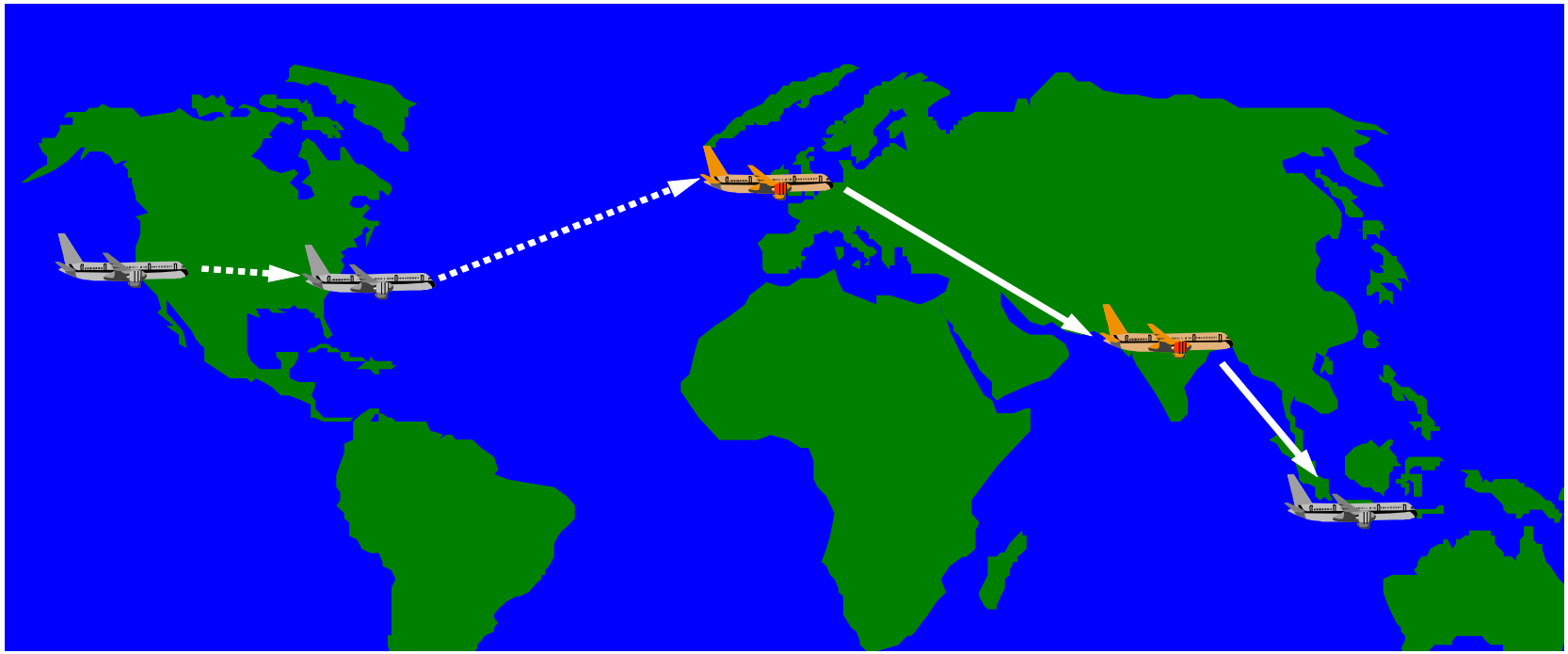
» Control: angle, velocity, acceleration, pitch, yaw…

» Noise: wind, measurement

$$V_t(x_t) = \max_u \left(C(x_t, u_t) + EV_{t+1}(x_{t+1}) \right)$$

Languages

- Where to send a plane:

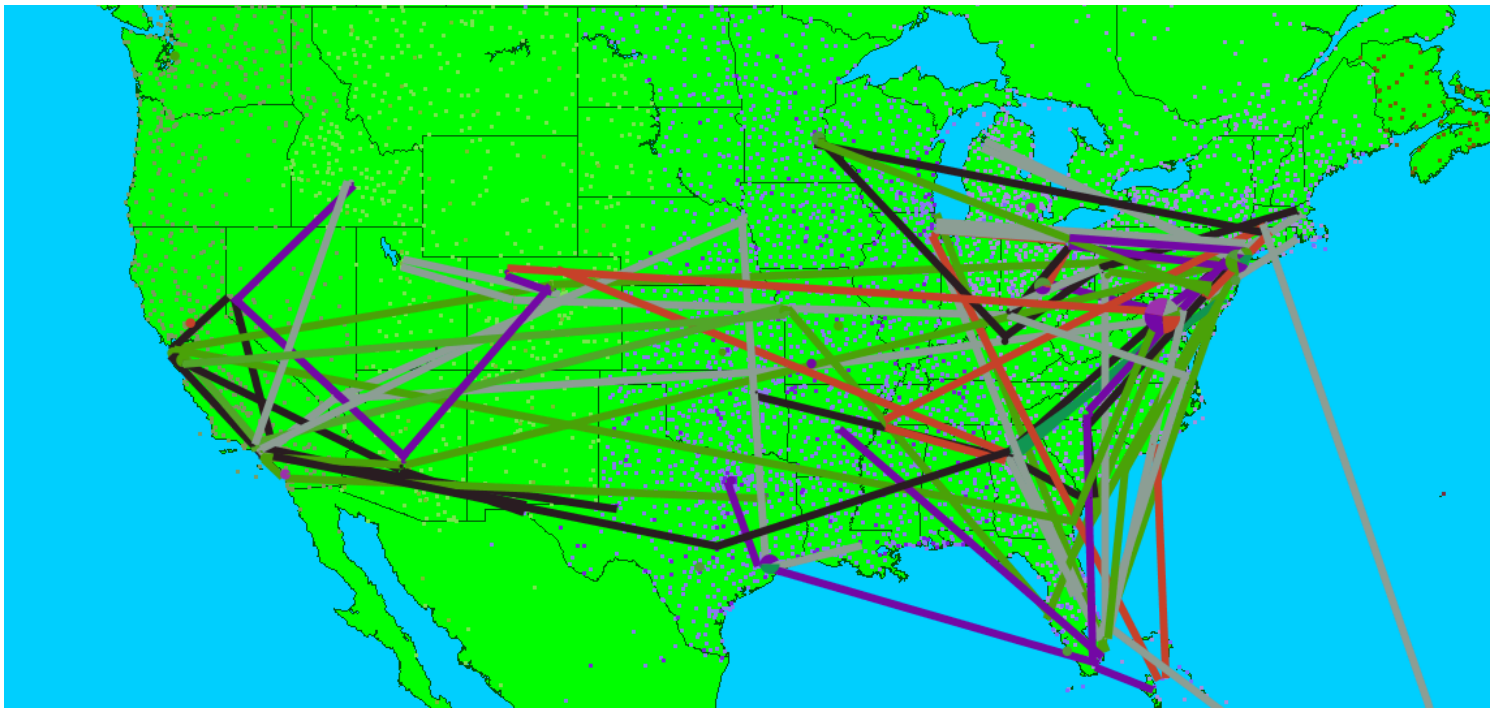


  - Control: Where to send the plane to accomplish a goal.
  - Noise: demands on the system, equipment failures.

$$V_t(S_t) = \max_a \left(C(S_t, a_t) + EV_{t+1}(S_{t+1}) \right)$$

Languages

- How to manage a fleet of aircraft:



  - Control: Which plane to assign to each customer.
  - Noise: demands on the system, equipment failures.

$$V_t(S_t) = \max_x \left(C(S_t, x_t) + EV_{t+1}(S_{t+1}) \right)$$

A progression of models

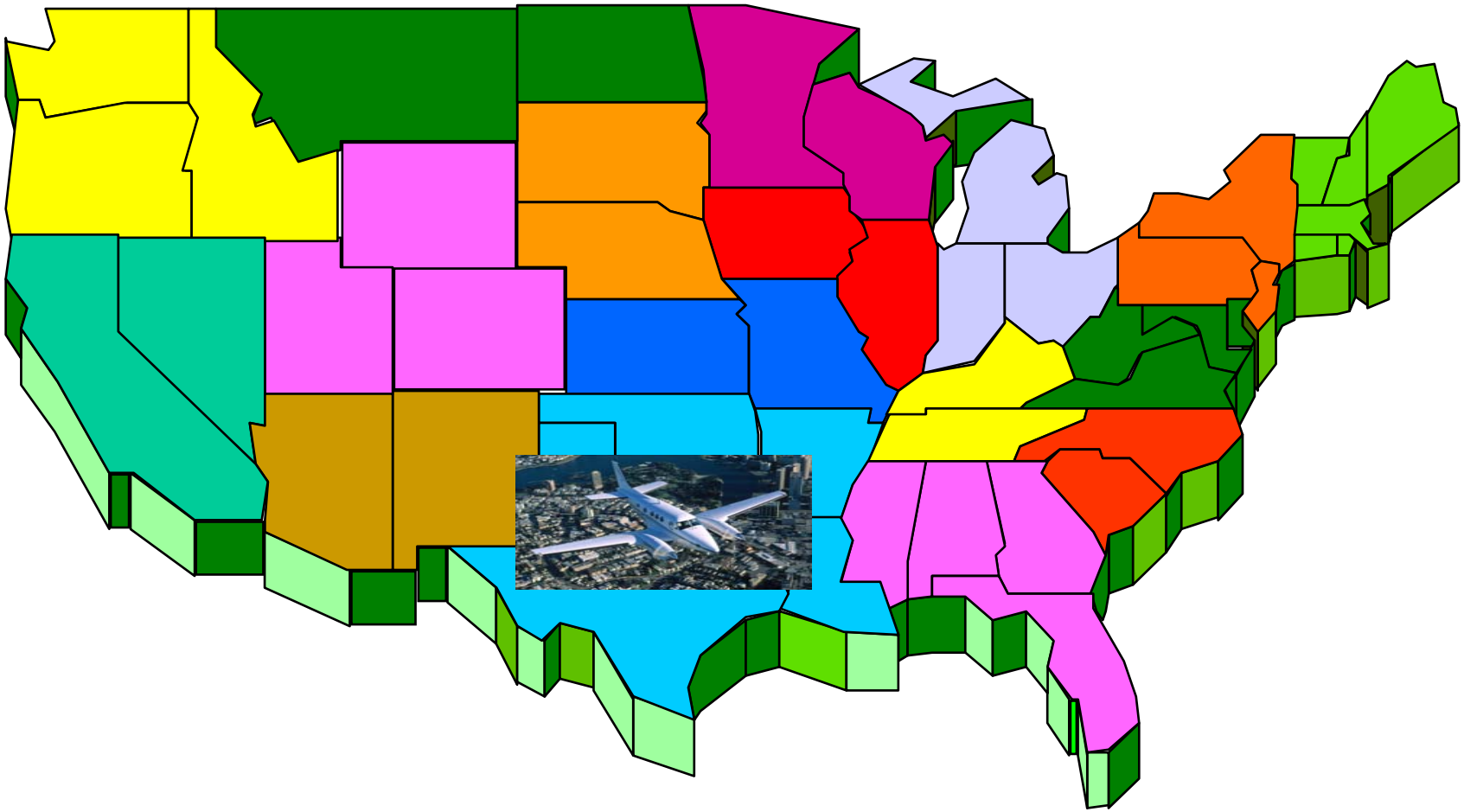
- Major problem classes

	Simple attributes	Complex attributes
Single entity	Textbook Markov decision process	Classical AI applications
Multiple entities	Classical OR applications	Opportunity for combining AI/OR

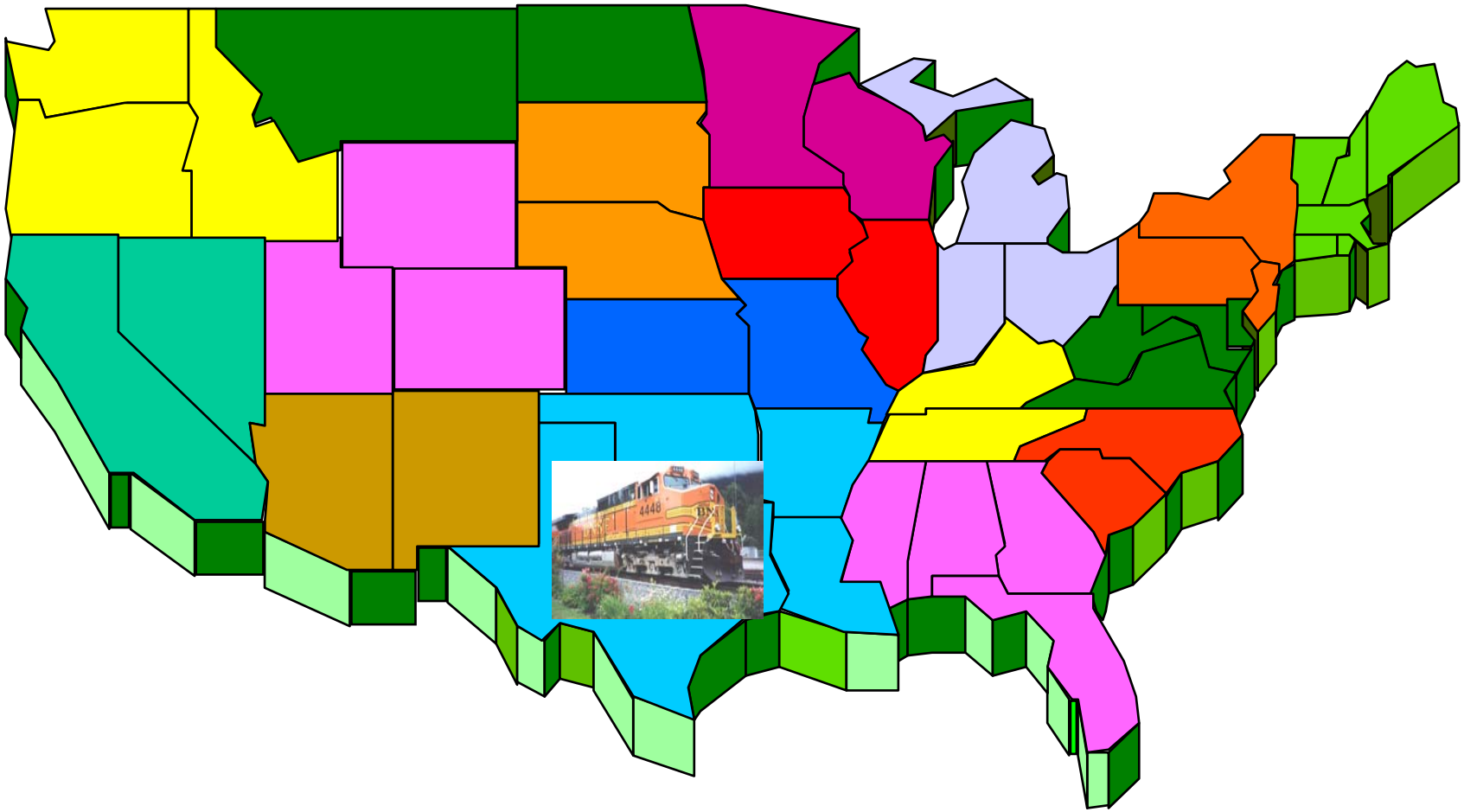
Sample applications

- Single entity problems
  - » Playing a board game
  - » Routing a truck around the country
  - » Planning a set of courses through college
- Storage problems (single resource class)
  - » Maintaining product inventories
  - » Purchasing commodity futures (oil, orange juice, …)
- Managing multiple resource classes
  - » Blood inventories
  - » Fleet management (with different equipment types)
- Managing multiple, discrete resources
  - » Locomotives, jets, people

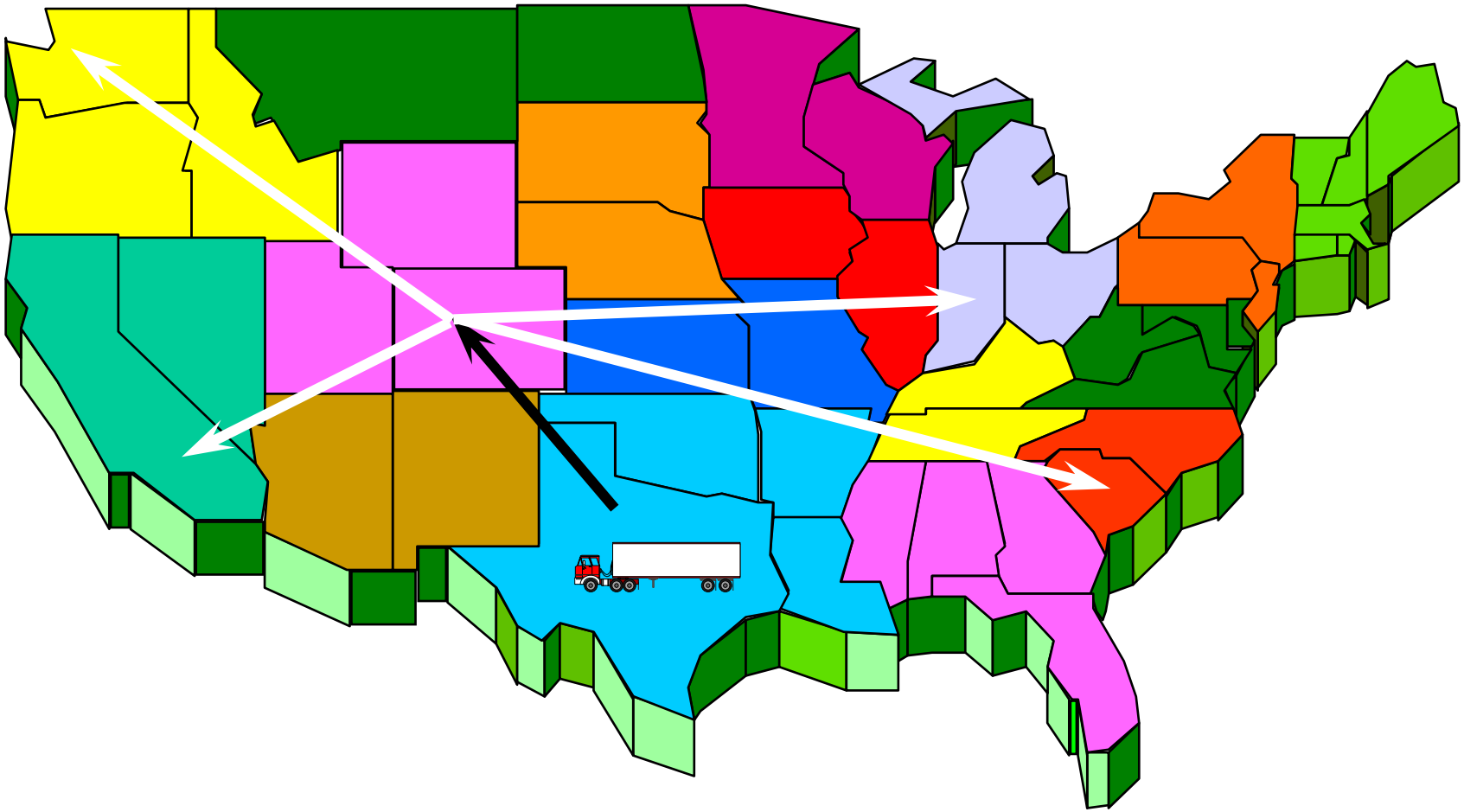
Single entity problems



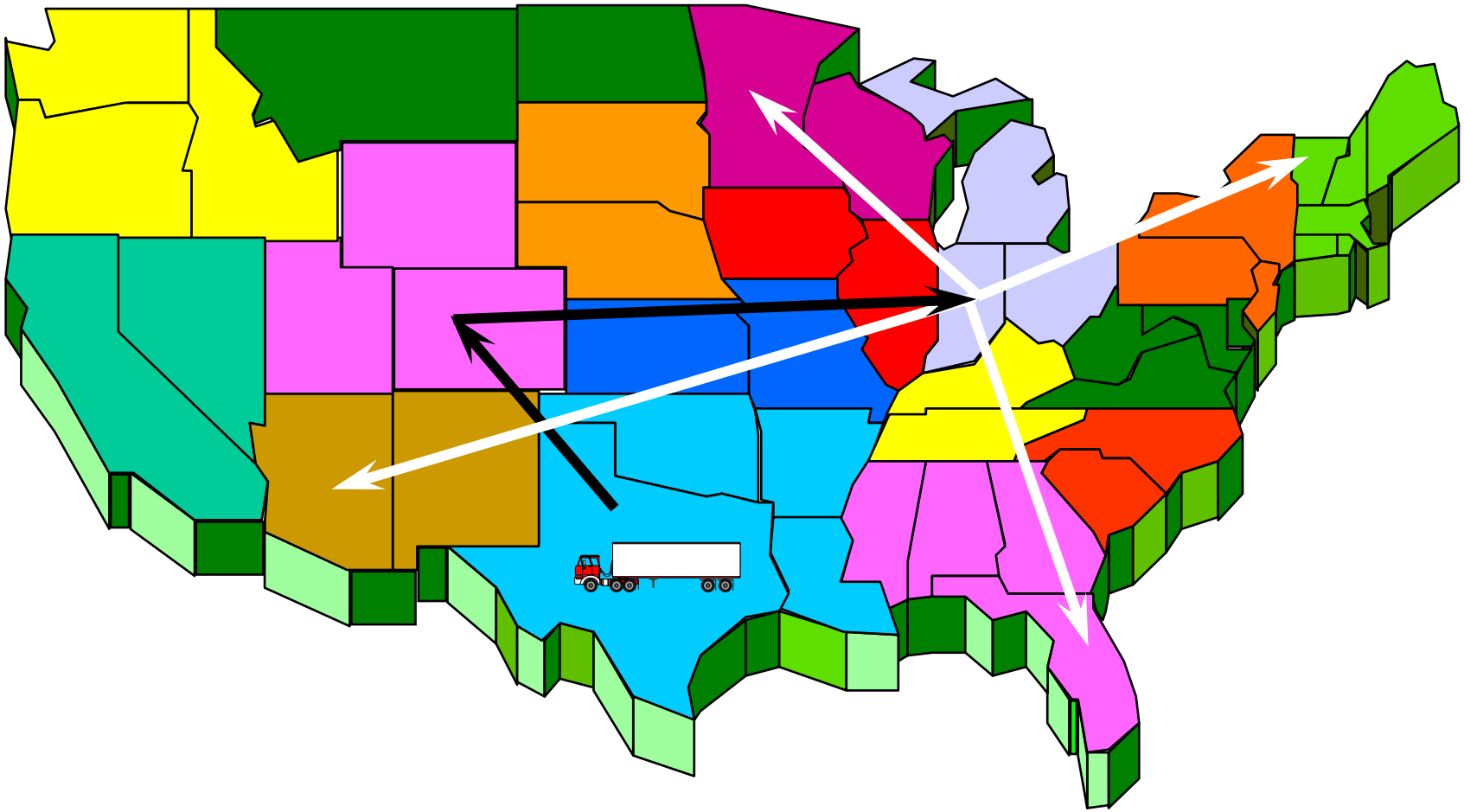
Single entity problems



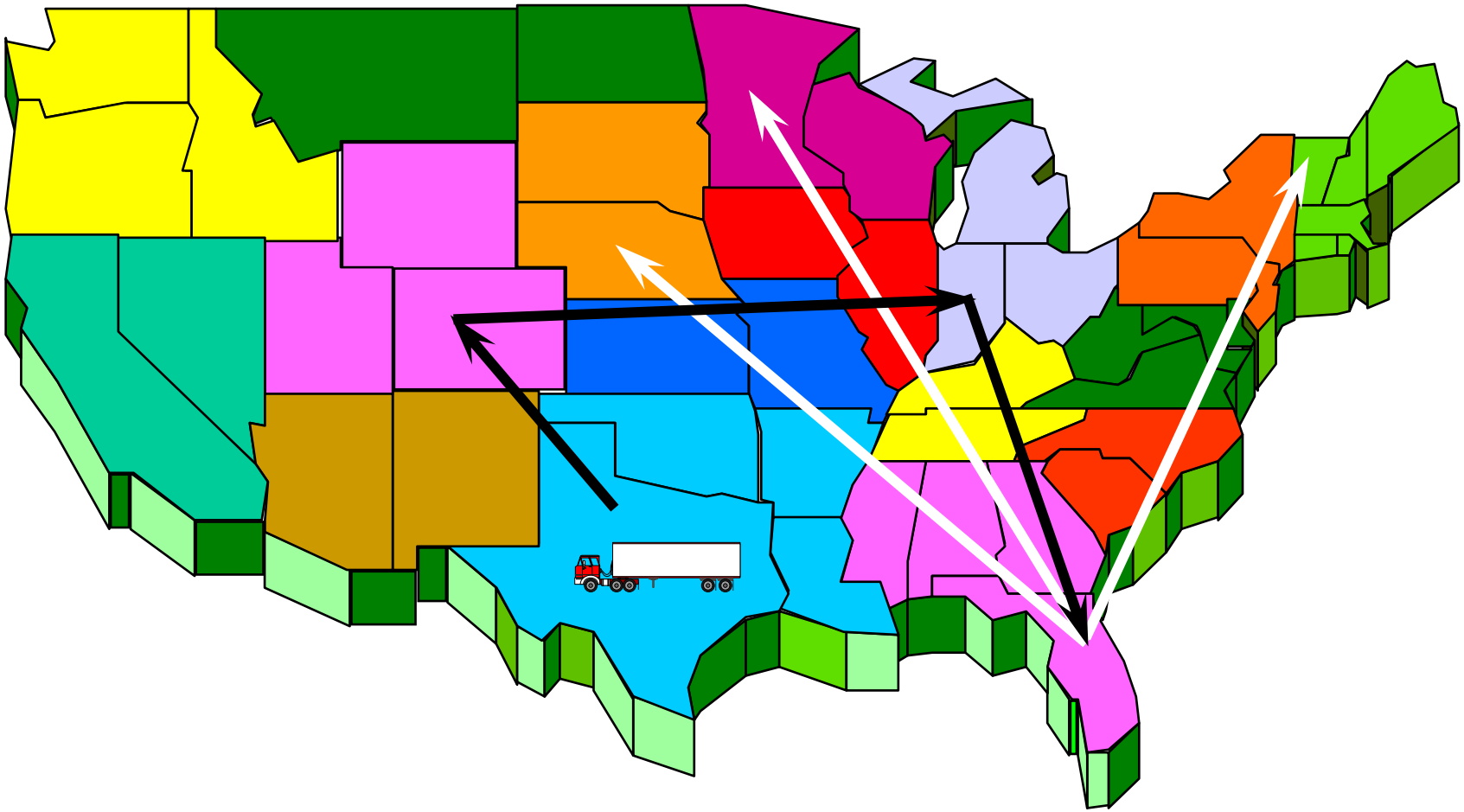
Single entity problems



Single entity problems

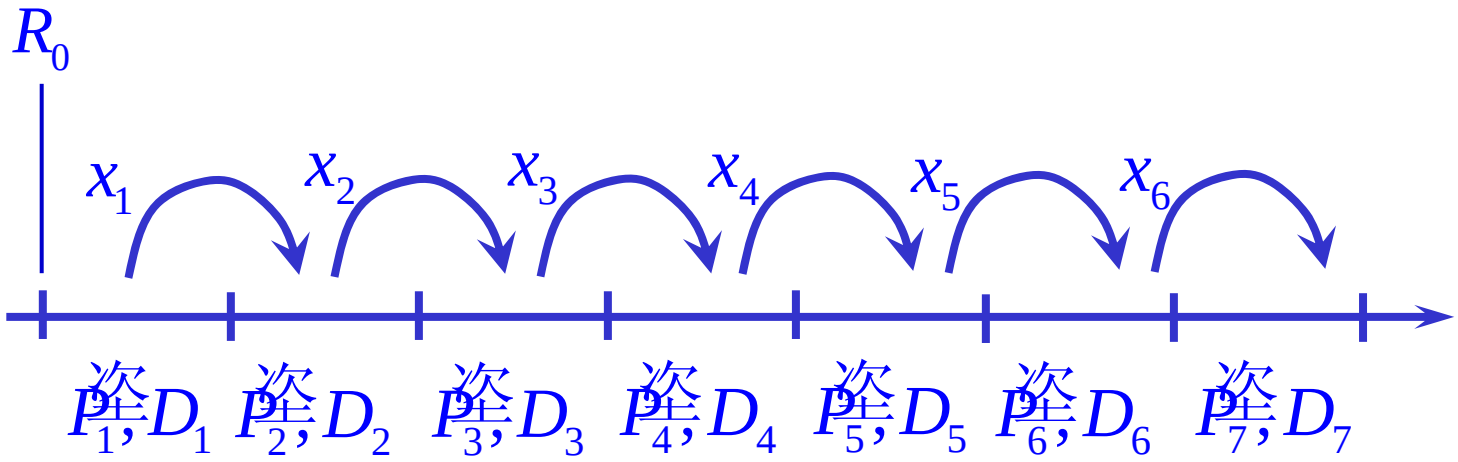


Single entity problems



Asset acquisition problems

- Storage problems

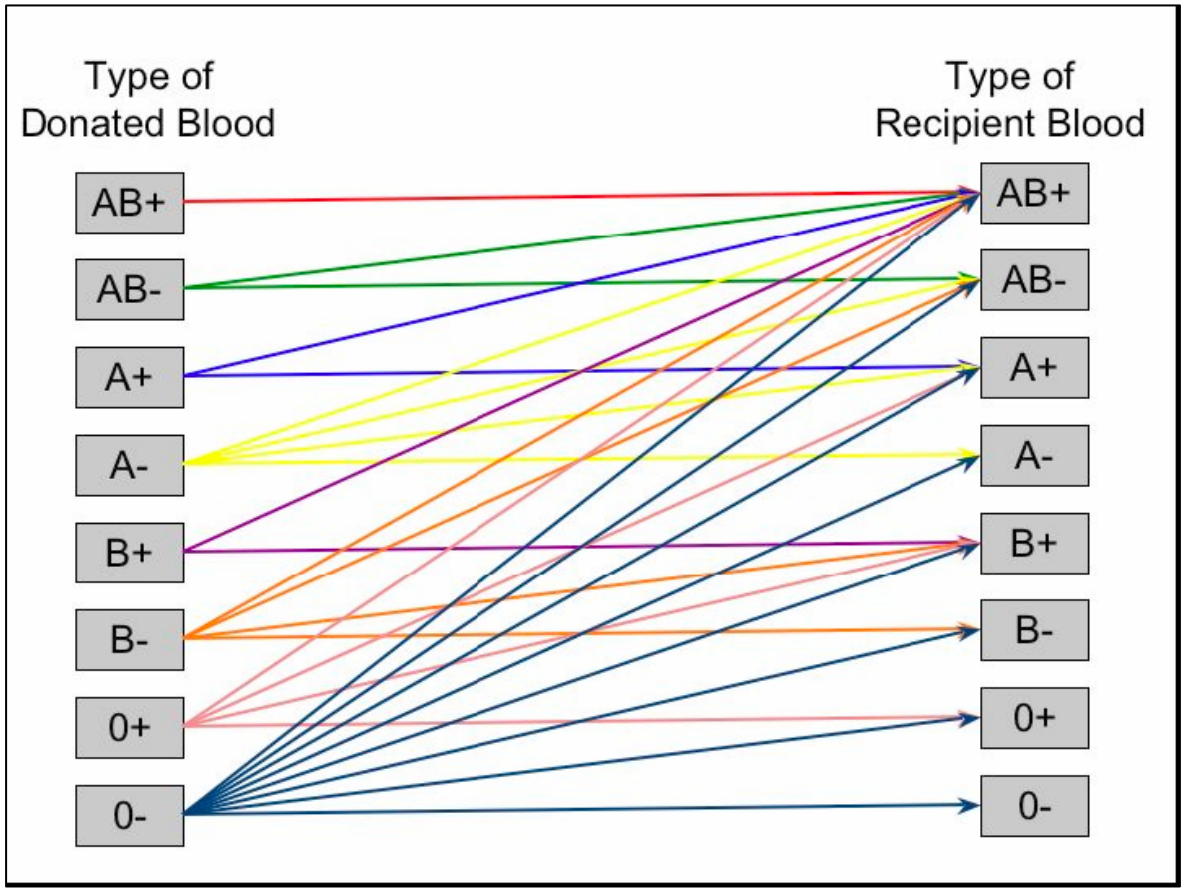


$$R_{t+1} = \left[R_t + x_t - \hat{D}_{t+1} \right]^+$$

$$P_{t+1} = P_t + \hat{P}_t$$

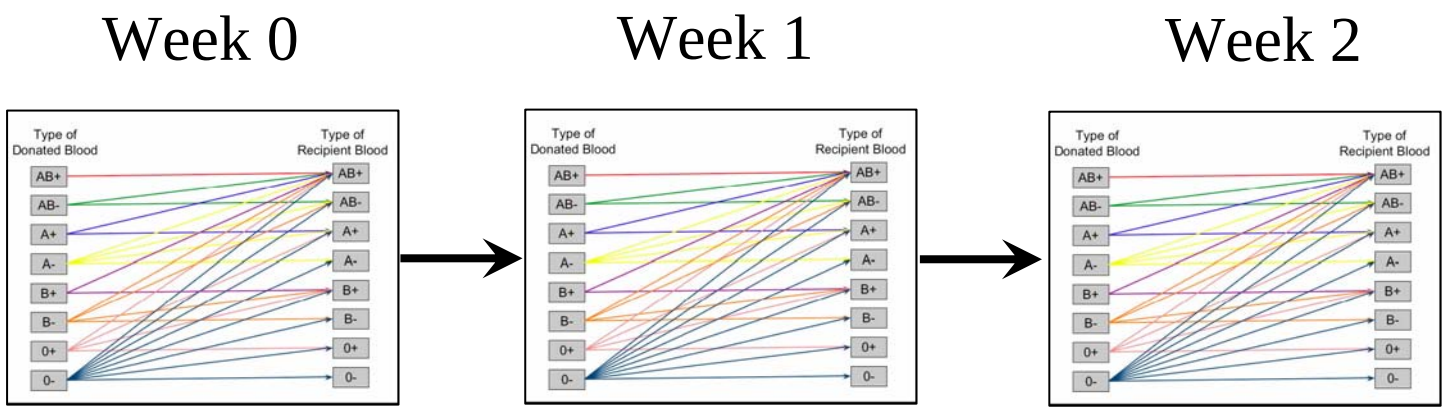
Multiple inventory types

- Managing blood inventories

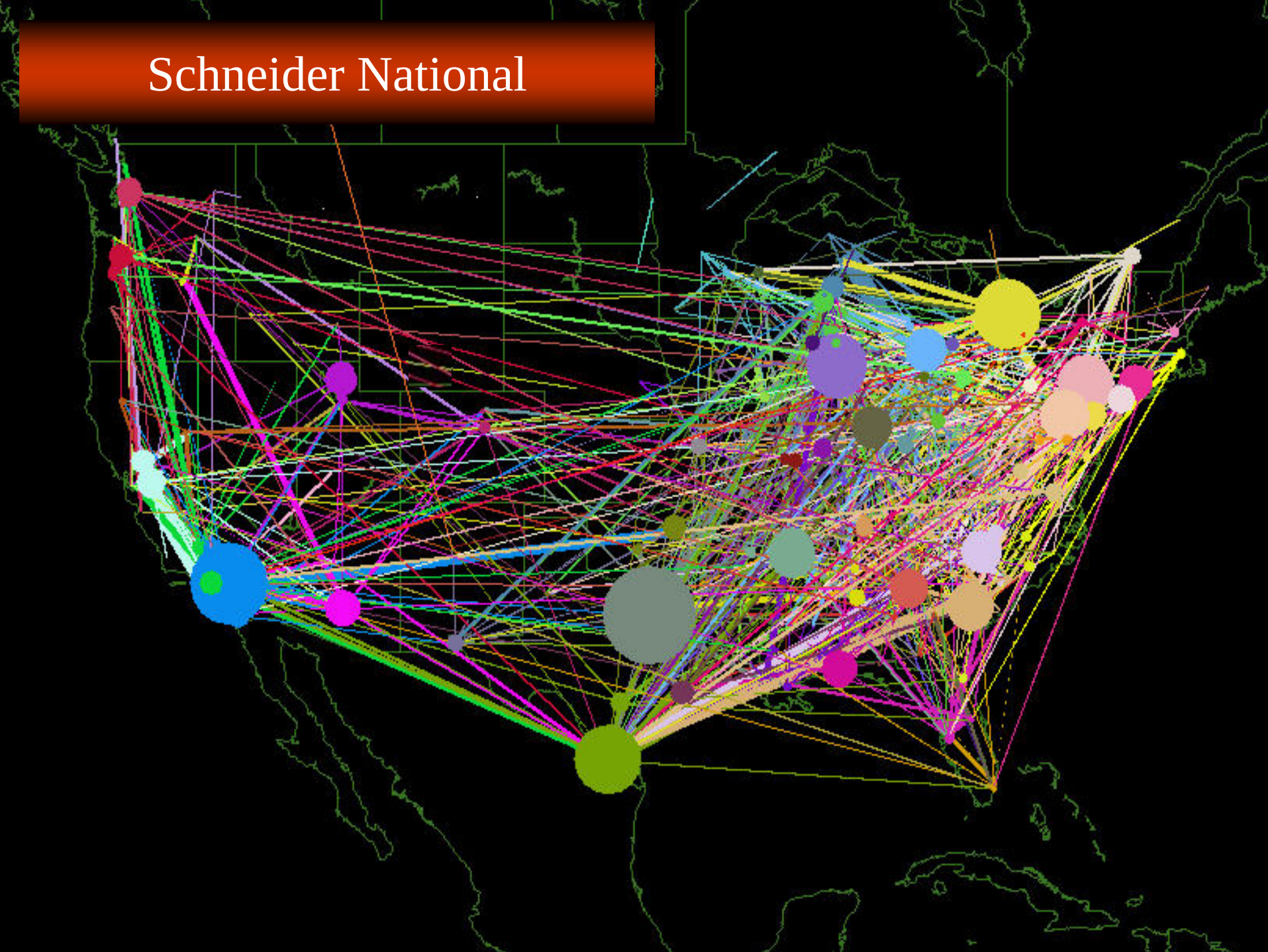


Multiple inventory types

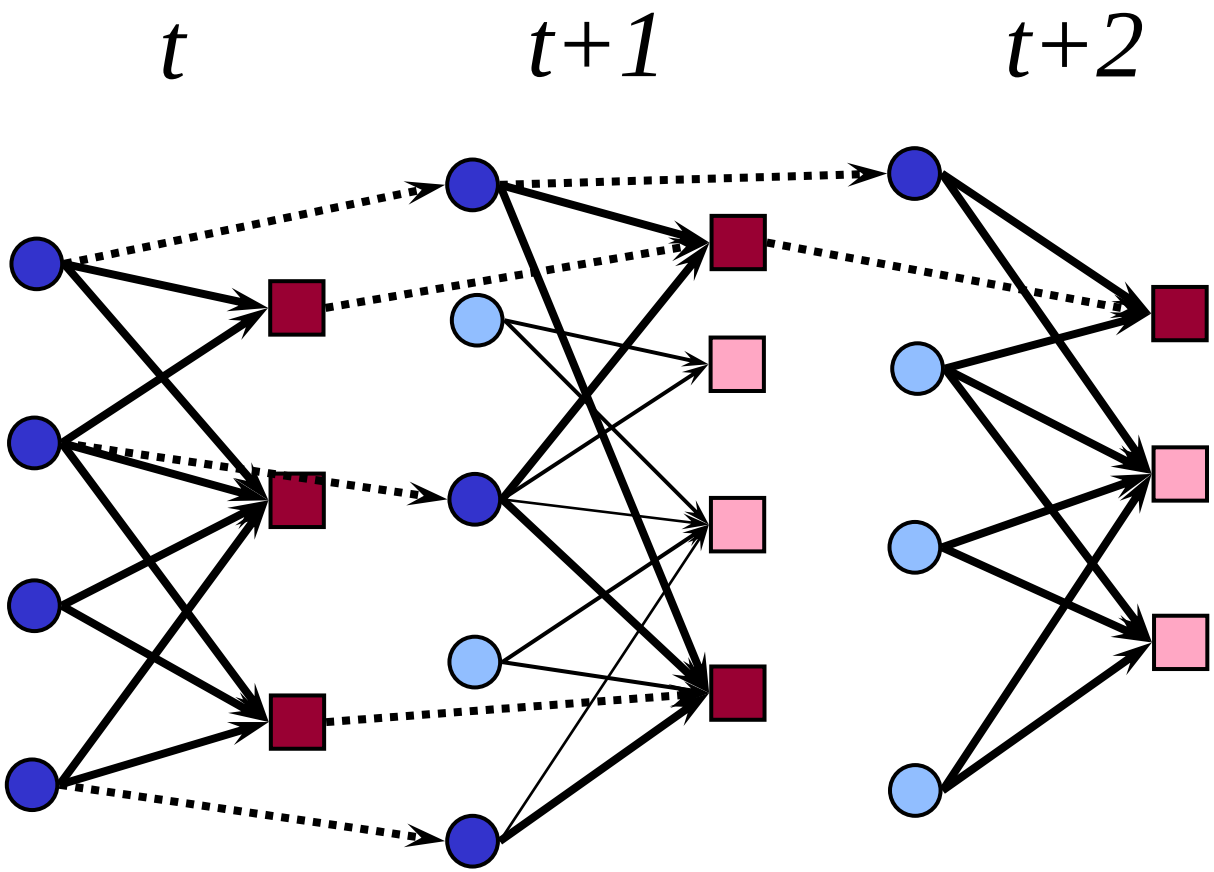
- Managing blood inventories over time



Schneider National



Multiple discrete assets



Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

A resource allocation model

- Modeling resources:
  - » The state of a single resource:

    $a =$ The attributes of a single resource

    $a \in \mathrm{A}$ The attribute space
  - » The state of multiple resources:

    $R_{ta} =$ The number of resources with attribute a

    $R_t = \left(R_{ta} \right)_{a \in \mathrm{A}}$ The resource state vector
  - » The information process:

    $\hat{R}_{ta} =$ The change in the number of resources with attribute a.

A resource allocation model

- Modeling demands:
  - The attributes of a single demand:

    $b =$ The attributes of a demand to be served.

    $b \in \mathrm{B}$ The attribute space
  - The demand state vector:

    $D_{tb} =$ The number of demands with attribute b

    $D_t = \left(D_{tb} \right)_{b \in \mathrm{B}}$ The demand state vector
  - The information process:

    $\hat{D}_{tb} =$ The change in the number of demands with attribute b.

A resource allocation model

- The system:
  - » The state vector

$$S_t = \left(R_t , D_t \right)$$

  - » The information process:

$$W_t = \text{Exogenous changes to resources and demands}$$
$$= \left(\hat{R}_t, \hat{D}_t \right)$$

A resource allocation model

- The three states of our system
  - The state of a single resource/entity

$$a_t = \begin{bmatrix} a_{t1} \\ a_{t2} \\ a_{t3} \end{bmatrix}$$

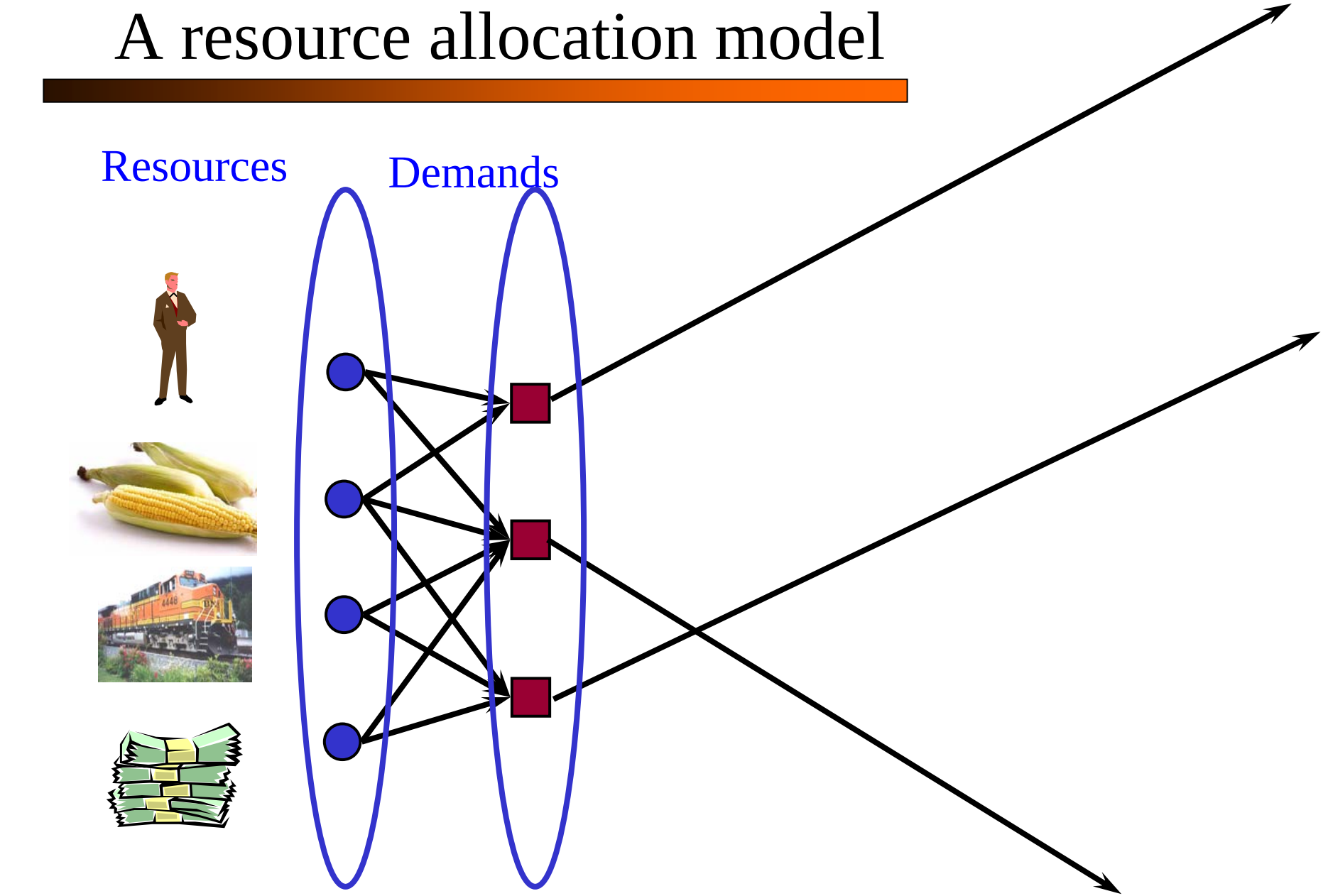
  - The resource state vector

$$R_t = \begin{bmatrix} R_{ta_1} \\ R_{ta_2} \\ R_{ta_3} \end{bmatrix}$$

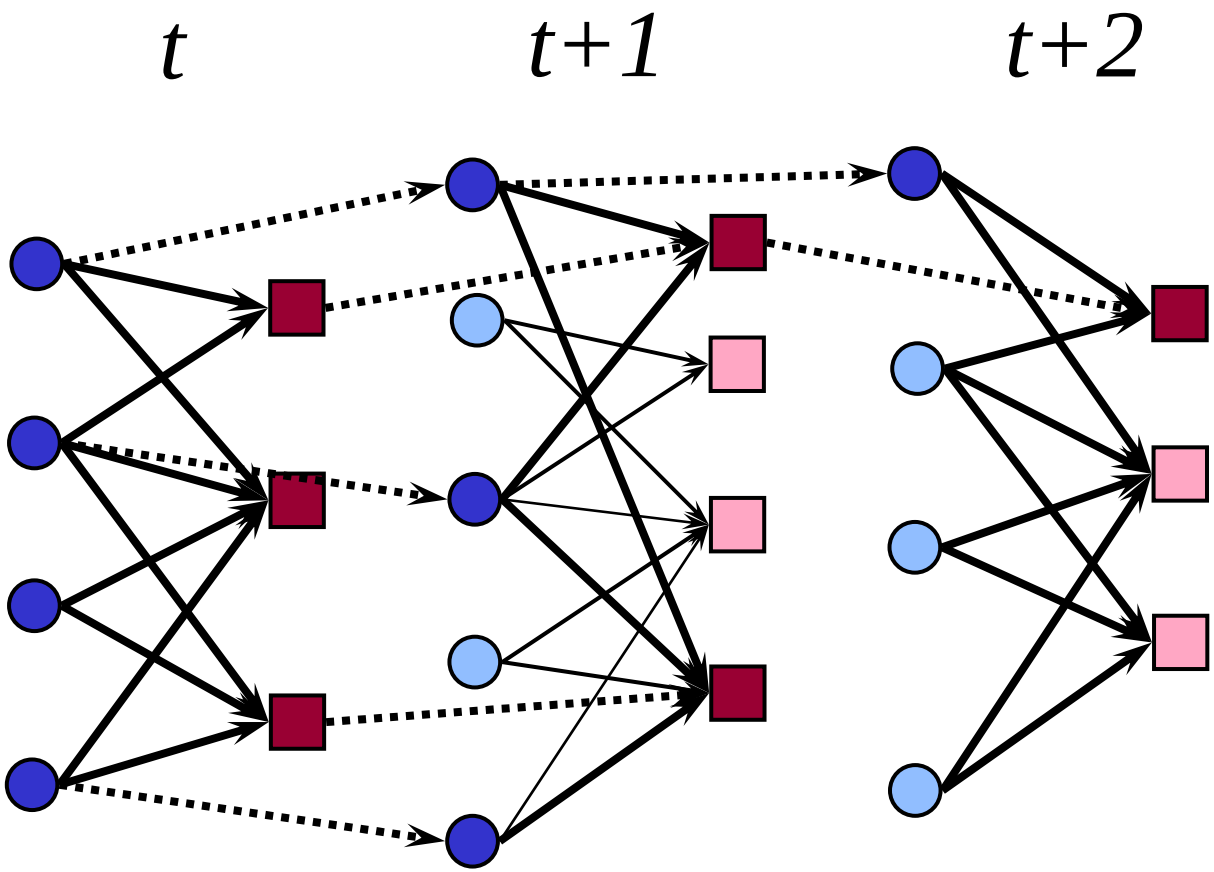
  - The system state vector

$$S_t = \left(R_t , D_t \right)$$

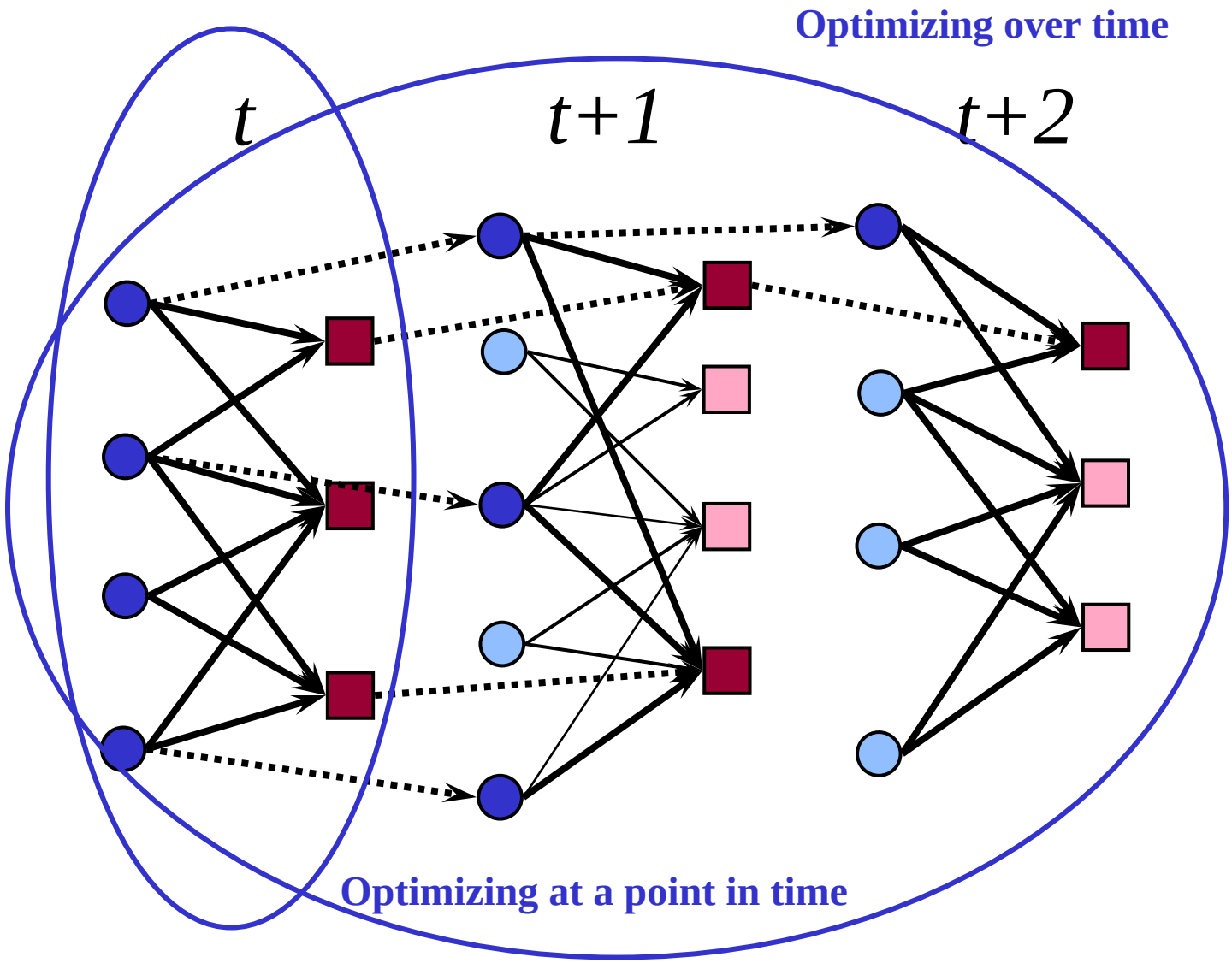
A resource allocation model



A resource allocation model

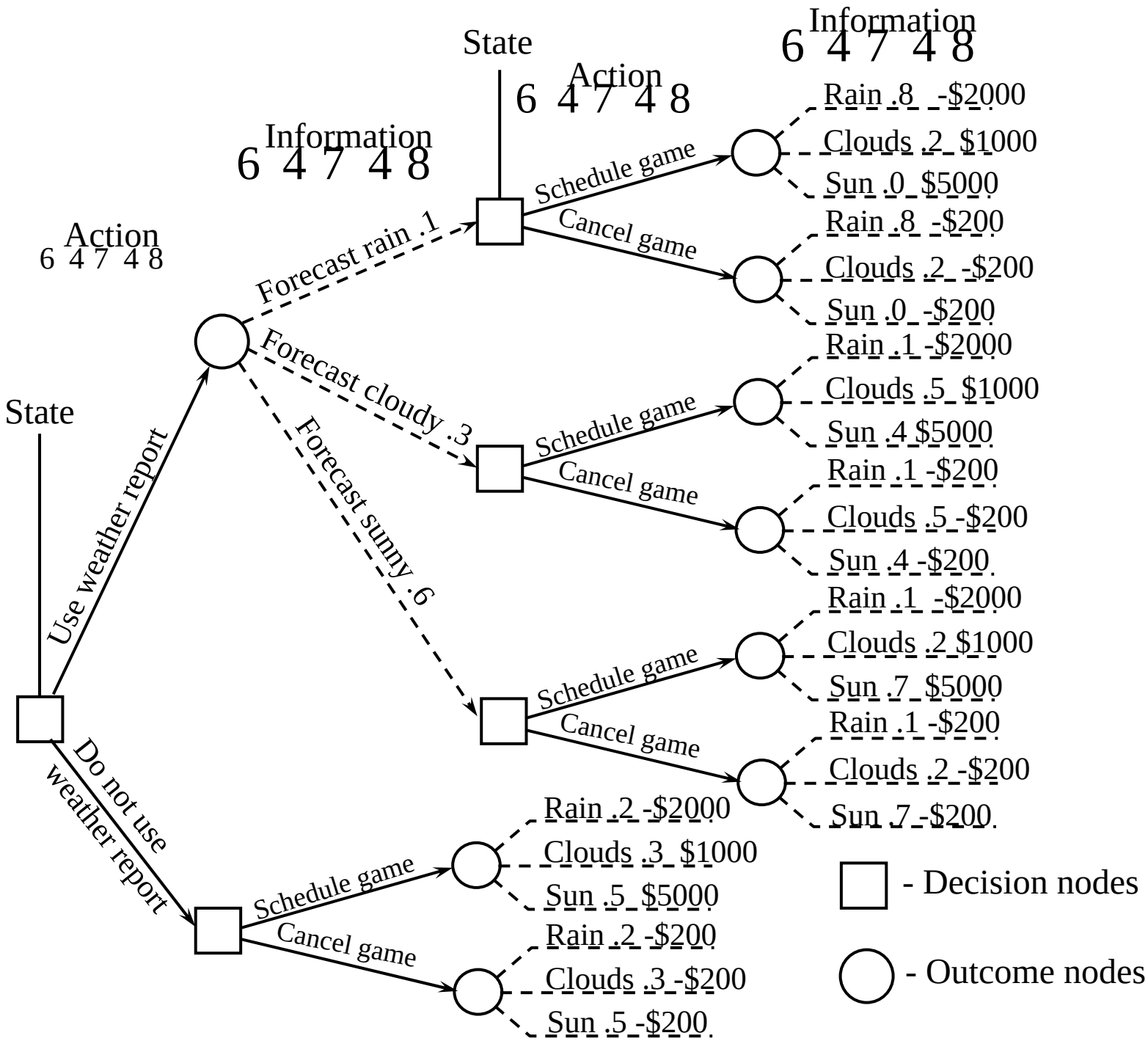


A resource allocation model



Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications


State
Action
6 4 7 4 8
Information
6 4 7 4 8
Use weather report
Do not use weather report
Forecast rain .1
Forecast cloudy .3
Forecast sunny .6
Schedule game
Cancel game
Rain .8 -$2000
Clouds .2 $1000
Sun .0 $5000
Rain .8 -$200
Clouds .2 -$200
Sun .0 -$200
Rain .1 -$2000
Clouds .5 $1000
Sun .4 $5000
Rain .1 -$200
Clouds .5 -$200
Sun .4 -$200
Rain .1 -$2000
Clouds .2 $1000
Sun .7 $5000
Rain .1 -$200
Clouds .2 -$200
Sun .7 -$200
Rain .2 -$2000
Clouds .3 $1000
Sun .5 $5000
Rain .2 -$200
Clouds .3 -$200
Sun .5 -$200
- Decision nodes
- Outcome nodes

Laying the foundation

- Dynamic programming review:
  - » Let:

S_t = "State" of our "system" at time t.

x_t = "Action" that we take to change the system.

$C(S_t, x_t)$ = Contribution earned when we take action x from state S_t.

  - » We model system dynamics using:

$p(S_{t+1} \mid S_t, x_t)$ = Probability that action x_t takes us from state S_t to state S_{t+1}

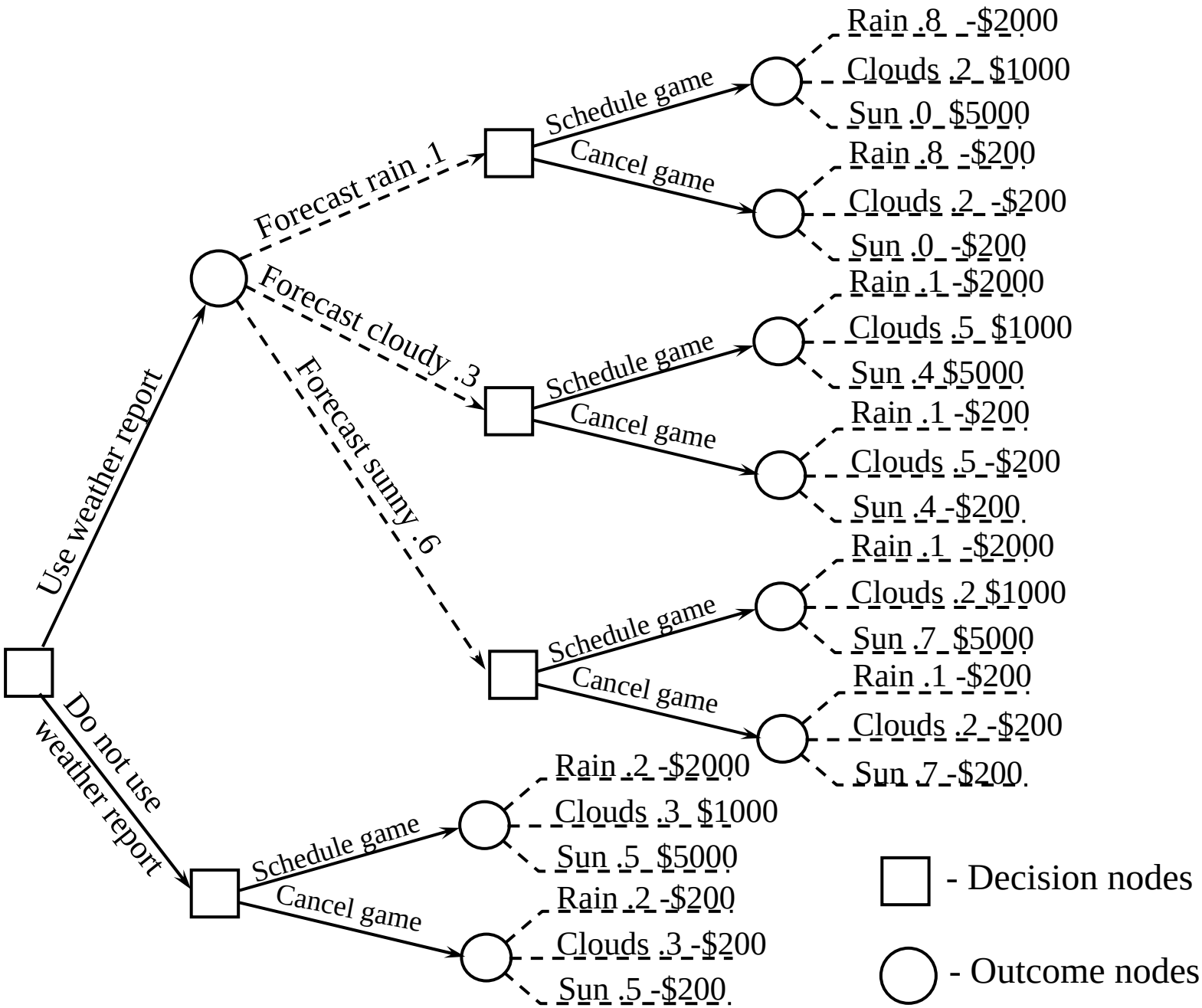
Laying the foundation

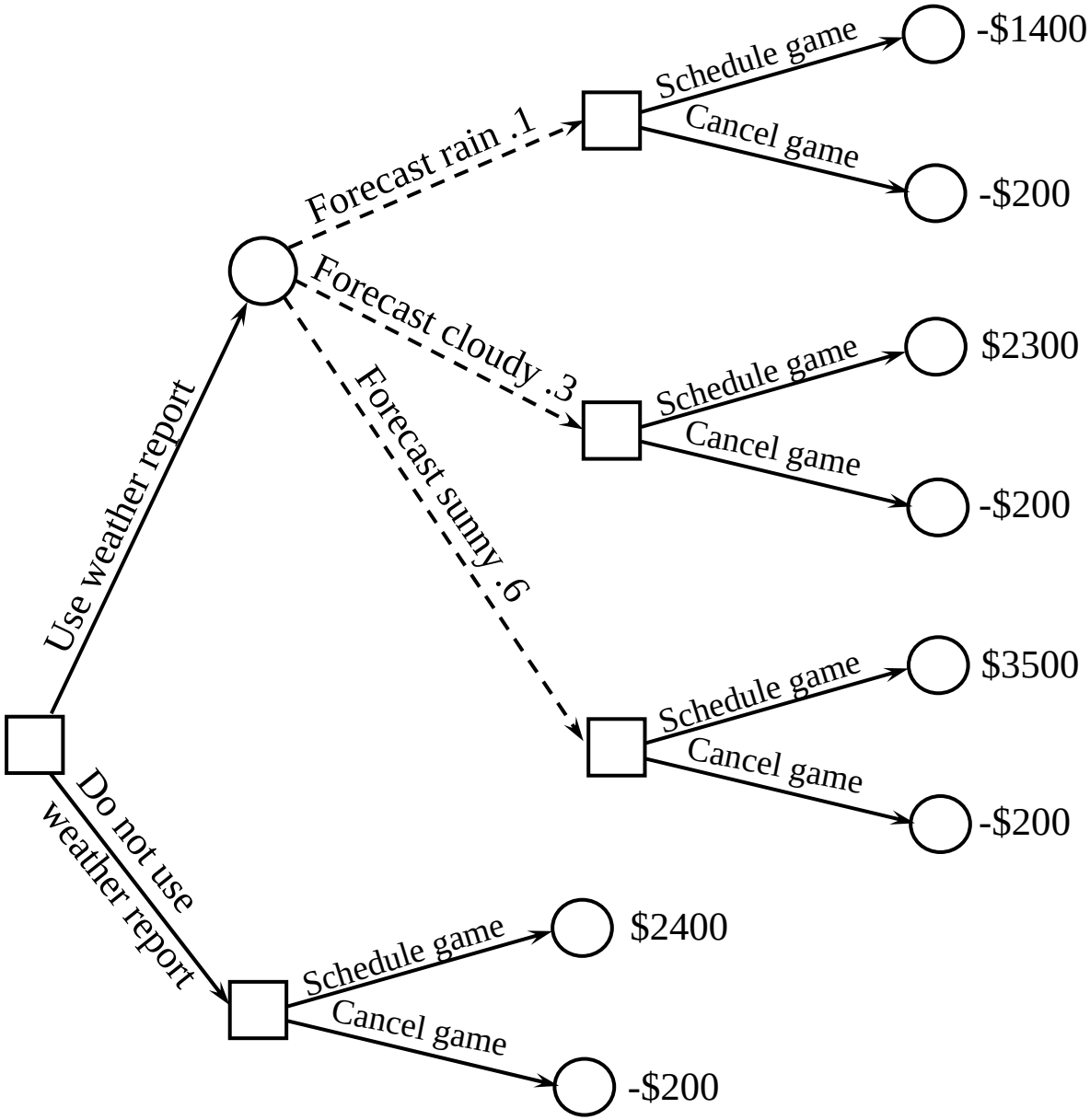
- Bellman's equation:
  - » Standard form:

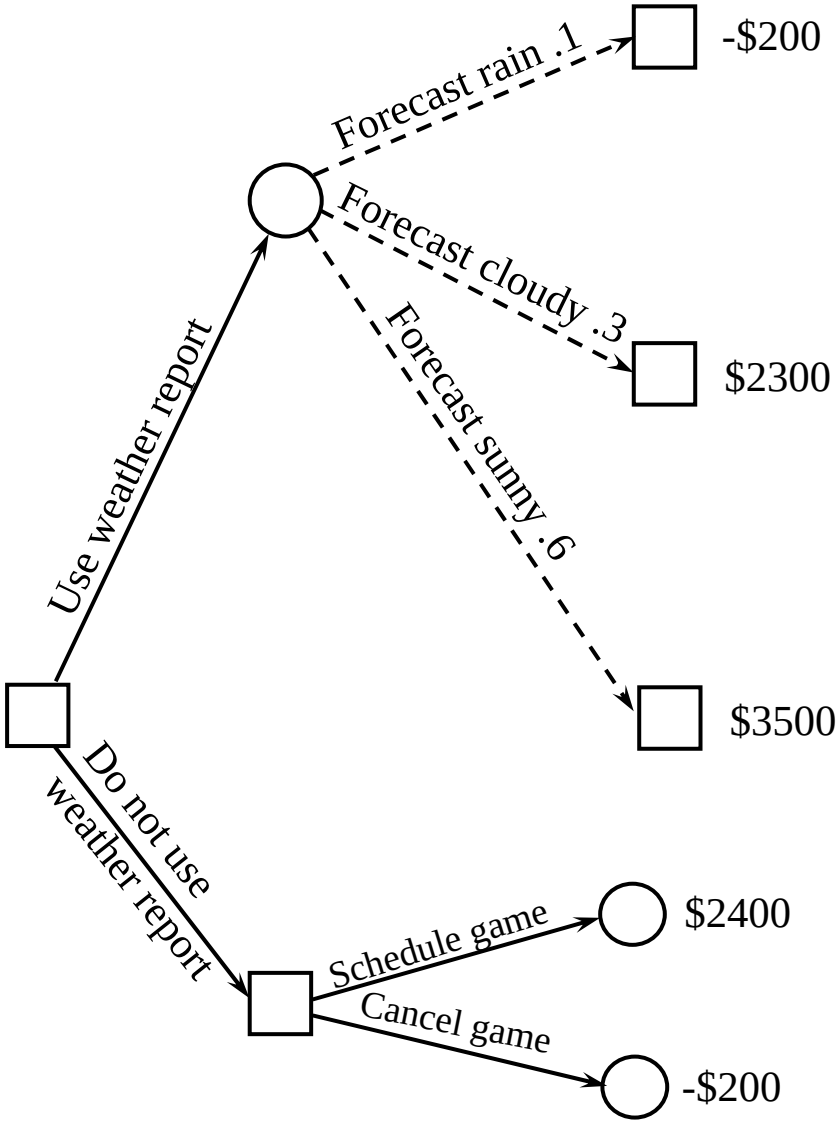
$$V_t(S_t) = \max_x \left(C_t(S_t, x_t) + \sum_{s'} p(s' \mid S_t, x_t) V_{t+1}(S_{t+1} = s') \right)$$

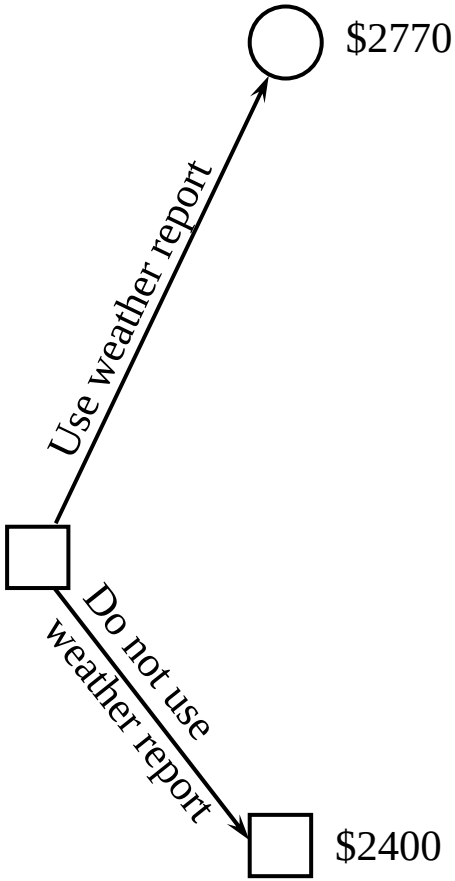
  - » Expectation form:

$$V_t(S_t) = \max_x \left(C_t(S_t, x_t) + E\left\{ V_{t+1}\left(S_{t+1}(S_t, x_t) \right) \mid S_t \right\} \right)$$


Use weather report
Do not use weather report
Forecast rain .1
Forecast cloudy .3
Forecast sunny .6
Schedule game
Cancel game
Rain .8 -$2000
Clouds .2 $1000
Sun .0 $5000
Rain .8 -$200
Clouds .2 -$200
Sun .0 -$200
Rain .1 -$2000
Clouds .5 $1000
Sun .4 $5000
Rain .1 -$200
Clouds .5 -$200
Sun .4 -$200
Rain .1 -$2000
Clouds .2 $1000
Sun .7 $5000
Rain .1 -$200
Clouds .2 -$200
Sun .7 -$200
Rain .2 -$2000
Clouds .3 $1000
Sun .5 $5000
Rain .2 -$200
Clouds .3 -$200
Sun .5 -$200
- Decision nodes
- Outcome nodes


Use weather report
Do not use weather report
Forecast rain .1
Forecast cloudy .3
Forecast sunny .6
Schedule game
Cancel game
-$1400
-$200
$2300
-$200
$3500
-$200
$2400
-$200


Forecast rain .1
-$200
Forecast cloudy .3
$2300
Forecast sunny .6
$3500
Use weather report
Do not use weather report
Schedule game
$2400
Cancel game
-$200


$2770
Use weather report
Do not use
weather report
$2400

Bellman's equation

- We just solved Bellman's equation:

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

  - » We found the value of being in each state by stepping backward through the tree.

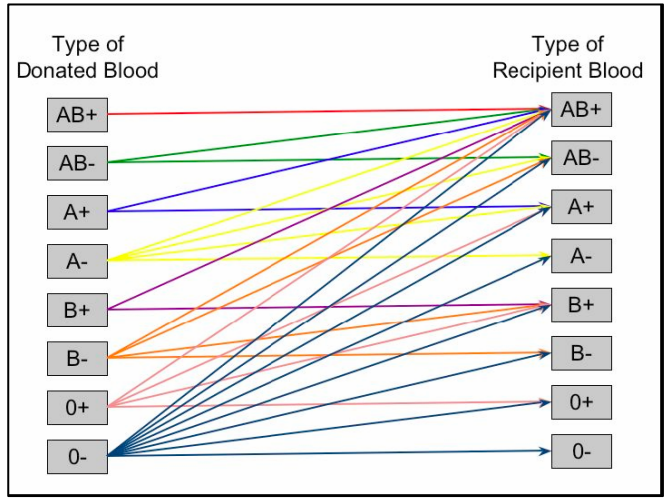
Bellman's equation

- The challenge of dynamic programming:

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

The curses of dimensionality

- What happens if we apply this idea to our blood problem?
  - » State variable is:
    - The supply of each type of blood, along with its age
      - – 8 blood types
      - – 6 ages
      - – = 48 “blood types”
    - The demand for each type of blood
      - – 8 blood types
  - » Decision variable is how much of 48 blood types to supply to 8 demand types.
    - 216- dimensional decision vector
  - » Random information
    - Blood donations by week (8 types)
    - New demands for blood (8 types)



The curses of dimensionality

- The challenge of dynamic programming:

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

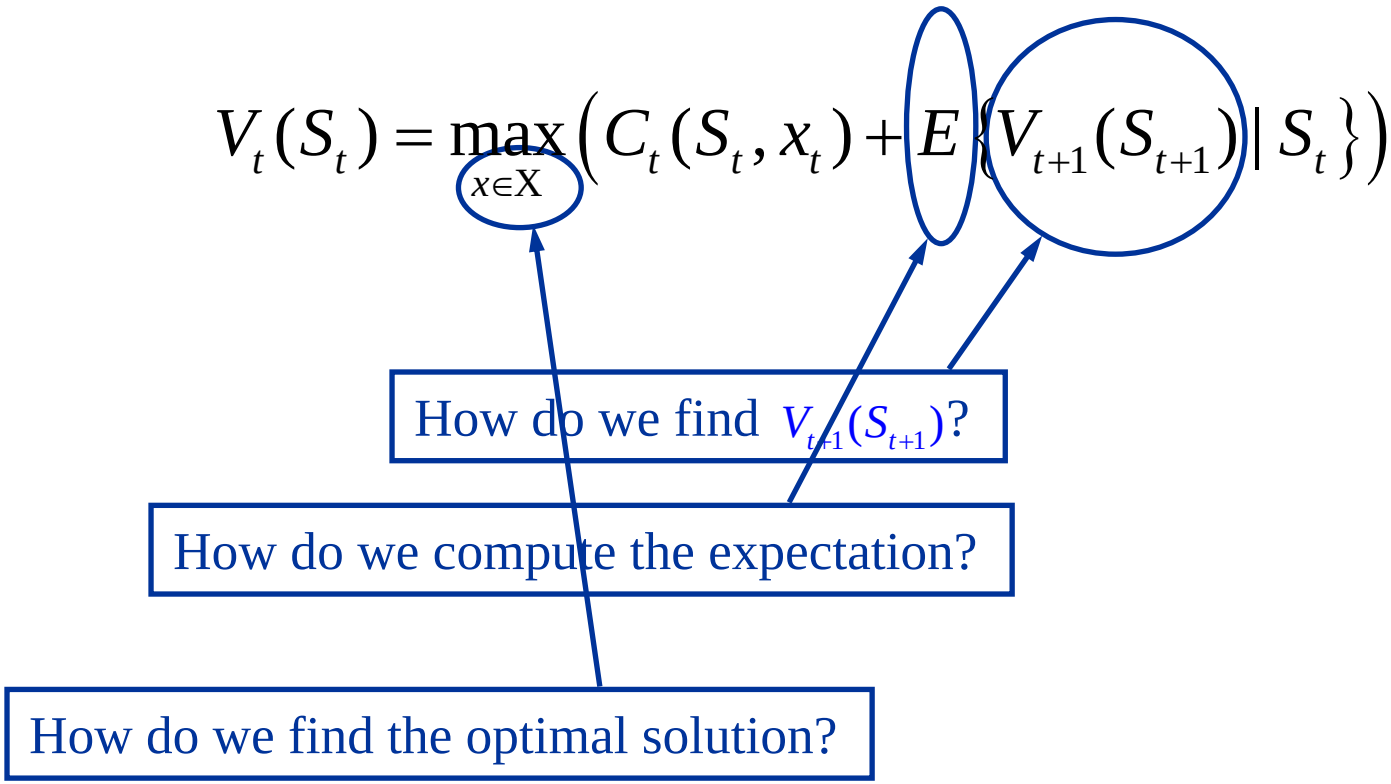
Three curses

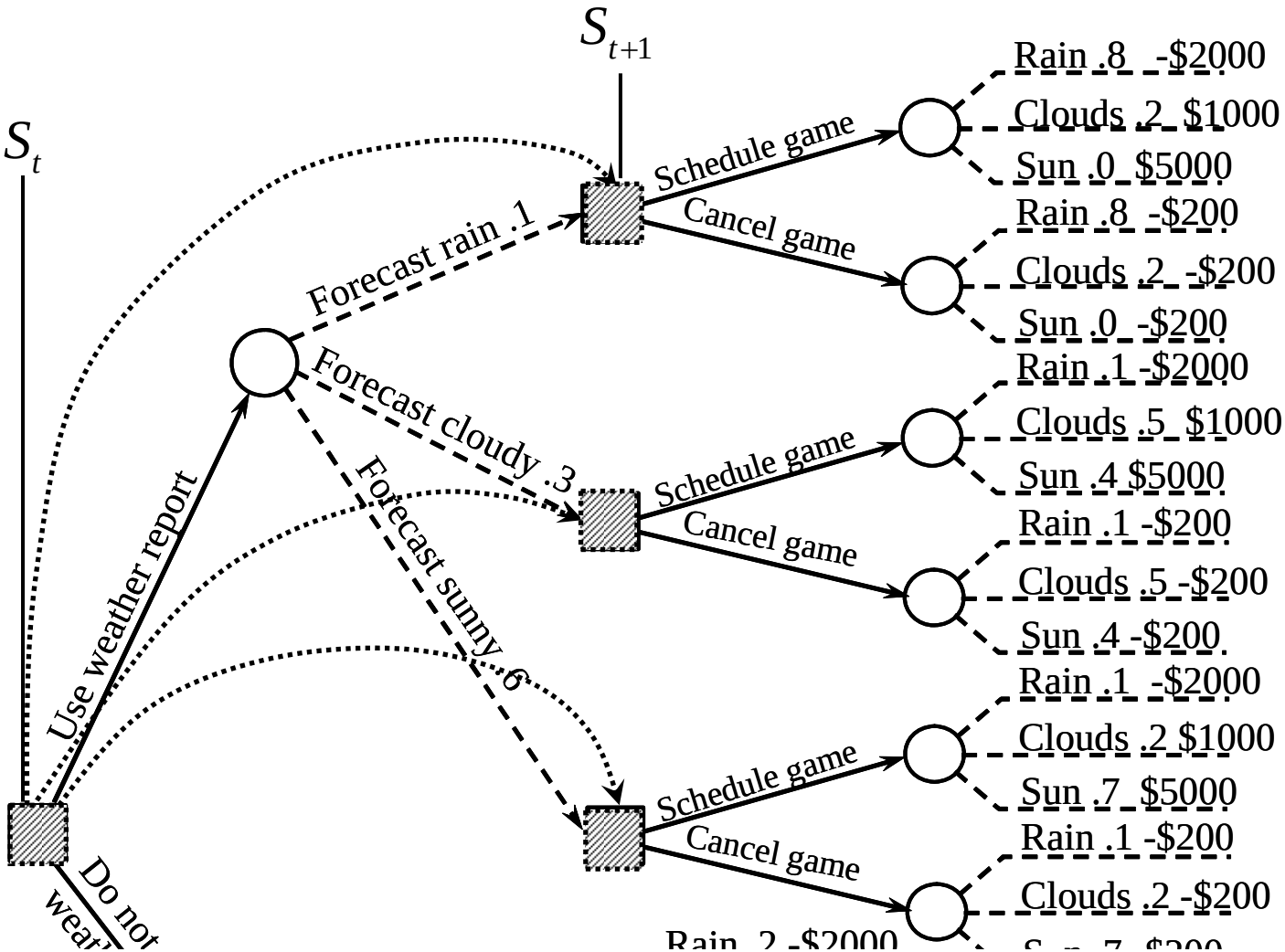
- Problem: Curse of dimensionality
  - State space
  - Outcome space
  - Action space (feasible region)

The curses of dimensionality

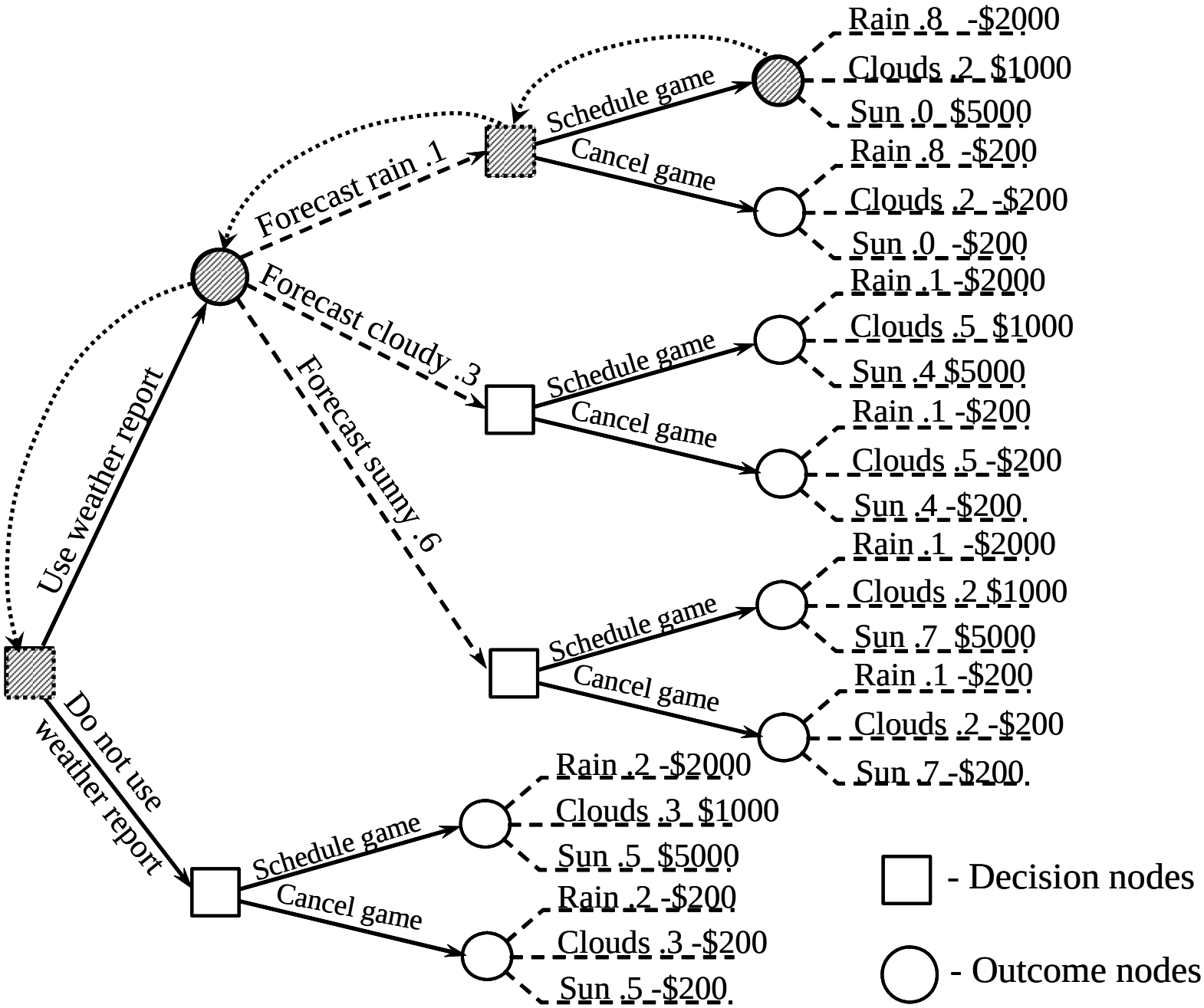
- The computational challenge:

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





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


Rain .8 -$2000
Clouds .2 $1000
Sun .0 $5000
Schedule game
Cancel game
Rain .8 -$200
Clouds .2 -$200
Sun .0 -$200
Forecast rain .1
Rain .1 -$2000
Clouds .5 $1000
Sun .4 $5000
Forecast cloudy .3
Schedule game
Cancel game
Rain .1 -$200
Clouds .5 -$200
Sun .4 -$200
Use weather report
Forecast sunny .6
Rain .1 -$2000
Clouds .2 $1000
Sun .7 $5000
Schedule game
Cancel game
Rain .1 -$200
Clouds .2 -$200
Sun .7 -$200
Do not use weather report
Rain .2 -$2000
Clouds .3 $1000
Sun .5 $5000
Schedule game
Cancel game
Rain .2 -$200
Clouds .3 -$200
Sun .5 -$200
- Decision nodes
- Outcome nodes

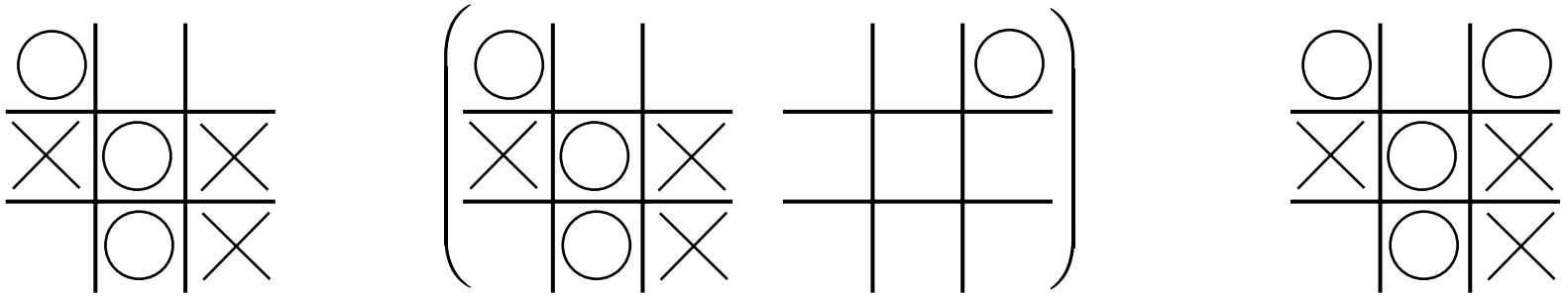
Pre- and post-decision states

- 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.

Pre- and post-decision states

- 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

A single, complex entity

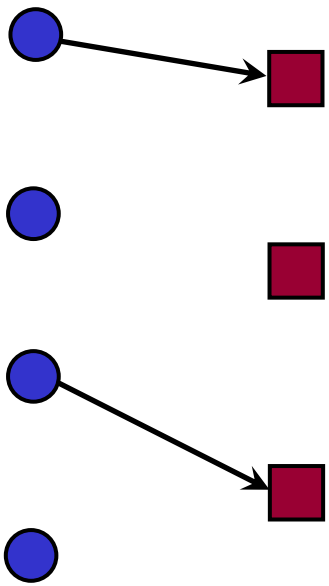
- Pre- and post-decision attributes for our nomadic truck driver:

	$t = 40$ Pre-decision	$t = 40$ Decision	$t = 40$ Post-decision		$t = 50$ Pre-decision
City	Dallas	Chicago	Chicago	$\cdots$	Chicago
ETA	41.2	-	54.7		56.2
Equip	Good	-	Good		Repair

Pre- and post-decision states

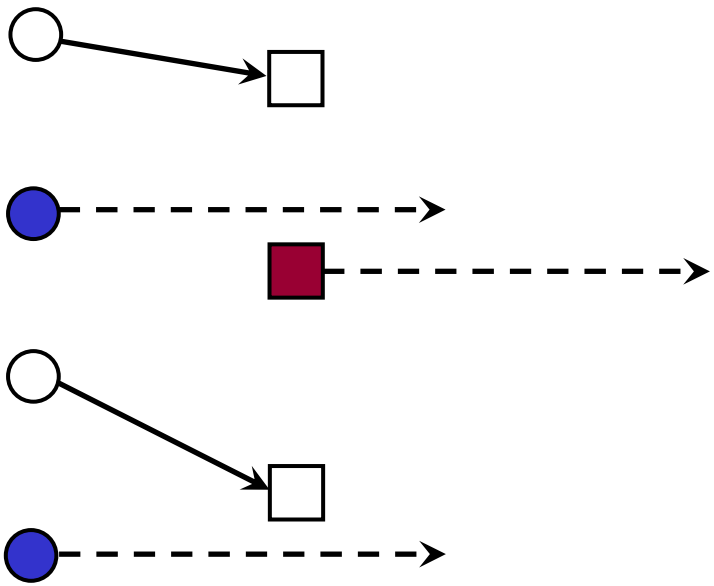
- Pre-decision: resources and demands

$$S_t = (R_t, D_t)$$



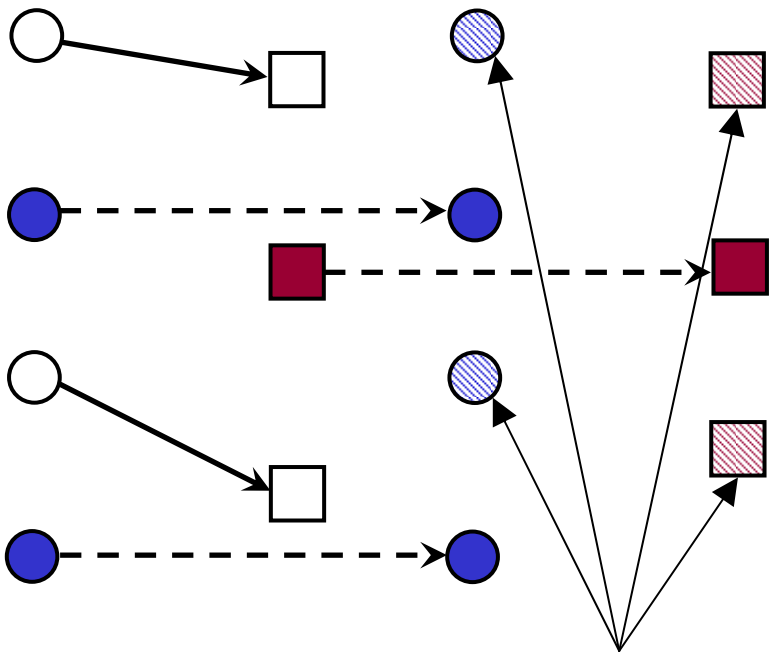
Pre- and post-decision states

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



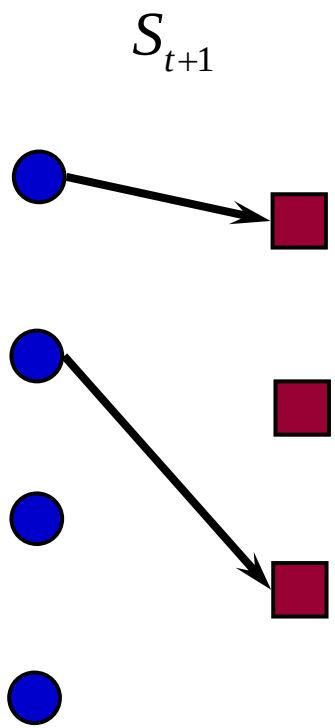
Pre- and post-decision states

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



$W_{t+1} = (R_{t+1}, D_{t+1})$

Pre- and post-decision states



System dynamics

- It is traditional to assume you are given the one-step transition matrix:

$$p(S_{t+1} \mid S_t, x_t) = \text{Probability that action } x_t \text{ takes us from state } S_t \text{ to state } S_{t+1}$$

  - » Computing the transition matrix is impossible for the vast majority of problems.

- We are going to assume that we are given a *transition function:*

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

  - » This is at the heart of any simulation model.
  - » Often rule-based. Very easy to compute, even for large-scale problems.

The transition function

- Working with pre- and post-decision states
  - » The "usual" transition function:

$$S_{t+1} = S^M\left(S_t, x_t, W_{t+1}\right) \qquad \text{From } S_t \text{ to } S_{t+1}.$$

  - » The transition function broken into two steps:

$$S_t^x = S^{M,x}\left(S_t, x_t\right) \qquad \text{The pure effect of a decision}$$

$$S_{t+1} = S^{M,W}\left(S_t^x, W_{t+1}\right) \qquad \text{The effect of the exogenous information}$$

The transition function

- Actually, we have three transition functions:
  - The attribute transition function:

$$a_t^x = a^{M,x}(a_t, x_t) \qquad \text{The pure effect of a decision}$$

$$a_{t+1} = a^{M,W}(a_t^x, W_{t+1}) \qquad \text{The effect of the exogenous information}$$

  - The resource transition function

$$R_t^x = R^{M,x}(R_t, x_t) \qquad \text{The pure effect of a decision}$$

$$R_{t+1} = R^{M,W}(R_t^x, W_{t+1}) \qquad \text{The effect of the exogenous information}$$

  - The general transition function:

$$S_t^x = S^{M,x}(S_t, x_t) \qquad \text{The pure effect of a decision}$$

$$S_{t+1} = S^{M,W}(S_t^x, W_{t+1}) \qquad \text{The effect of the exogenous information}$$

Bellman's equations with the post-decision state

- Bellman's equations broken into stages:

  » Optimization problem (making the decision):

$$V_t(S_t) = \max_x \left(C_t(S_t, x_t) + V_t^x \left(S_t^{M,x}(S_t, x_t) \right) \right)$$

  - Note: this problem is deterministic!

  » Simulation problem (the effect of exogenous information):

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

Bellman's equations with the post-decision state

- Challenges
  - » For most practical problems, we are not going to be able to compute $V_t^x(S_t^x)$.

$$V_t(S_t) = \max_x \left(C_t(S_t, x_t) + V_t^x(S_t^x) \right)$$

  - » Concept: replace it with an approximation $\bar{V}_t(S_t^x)$ and solve

$$V_t(S_t) = \max_x \left(C_t(S_t, x_t) + \bar{V}_t(S_t^x) \right)$$

  - » So now we face:
    - What should the approximation look like?
    - How do we estimate it?

Approximating the value function

- For "resource allocation" problems, we have been using:
  - » Linear (in the resource state):

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

Best when assets are complex, which means that R_{ta} is small (typically 0 or 1).

  - » Piecewise linear, separable:

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

Best when assets are simple, which means that R_{ta} may be larger.

Our general algorithm

Step 1: Start with a post-decision state $S_{t-1}^{x,n}$

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

$$S_t^n = S^{M,W}\left(S_{t-1}^{x,n}, W_t(\omega^n)\right)$$ Simulation

Step 3: Solve the deterministic optimization using an approximate value function:

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

to obtain x_t^n.

Step 4: Update the value function approximation

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

Step 5: Find the next post-decision state:

$$S_t^{x,n} = S^{M,x}(S_t^n, x_t^n)$$

Competing updating methods

- Comparison to other methods:
  - » Classical MDP (value iteration)

$$V^n(S) = \max_x \left(C(S,x) + EV^{n-1}(S_{t+1}) \right)$$

  - » Classical ADP (pre-decision state):

$$\hat{v}_t^n = \max_x \left(C_t(S_t^n, x_t) + \sum_{s'} p(s' \mid S_t^n, x_t) \bar{V}_{t+1}^{n-1}(s') \right)$$

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

$\hat{v}_t$ updates $\bar{V}_t(S_t)$

  - » Our method (update $\bar{V}_t^{x,n-1}$ around post-decision state):

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

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

$\hat{v}_t$ updates $\bar{V}_{t-1}(S_{t-1}^x)$

APPROXIMATE DYNAMIC PROGRAMMING

Solving the curses of dimensionality

Warren B. Powell

WILEY-INTERSCIENCE

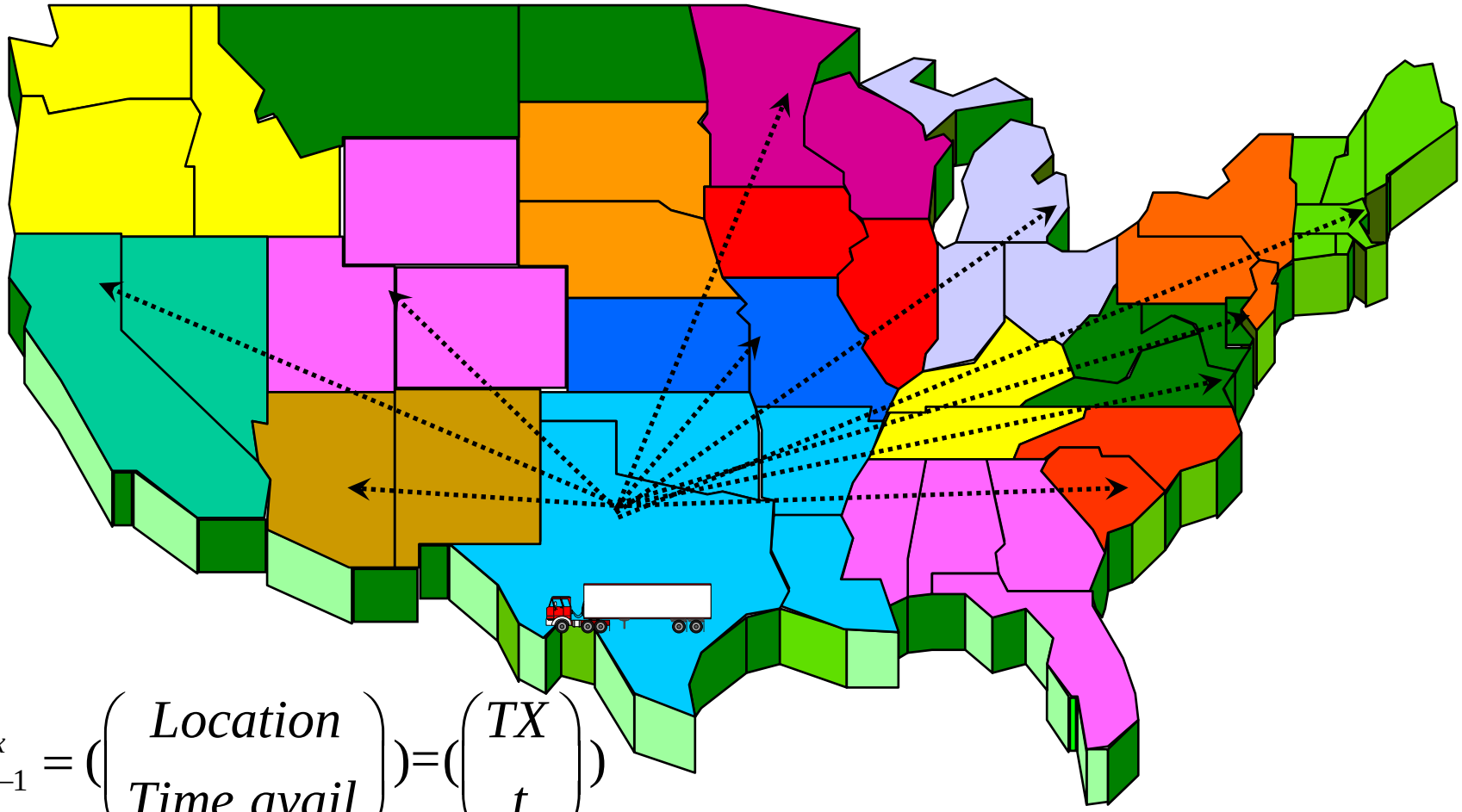
A JOHN WILEY & SONS, INC., PUBLICATION

Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

Nomadic trucker illustration

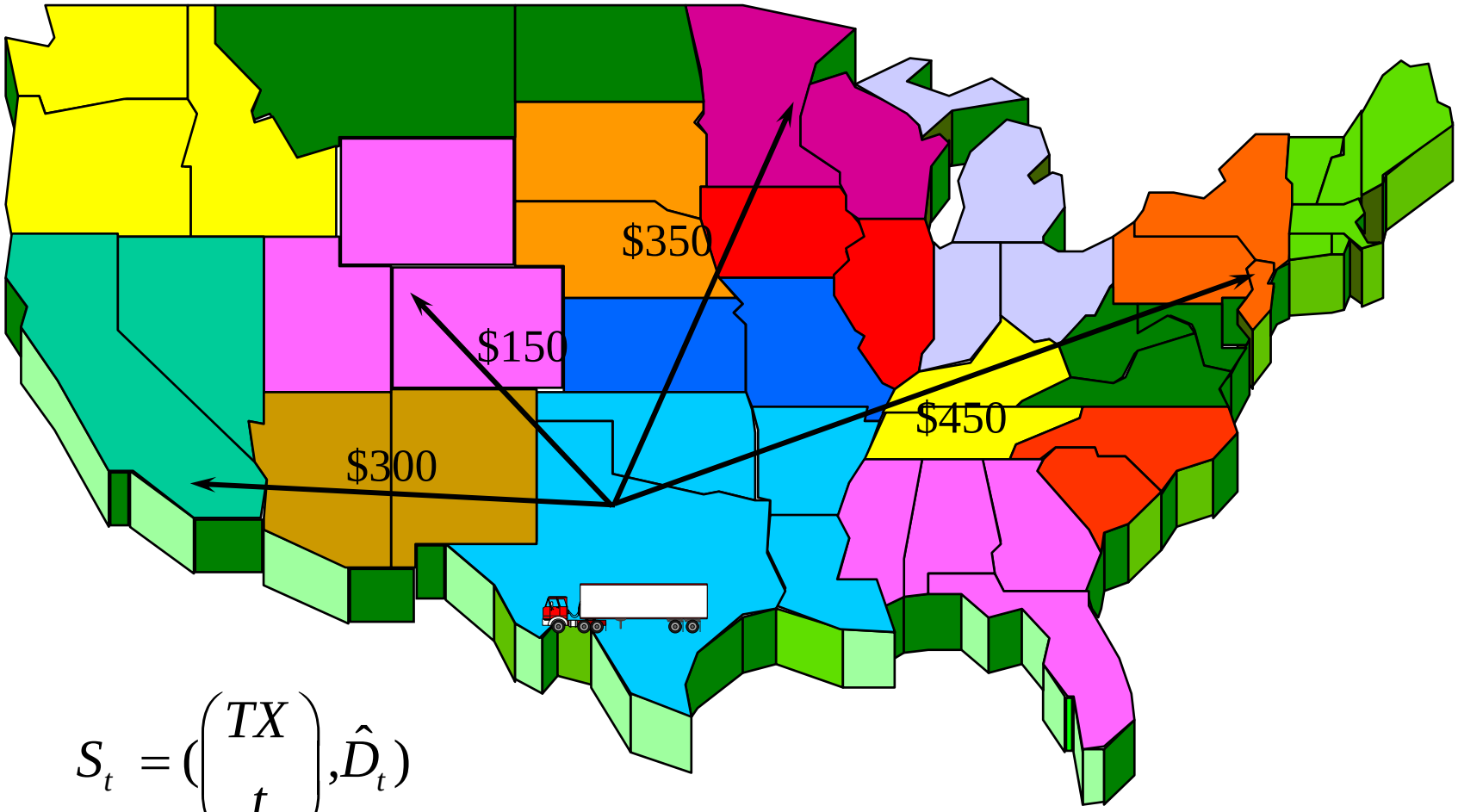
- The previous post-decision state: trucker in Texas



$$S_{t-1}^{x} = \left(\begin{pmatrix} Location \\ Time\ avail \end{pmatrix}\right) = \left(\begin{pmatrix} TX \\ t \end{pmatrix}\right)$$

Nomadic trucker illustration

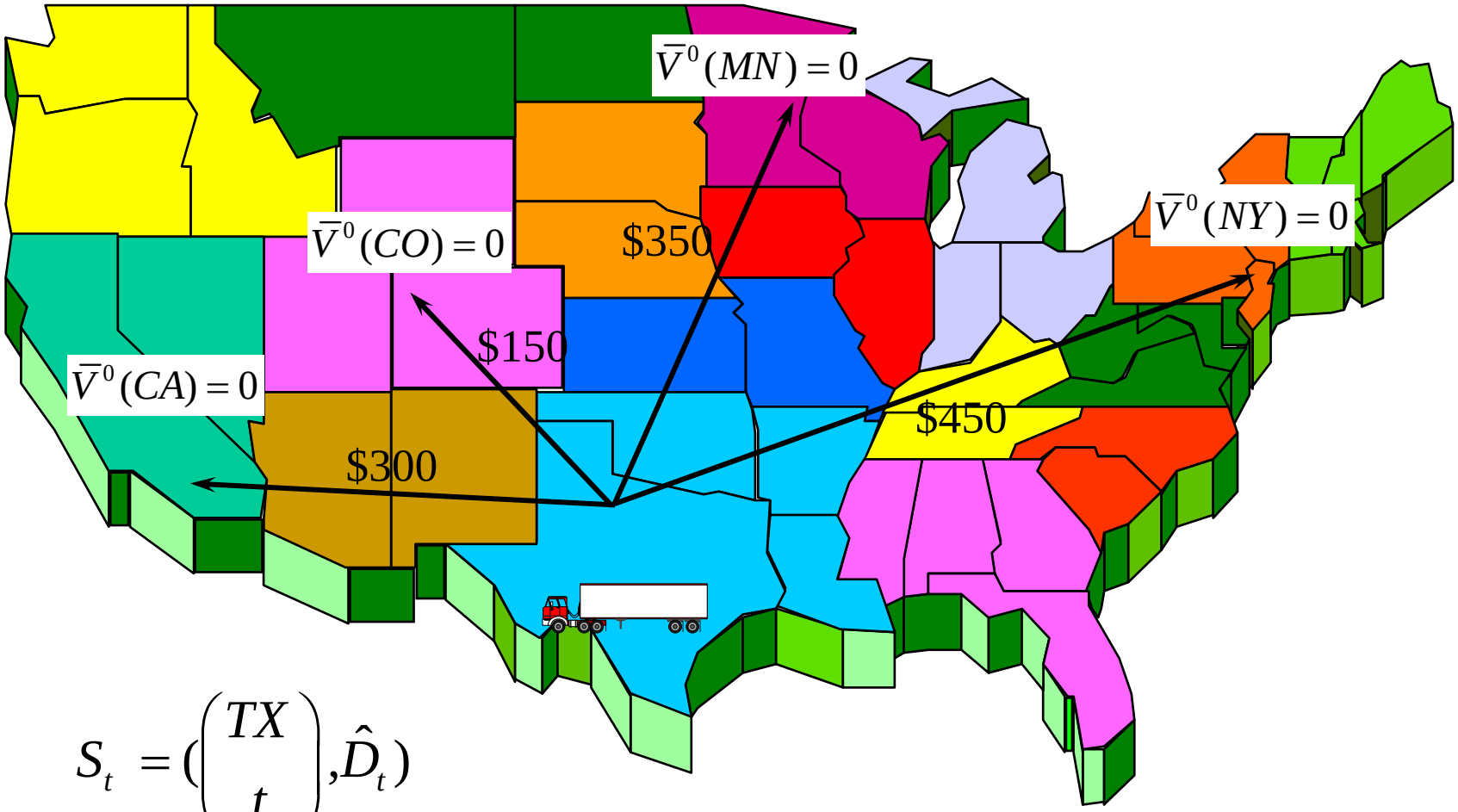
- Pre-decision state: we see the demands



$$S_t = \left(\begin{pmatrix} TX \\ t \end{pmatrix}, \hat{D}_t\right)$$

Nomadic trucker illustration

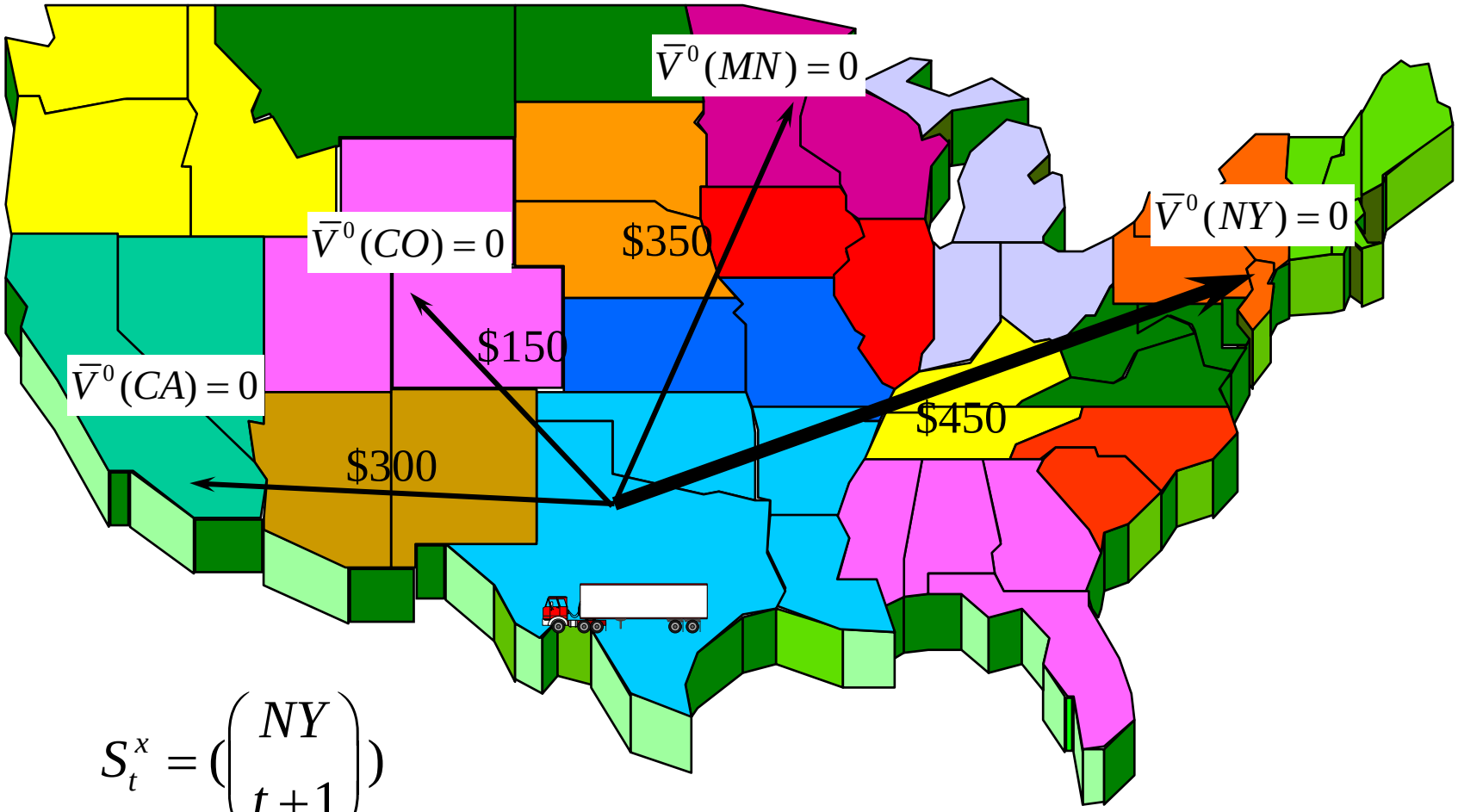
- We use initial value function approximations…



$$S_t = \left(\begin{pmatrix} TX \\ t \end{pmatrix}, \hat{D}_t\right)$$

Nomadic trucker illustration

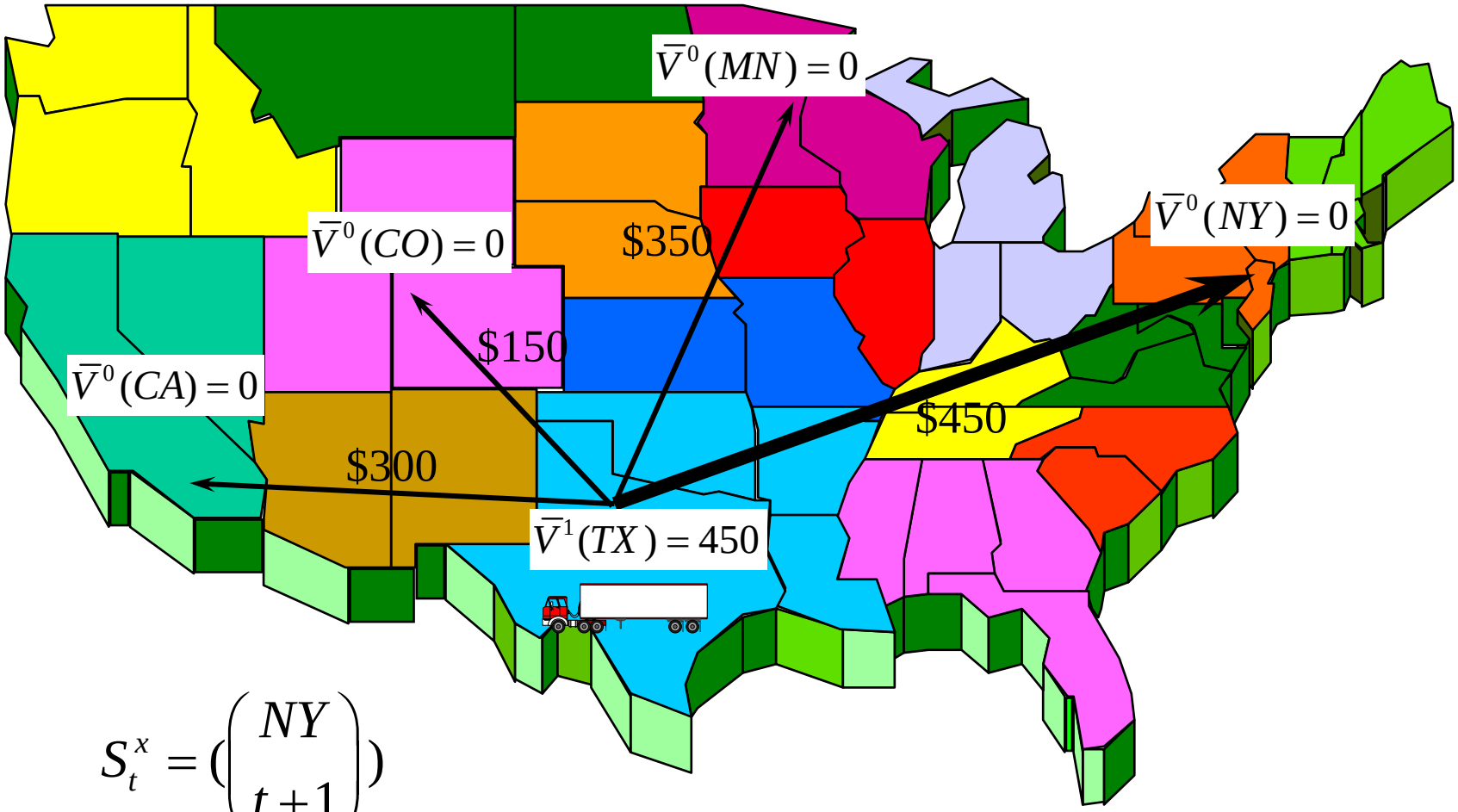
- … and make our first choice: x^1



$$S_t^x = \begin{pmatrix} NY \\ t+1 \end{pmatrix}$$

Nomadic trucker illustration

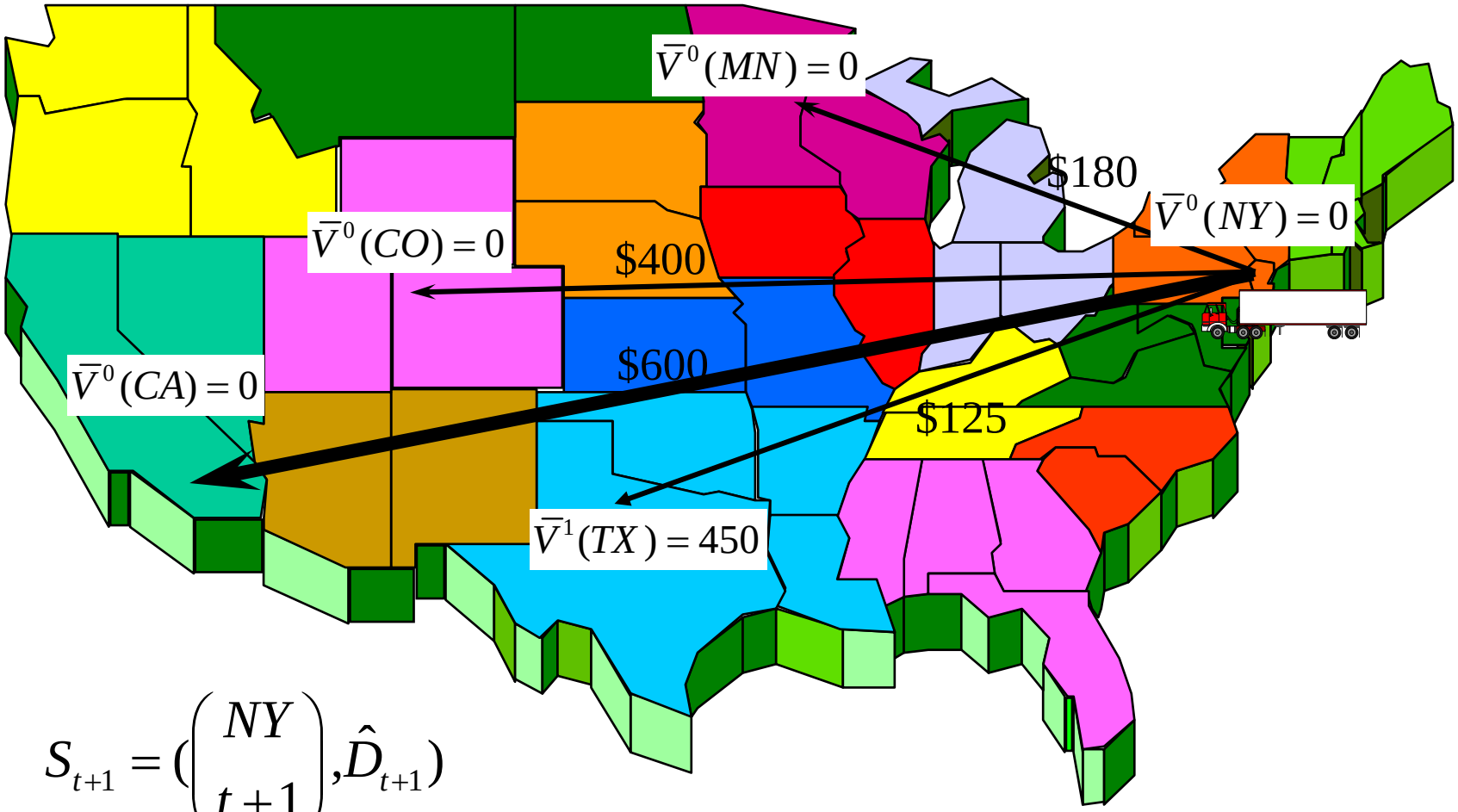
- Update the value of being in Texas.



$$S_t^x = \begin{pmatrix} NY \\ t+1 \end{pmatrix}$$

Nomadic trucker illustration

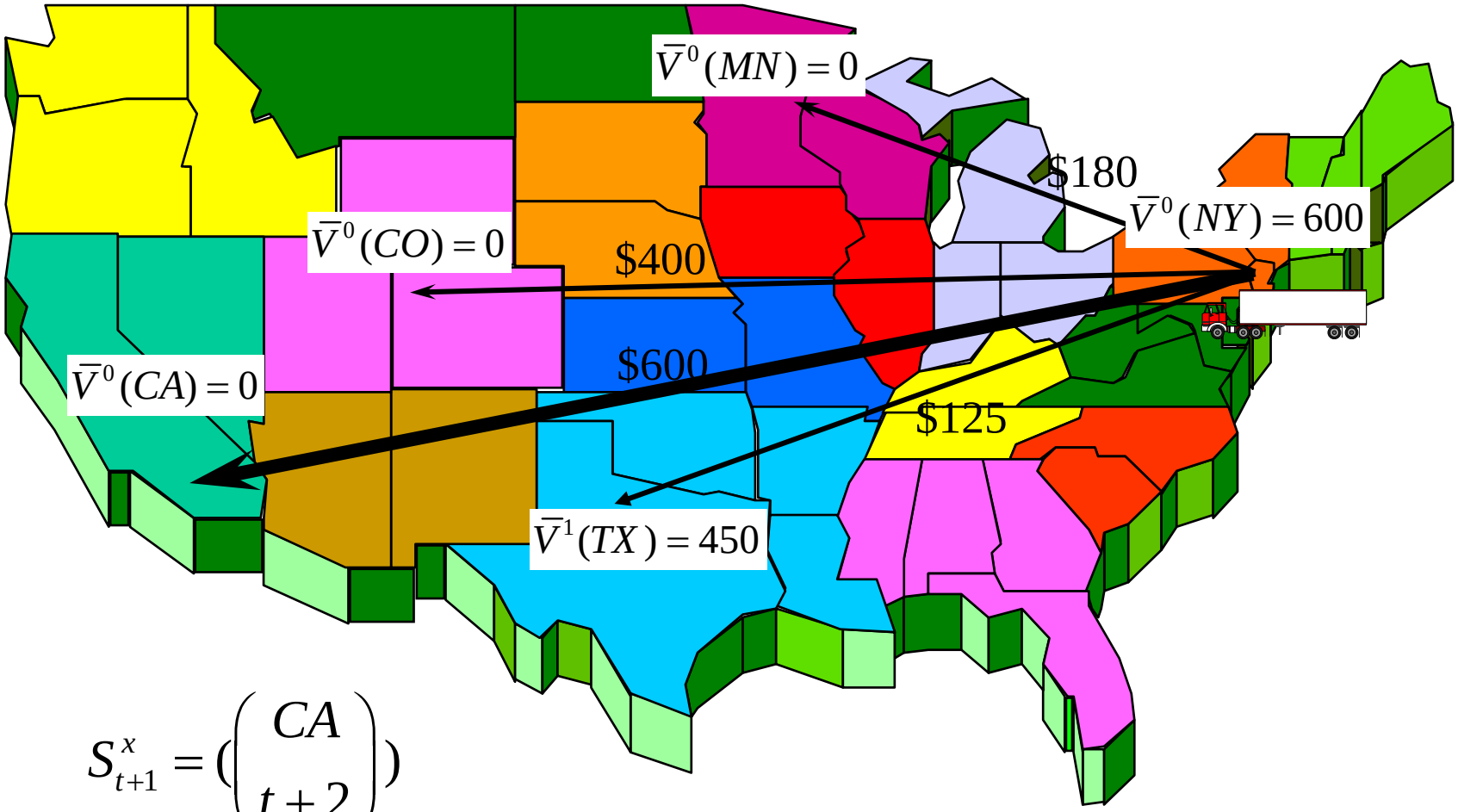
- Now move to the next state, sample new demands and make a new decision



$$S_{t+1} = \left(\begin{pmatrix} NY \\ t+1 \end{pmatrix}, \hat{D}_{t+1}\right)$$

Nomadic trucker illustration

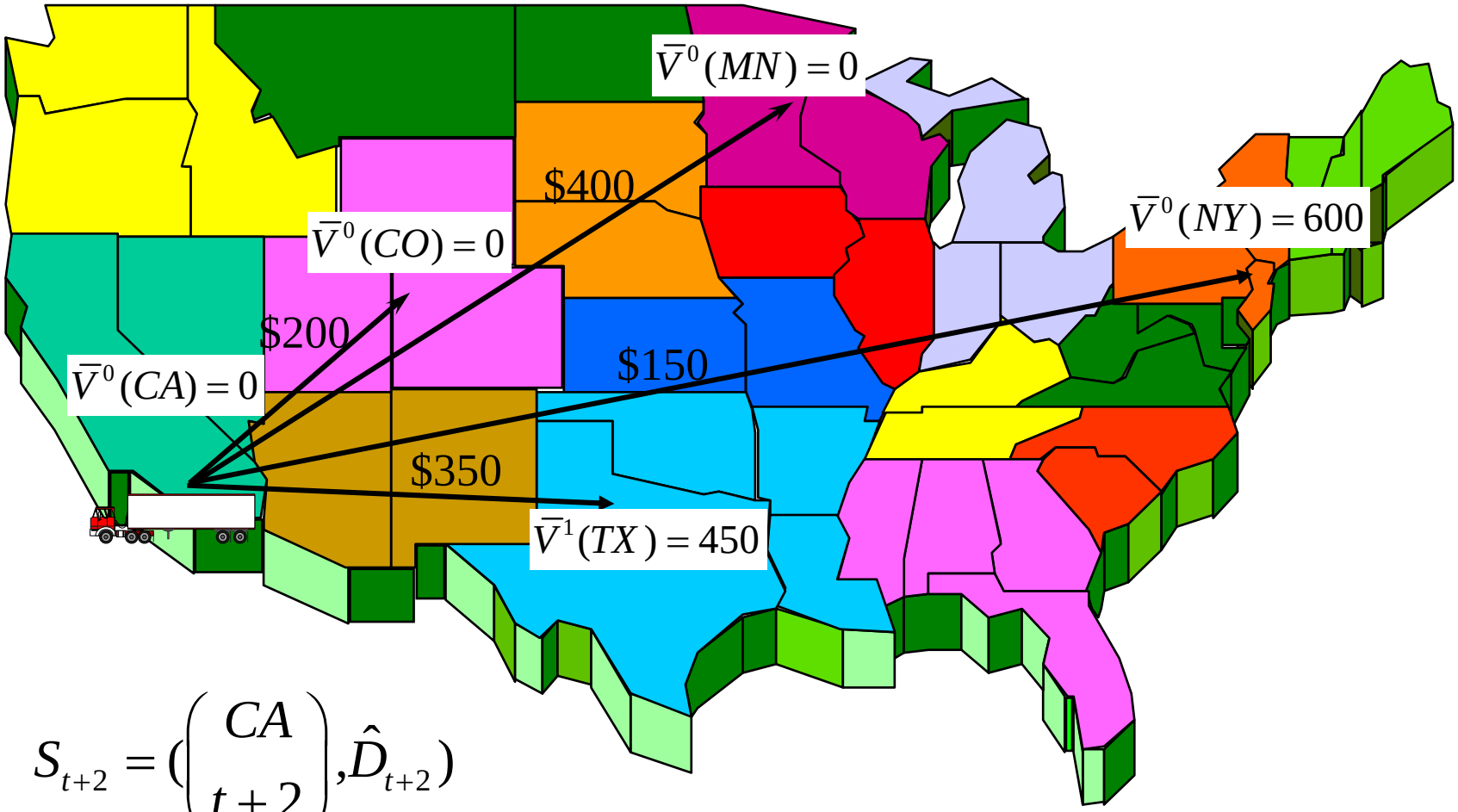
- Update value of being in NY



$$S_{t+1}^x = \begin{pmatrix} CA \\ t+2 \end{pmatrix}$$

Nomadic trucker illustration

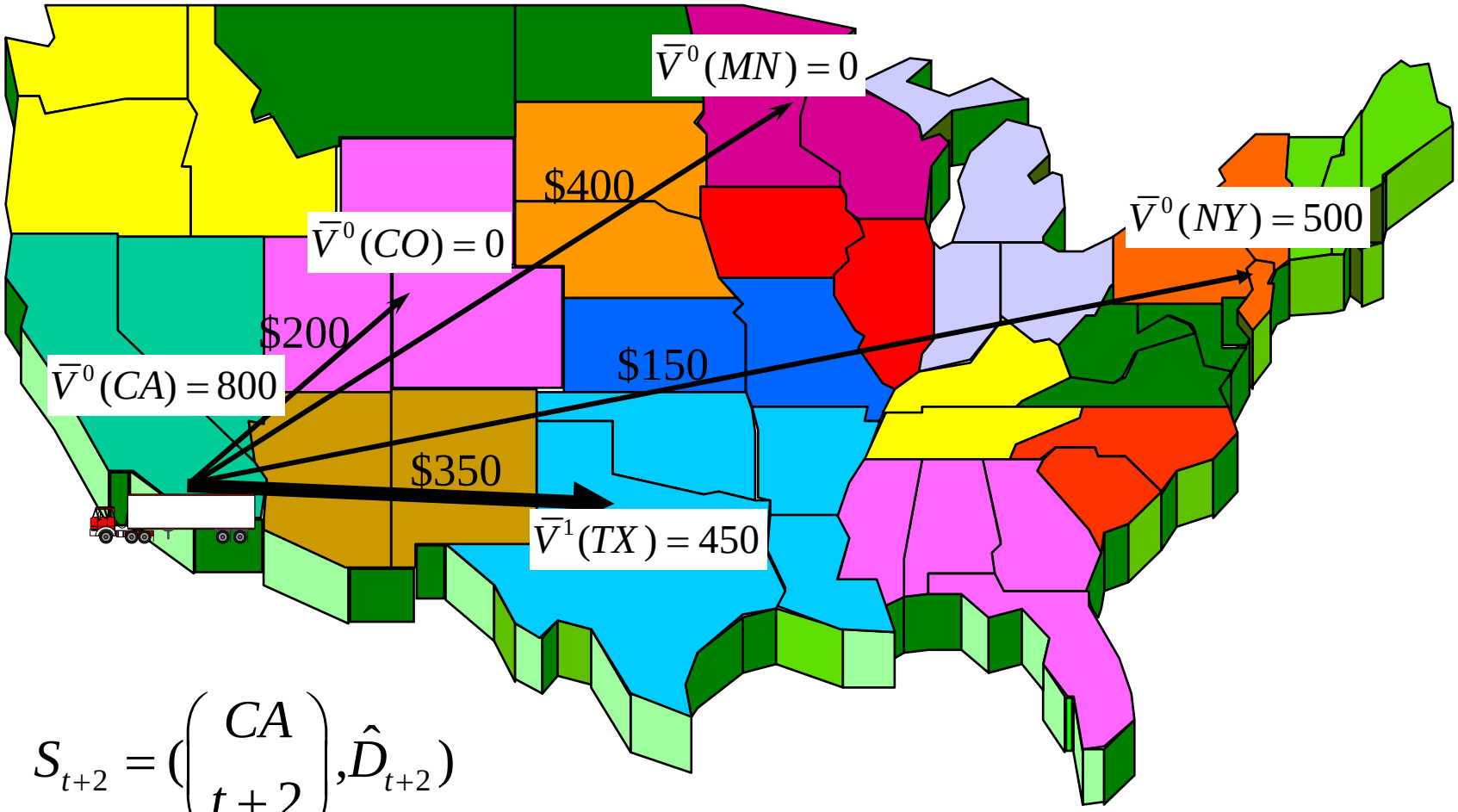
- Move to California.



$$S_{t+2} = \left(\begin{pmatrix} CA \\ t+2 \end{pmatrix}, \hat{D}_{t+2}\right)$$

Nomadic trucker illustration

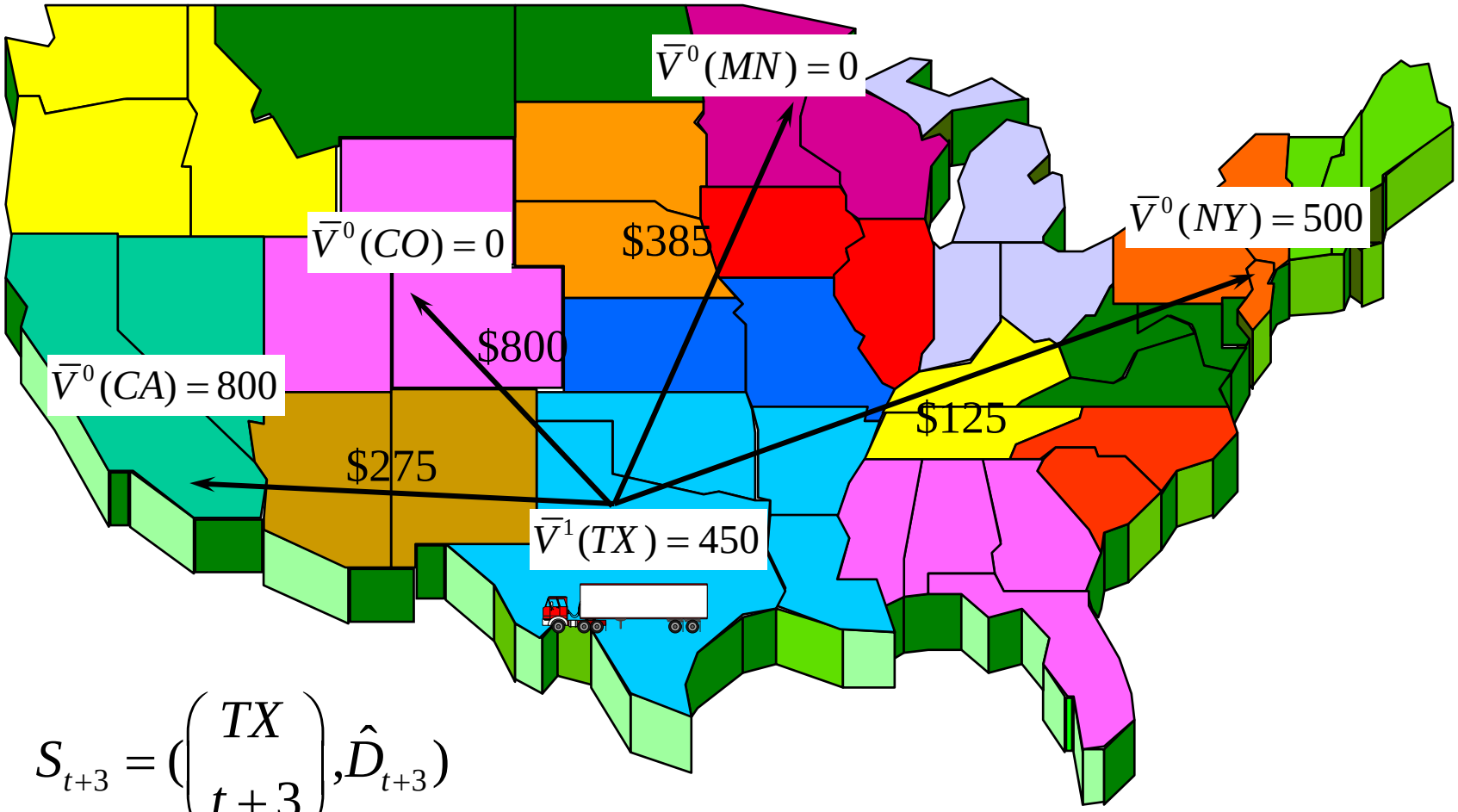
- Make decision to return to TX and update value of being in CA



$$S_{t+2} = \left(\begin{pmatrix} CA \\ t+2 \end{pmatrix}, \hat{D}_{t+2}\right)$$

Nomadic trucker illustration

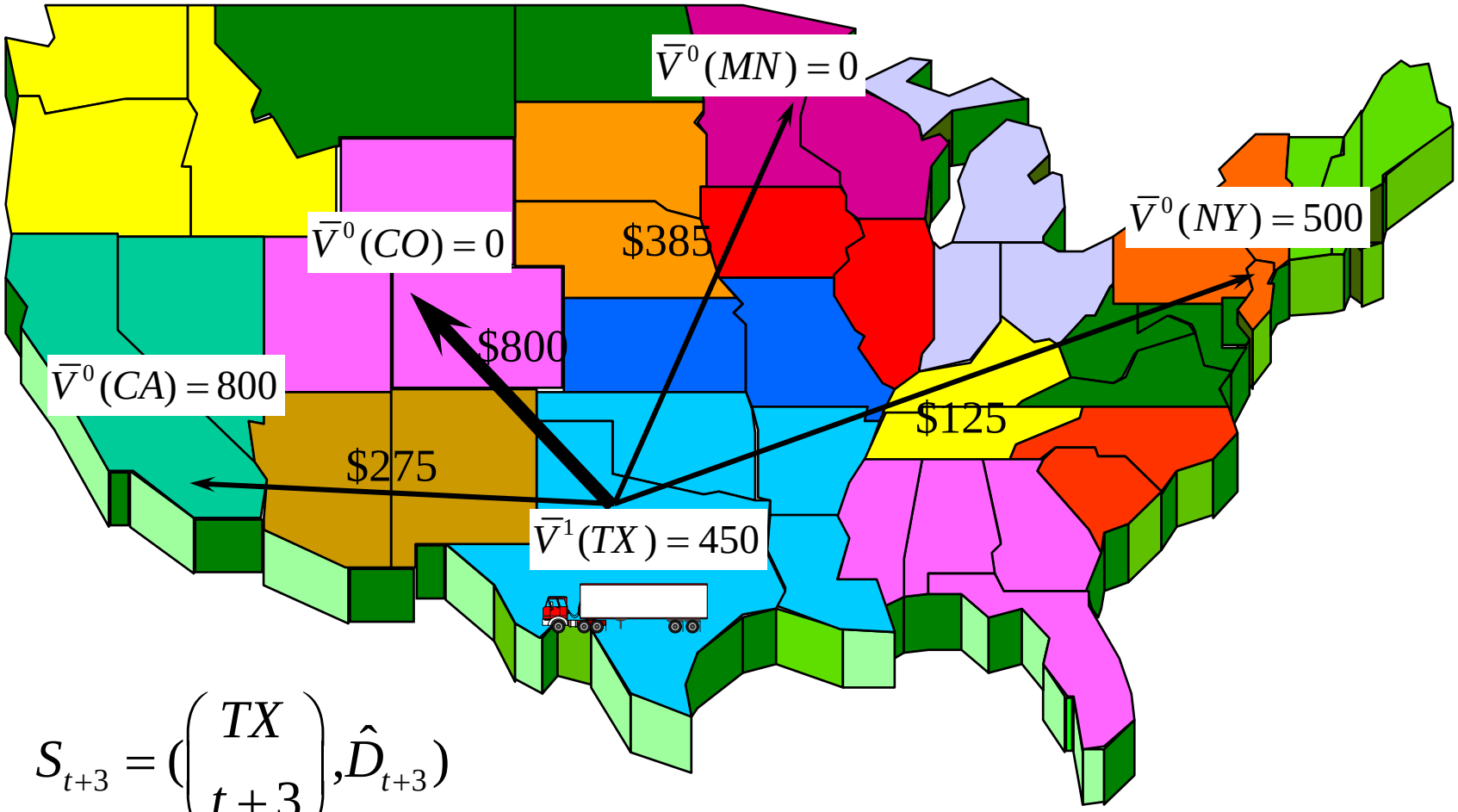
- Back in TX, we repeat the process, observing a different set of demands.



$$S_{t+3} = \left(\begin{pmatrix} TX \\ t+3 \end{pmatrix}, \hat{D}_{t+3}\right)$$

Nomadic trucker illustration

- We get a different decision and a new estimate of the value of being in TX



$$S_{t+3} = \left(\begin{pmatrix} TX \\ t+3 \end{pmatrix}, \hat{D}_{t+3}\right)$$

Nomadic trucker illustration

- Updating the value function:

Old value:

$$\bar{V}^1(TX) = \$450$$

New estimate:

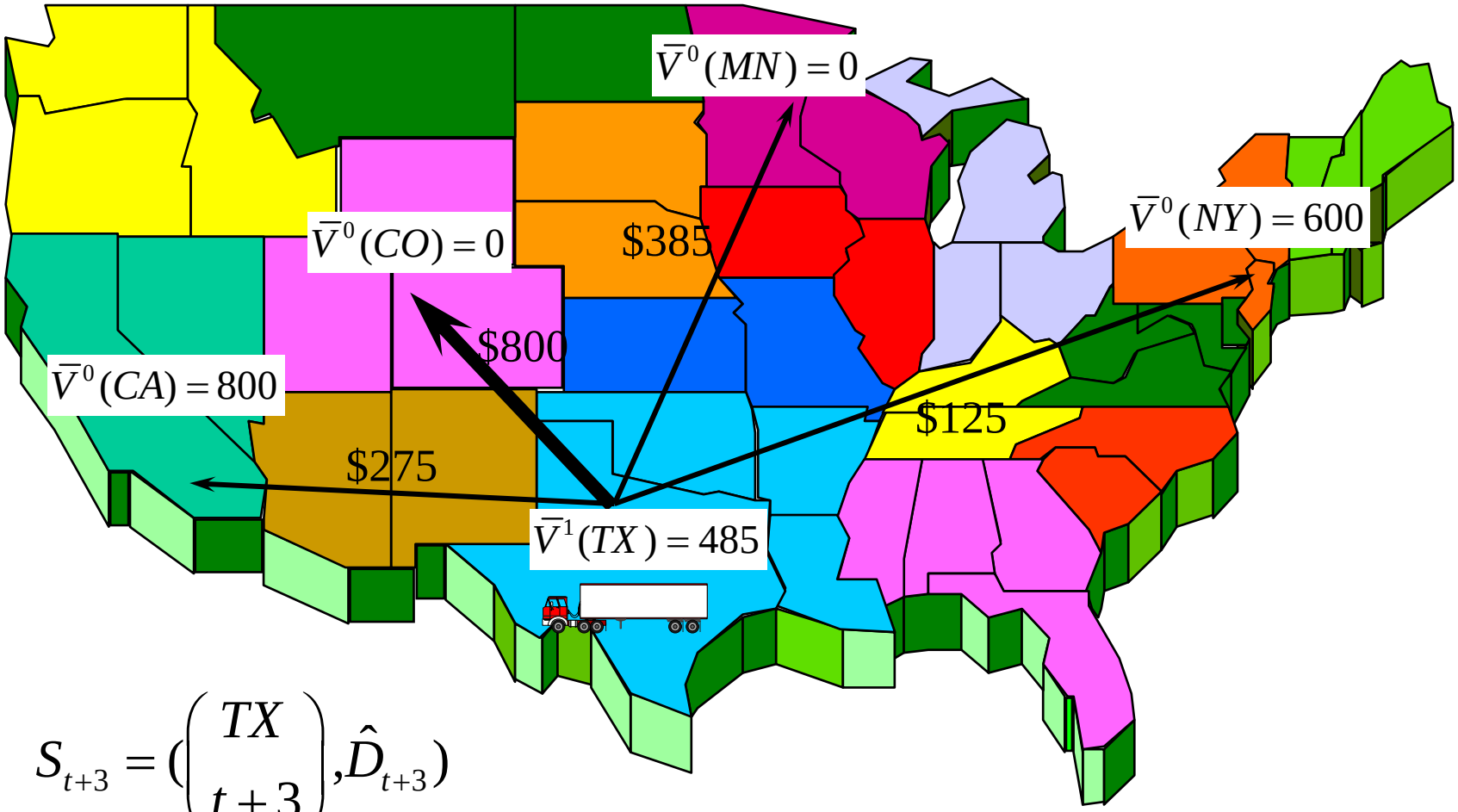
$$\hat{v}^2(TX) = \$800$$

How do we merge old with new?

$$\begin{aligned}\bar{V}^2(TX) &= (1-\alpha)\bar{V}^1(TX) + (\alpha)\hat{v}^2(TX) \\ &= (0.90)\$450 + (0.10)\$800 \\ &= \$485\end{aligned}$$

Nomadic trucker illustration

- An updated value of being in TX



$$S_{t+3} = \left(\begin{pmatrix} TX \\ t+3 \end{pmatrix}, \hat{D}_{t+3}\right)$$

Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

The states of our system

- Now let's take a look at what we just did:

a_t = Attribute of our nomadic trucker at time t before the decision is made.

$\hat{D}_t$ = Vector of demands that are revealed at time t.

$S_t = \left(a_t, \hat{D}_t\right)$ Pre-decision state variable.

a_t^x = Attribute of our nomadic trucker at time t after the decision is made.

$S_t^x = a_t^x$ Post-decision state variable.

A single, complex entity

- Pre- and post-decision attributes for our nomadic truck driver:

	$t = 40$ Pre-decision	$t = 40$ Decision	$t = 40$ Post-decision		$t = 50$ Pre-decision
City	Dallas	Chicago	Chicago	$\cdots$	Chicago
ETA	41.2	-	54.7		56.2
Equip	Good	-	Good		Repair

Multiple, complex entities

- Notation for multiple entities:
  - The truck state vector:

$a =$ The attributes of the truck

$a \in \mathrm{A}$ The attribute space

$R_{ta}^{truck} =$ The number of trucks with attribute a

$R_t^{truck} = \left(R_{ta}^{truck} \right)_{a \in \mathrm{A}}$ The truck state vector

  - The information process:

$\hat{R}_{ta}^{truck} =$ The change in the number of trucks with attribute a.

Multiple, complex entities

- Modeling the fleet management problem:
  - The load state vector:

$b =$ The attributes of a load to be moved.

$b \in \mathrm{B}$ The attribute space

$R_{tb}^{load} =$ The number of tasks with attribute b

$R_{t}^{load} = \left(R_{tb}^{load} \right)_{b \in \mathrm{B}}$ The load state vector

  - The information process:

$\hat{R}_{tb}^{load} =$ The change in the number of loads with attribute b.

Multiple, complex entities

- Modeling the fleet management problem:
  - The resource state vector (a.k.a. "physical state")

$$R_t = \left(R_t^{truck}, R_t^{load} \right)$$

  - The information process:

$$\hat{R}_t = \text{The number of new arrivals (of drivers and loads) during time interval } t.$$

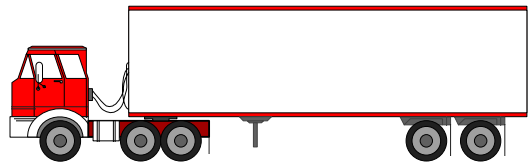
$$= \left(\hat{R}_t^{truck}, R_t^{load} \right)$$

$$= W_t$$

The states of our system

- The state of a single, simple entity:

$$a = [\text{Location}]$$



$$a \in \mathbf{A} \qquad |\mathbf{A}| \approx 100 - 10,000$$

The states of our system

- The state of a single, complex entity:

$$a = \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Equipment type} \\ \text{Home base} \\ \text{Operator attributes} \\ \text{Time in service} \\ \text{Maintenance status} \end{bmatrix}$$

$$a \in \mathbf{A} \qquad |\mathbf{A}| \approx 10^{10} - 10^{100}$$

The curse of dimensionality!

The states of our system

- Multiple, complex entities

$$R_t = \begin{bmatrix} R_{ta_1} \\ R_{ta_2} \\ R_{ta_3} \\ \text{M} \\ \text{M} \\ R_{ta_n} \end{bmatrix}$$

The number of *dimensions* of our state variable is equal to the size of the *state space* for a single entity problem.

The curse of curses.

The states of our system

- What is a state variable?

  - *A minimally dimensioned function of history that necessary and sufficient to compute the decision function, the transition function and the contribution function.*

The states of our system

- The three states of our system
  - » The state of a single resource/entity

$$a_t = \begin{bmatrix} a_{t1} \\ a_{t2} \\ a_{t3} \end{bmatrix}$$

  - » The state of all our resources

$$R_t = \begin{bmatrix} R_{ta_1} \\ R_{ta_2} \\ R_{ta_3} \end{bmatrix}$$

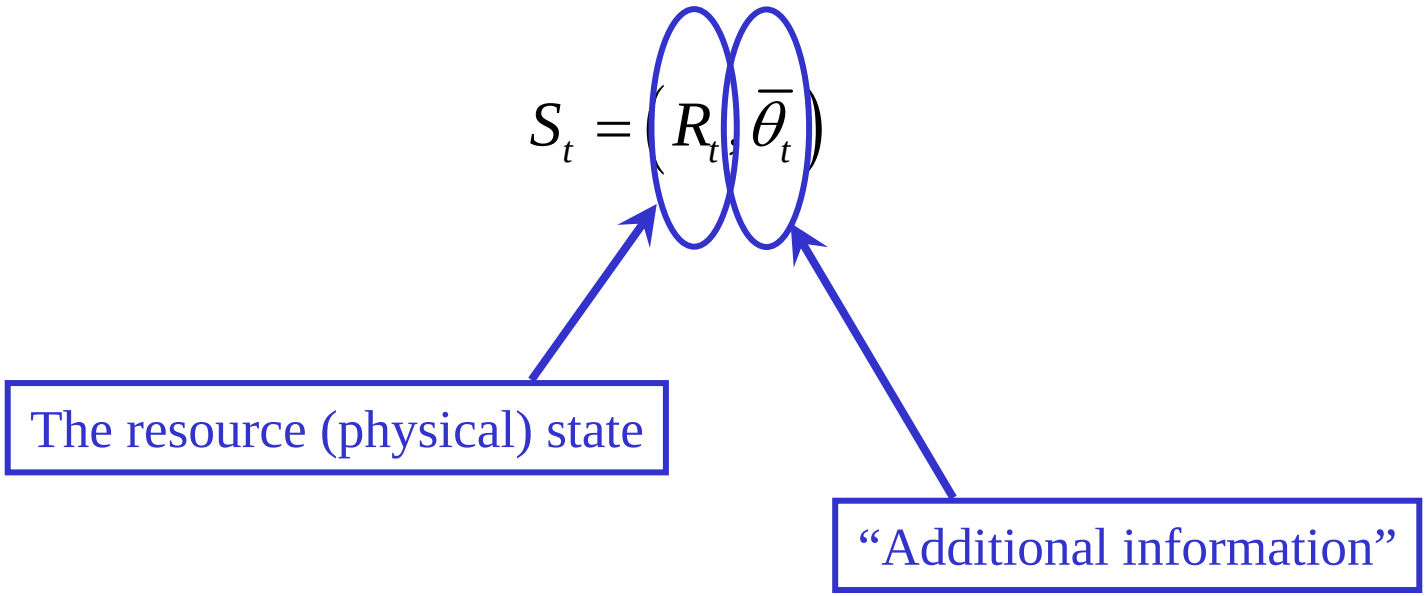
  - » The state of knowledge

$$S_t = \left(R_t, \bar{\theta}_t\right) \qquad \bar{\theta}_t = \text{Estimates of "other parameters"}$$

The states of our system

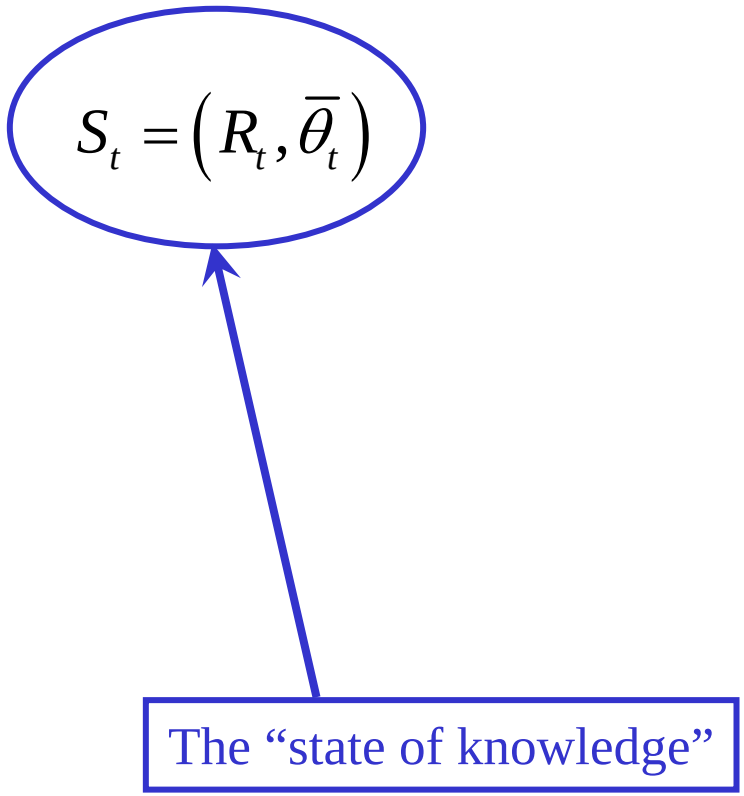
- The state variable

$$S_t = (R_t, \bar{\theta}_t)$$



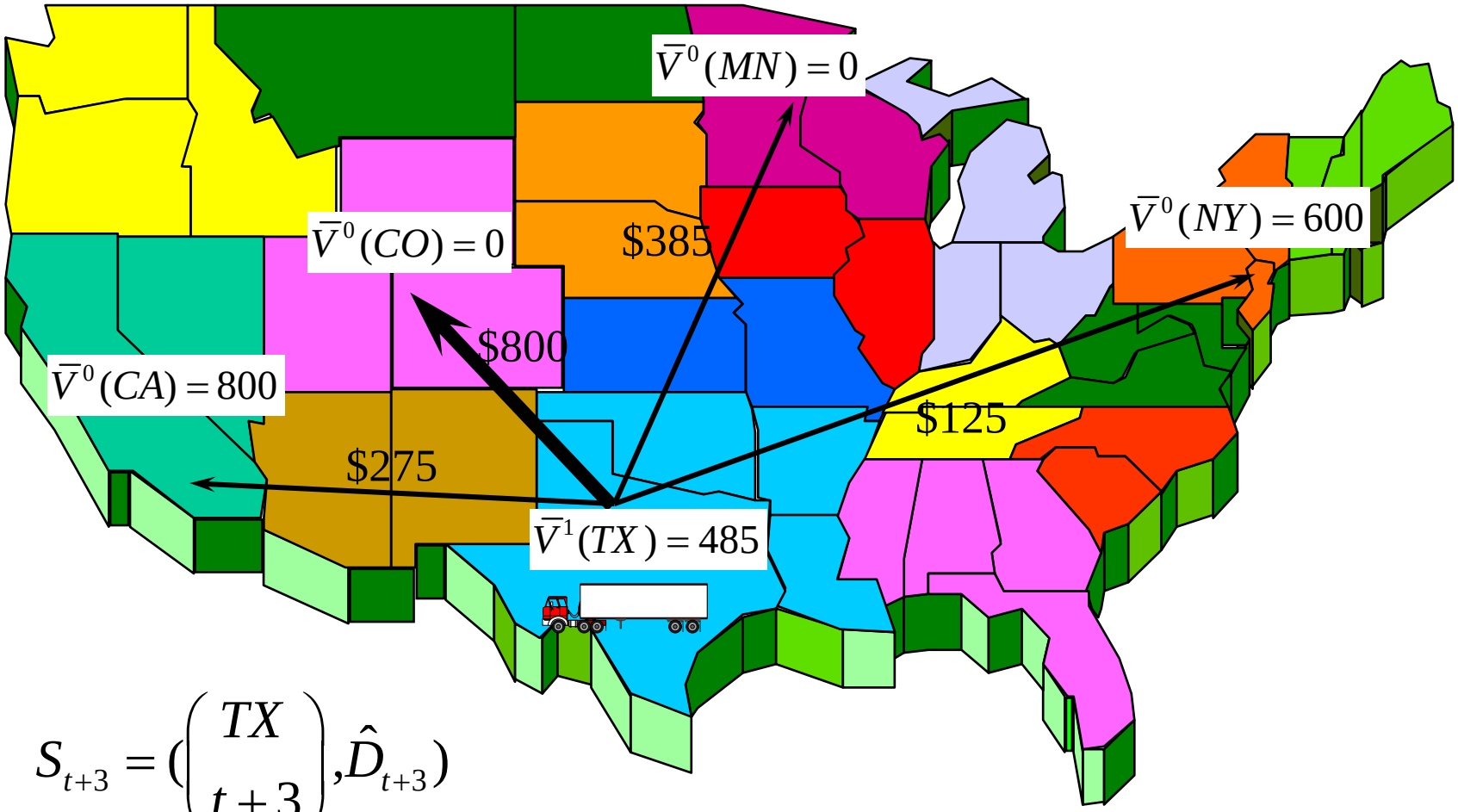
The states of our system

- The state variable



Nomadic trucker illustration

- What is missing from our state variable?



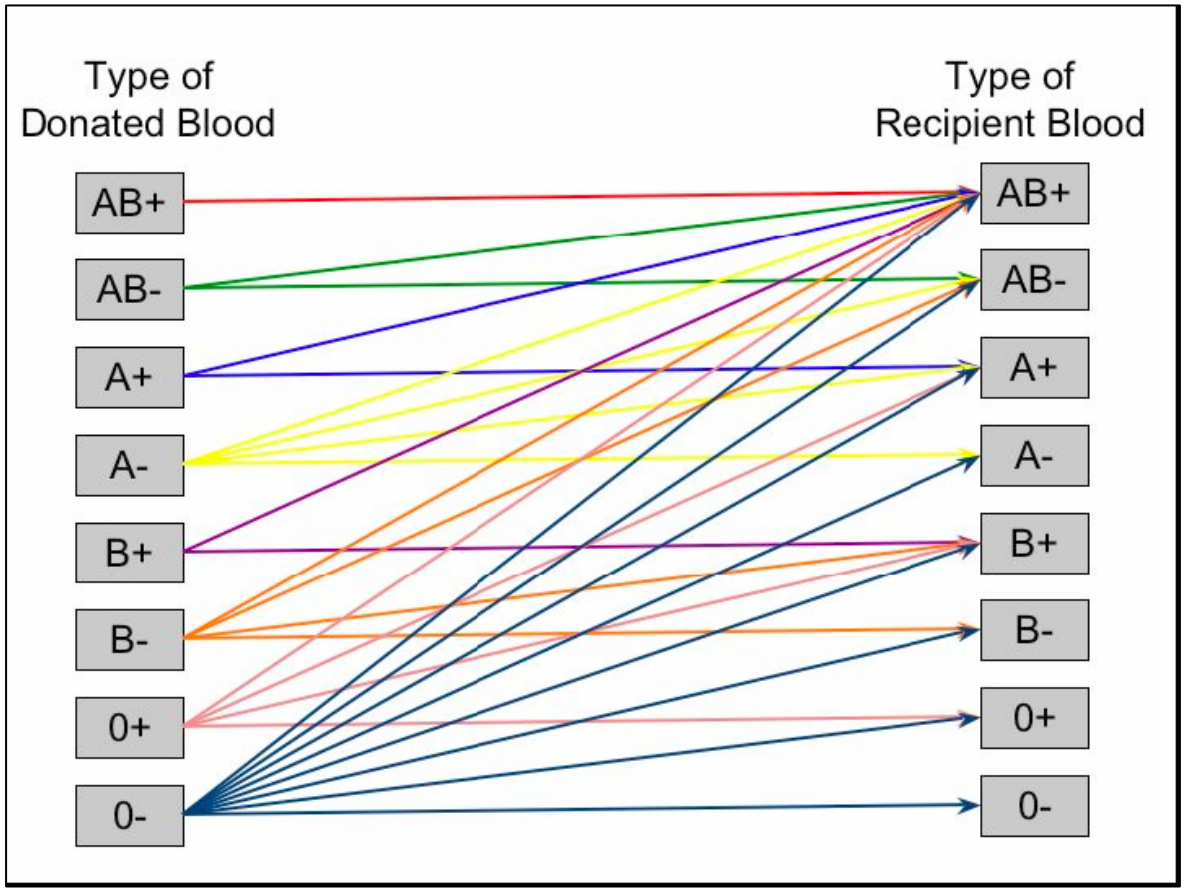
$$S_{t+3} = \left(\begin{pmatrix} TX \\ t+3 \end{pmatrix}, \hat{D}_{t+3}\right)$$

Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

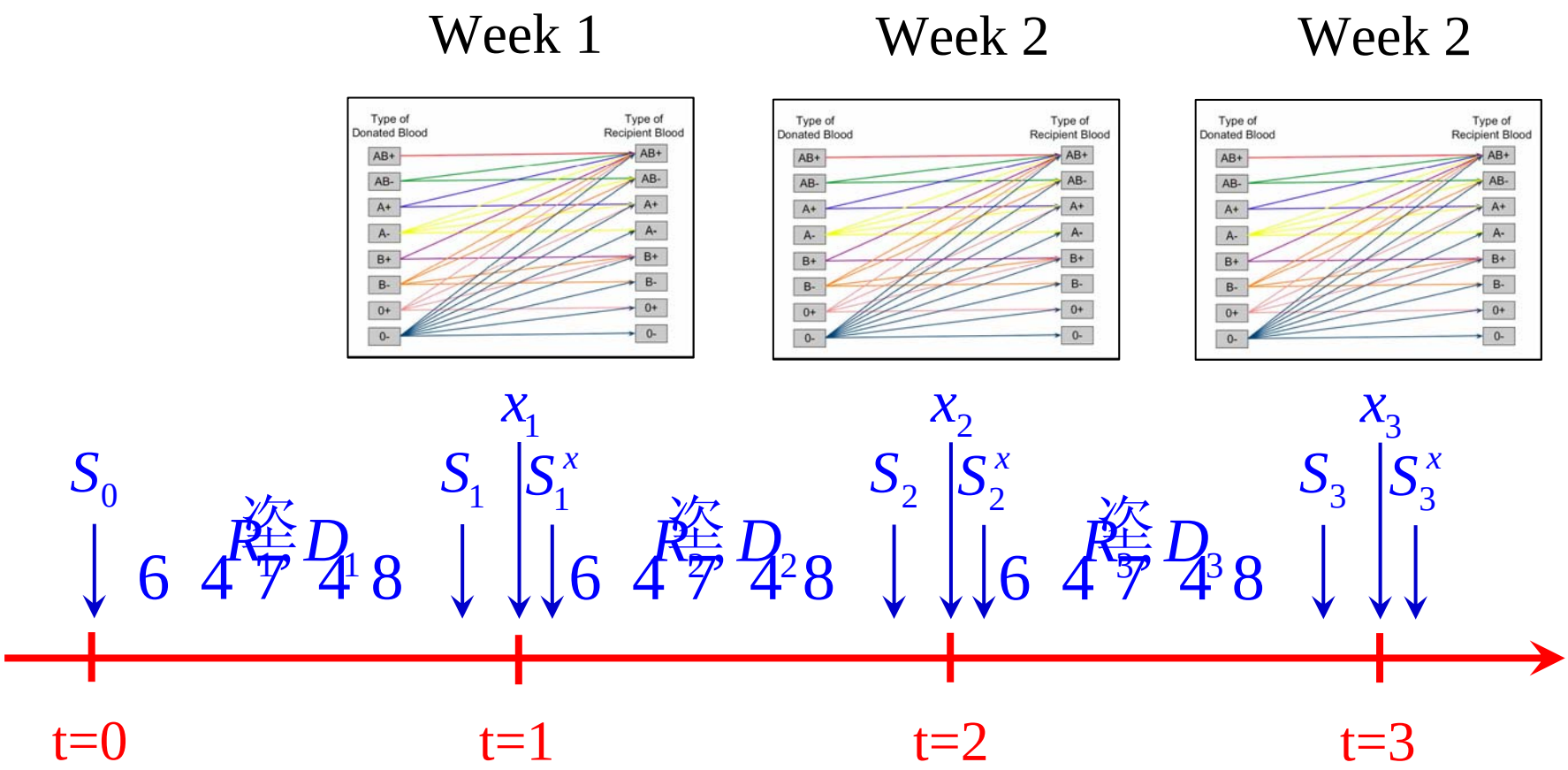
Blood management

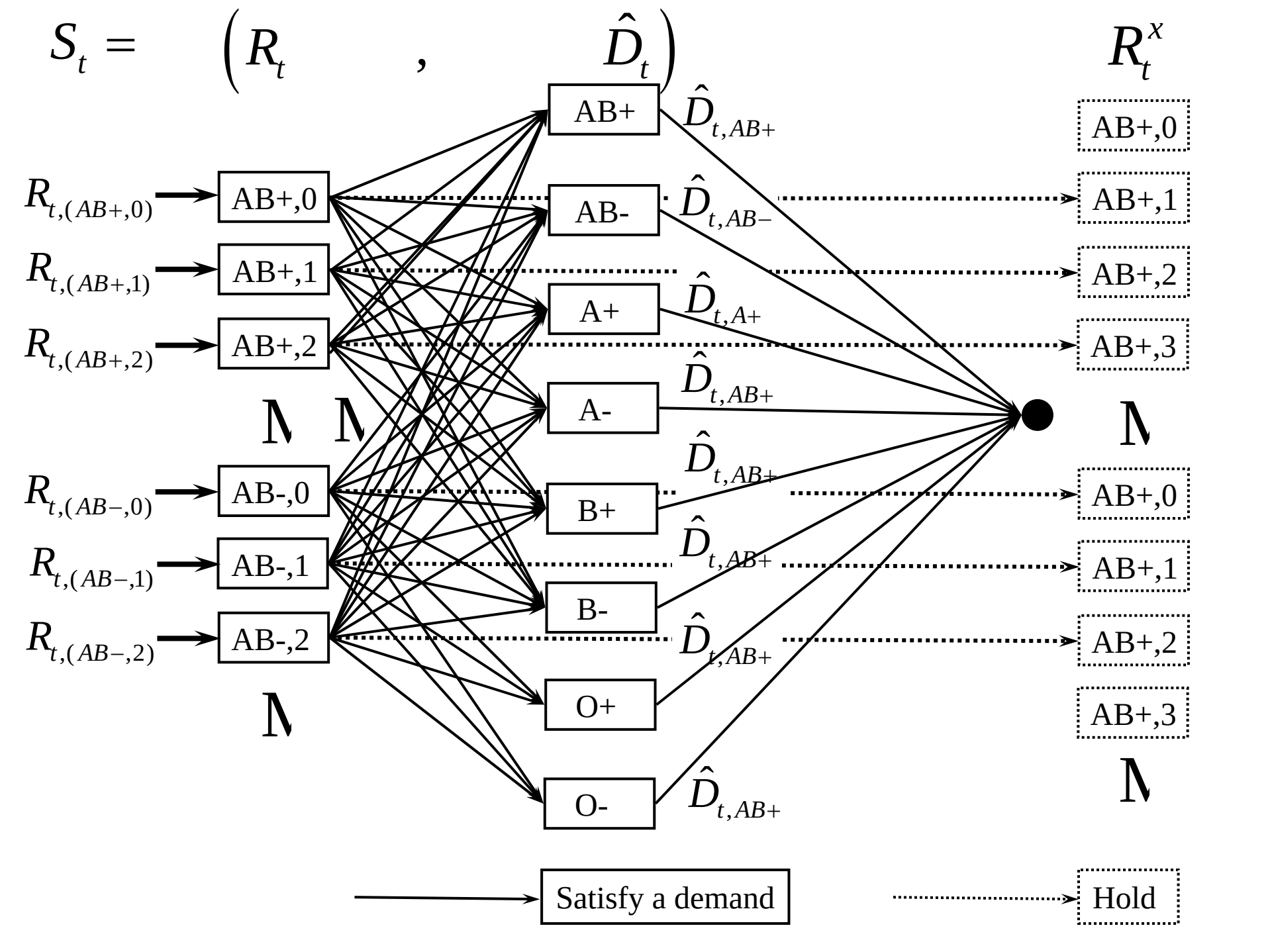
- Managing blood inventories

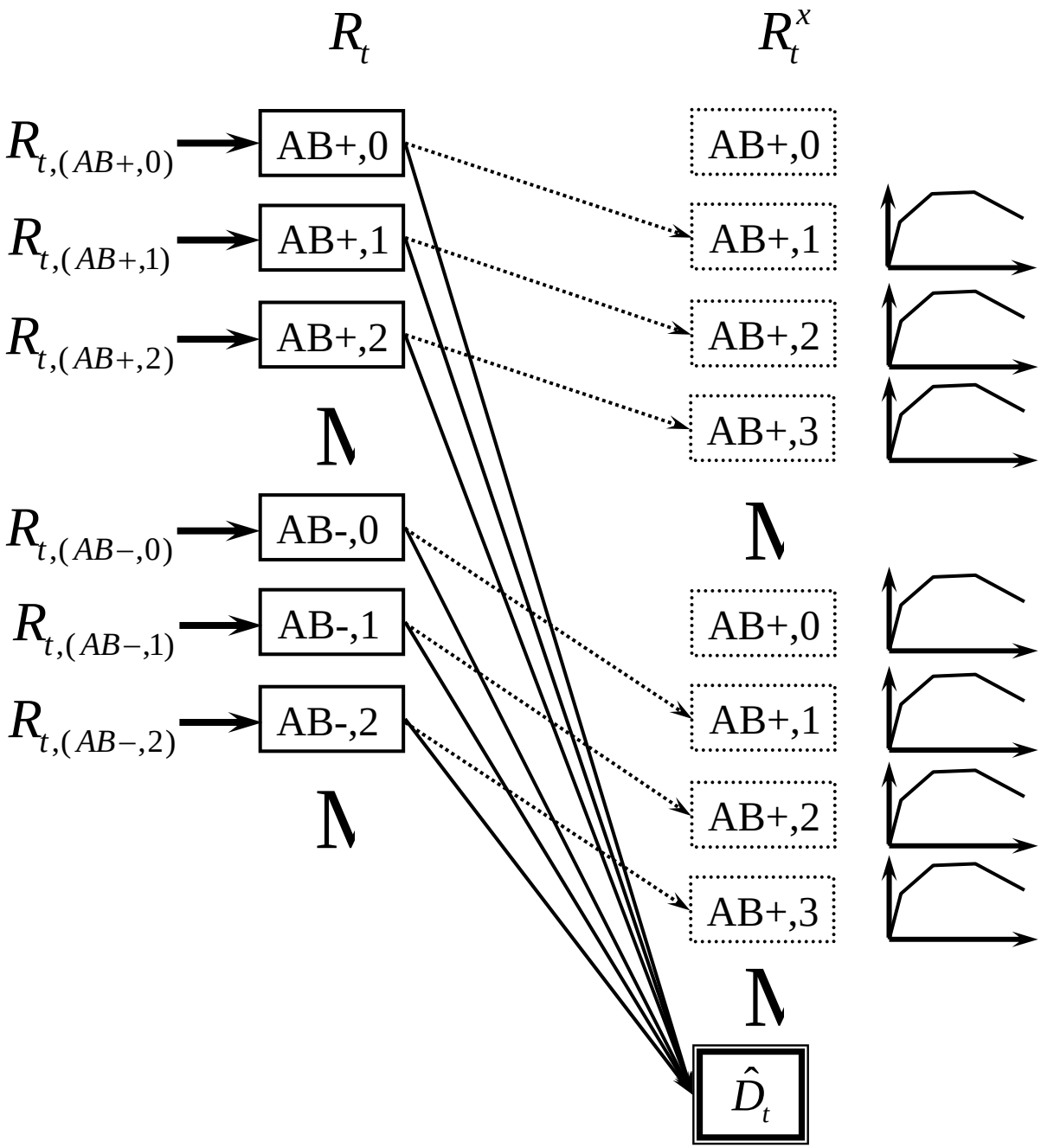


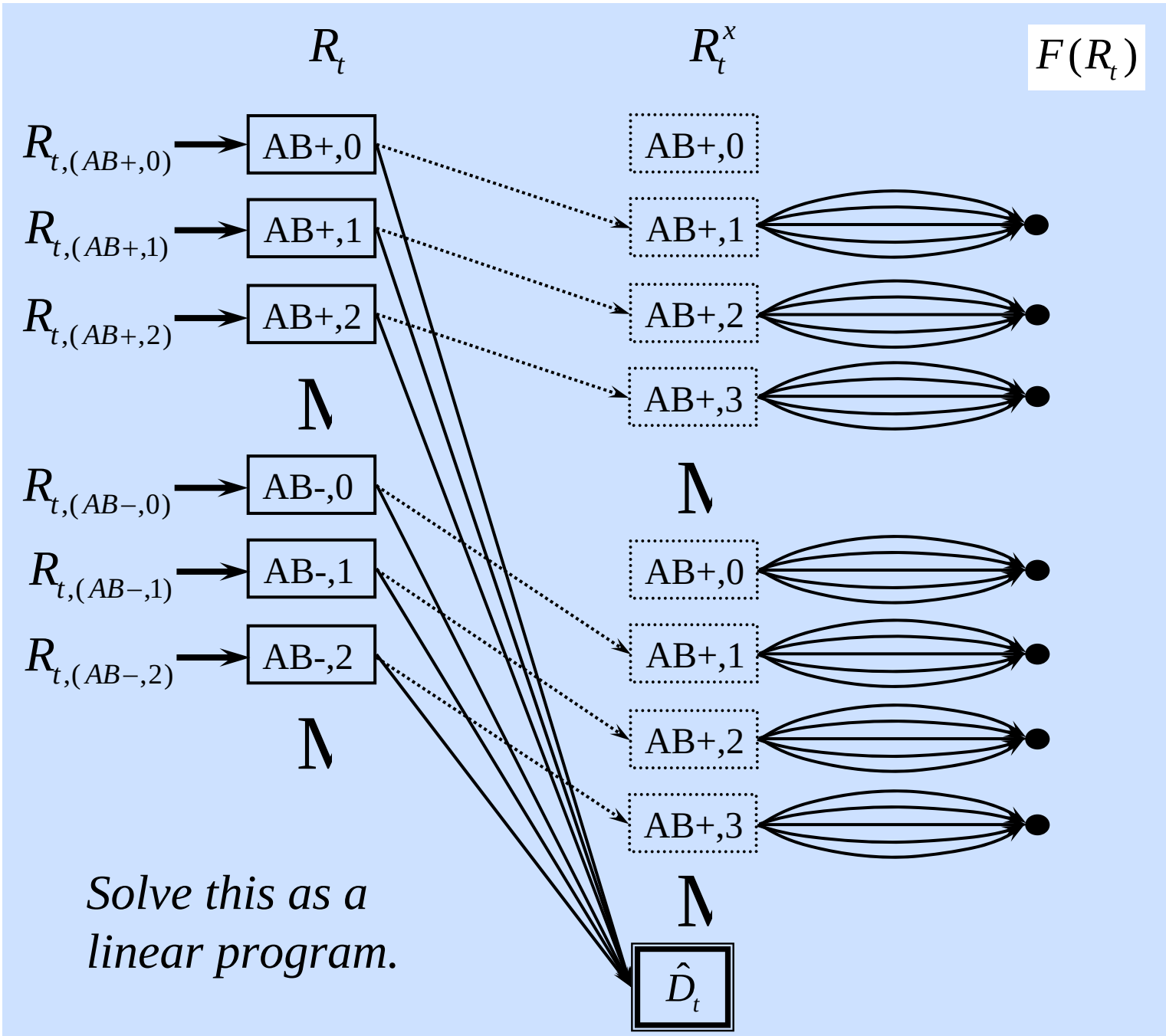
Blood management

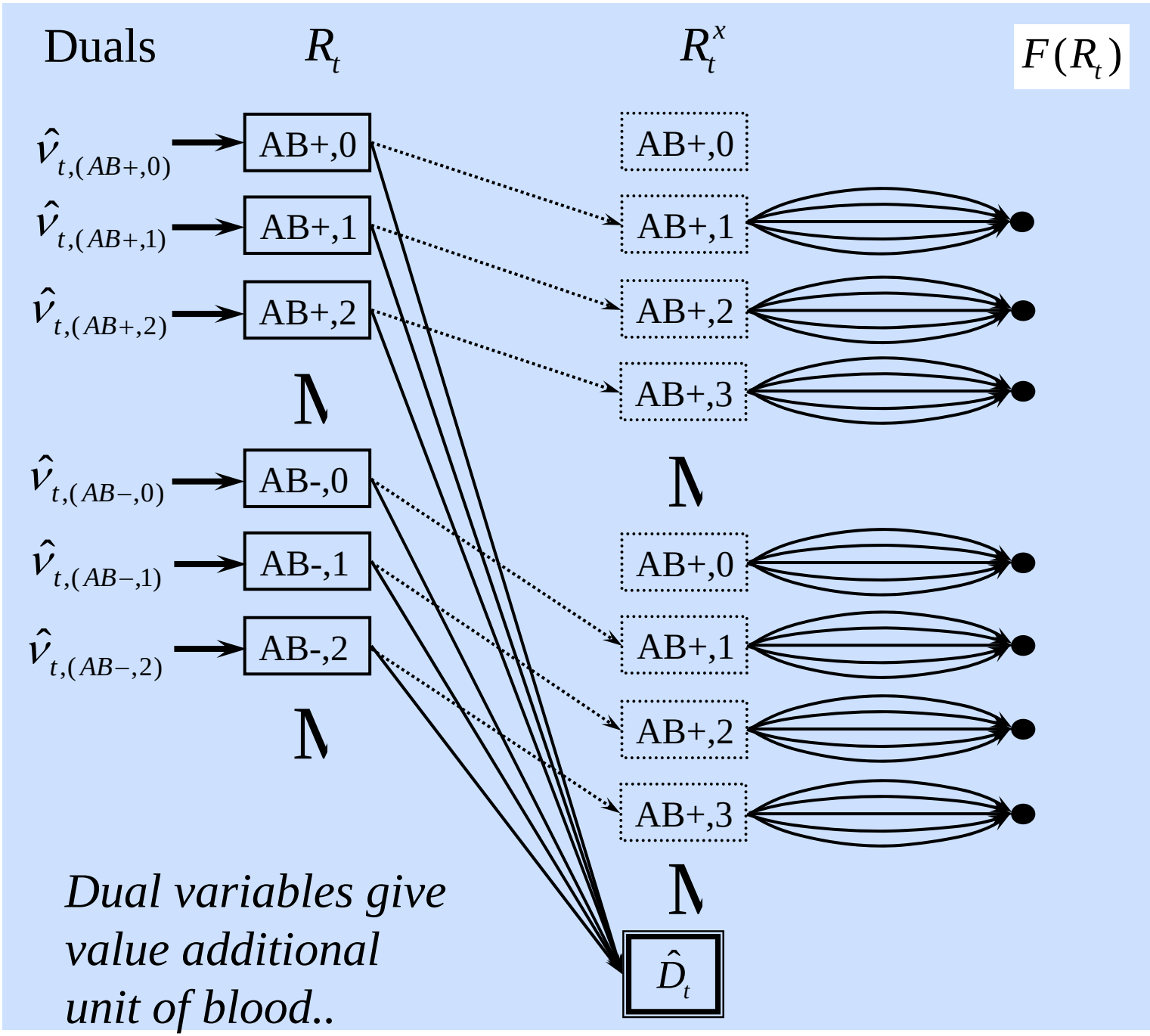
- Managing blood inventories over time




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

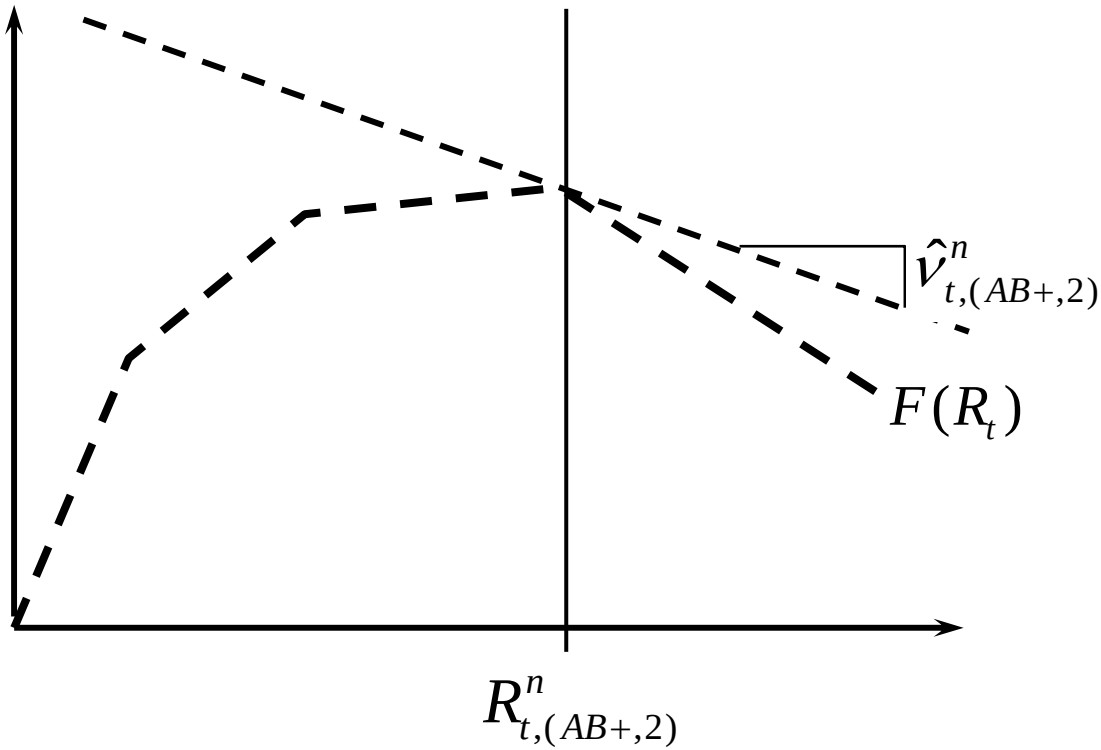

R_t
R_t^x
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(AB-,0)}$
$R_{t,(AB-,1)}$
$R_{t,(AB-,2)}$
AB+,0
AB+,1
AB+,2
AB-,0
AB-,1
AB-,2
AB+,0
AB+,1
AB+,2
AB+,3
AB+,0
AB+,1
AB+,2
AB+,3
$\hat{D}_t$


R_t
R_t^x
$F(R_t)$
$R_{t,(AB+,0)}$
$R_{t,(AB+,1)}$
$R_{t,(AB+,2)}$
$R_{t,(AB-,0)}$
$R_{t,(AB-,1)}$
$R_{t,(AB-,2)}$
AB+,0
AB+,1
AB+,2
AB-,0
AB-,1
AB-,2
AB+,0
AB+,1
AB+,2
AB+,3
AB+,0
AB+,1
AB+,2
AB+,3
$\hat{D}_t$
Solve this as a linear program.


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

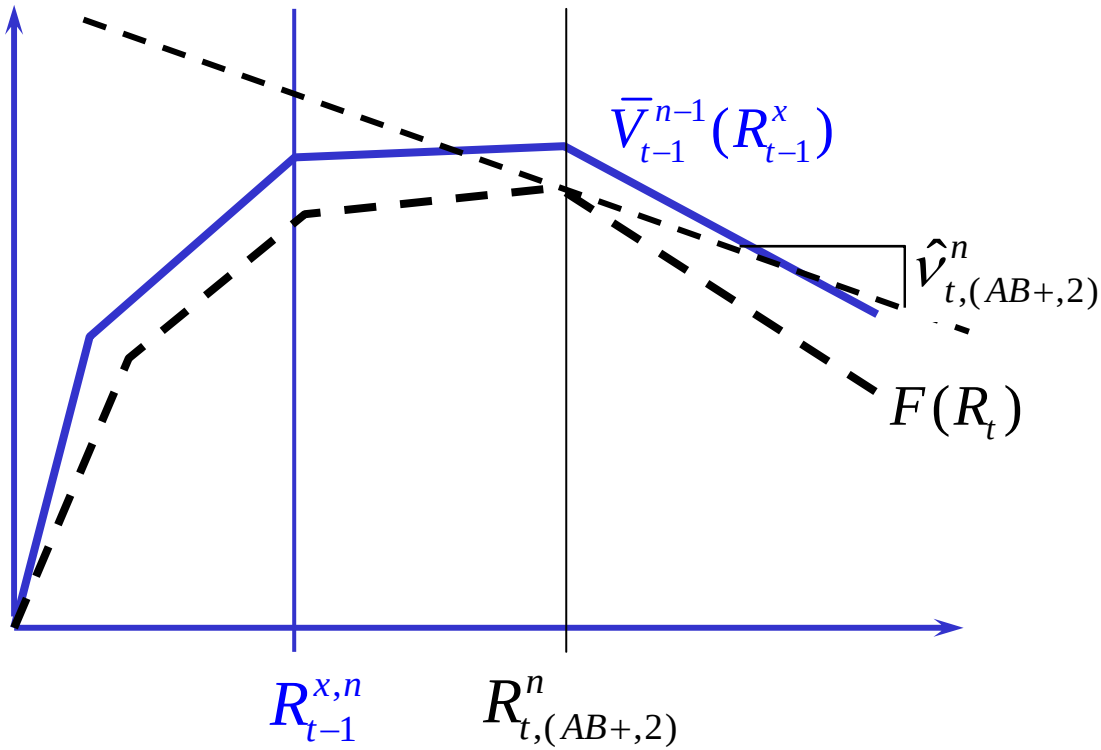
Updating the value function approximation

- Estimate the gradient at R_t^n



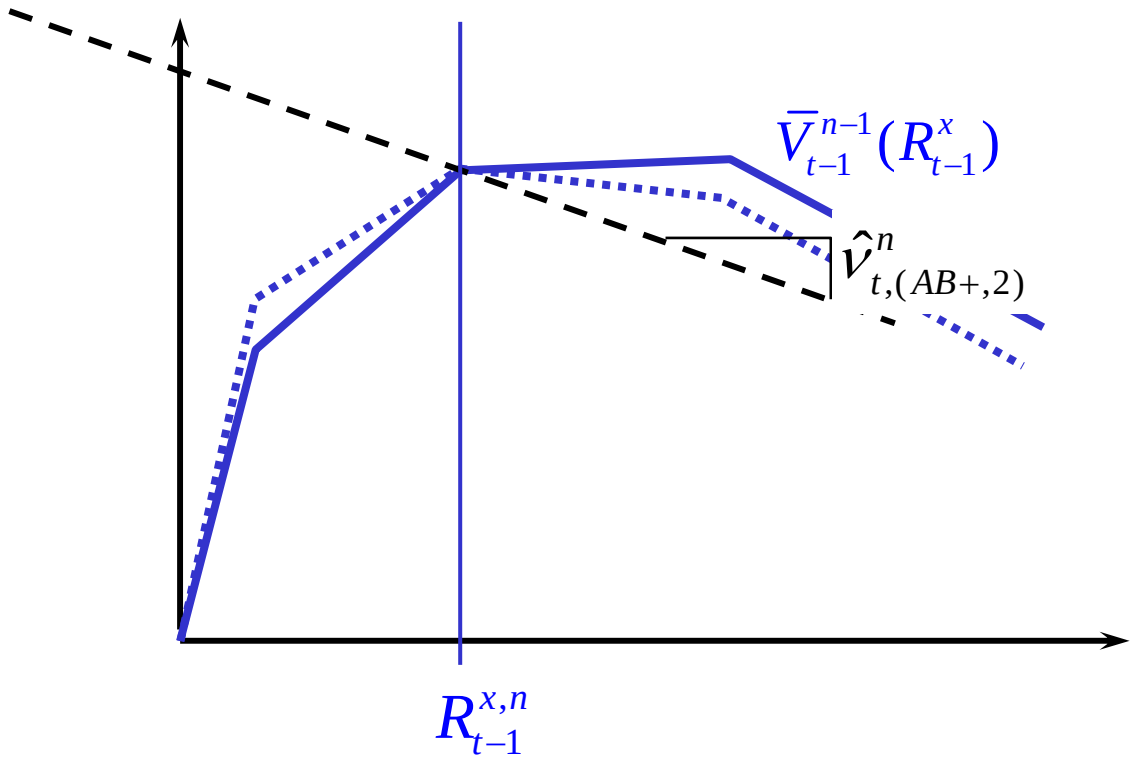
Updating the value function approximation

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



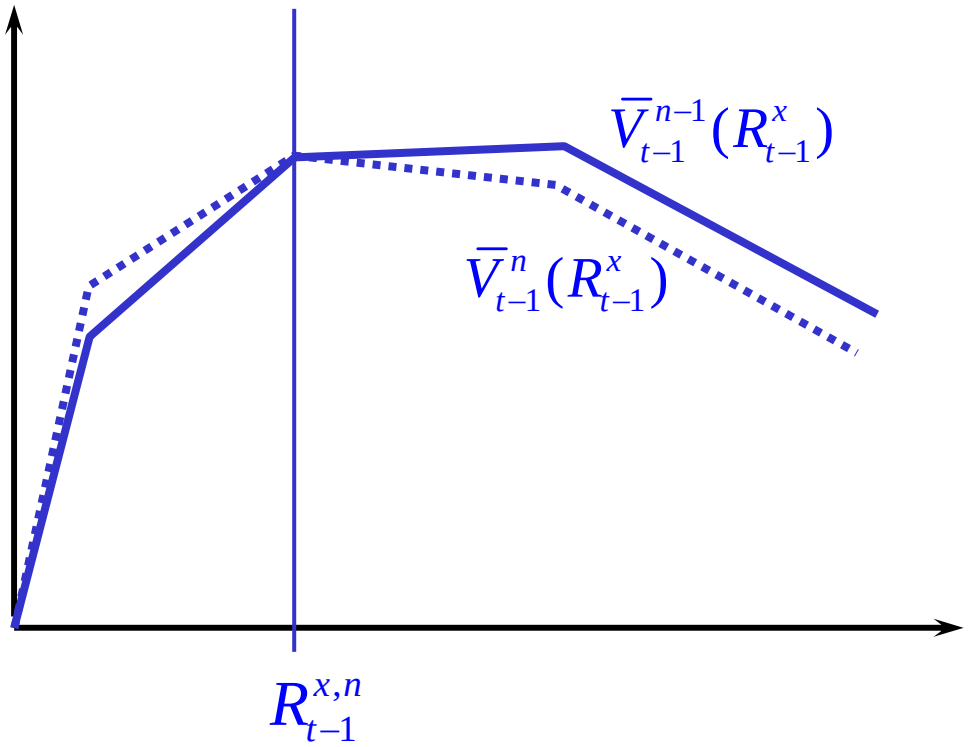
Updating the value function approximation

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



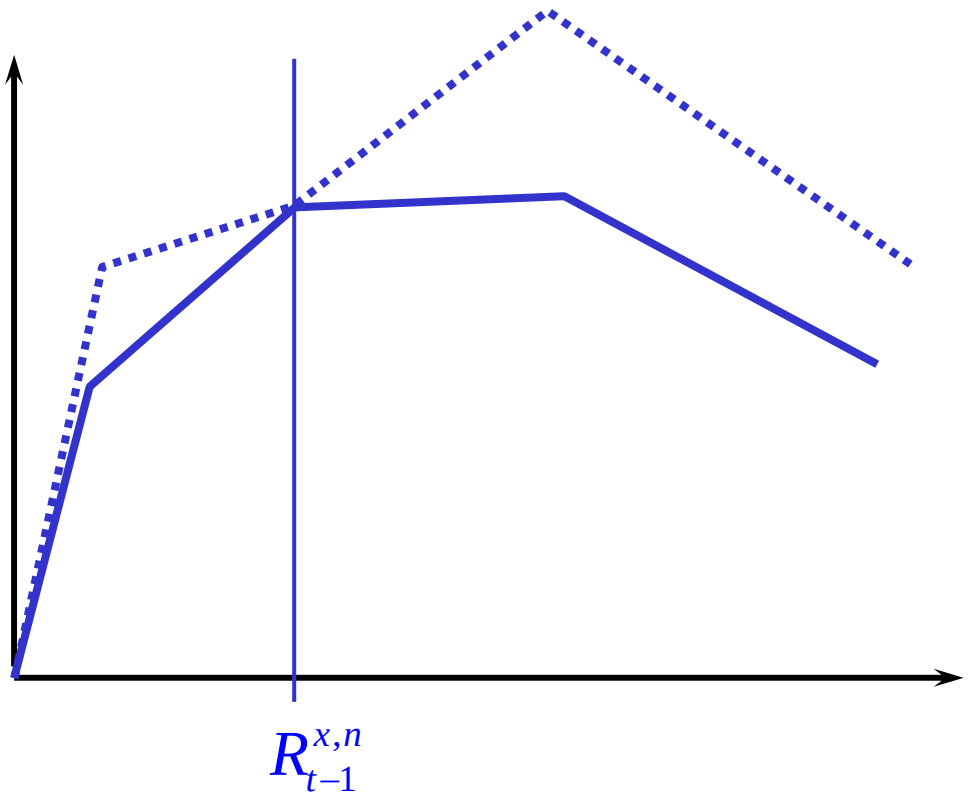
Updating the value function approximation

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



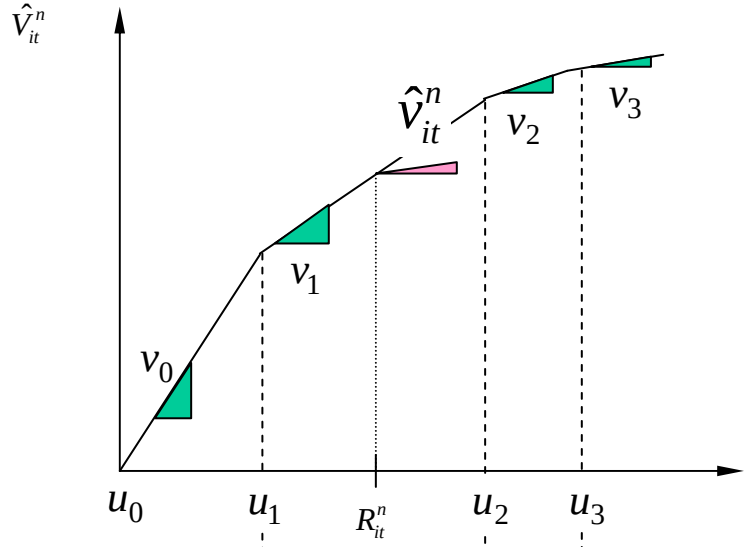
Updating the value function approximation

- The updated function may not be concave:

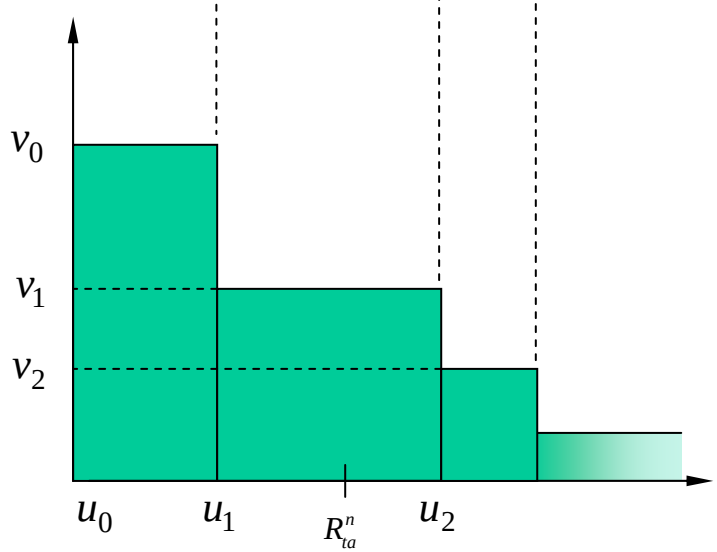


Maintaining concavity

Value function



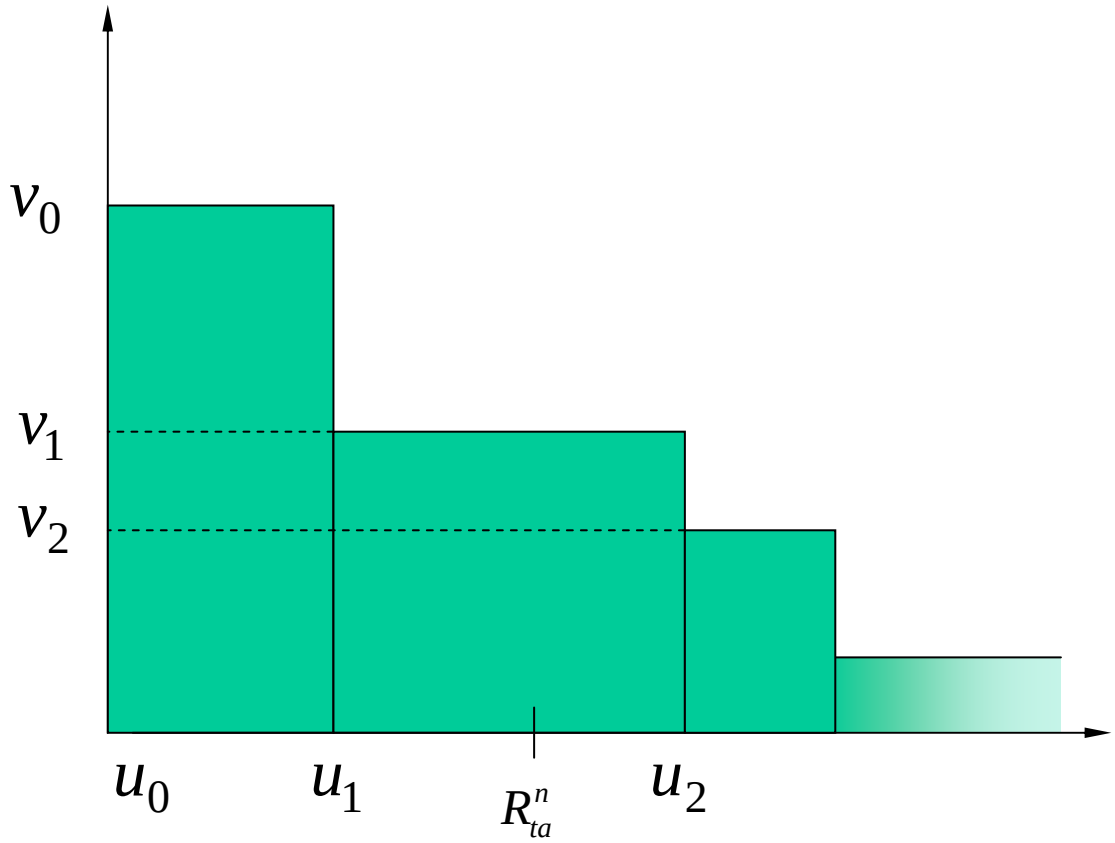
Slopes



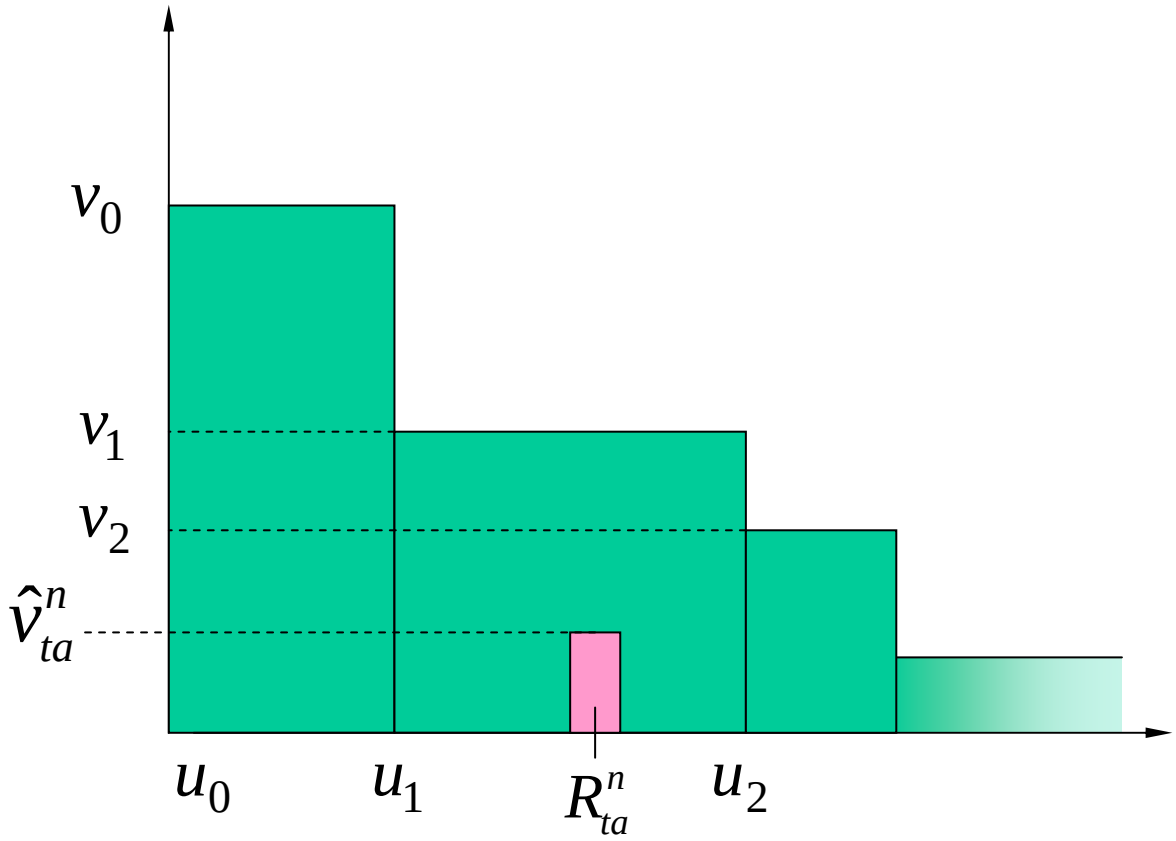
A concave function…

… has monotonically decreasing slopes. But updating the function with a stochastic gradient may violate this property.

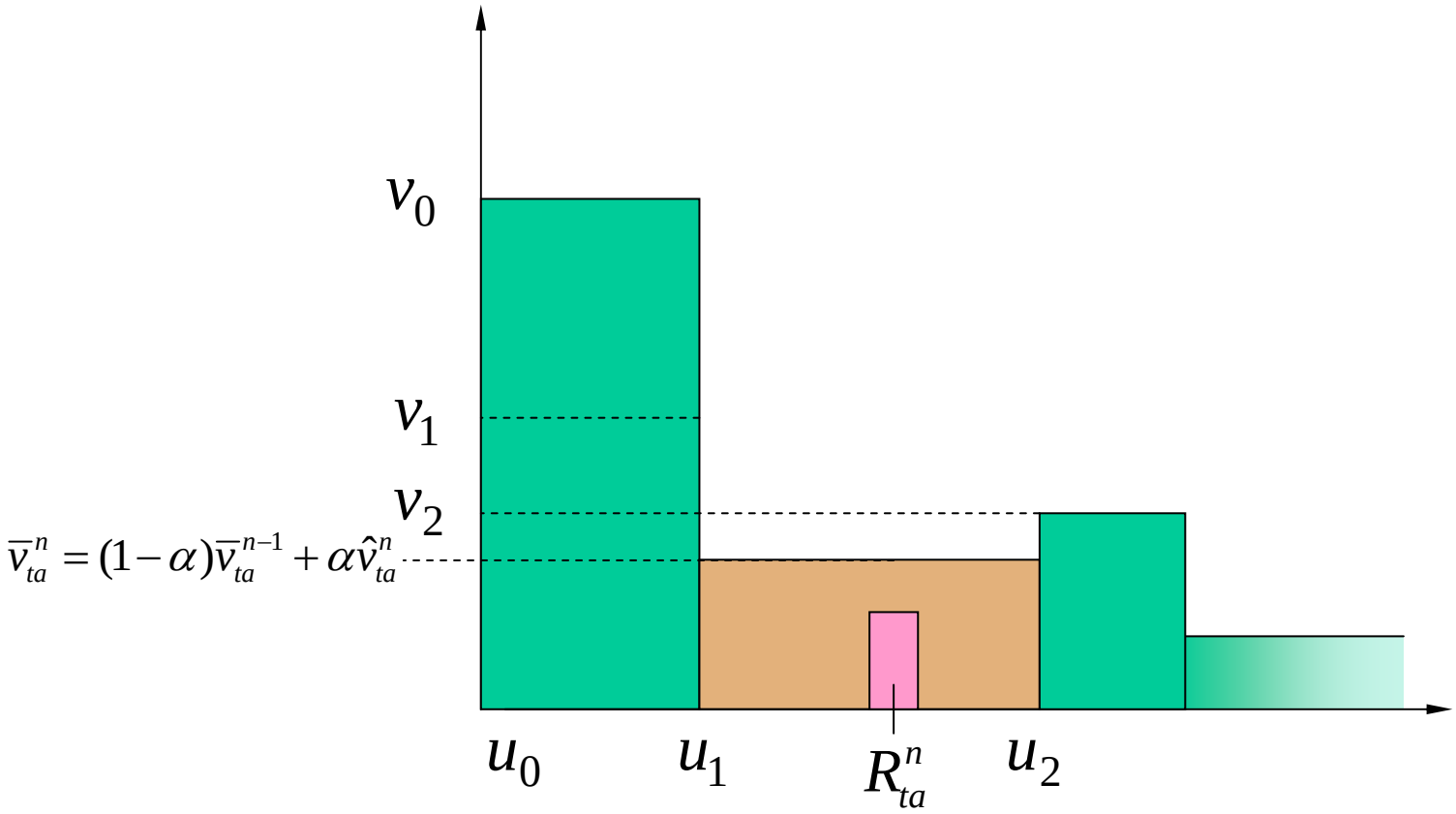
Maintaining concavity



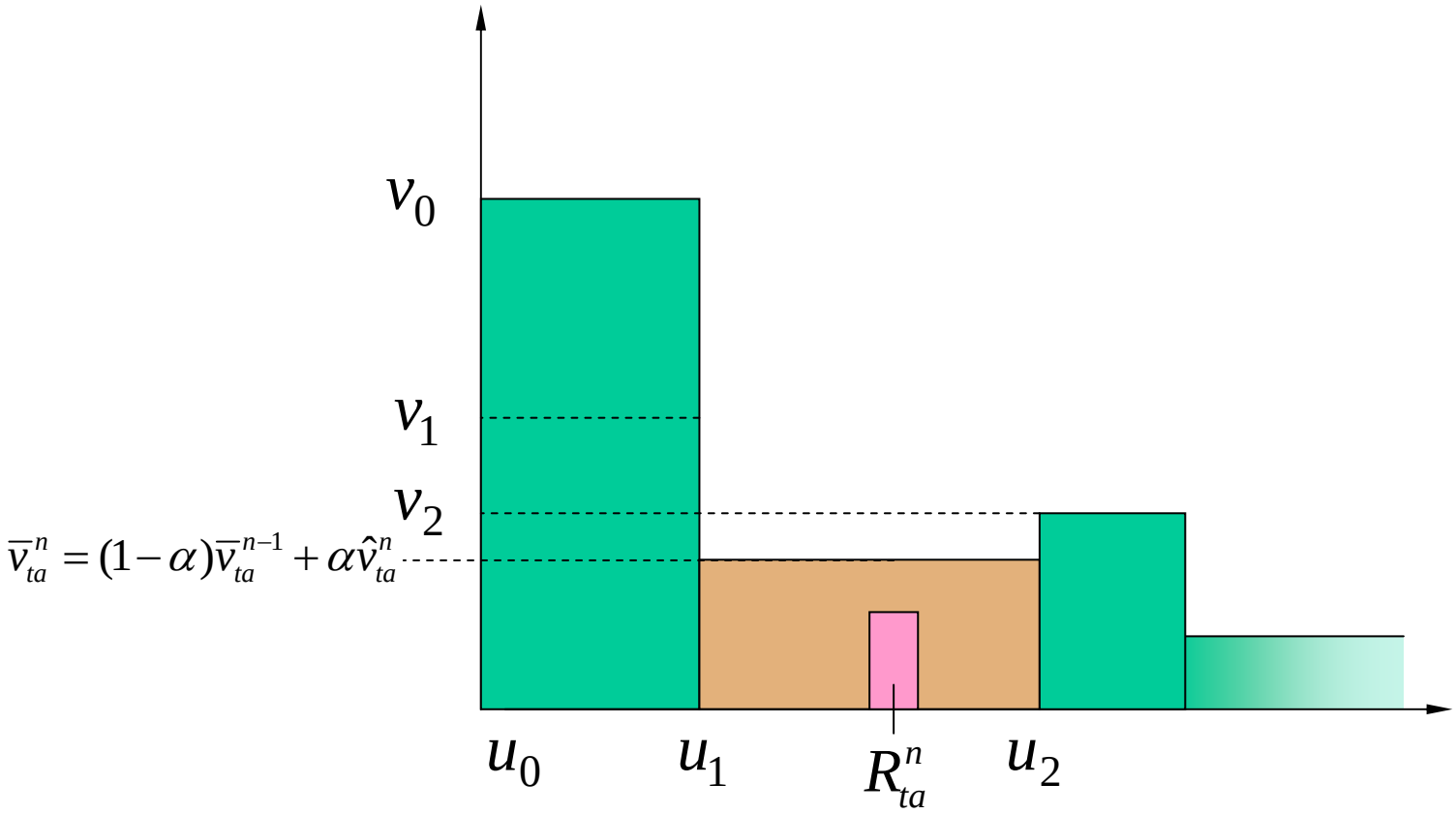
Maintaining concavity



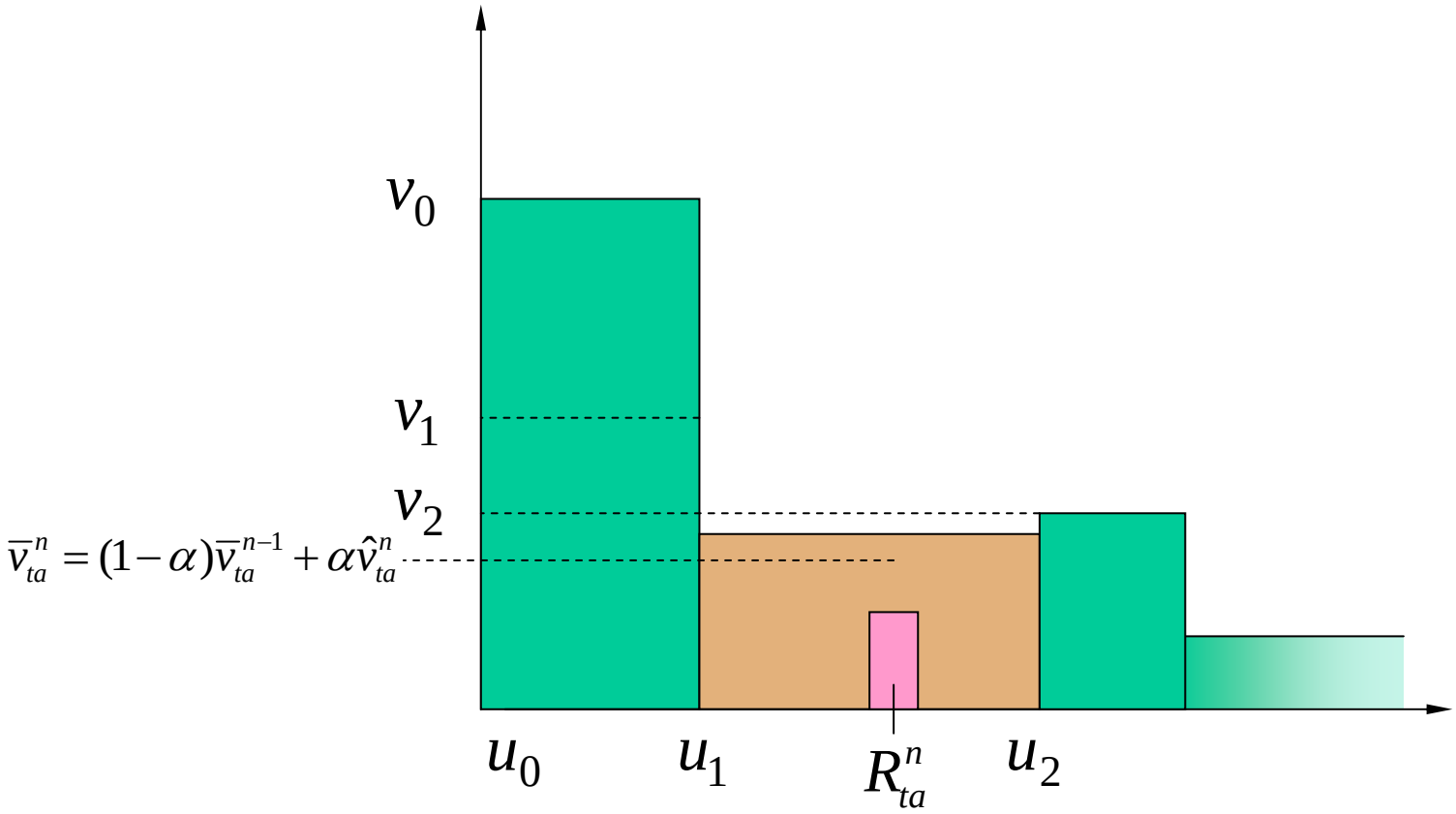
Maintaining concavity



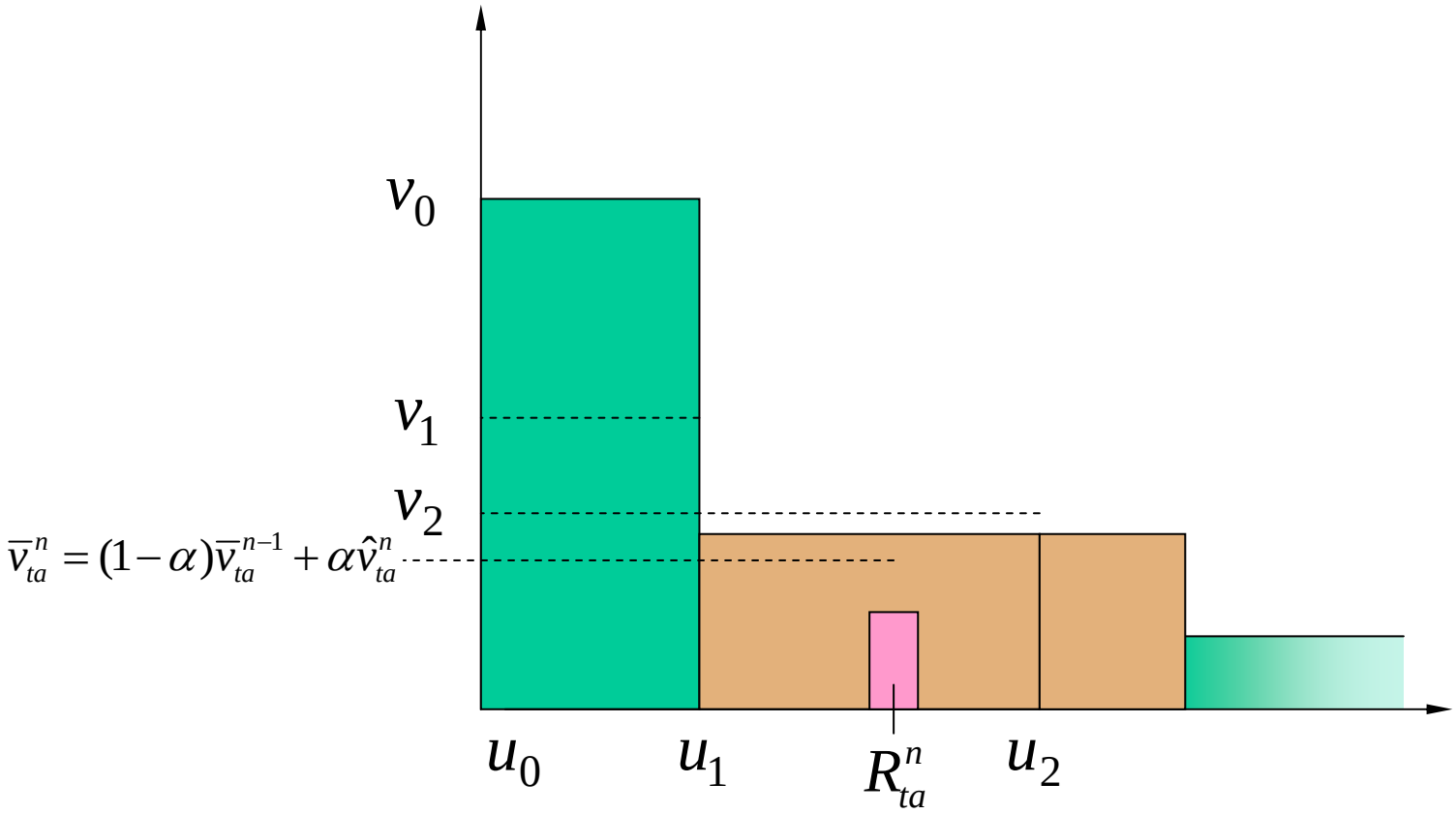
A projection algorithm (SPAR)



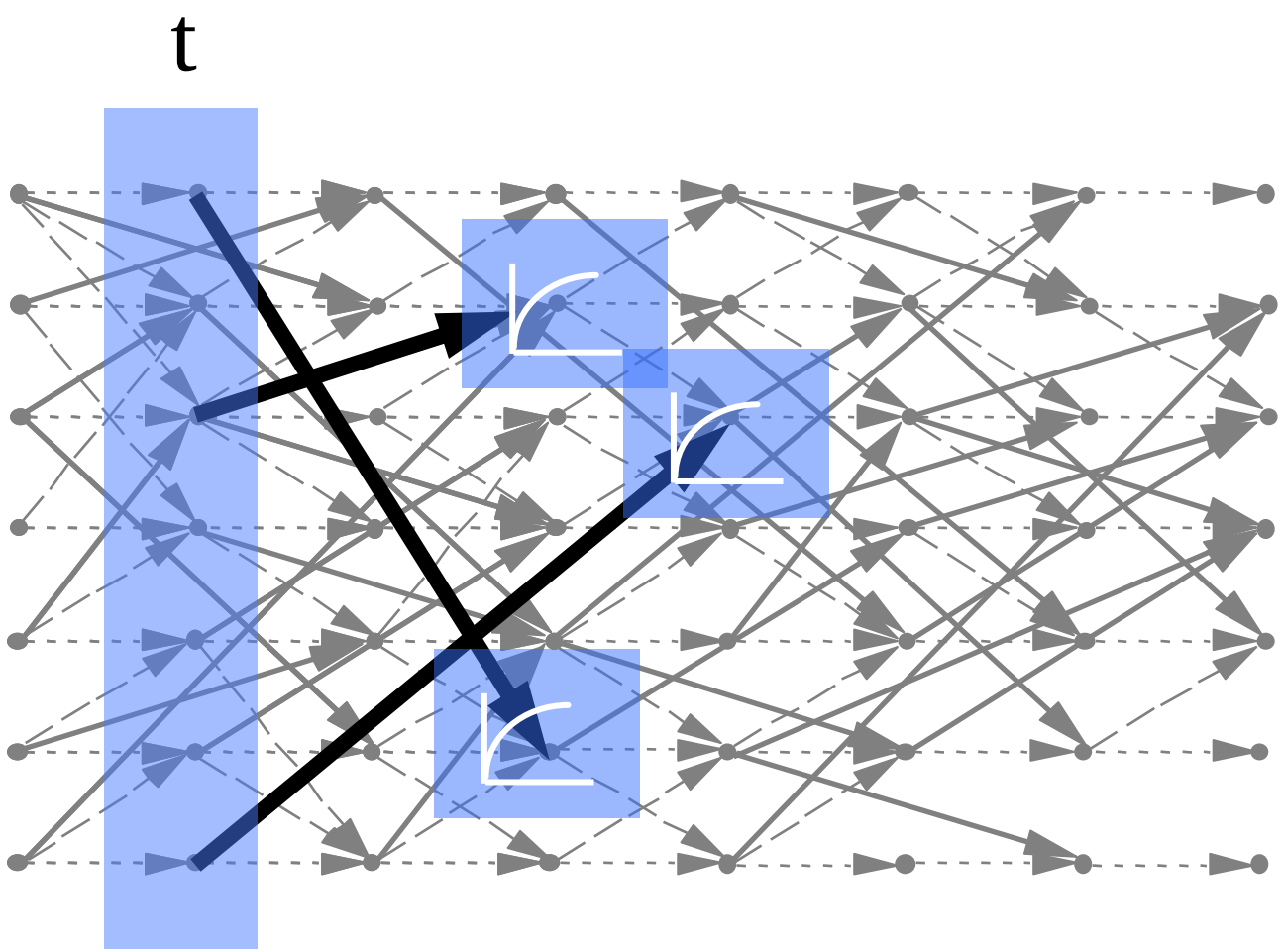
A projection algorithm (SPAR)



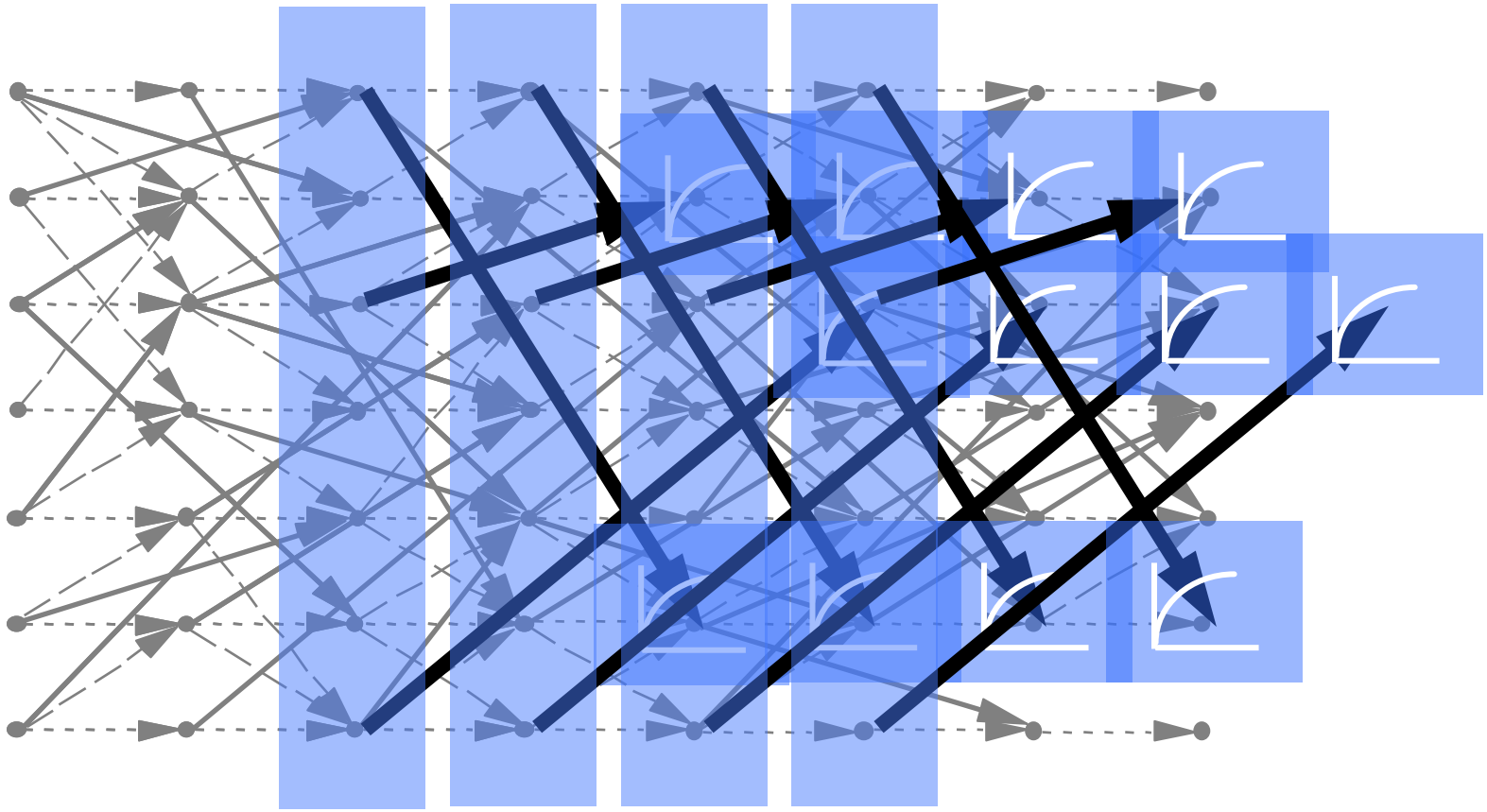
A projection algorithm (SPAR)



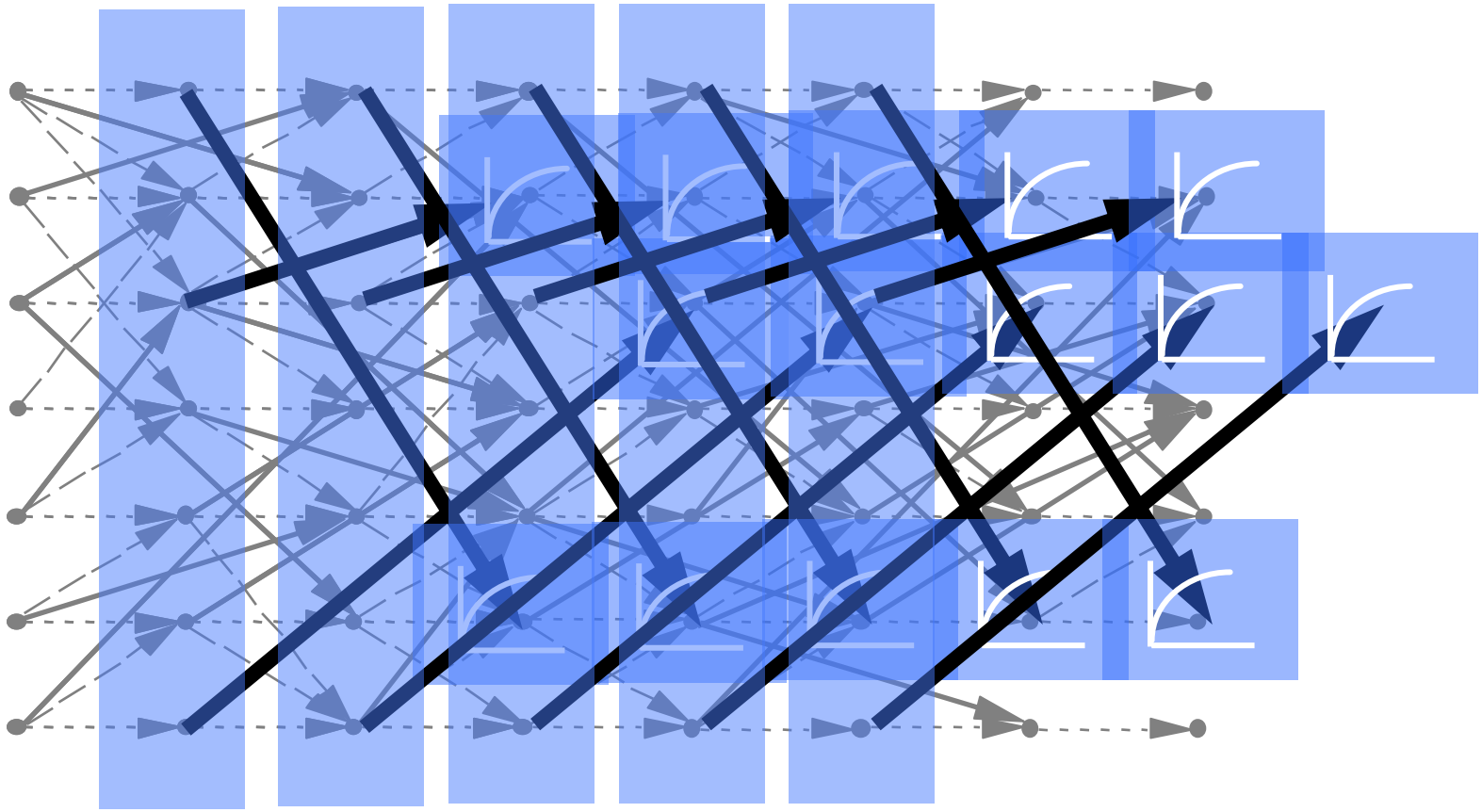
Blood management



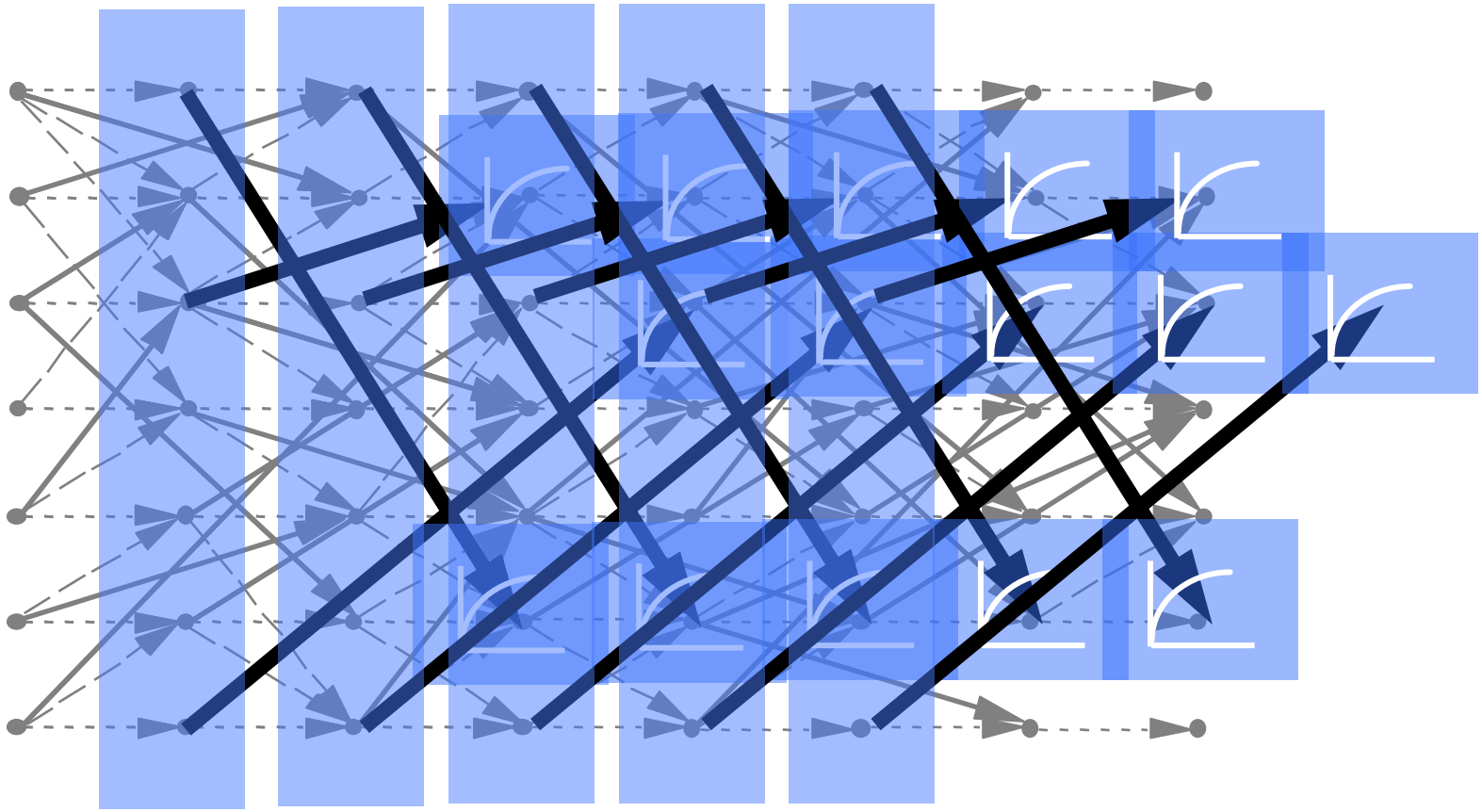
Blood management



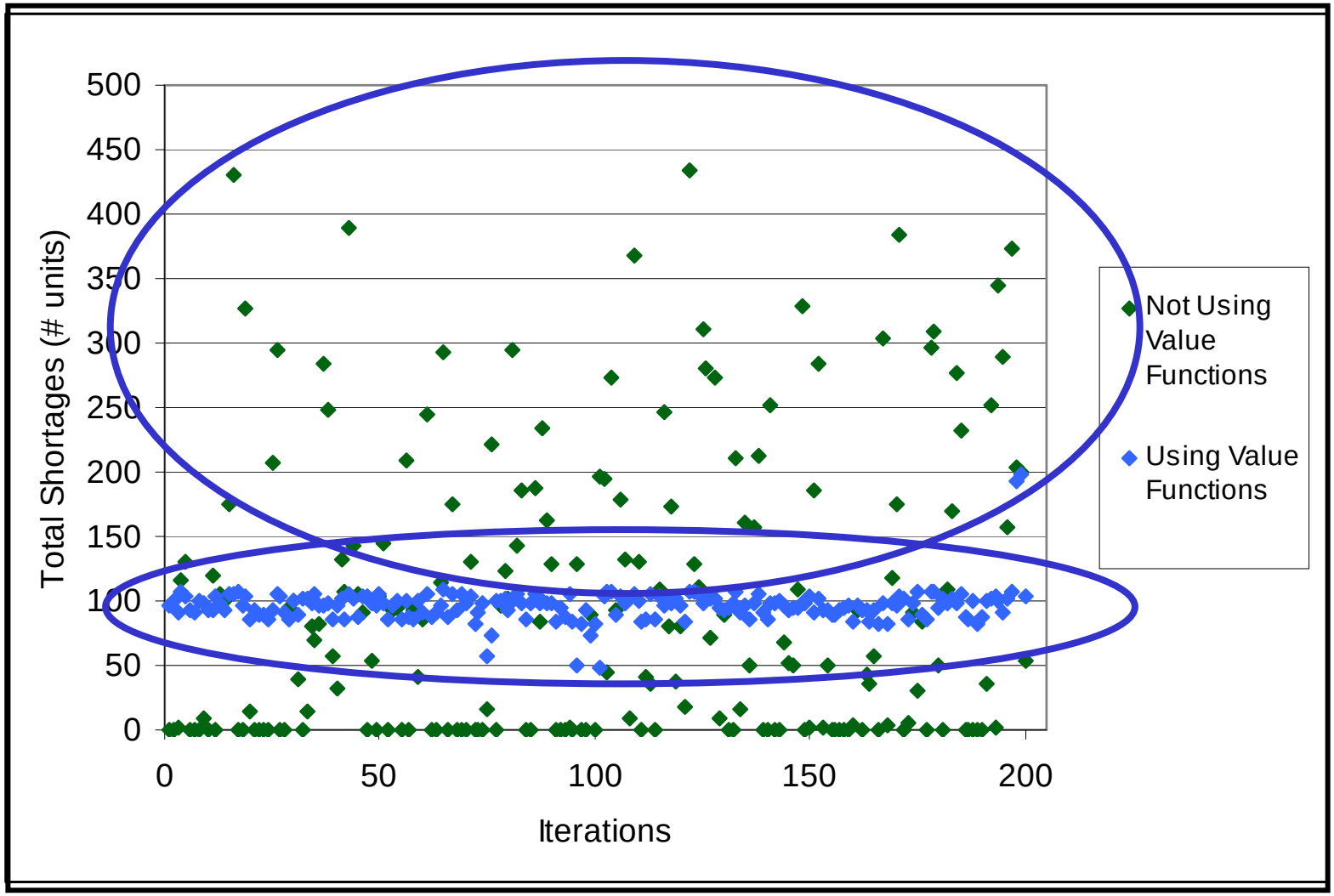
Blood management



Blood management



Blood management



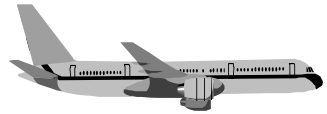
Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

Multiattribute resources

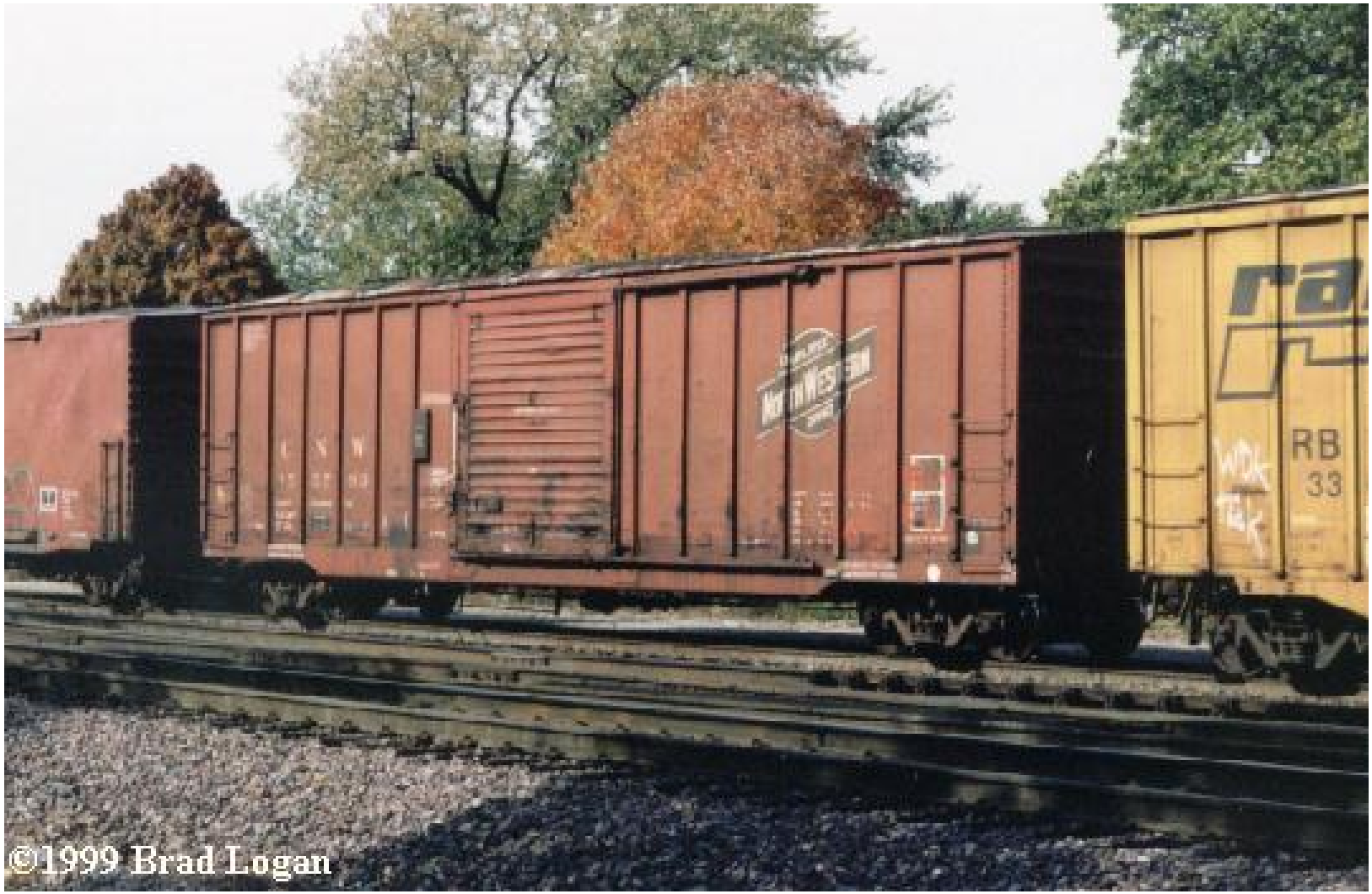
- Assets can have a number of attributes:



$$a = \begin{bmatrix} \text{Location} \\ \text{Equipment type} \end{bmatrix} \begin{bmatrix} \text{Location} \\ \text{ETA} \\ \text{Equipment type} \\ \text{Train priority} \\ \text{Pool} \\ \text{Due for maint} \\ \text{Home shop} \end{bmatrix} \begin{bmatrix} \text{Location} \\ \text{ETA} \\ \text{Bus. segment} \\ \text{Single/team} \\ \text{Domicile} \\ \text{Drive hours} \\ \text{Duty hours} \\ \text{8 day history} \\ \text{Days from home} \end{bmatrix} \begin{bmatrix} \text{Location} \\ \text{ETA} \\ \text{A/C type} \\ \text{Fuel level} \\ \text{Home shop} \\ \text{Crew} \\ \text{Eqpt1} \\ \text{M} \\ \text{Eqpt100} \end{bmatrix}$$

Boxcars



Multiattribute resources

- The evolution of attributes:

$$a = \begin{bmatrix} \text{Time} \\ \text{Location} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Boxcar type} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Boxcar type} \\ \text{Time to dest.} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Boxcar type} \\ \text{Time to dest.} \\ \text{Repair status} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Boxcar type} \\ \text{Time to dest.} \\ \text{Repair status} \\ \text{Shipper pool} \end{bmatrix}$$

$$|\mathbf{A}| \approx \quad \mathbf{4{,}000} \qquad \mathbf{40{,}000} \qquad \mathbf{1{,}680{,}000} \qquad \mathbf{5{,}040{,}000} \qquad \mathbf{50{,}400{,}000}$$

The states of our system

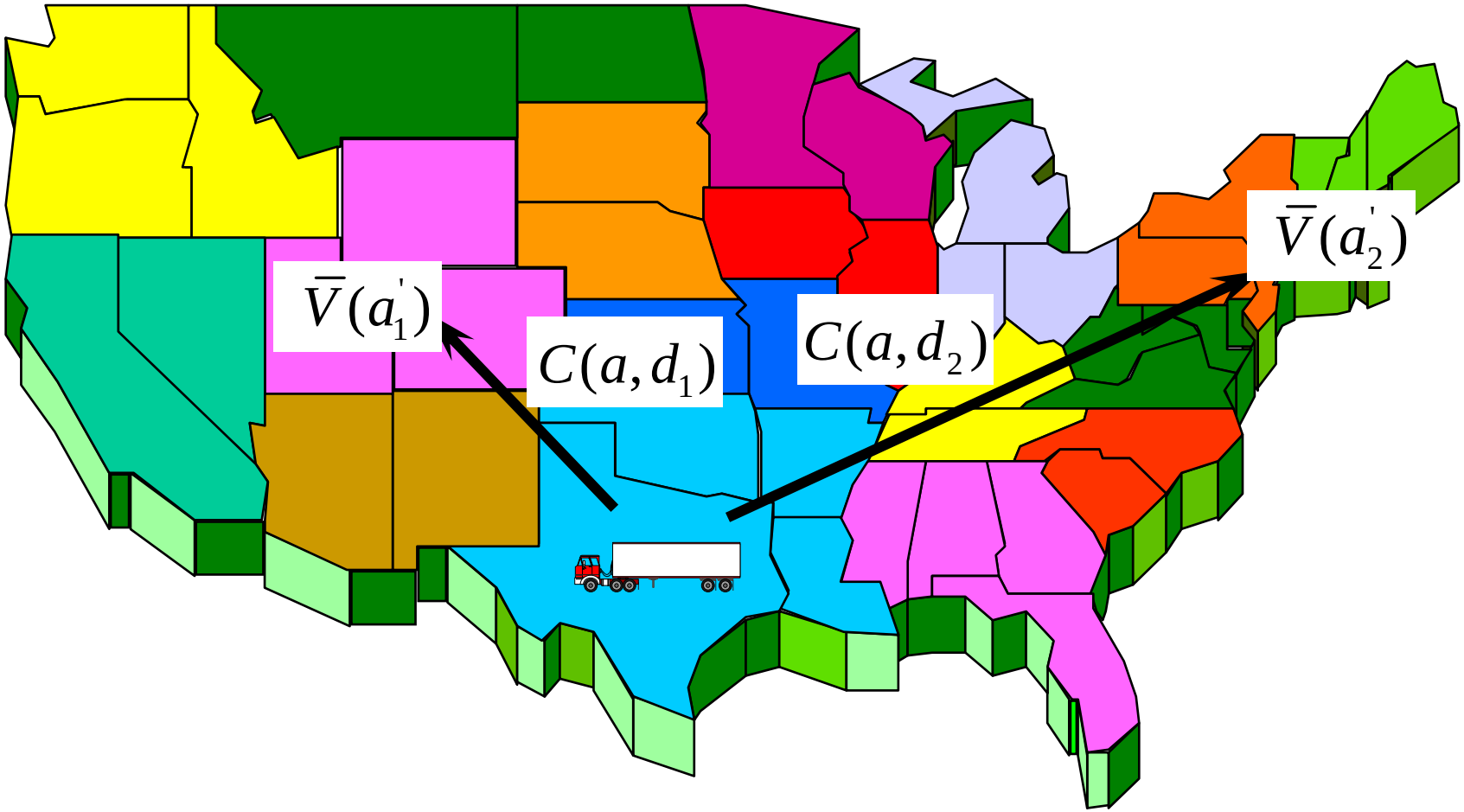
- Multiple, complex entities

$$R_t = \begin{bmatrix} R_{ta_1} \\ R_{ta_2} \\ R_{ta_3} \\ \text{M} \\ \text{M} \\ R_{ta_n} \end{bmatrix}$$

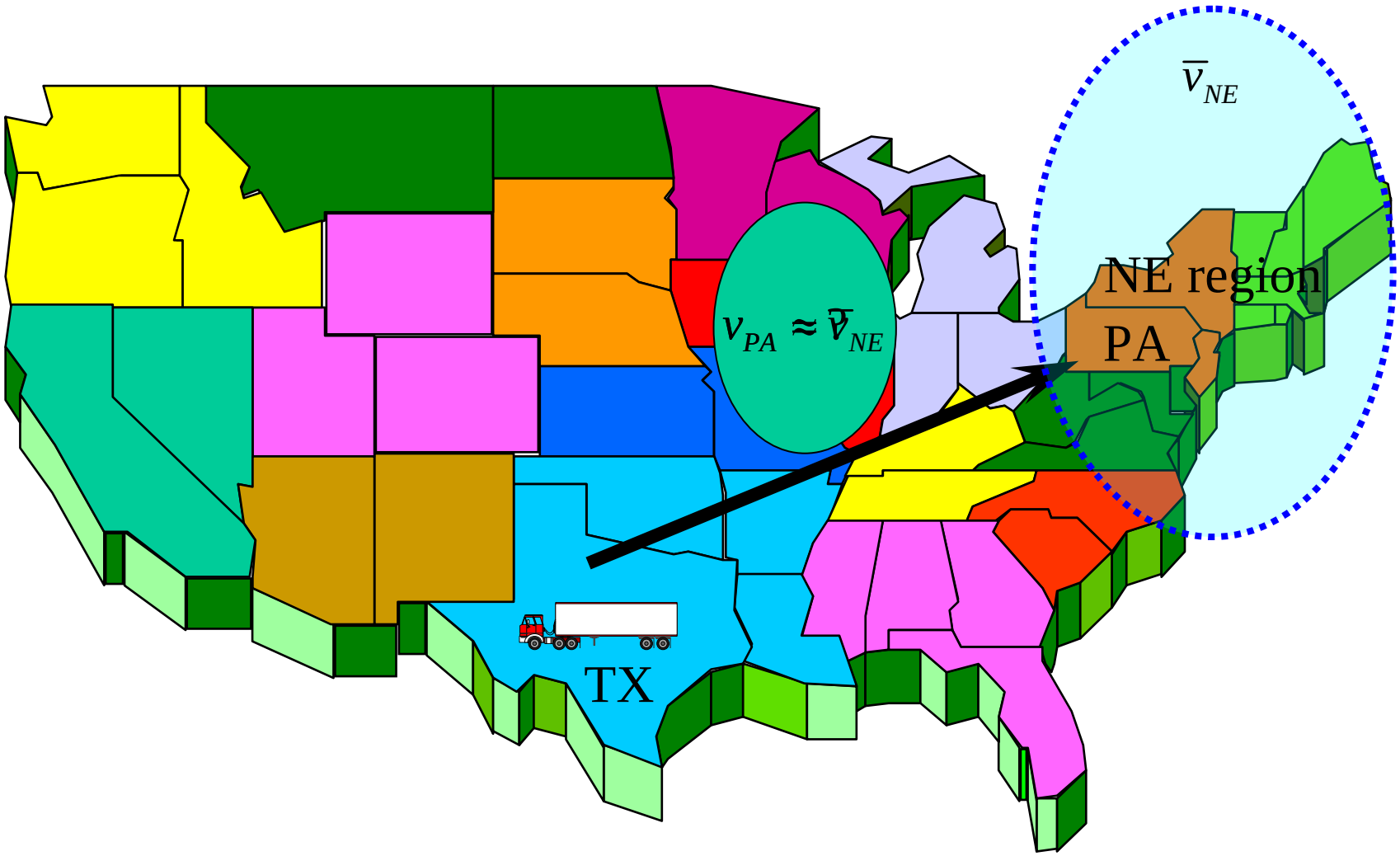
The number of *dimensions* of our state variable is equal to the size of the *state space* for a single entity problem.

The curse of curses.

Aggregation for table lookup



Aggregation for table lookup



Aggregation for table lookup

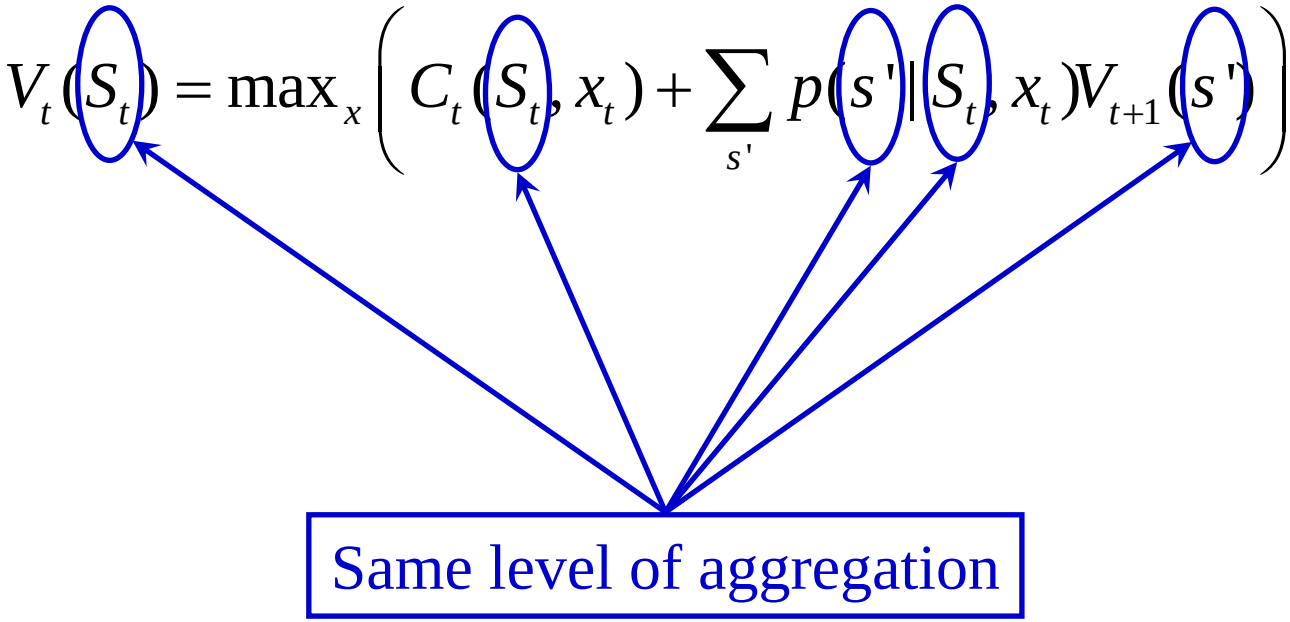
- Different levels of aggregation

$$a = \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Fleet} \\ \text{Domicile} \\ \text{Days from home} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Fleet} \\ \text{Domicile} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \\ \text{Fleet} \end{bmatrix} \begin{bmatrix} \text{Time} \\ \text{Location} \end{bmatrix}$$

$$|\mathbf{A}| \approx \quad \mathbf{12{,}000{,}000} \quad \mathbf{600{,}000} \quad \mathbf{6{,}000} \quad \mathbf{2{,}000}$$

Aggregation

- Aggregation:
  - » Exact methods
    - We have to use the same level of aggregation throughout (in particular, the transition matrix and value function).



Aggregation

- Approximate DP
  - » We only need to discretize the value function. We can capture the full state variable in the transition function:
    - Decision function:

$$x_t = \arg\max_x \left(C_t(S_t, x_t) + \bar{V}_t\ (S_t^x) \right)$$

Value function using aggregated state

    - Transition functions

$$S_t = S^{M,W}(S_{t-1}^x, W_t(\omega))$$
$$S_t^x = S^{M,x}(S_t, x_t)$$

No aggregation

Aggregation for table lookup

- Updating the value of a driver:

$$\$2050 = (1-0.10)\times \$2000 + (0.10)\ \$2500$$

$$\bar{v}^n\left(\begin{bmatrix}\text{Location}\\ \text{Fleet}\\ \text{Domicile}\end{bmatrix}\right) = (1-\alpha)\bar{v}^{n-1}\left(\begin{bmatrix}\text{Location}\\ \text{Fleet}\\ \text{Domicile}\end{bmatrix}\right) + \alpha\hat{v}\left(\begin{bmatrix}\text{Location}\\ \text{Fleet}\\ \text{Domicile}\\ \text{DOThrs}\\ \text{DaysFromHome}\end{bmatrix}\right)$$

Value function Approximation may have fewer attributes than driver.

Drivers may have very detailed attributes

Aggregation for table lookup

- Estimating value functions
  - » Most disaggregate level

$$\bar{v}^n\left(\begin{bmatrix} Location \\ Fleet \\ Domicile \end{bmatrix}\right) = (1-\alpha)\bar{v}^{n-1}\left(\begin{bmatrix} Location \\ Fleet \\ Domicile \end{bmatrix}\right) + \alpha\hat{v}\left(\begin{bmatrix} Location \\ Fleet \\ Domicile \\ DOThrs \\ DaysFromHome \end{bmatrix}\right)$$

Aggregation for table lookup

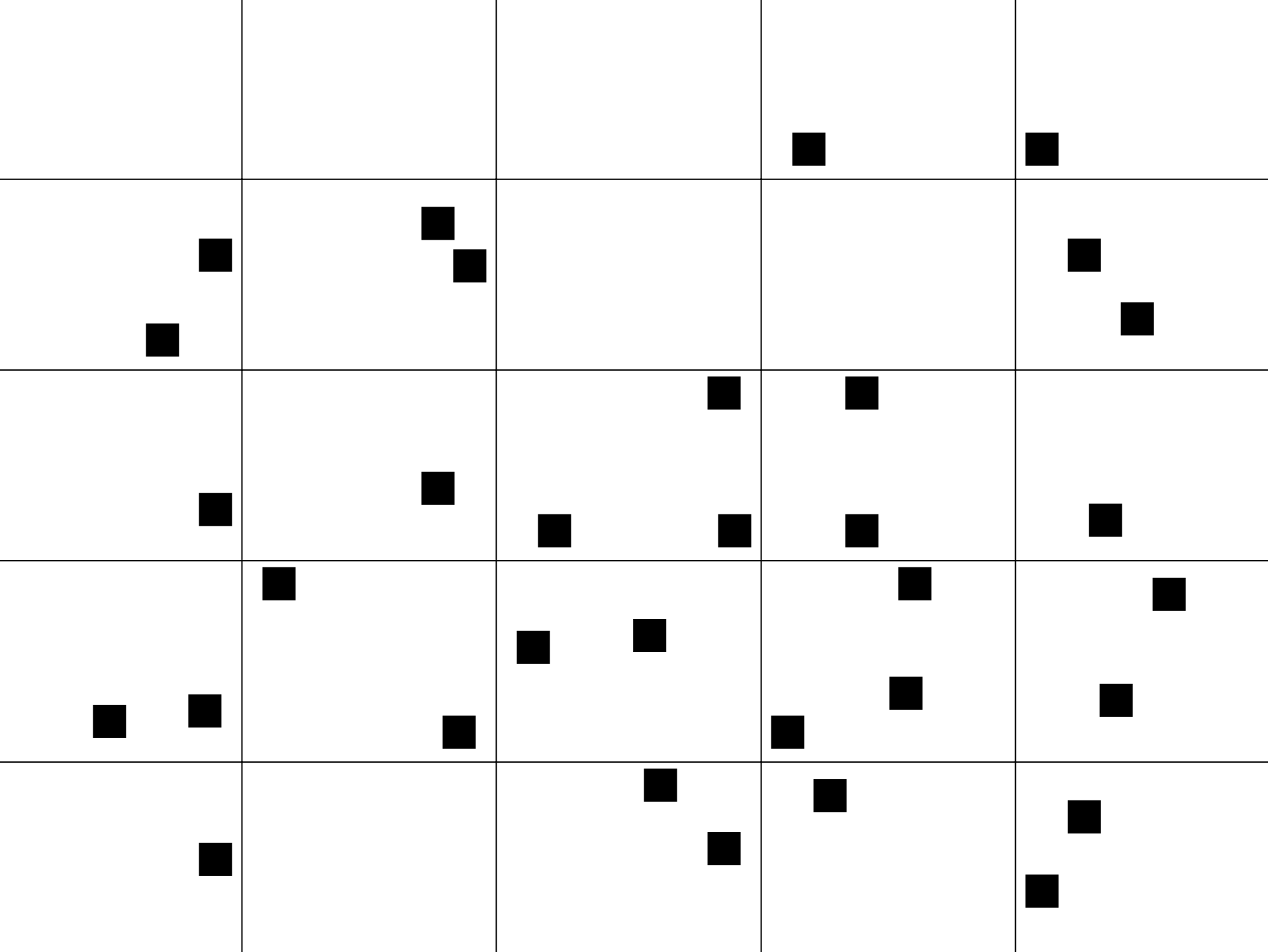
- Estimating value functions
  - » Middle level of aggregation

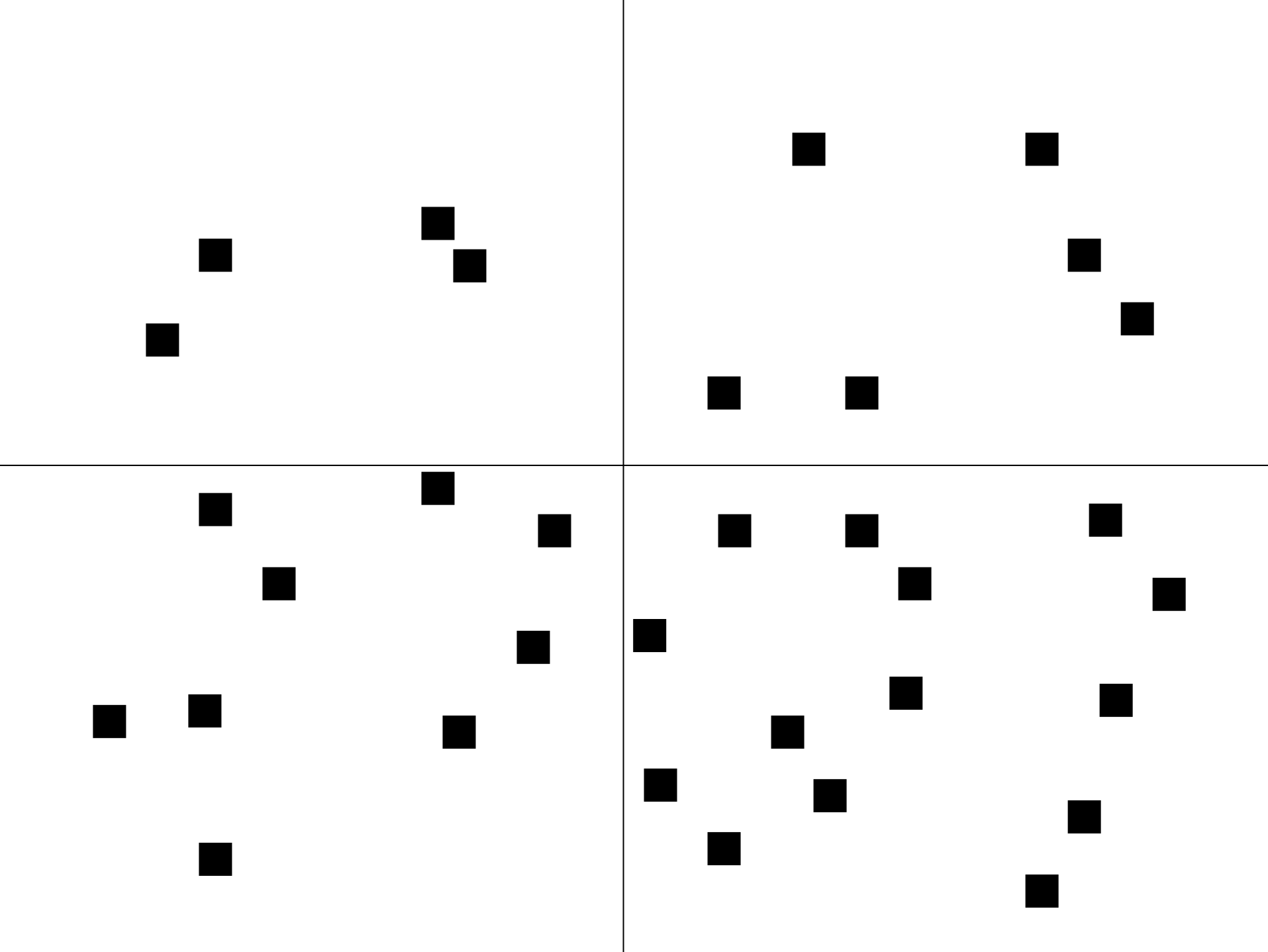
$$\bar{v}^n\left(\begin{bmatrix} Location \\ Fleet \end{bmatrix}\right) = (1-\alpha)\bar{v}^{n-1}\left(\begin{bmatrix} Location \\ Fleet \end{bmatrix}\right) + \alpha\hat{v}\left(\begin{bmatrix} Location \\ Fleet \\ Domicile \\ DOThrs \\ DaysFromHome \end{bmatrix}\right)$$

Aggregation for table lookup

- Estimating value functions
  - » Most aggregate level

$$\bar{v}^n(\left[Location\right]) = (1-\alpha)\bar{v}^{n-1}(\left[Location\right]) + \alpha\hat{v}(\begin{bmatrix} Location \\ Fleet \\ Domicile \\ DOThrs \\ DaysFromHome \end{bmatrix})$$





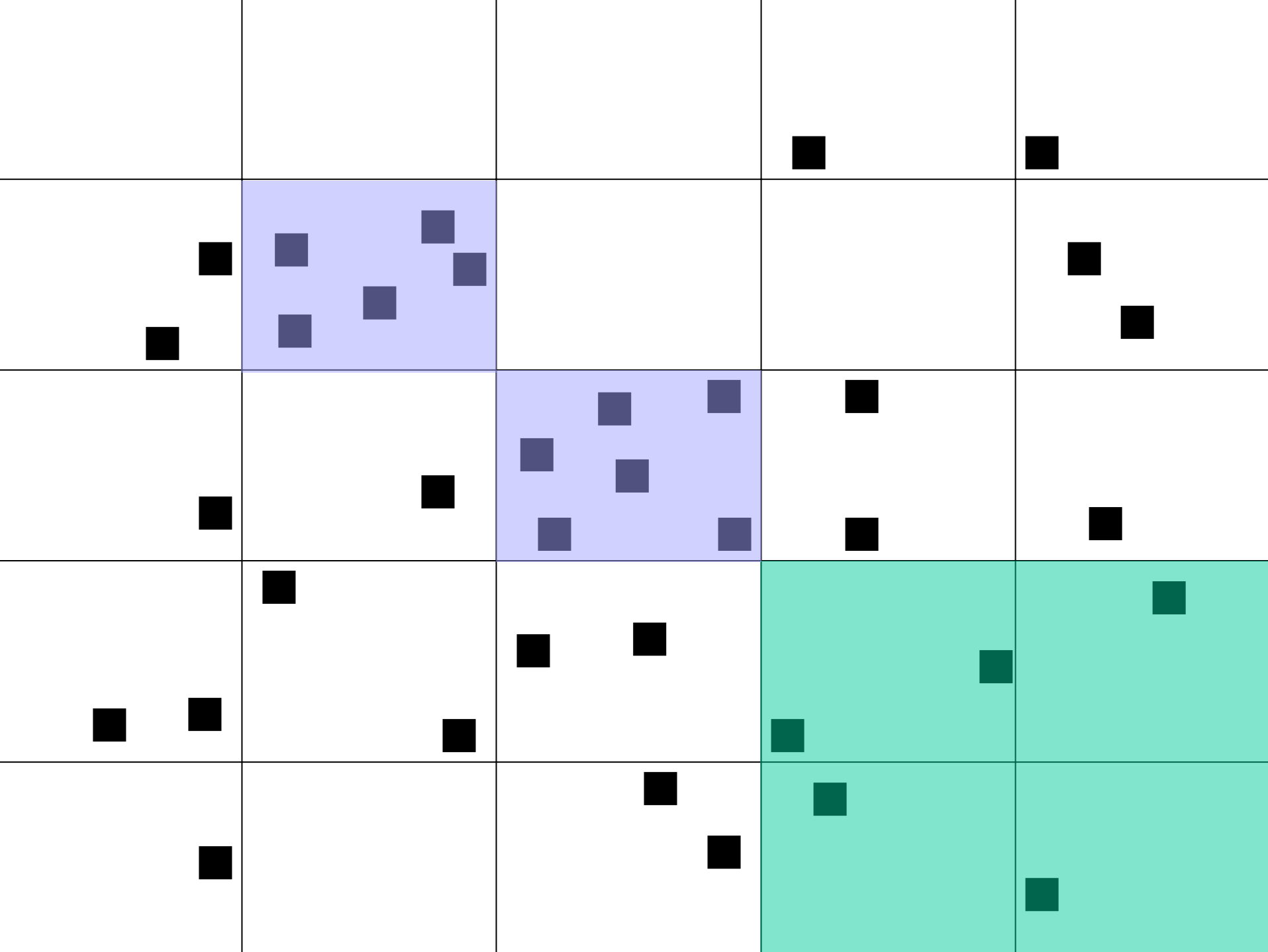
Aggregation for table lookup

- Using different levels of aggregation:
  - » Pick the (single) level of aggregation that produces the best overall results.
  - » Pick the level of aggregation that produces the lowest variance for each state.
  - » Use a weighted sum of estimates at each level of aggregation (weight depends only on the level of aggregation):

$$\bar{v}_a^n = \sum_g w^{(g,n)} \bar{v}_a^{(g,n)}$$

  - » Use a weighted sum, but where the weights depend on the state (attribute):

$$\bar{v}_a^n = \sum_g w_a^{(g,n)} \bar{v}_a^{(g,n)}$$



Aggregation for table lookup

- State-dependent weighted aggregation:
  - » There may be hundreds of thousands of weights, so these have to be easy to compute.

$$\bar{v}_a = \sum_g w_a^{(g)} \bar{v}_a^{(g)} \qquad \sum_g w_a^{(g)} = 1$$

where

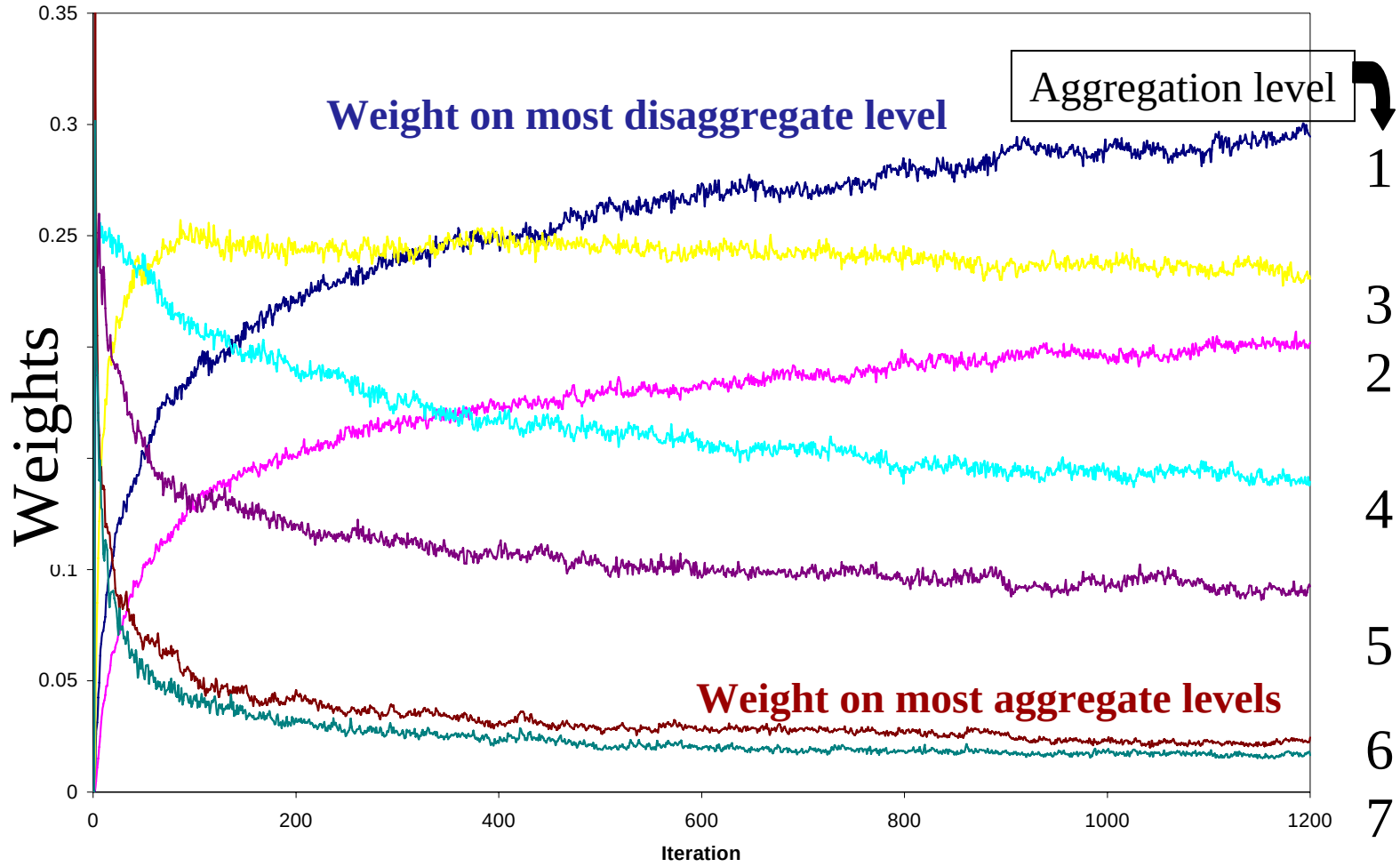
$$w_a^{(g)} \propto \left(Var\left(\bar{v}_a^{(g)}\right) + \left(\beta_a^{(g)}\right)^2 \right)^{-1}$$

Estimate of variance

Estimate of bias

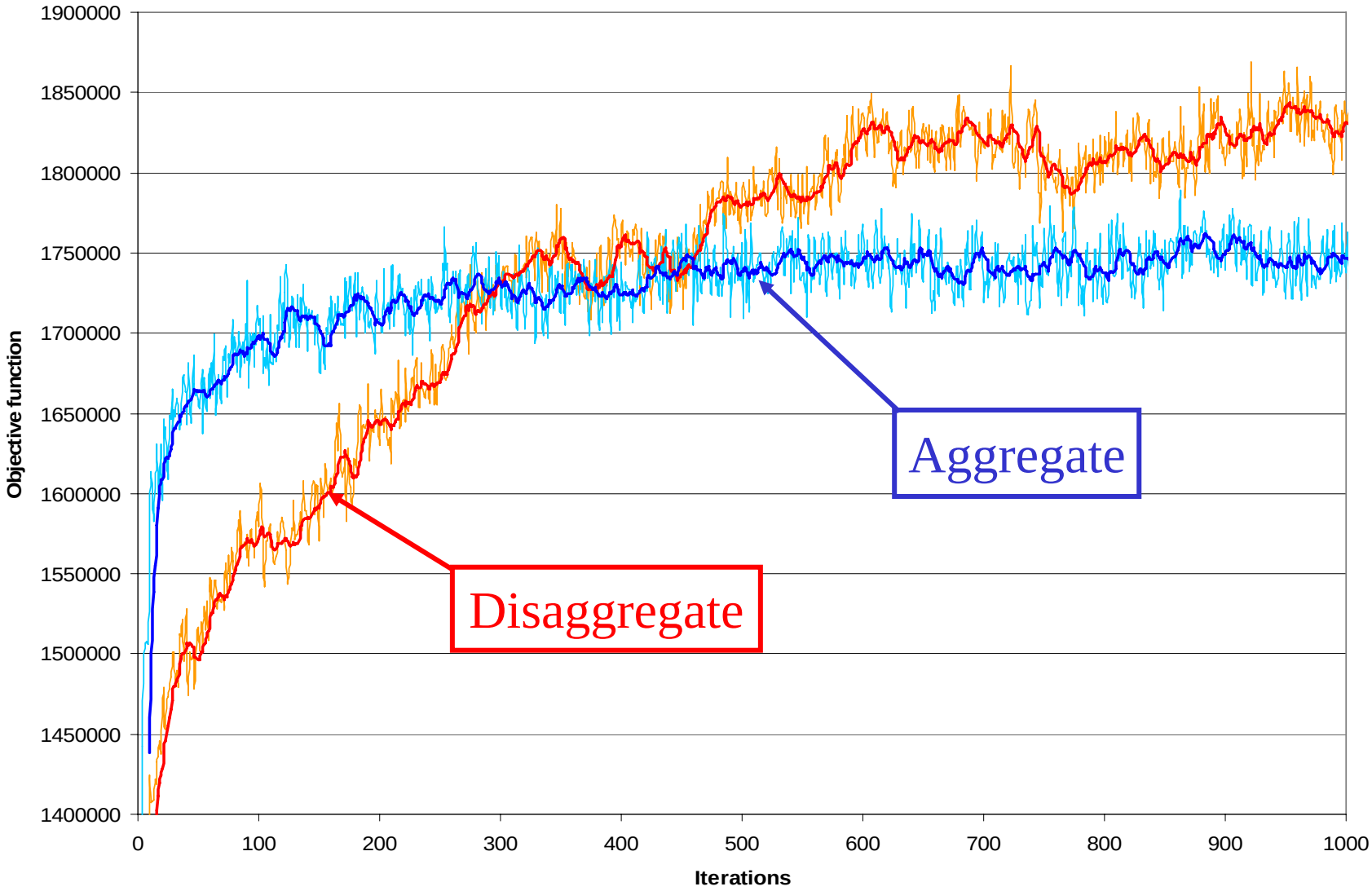
Both can be computed using simple recursive formulas.

Aggregation for table lookup

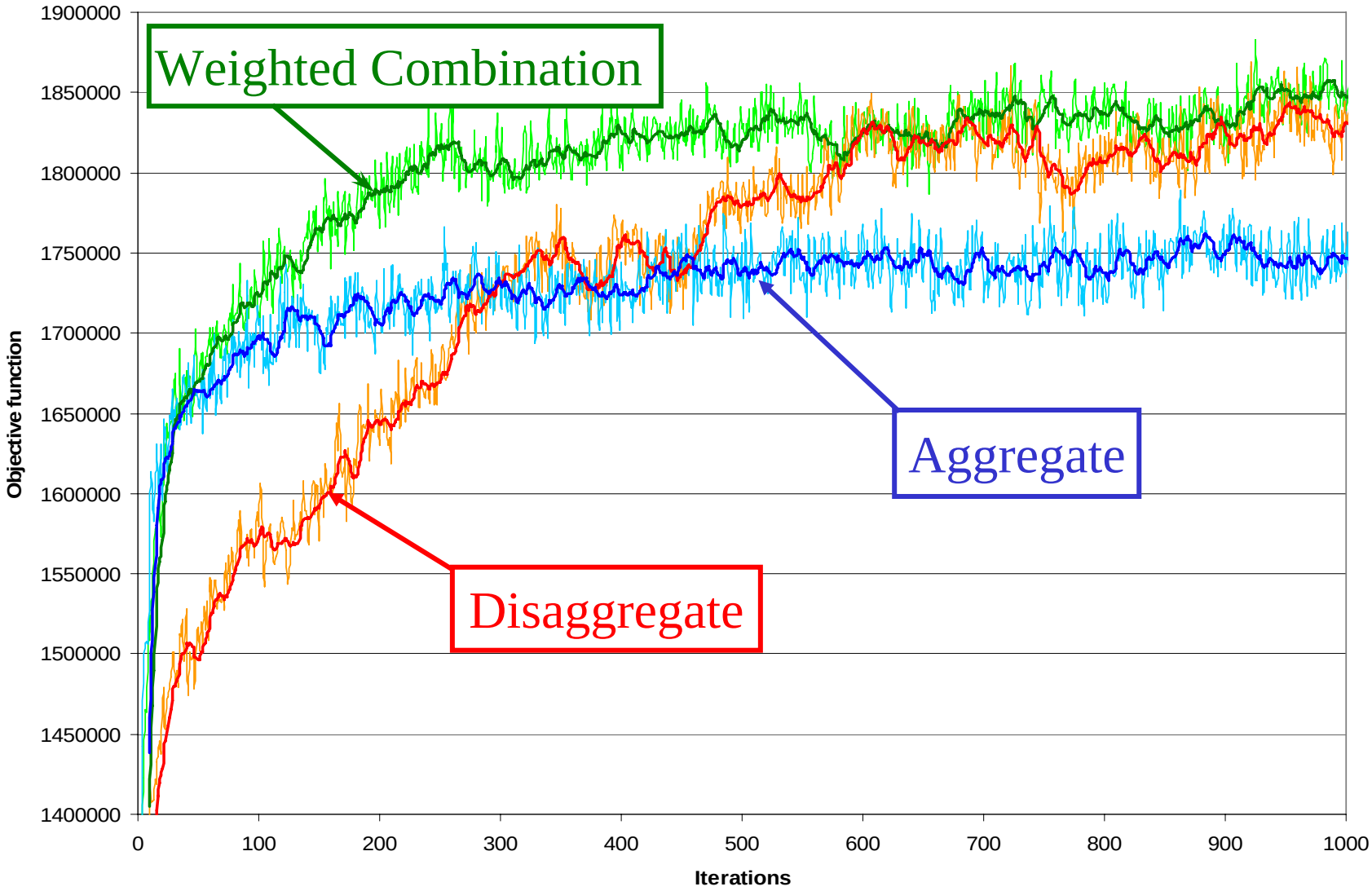


Iterations

Aggregation for table lookup



Aggregation for table lookup

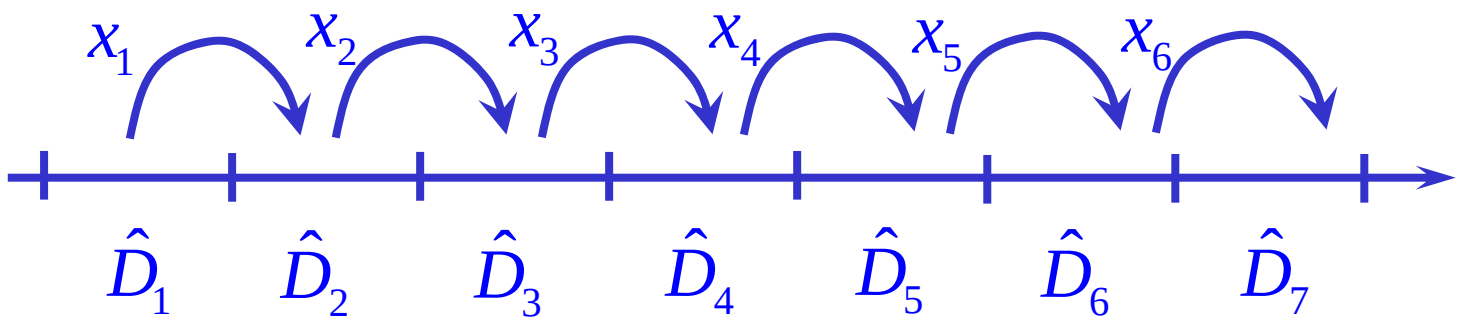


Outline

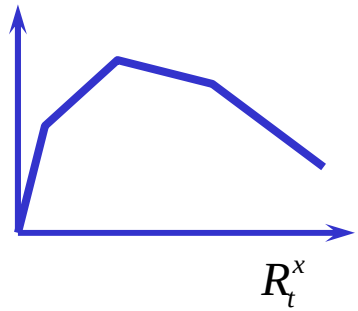
- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

Basis functions

- Aggregation works well when we have a state (attribute) space with very little structure.
- But what if we have some structure? Consider our inventory problem:



$$x_t^n = \arg\max_x \left(p x_t^D - c x_t^O + \bar{V}_t\ (R_t^x) \right)$$



Basis functions

- Approximating the value function:
  - » We have to exploit the structure of the value function (e.g. concavity).
  - » We might approximate the value function using a simple polynomial

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

  - » .. or a complicated one:

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

  - » Sometimes, they get really messy:

$$\bar{V}_t(R\mid\theta)=\theta^{(0)}+\sum_s\sum_{t'=t}^{t+2}\theta^{(1)}_{t,st'}R_{t,st'}+\sum_w\sum_{t'=t}^{t+3}\theta^{(1)}_{t,wt'}R_{t,wt'}$$ 1,2,3 4,5,6,7

$$+\sum_w\sum_{t'}\theta^{(2)}_{t,wt'}R^2_{t,wt'}+\sum_s\sum_{t'}\theta^{(2)}_{t,st'}R^2_{t,st'}$$ 8-14

$$+\sum_s\theta^{(3)}_{ts}\left(R_{t,st}-\frac{1}{S}\sum_{s'}R_{t,s't}\right)^2$$ 15

$$+\sum_s\theta^{(4)}_{ts}\left(\left(\sum_{t'=t}^{t+1}R_{t,st'}\right)-\frac{1}{2S}\sum_{s'}\sum_{t'=t}^{t+1}R_{t,s't'}\right)^2$$ 16

$$+\sum_s\theta^{(5)}_{ts}\left(\left(\sum_{t'=t}^{t+2}R_{t,st'}\right)-\frac{1}{3S}\sum_{s'}\sum_{t'=t}^{t+2}R_{t,s't'}\right)^2$$ 17

$$+\sum_w\sum_s\theta^{(ws,1)}_{t,ws}R_{t,wt}R_{t,st}$$ 18

$$+\sum_w\sum_s\theta^{(ws,2)}_{t,ws}R_{t,wt}\left(\sum_{t'=t}^{t+2}R_{t,st'}\right)$$ 19

$$+\sum_w\sum_s\theta^{(ws,3)}_{t,ws}\left(\sum_{t'=t}^{t+3}R_{t,wt'}\right)\left(\sum_{t'=t}^{t+2}R_{t,st'}\right)$$ 20

$$+\sum_w\sum_s\theta^{(ws,4)}_{t,ws}\left(\sum_{t'=t}^{t+2}R_{t,wt'}\right)\left(\sum_{t'=t}^{t+2}R_{t,st'}\right)$$ 21

$$+\sum_w\sum_s\theta^{(ws,5)}_{t,sw}R_{t,s,t+2}R_{t,wt}R_{t,w,t+2}$$ 22

Basis functions

- We can write a model of the observed value of being in a state as:

$$\hat{v} = \theta_0 + \theta_1 R_t + \theta_2 R_t^2 + \theta_3 \ln\left(R_t\right) + \theta_4 \sin(R_t) + \varepsilon$$

- This is often written as a generic regression model:

$$Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \theta_4 X_4$$

- The ADP community refers to the independent variables as *basis functions*:

$$Y = \theta_0 \varphi_0(R) + \theta_1 \varphi_1(R) + \theta_2 \varphi_2(R) + \theta_3 \varphi_3(R) + \theta_4 \varphi_4(R)$$

$$= \sum_{f \in \mathrm{F}} \theta_f \varphi_f(R)$$

$\varphi_f(R)$ are also known as *features*.

Basis functions

- Methods for estimating θ
  - » Generate observations $\hat{v}^1, \hat{v}^2, \ldots, \hat{v}^N$, and use traditional regression methods to fit θ.
  - » Use recursive statistics - update θ^n after each iteration:

$$\theta^n = \theta^{n-1} - \alpha_{n-1}\left(\bar{V}^{n-1}(S^n \mid \theta^{n-1}) - \hat{v}^n\right)\nabla \bar{V}^{n-1}(S^n \mid \theta^{n-1})$$

$$= \theta^{n-1} - \alpha_{n-1}\underbrace{\left(\bar{V}^{n-1}(S^n \mid \theta^{n-1}) - \hat{v}^n\right)}_{\text{Error}}\underbrace{\begin{pmatrix}\varphi_1(S)\\ \varphi_2(S)\\ \\ \varphi_F(S)\end{pmatrix}}_{\text{Basis functions}}$$

Basis functions

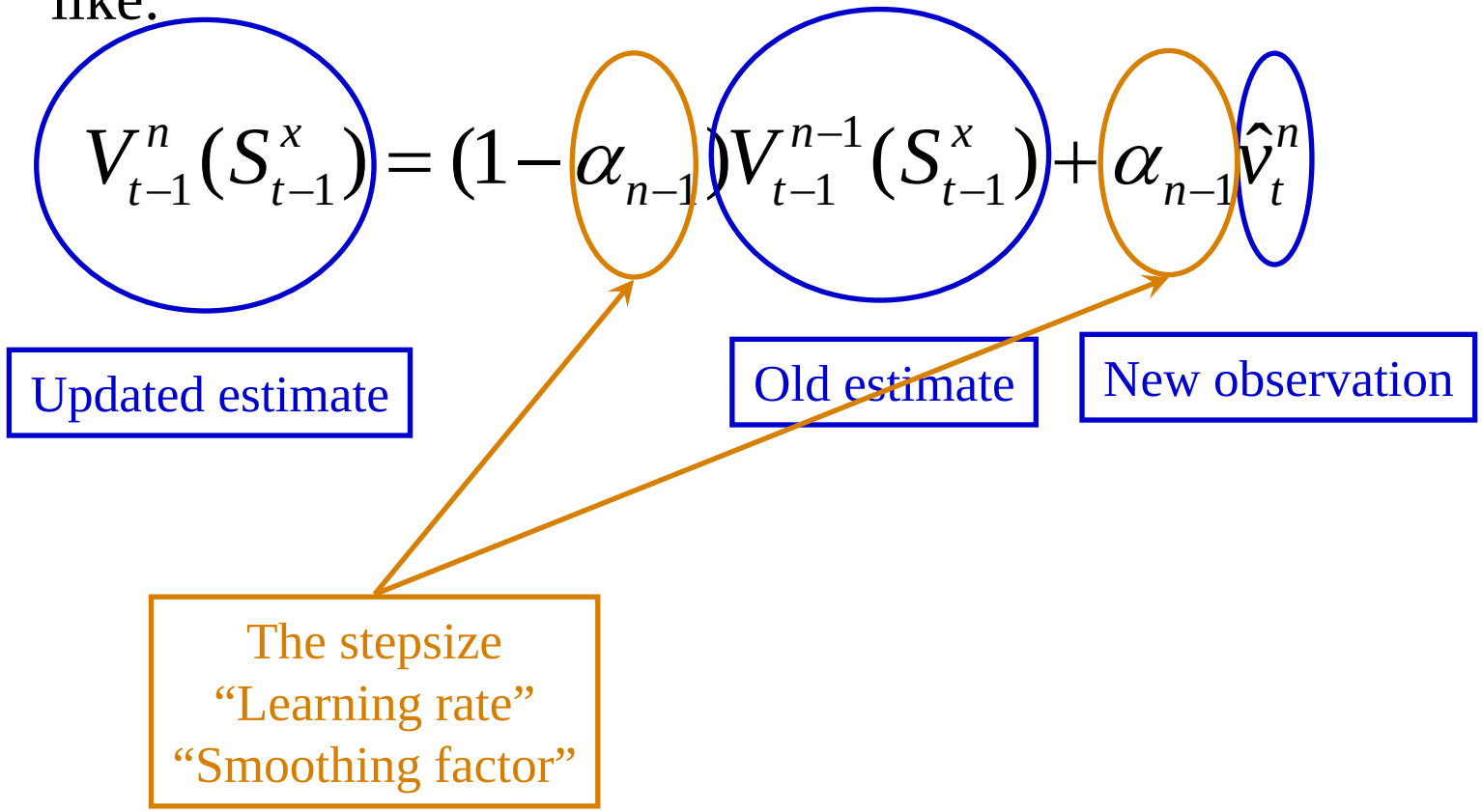
- Notes:
  - When using basis functions, we are basically drawing on the entire field of statistics.
  - Designing basis functions (independent variables) is mostly art.
  - In special cases, the resulting algorithm can produce optimal solutions.
  - Most of the time, we are hoping for "good" solutions.
  - In some cases, it can work terribly.
  - As a general rule – you have to use problem structure. Value function approximations have to capture the right structure. Blind use of polynomials will rarely be successful.

Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

Stepsizes

- Stepsizes:
  - » Fundamental to ADP is an updating equation that looks like:



Stepsizes

- Theory:
  - » Many convergence results require:

$$\sum_{n=1}^{\infty} \alpha_{n-1} = \infty$$

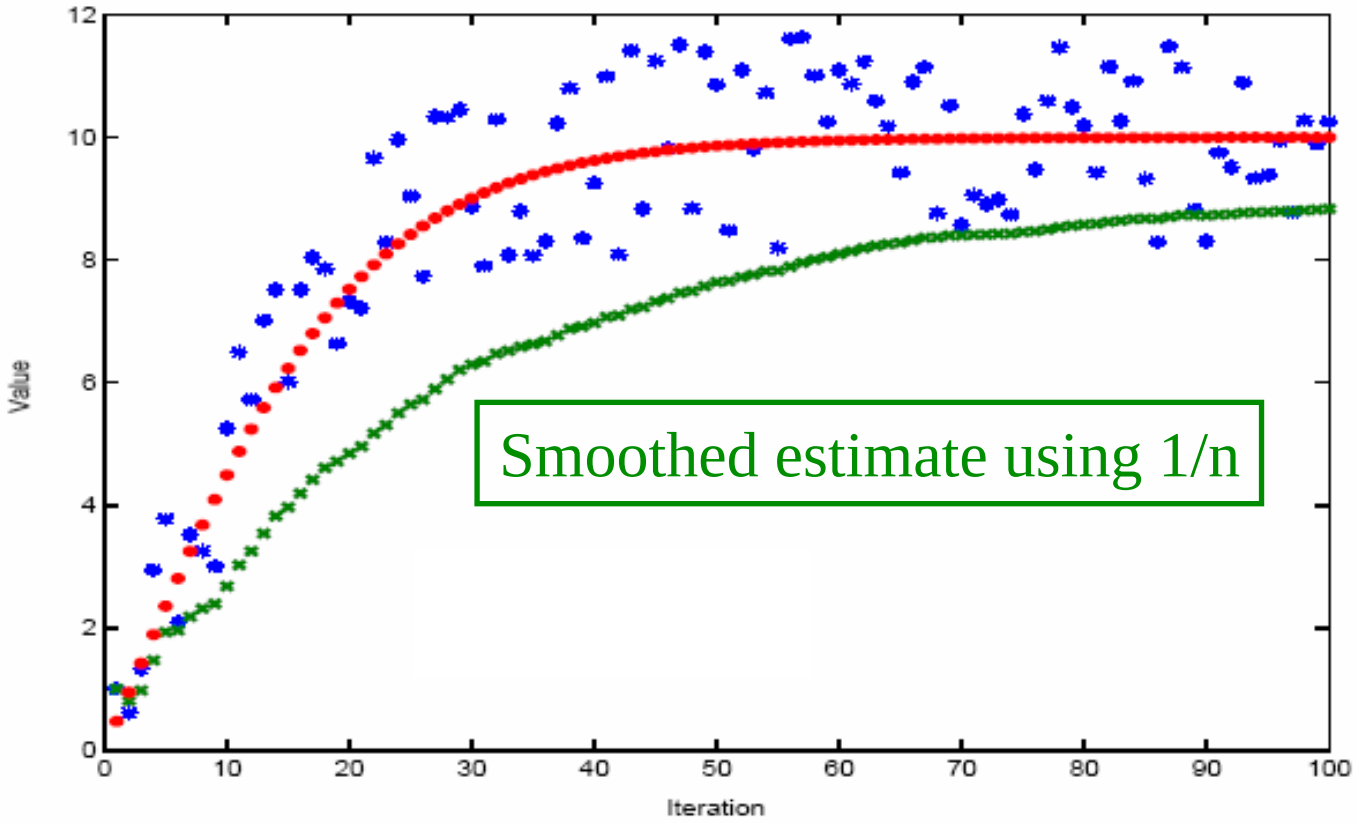
$$\sum_{n=1}^{\infty} \left(\alpha_{n-1}\right)^2 < \infty$$

  - » For example:

$$\alpha_{n-1} = \frac{1}{n}$$

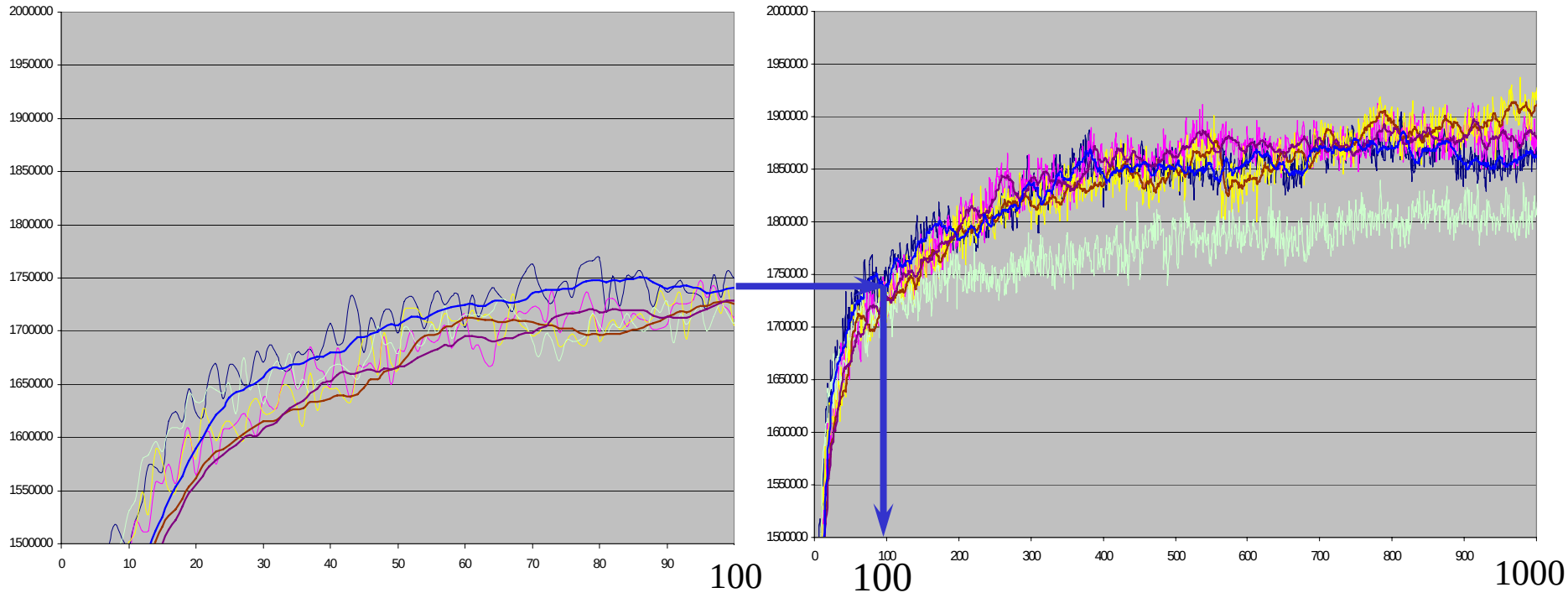
- Practice
  - » 1/n “doesn’t work”
  - » Constant stepsizes
  - » Various stepsize rules
    - Deterministic
    - Stochastic

Rate of convergence



Rate of convergence

- The challenge of stepsizes:
  - When have we converged?



We need to improve our understanding of adaptive stepsizes.

Stepsizes

- Deterministic stepsize rules:

$$\alpha_n = \frac{1}{n} \text{ or } \frac{1}{n^\beta}$$ Basic averaging

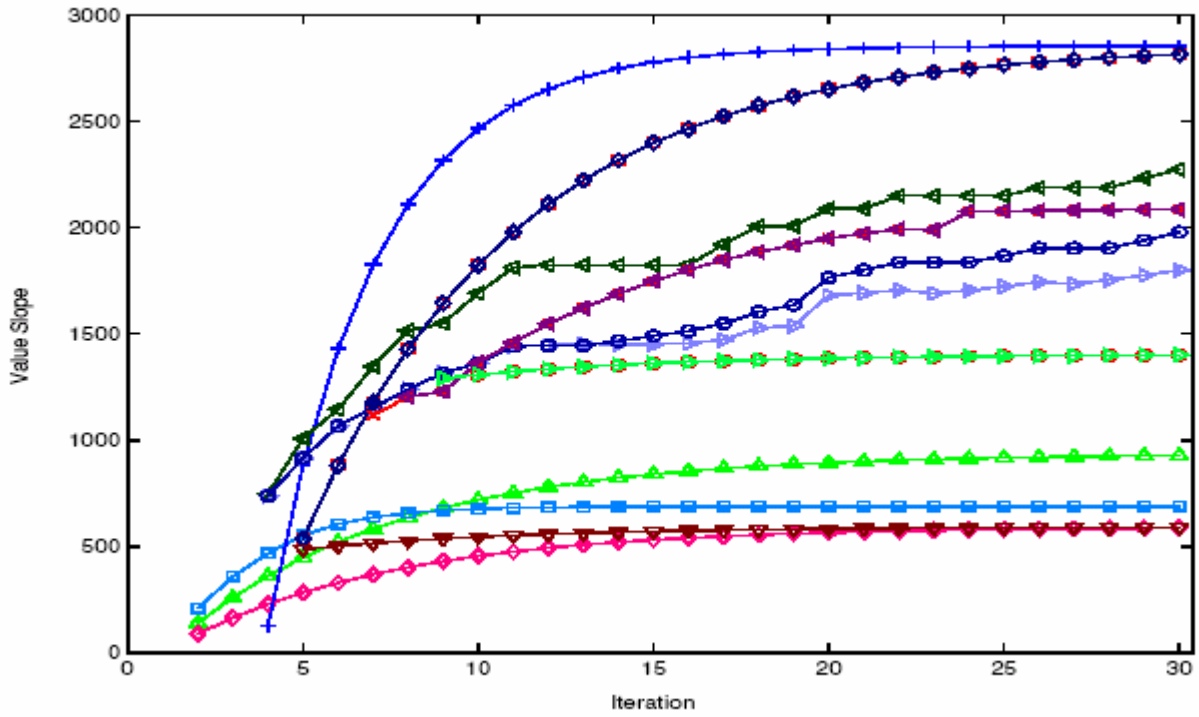
$$\alpha_n = \frac{a}{a+n-1}$$ Slows the rate of descent

$$\alpha_n = \frac{\frac{b}{n}+a}{\frac{b}{n}+a+n^\beta}$$ "Search then converge"

$$\alpha_n = \frac{\alpha_n}{1+\alpha_n-\bar{\alpha}}$$ McClain's formula

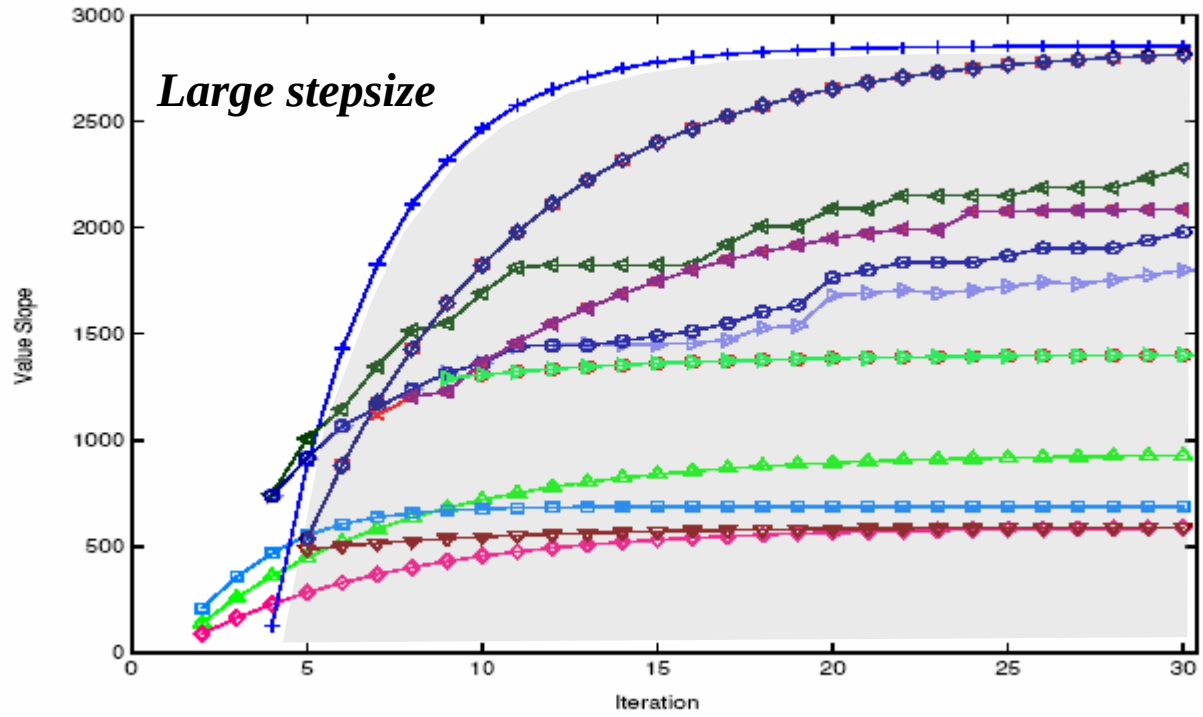
Stepsizes

- The right stepsize rule depends on the rate of change in the value function.
- This varies widely for different parameters in the same problem:



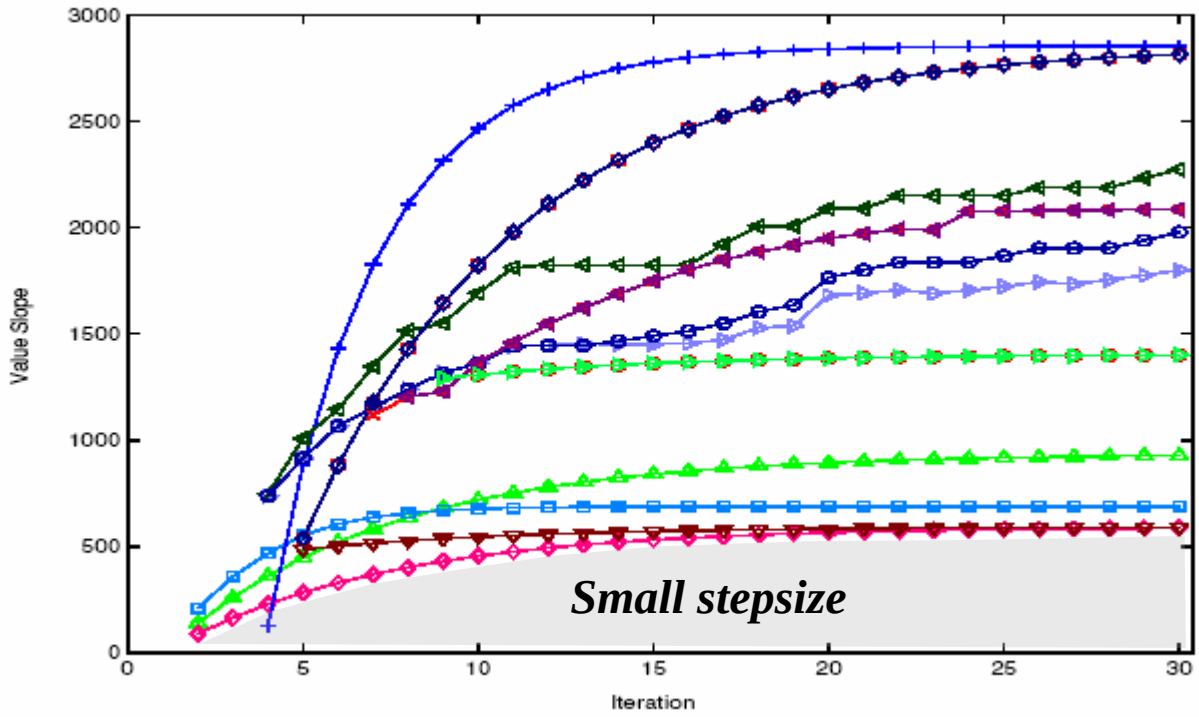
Stepsizes

- The right stepsize rule depends on the rate of change in the value function.
- This varies widely for different parameters in the same problem:



Stepsizes

- The right stepsize rule depends on the rate of change in the value function.
- This varies widely for different parameters in the same problem:



Stepsizes

- Stochastic stepsize rules:

Let: ε^n = New observation - old estimate

Kesten's rule: Tunable parameters

$$\alpha_n = \alpha^0 \frac{a}{a + K^n} \qquad K^n = K^{n-1} + 1_{\{\varepsilon^n \varepsilon^{n-1} < 0\}} \qquad a$$

Stochastic gradient adaptive stepsize (Benveniste et al.)

$$\alpha_n = \left[\alpha^{n-1} + a\varepsilon^n \ \varphi^n \right]_{\alpha^{\min}}^{\alpha^{\max}} \qquad a,\ \alpha^{\max},\ \alpha^{\min}$$

where:

$$\varphi^n = \left(1 - \alpha^{n-1}\right)\varphi^n + \varepsilon^n$$

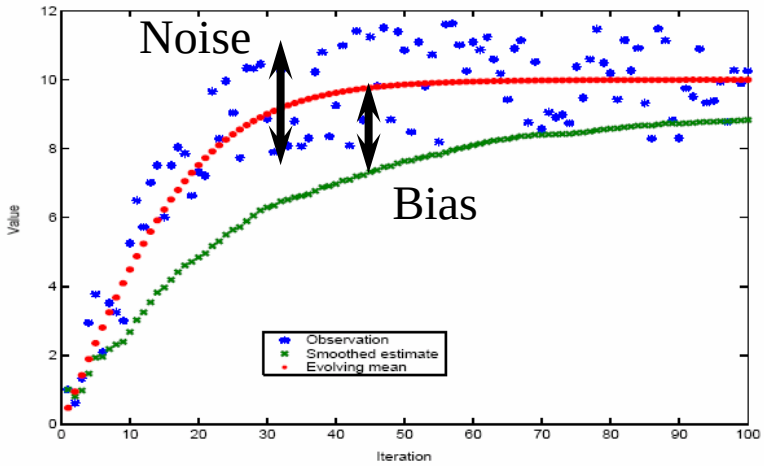
Stepsizes

- Bias-adjusted Kalman filter

Estimate of the variance

$$\alpha_n = 1 - \frac{\sigma^2}{\left(1+\lambda^{n-1}\right)\sigma^2 + \left(\beta^n\right)^2}$$

Estimate of the bias



where:

$$\lambda^n = \left(1-\alpha_n\right)^2 \lambda^{n-1} + \left(\alpha_n\right)^2$$

As σ^2 increases, stepsize decreases

As β^n increases, stepsize increases.

Stepsizes

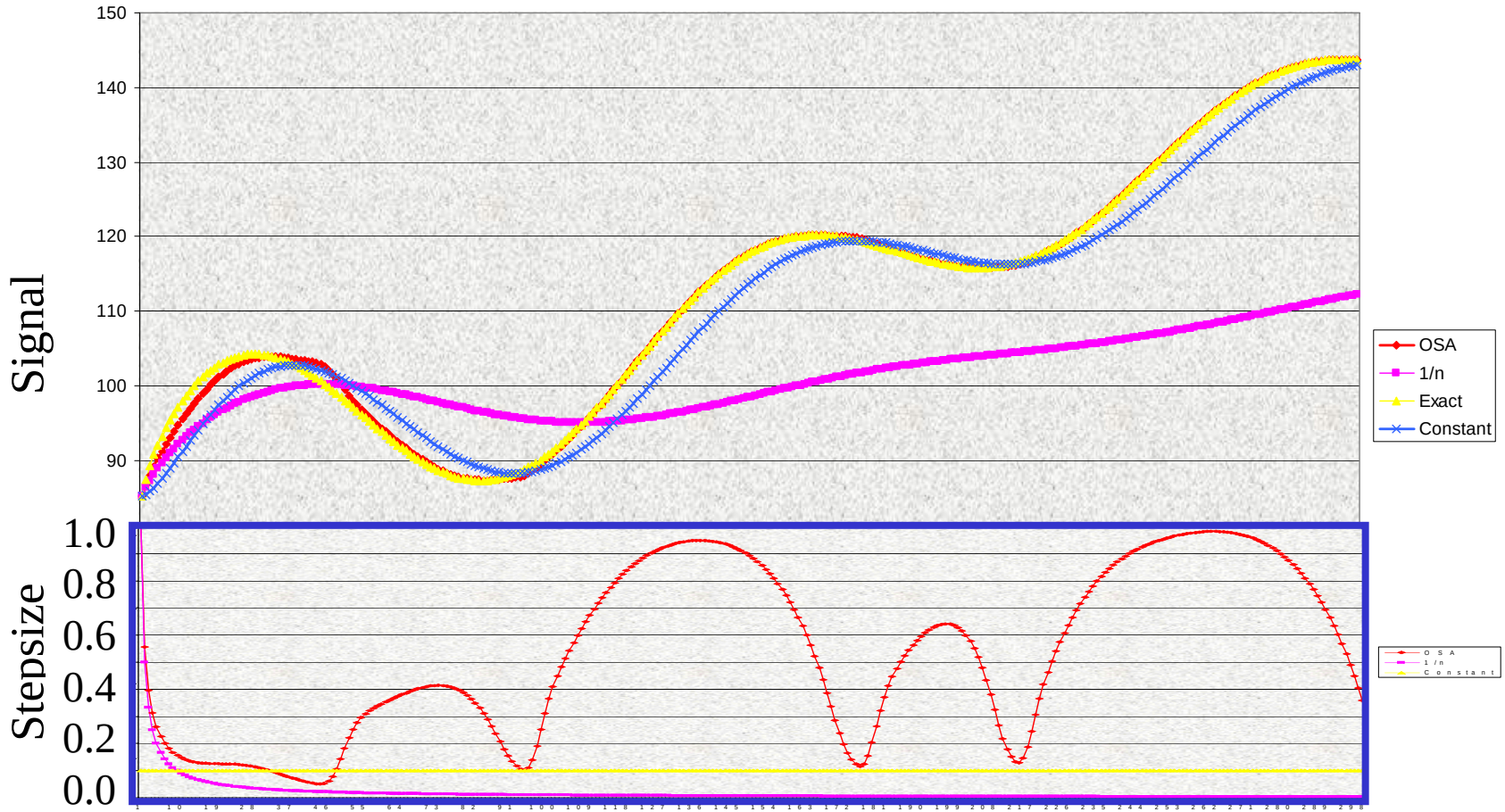
- Bias-adjusted Kalman filter
  - » Properties:

$$\alpha_n \to 1 \quad \text{as } \sigma^2 \to 0$$

$$\alpha_n \to 1/n \quad \text{as } \beta \to 0 \text{ or } \sigma^2 \to \infty$$

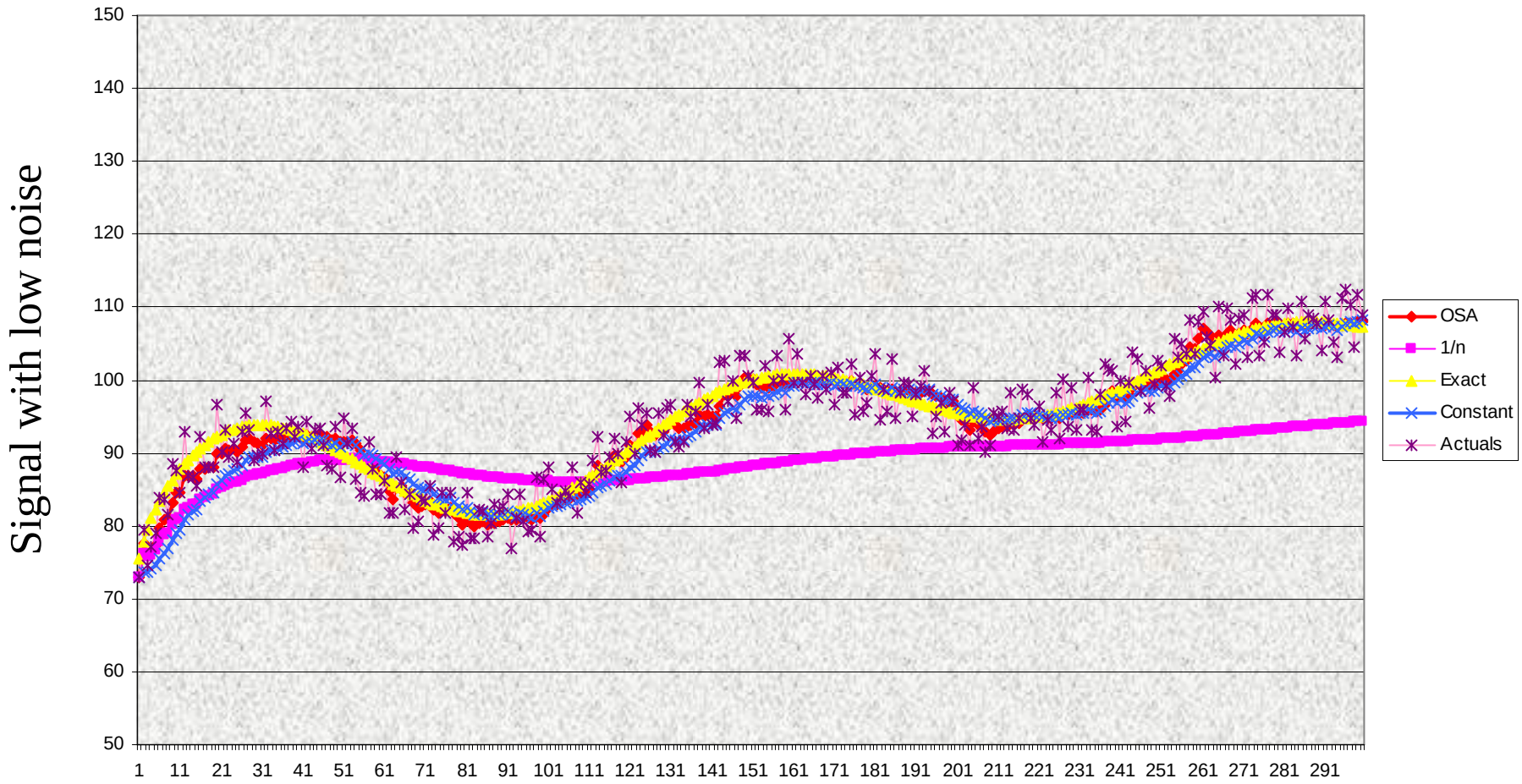
Stepsizes

- Deterministic data: predictions and stepsizes



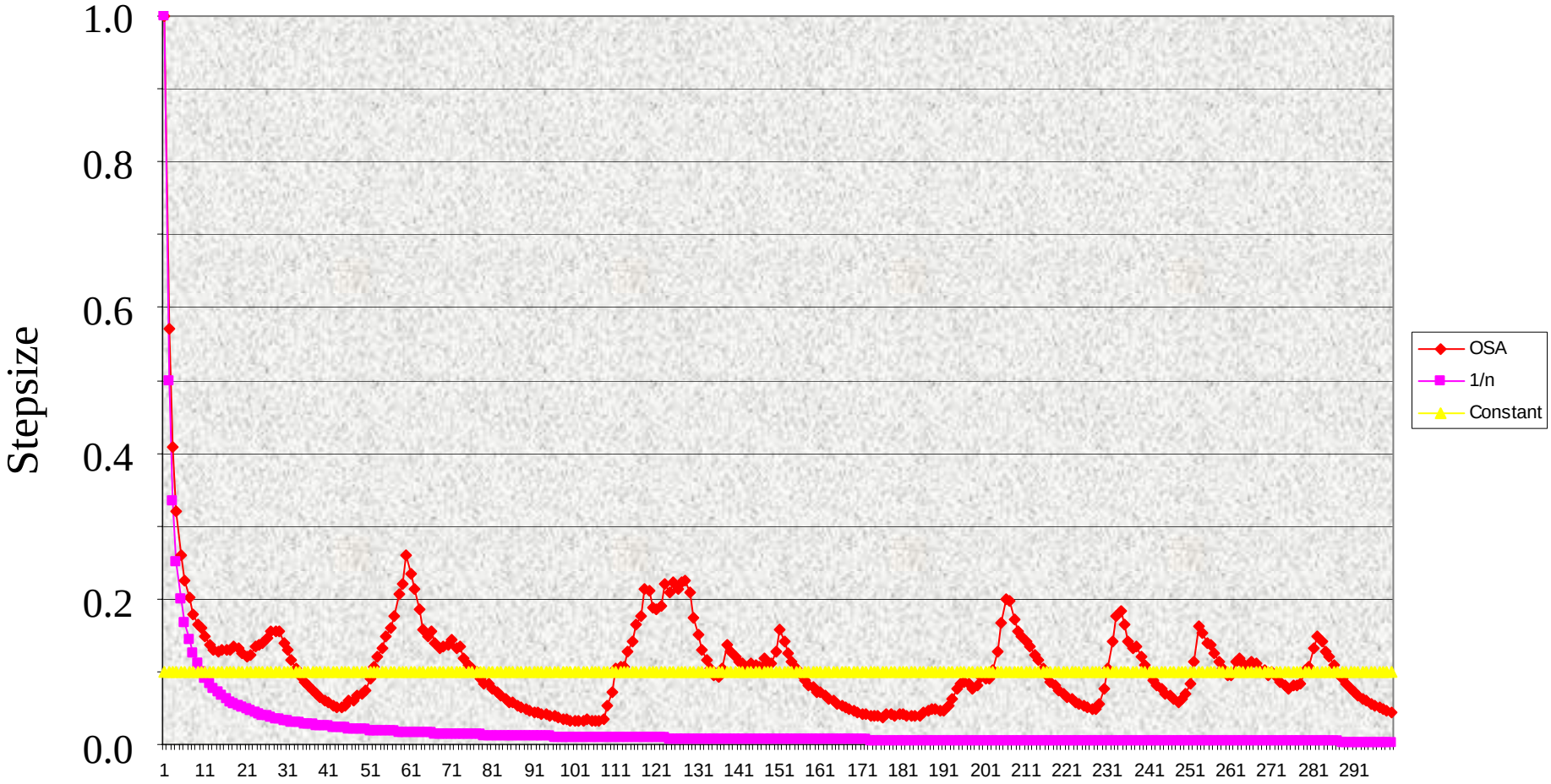
Stepsizes

- Low noise stochastic data: predictions



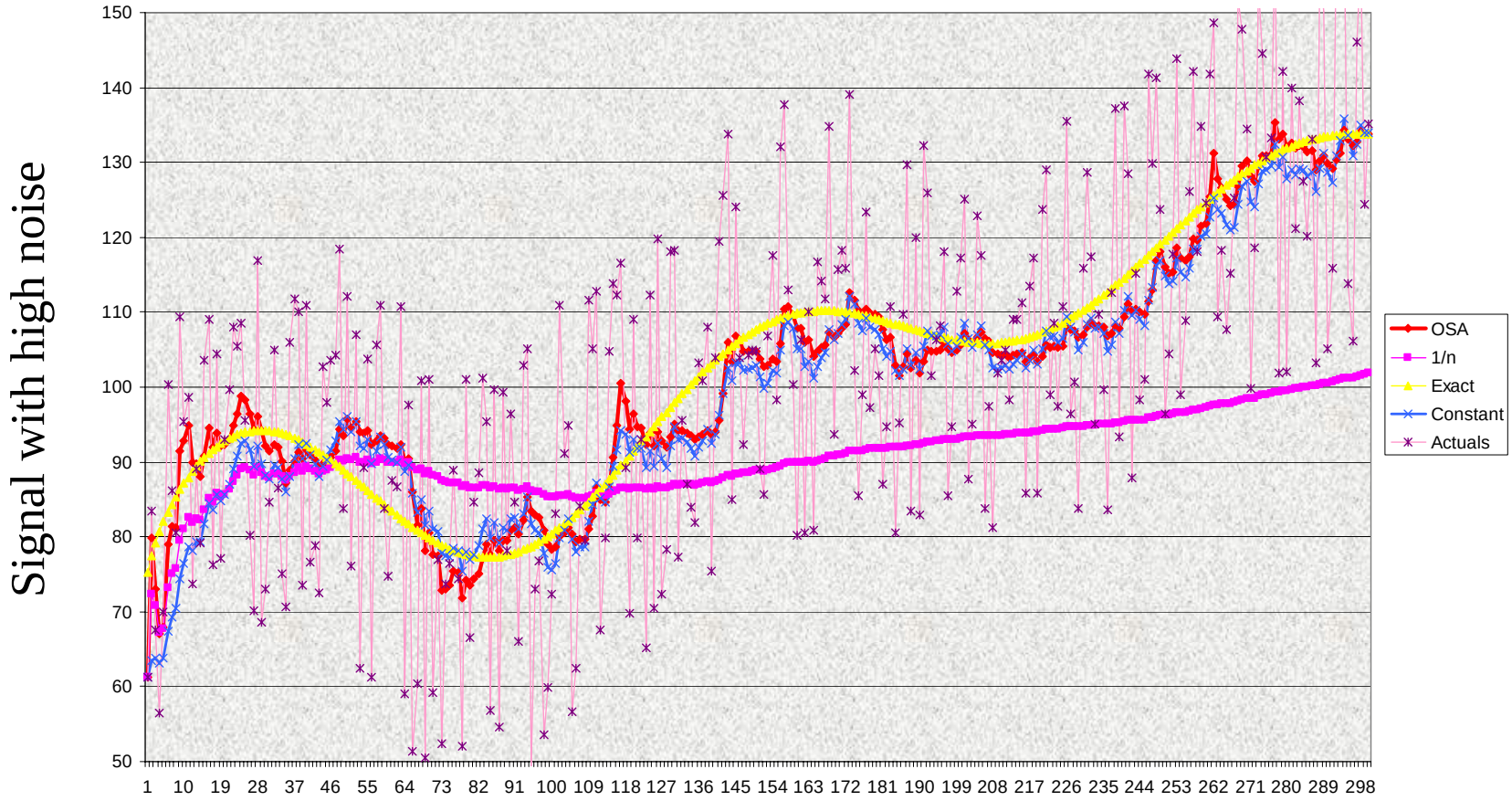
Stepsizes

- Low noise stochastic data: stepsizes



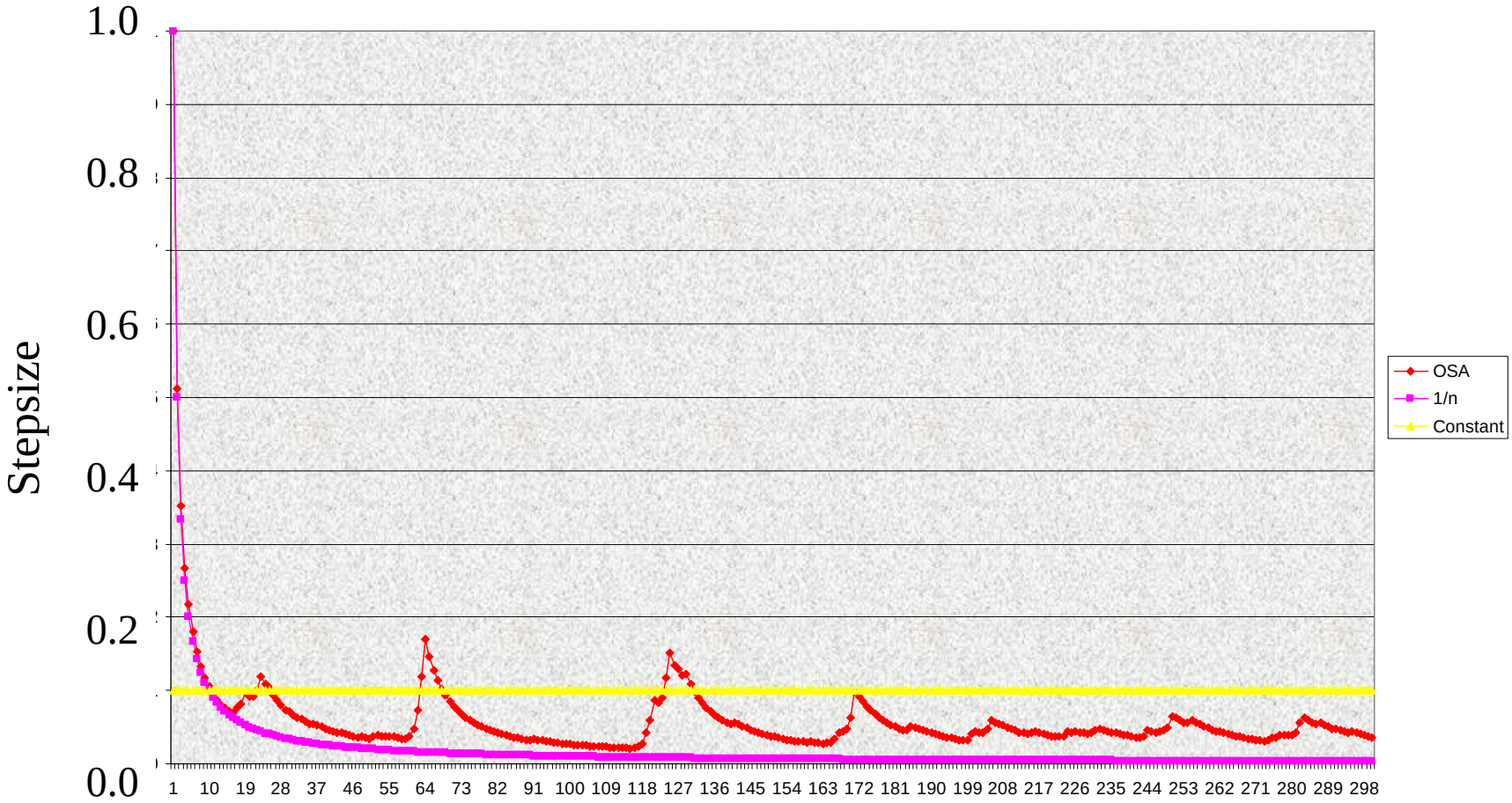
Stepsizes

- High noise stochastic data: predictions



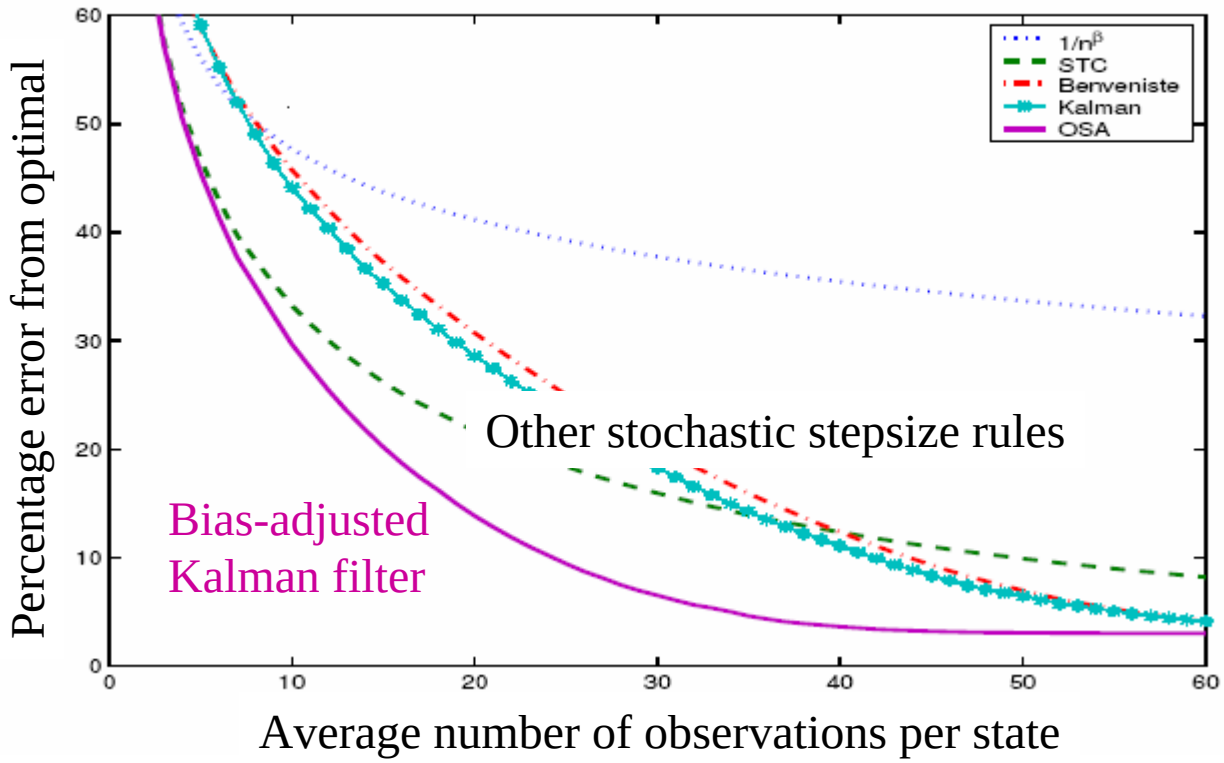
Stepsizes

- High noise stochastic data: stepsizes



Stepsizes

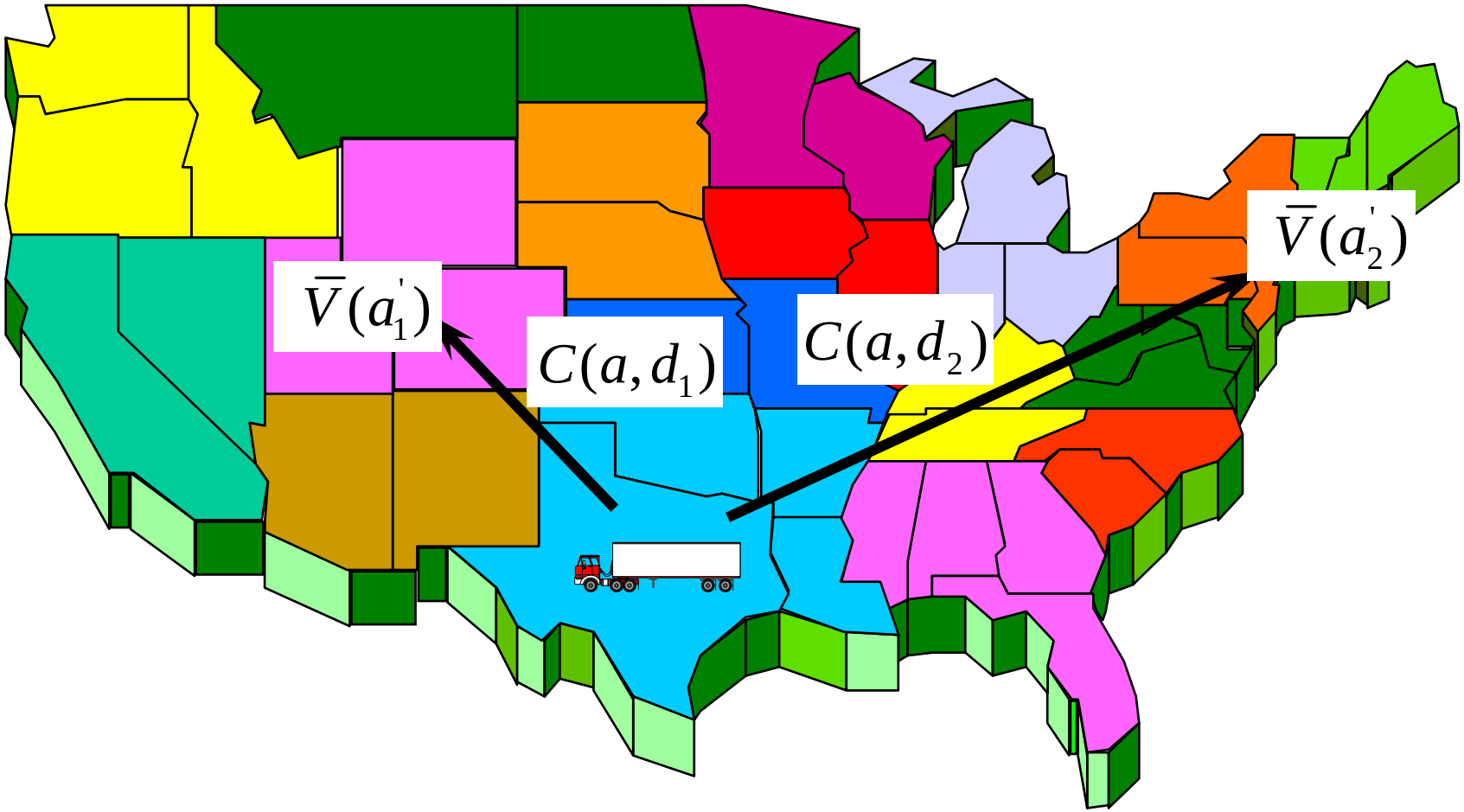
- Bias-adjusted Kalman filter learning rate vs. other



Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

Exploration vs. exploitation



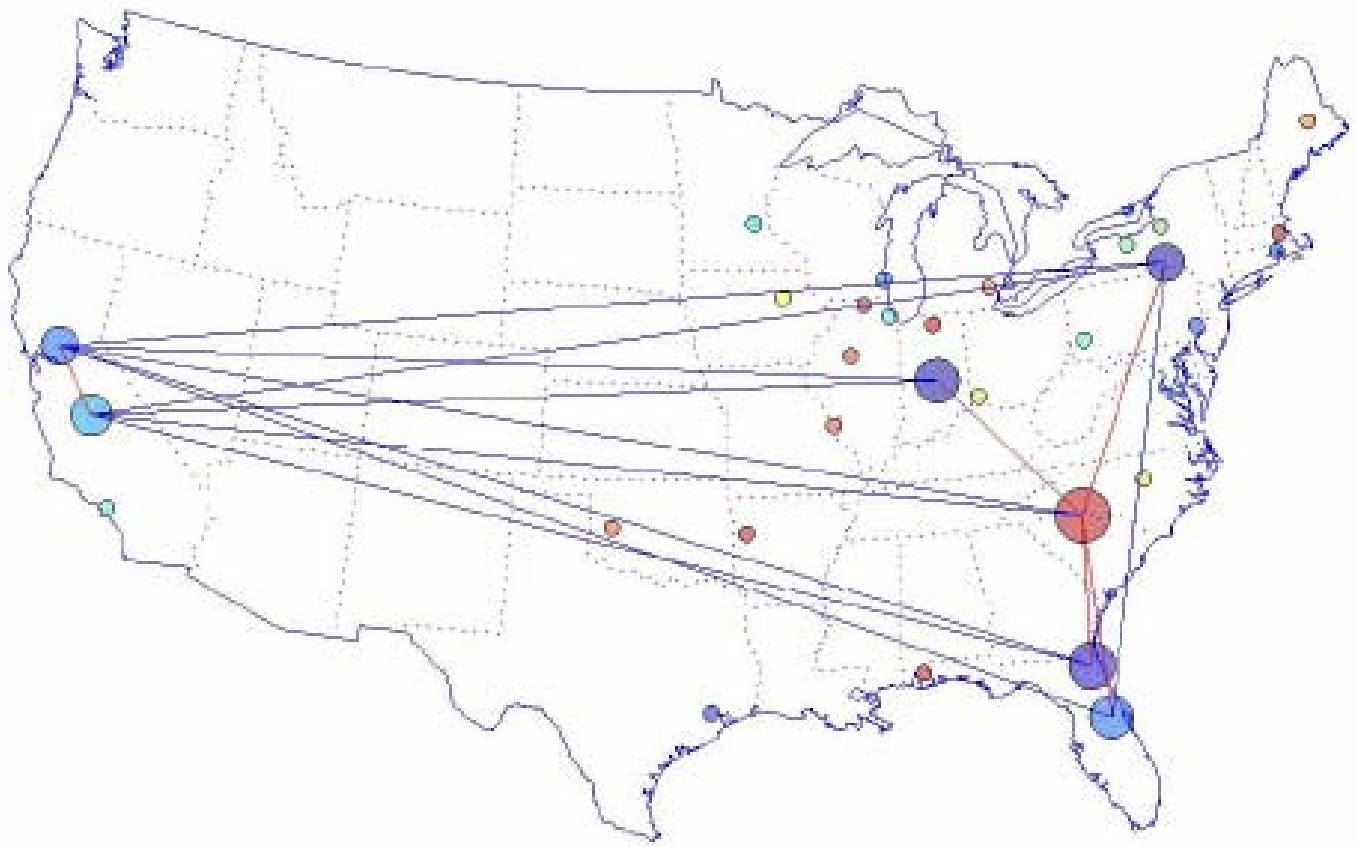
Exploration vs. exploitation

- What decision do we make?
  - » The one we think is best?
    - Exploitation
  - » Or do we make a decision just to try something and learn more about the result?
    - Exploration

Exploration vs. exploitation

- Exploration vs. exploitation with the nomadic trucker
  - » Pure exploitation

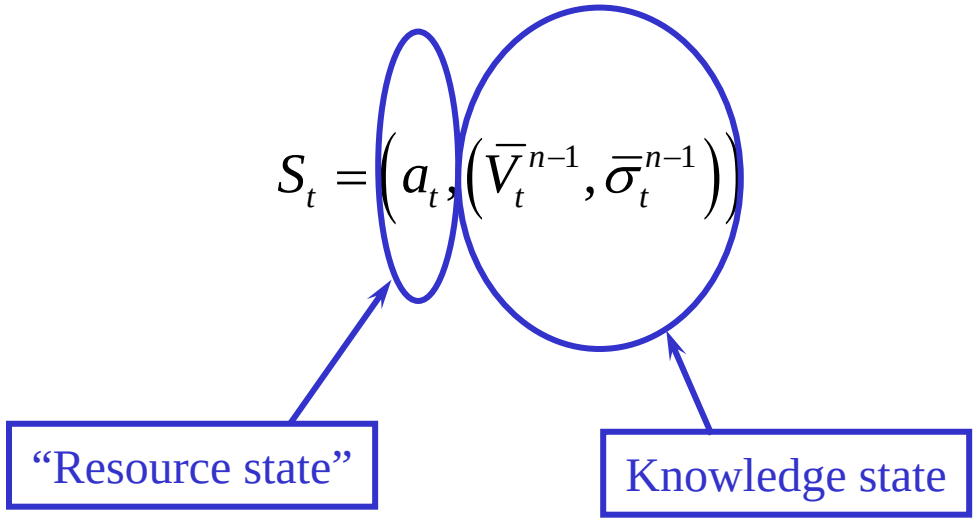
Information collection



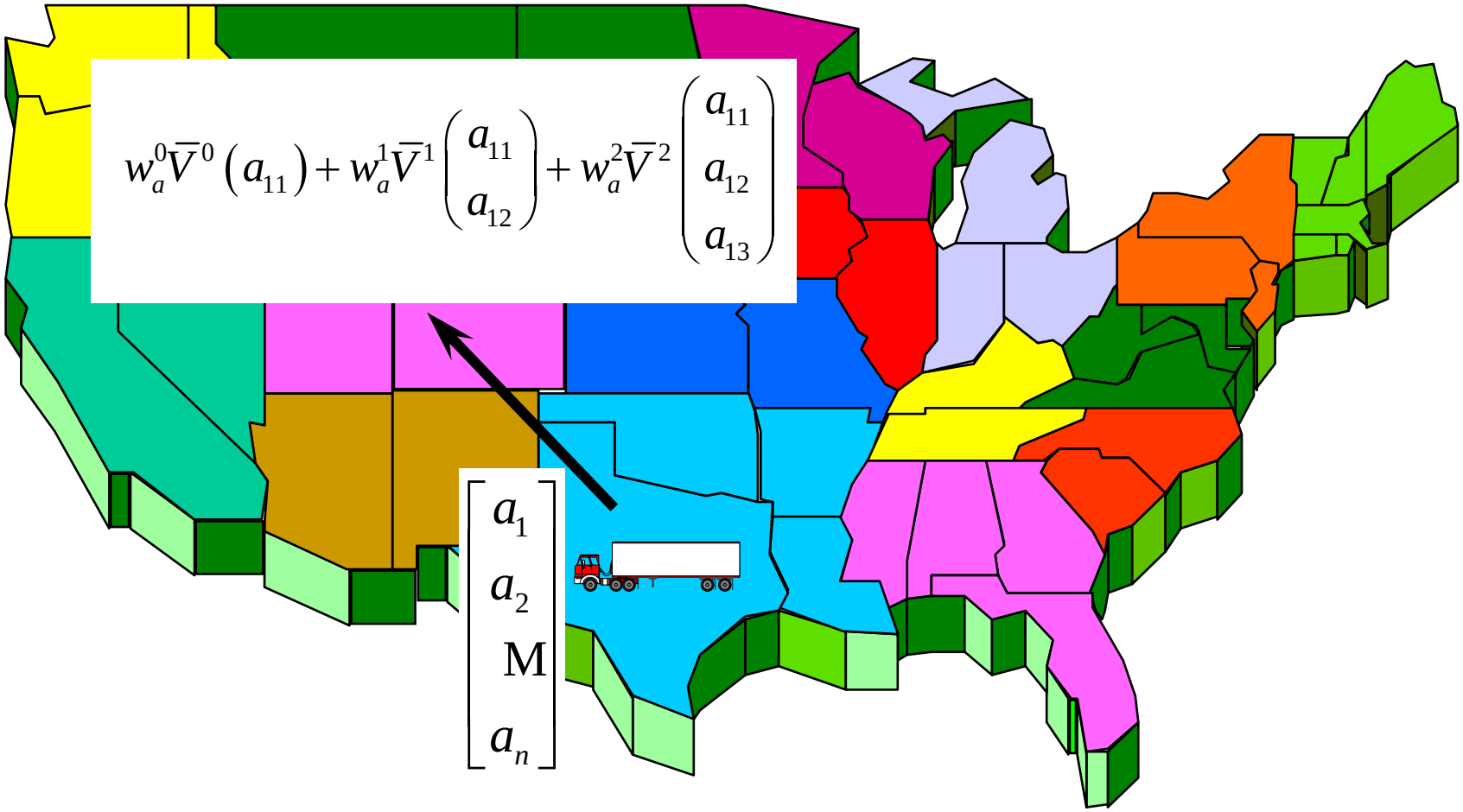
Pure exploitation

Exploration vs. exploitation

- The state variable:



Resource allocation

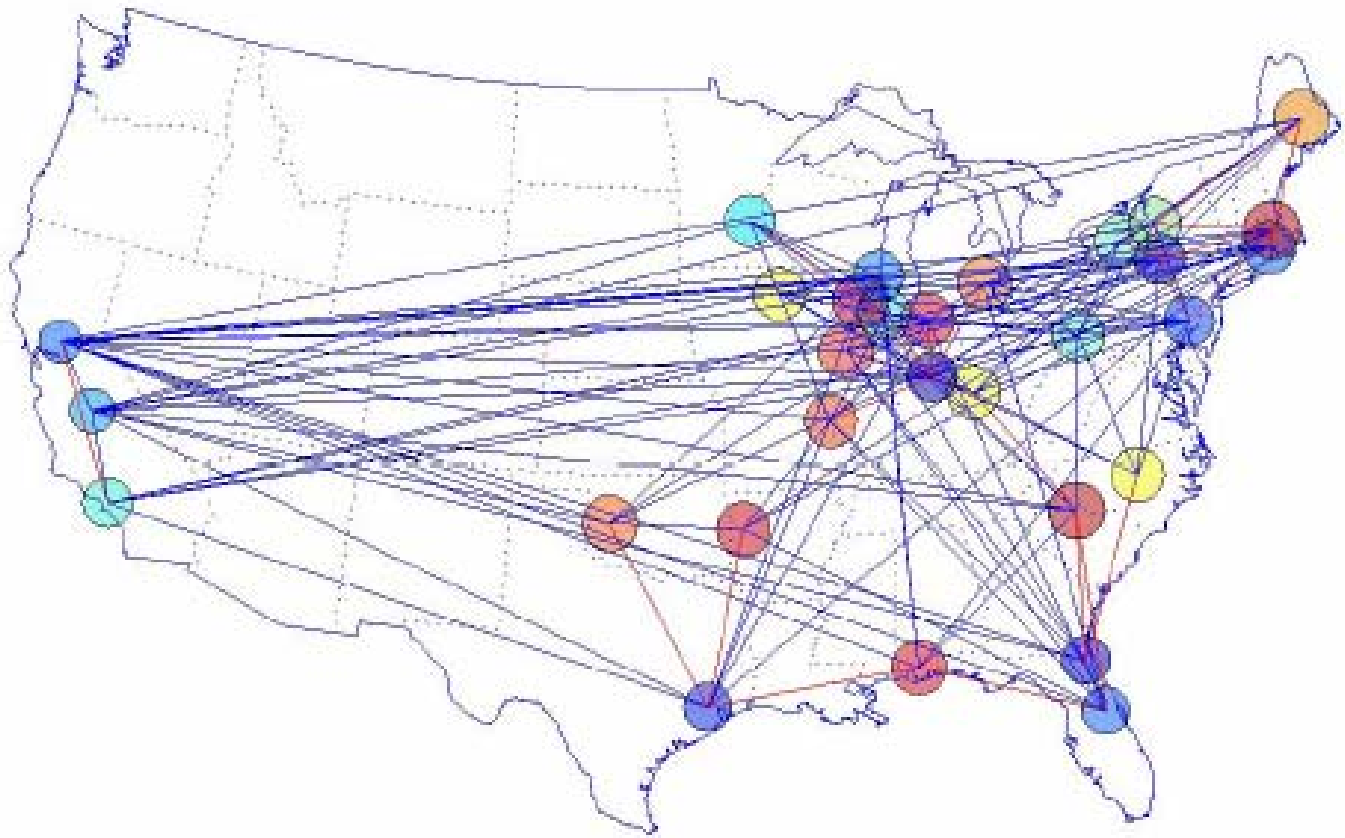


Exploration vs. exploitation

- Strategies for overcoming the exploitation trap

  - » Generalization:
    - Visit one state, learn something about other states
    - Exploitation with more general learning

Information collection

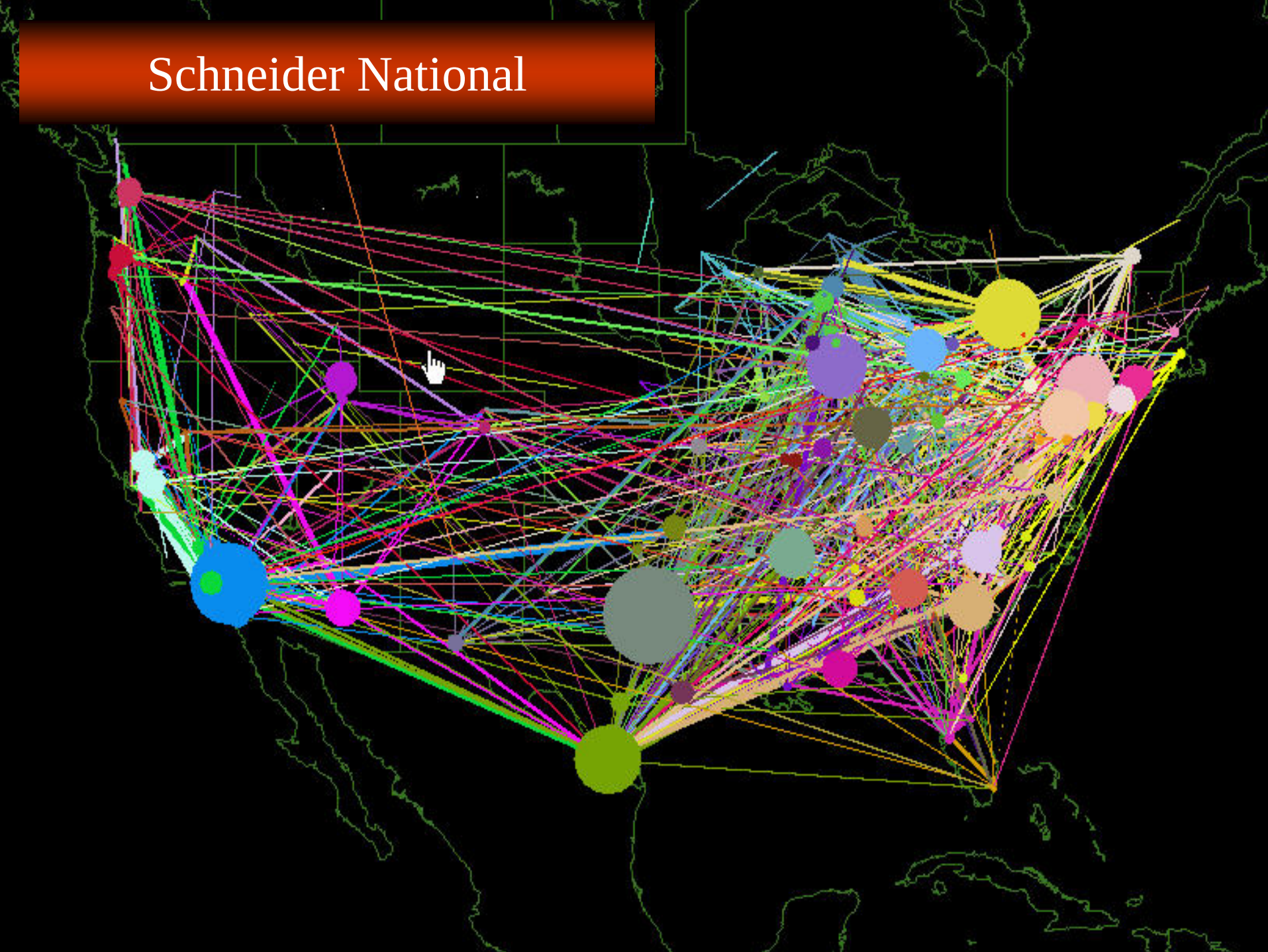


Pure exploration

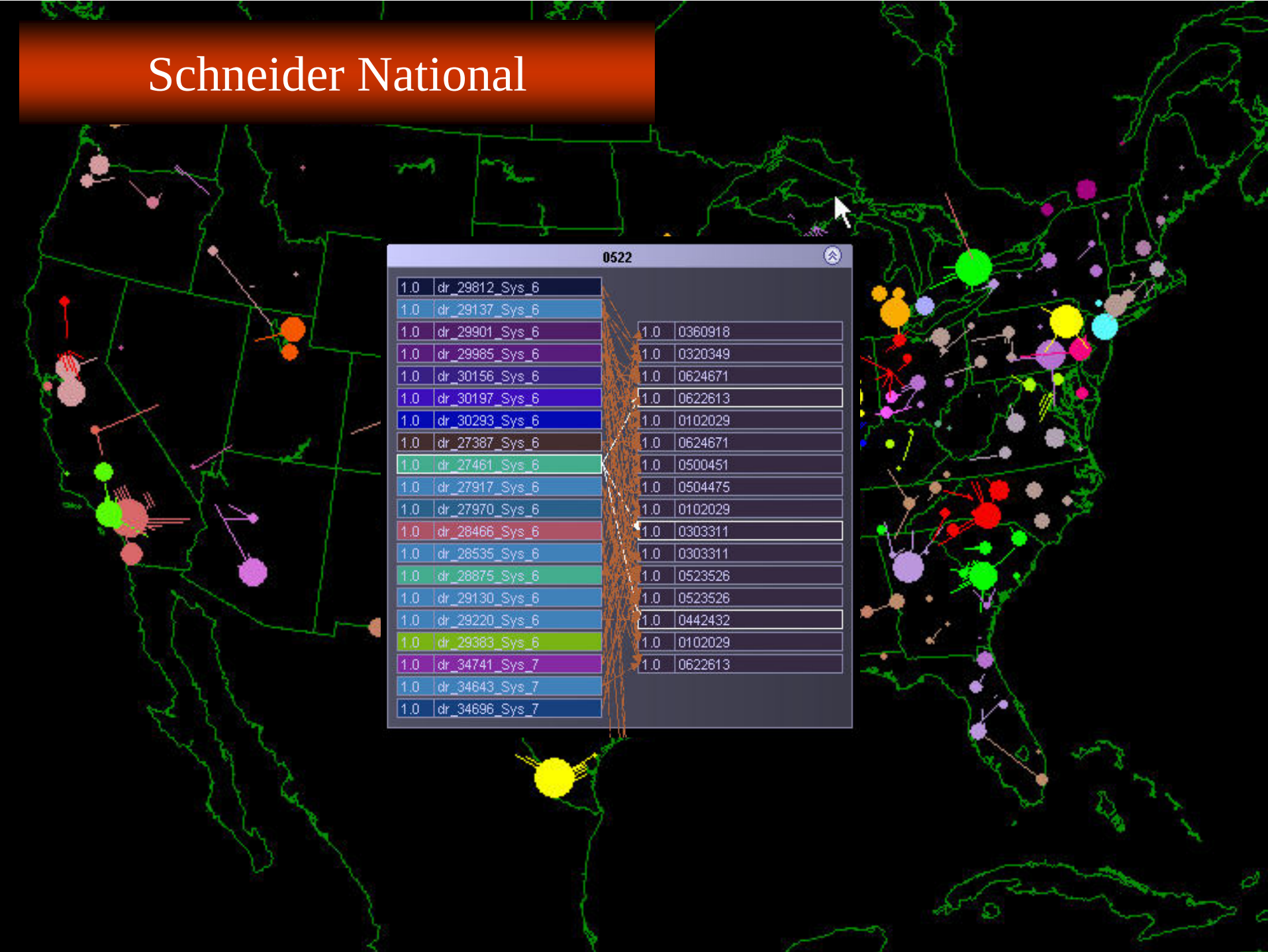
Outline

- The languages of dynamic programming
- A resource allocation model
- The post-decision state variable
- Example: A discrete resource: the nomadic trucker
- The states of our system
- Example: A continuous resource: blood inventory management
- Approximation methods
  - » Lookup tables and aggregation
  - » Basis functions
- Stepsizes
- Exploration vs. exploitation
- Applications

Schneider National

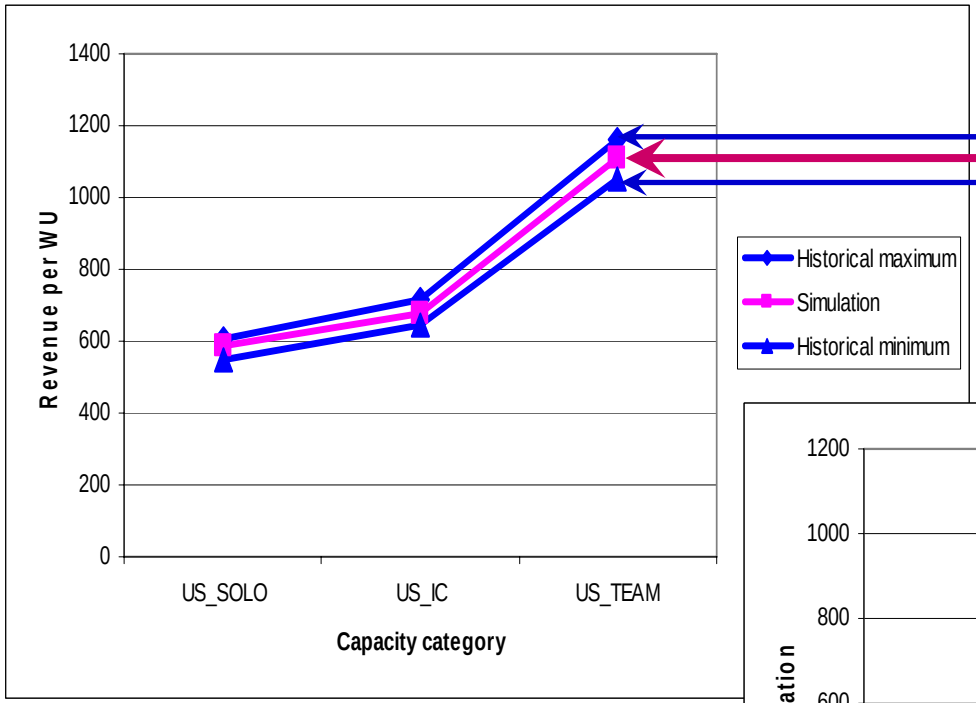


Schneider National




SCHNEIDER
Week 45
Miles
Working units
By industry
Miles
Rsv YOY
Growth
SCHNEIDER

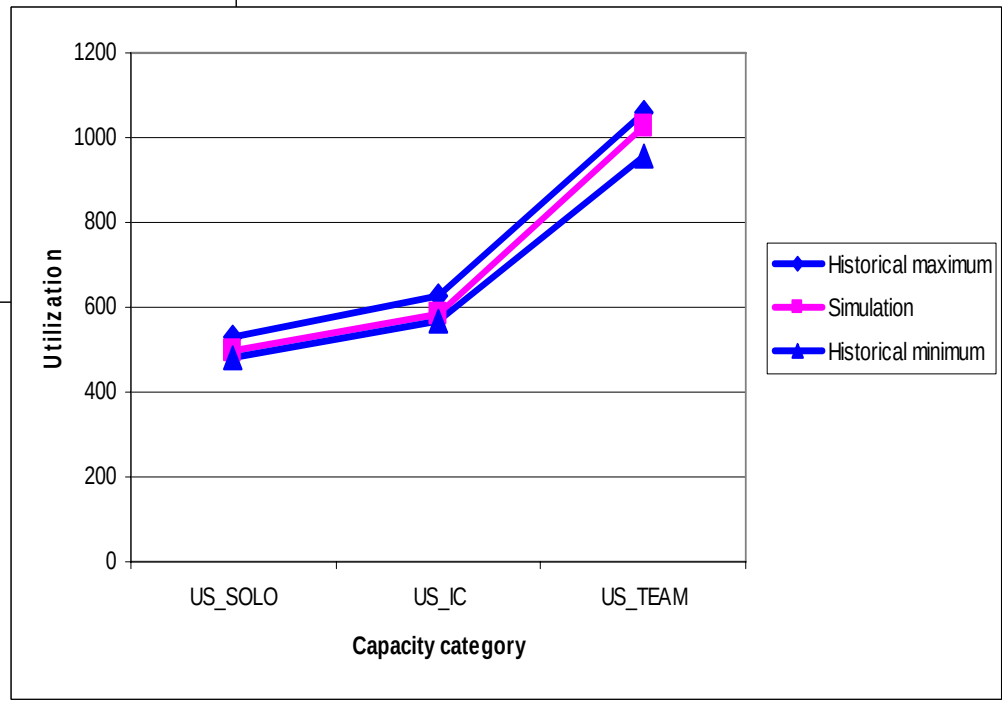
Case study: truckload trucking



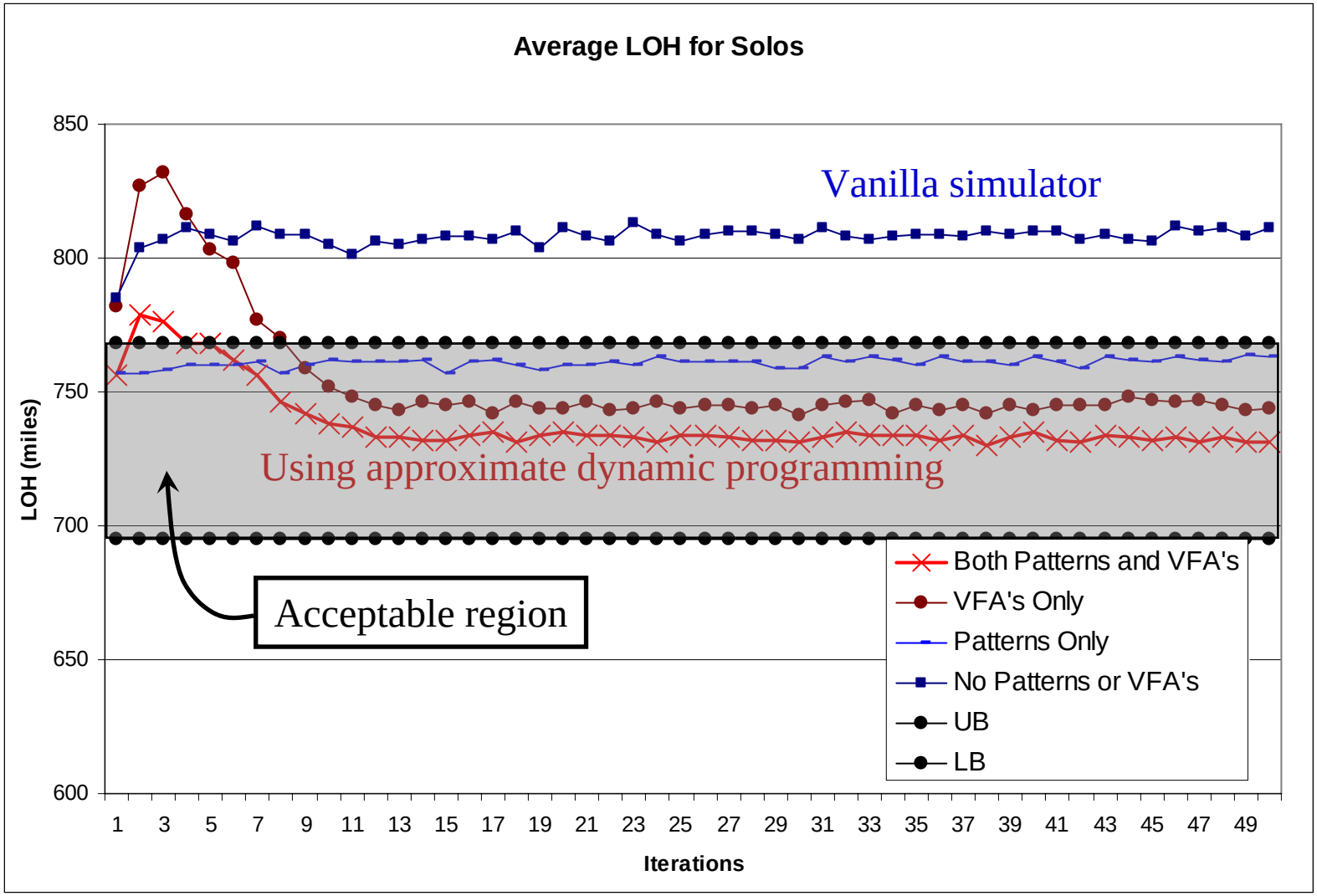
Historical min and max

Revenue per WU

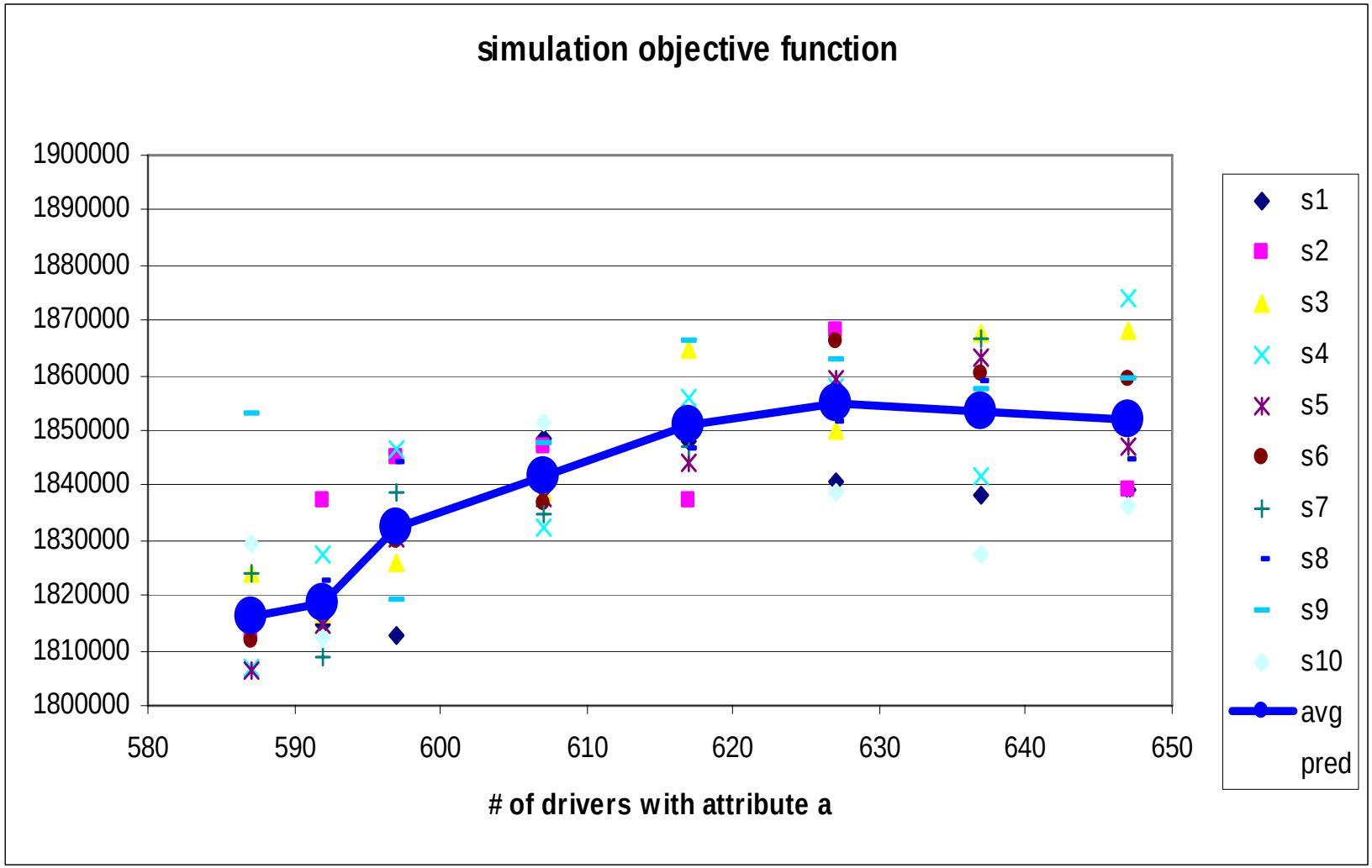
Utilization



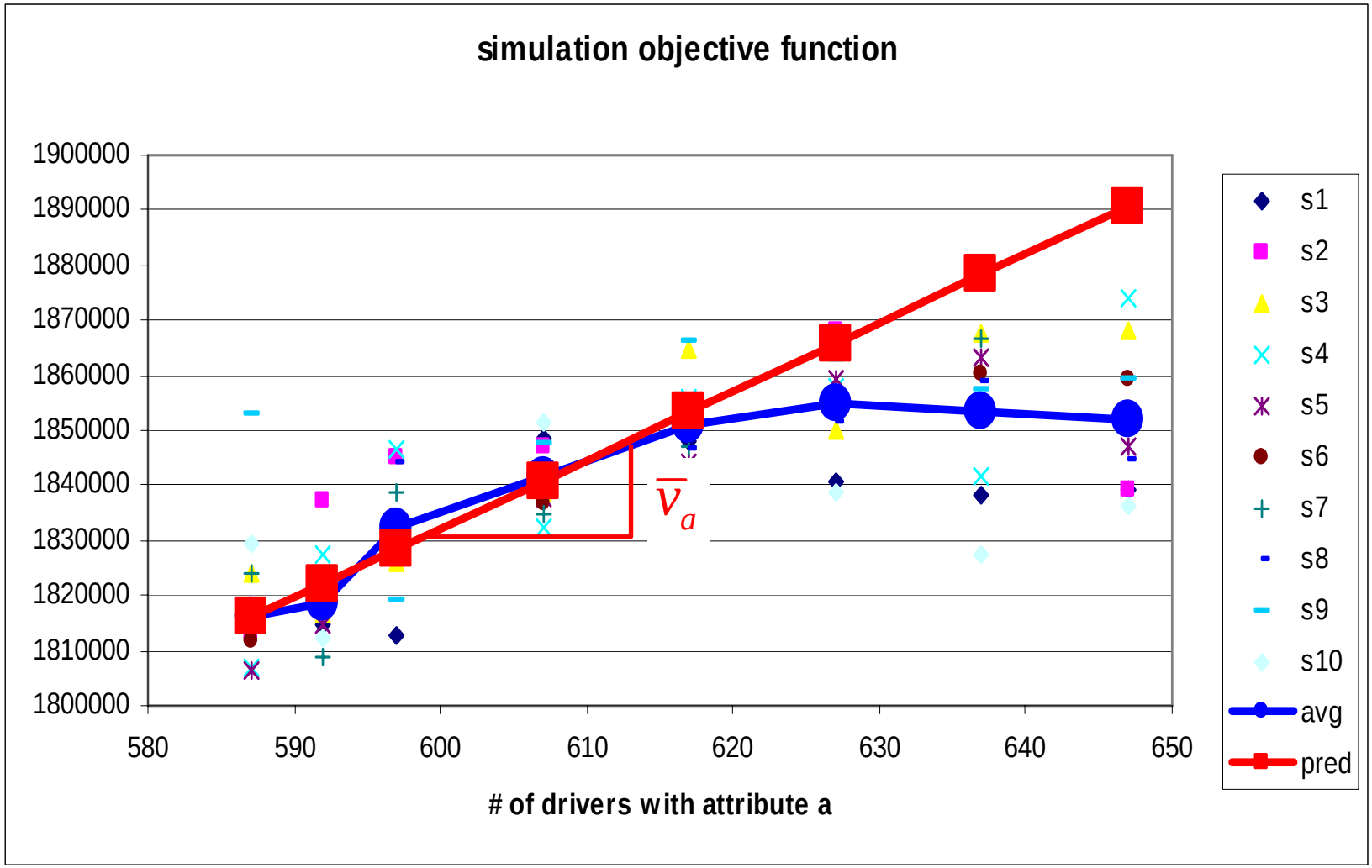
Case study: truckload trucking



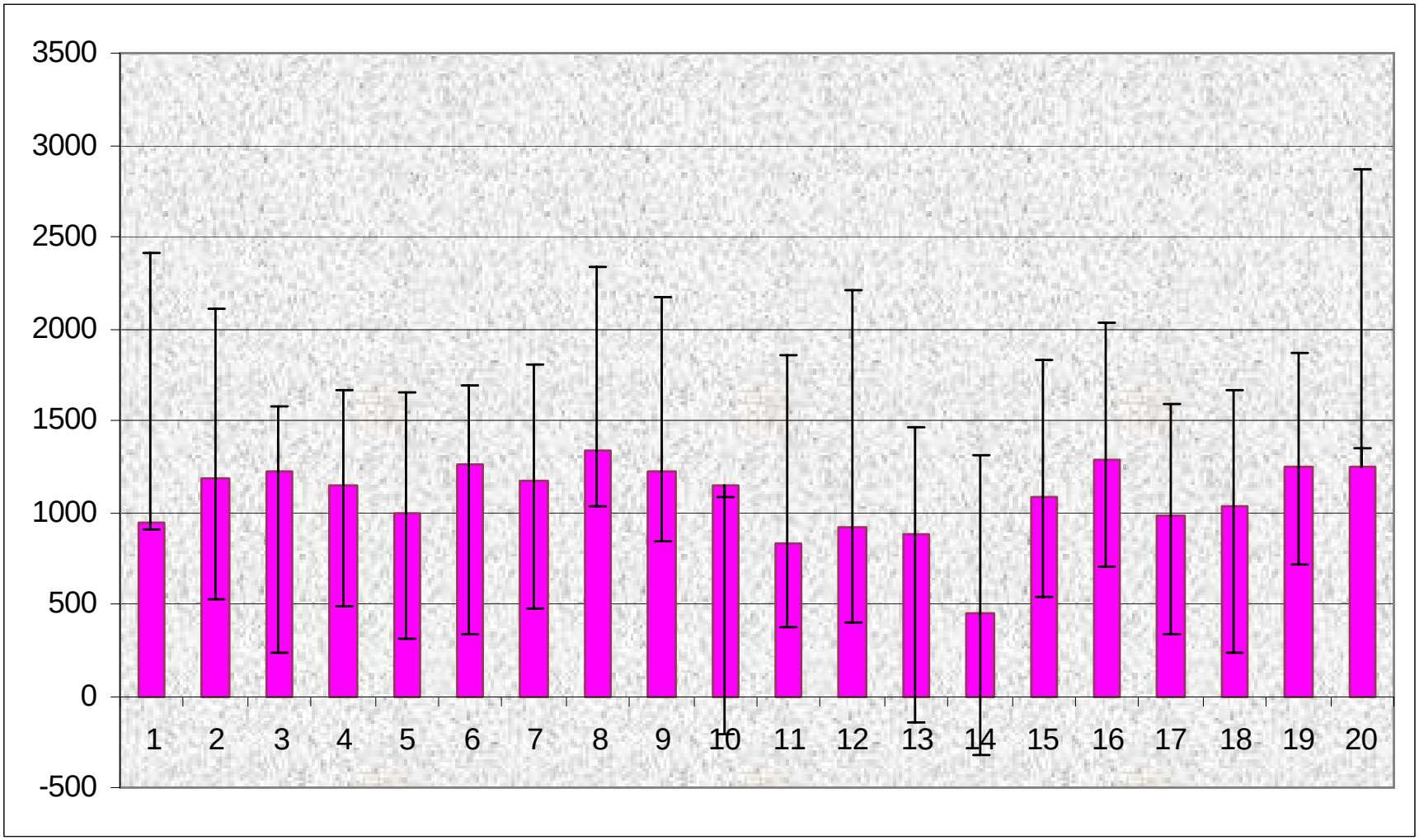
Case study: truckload trucking



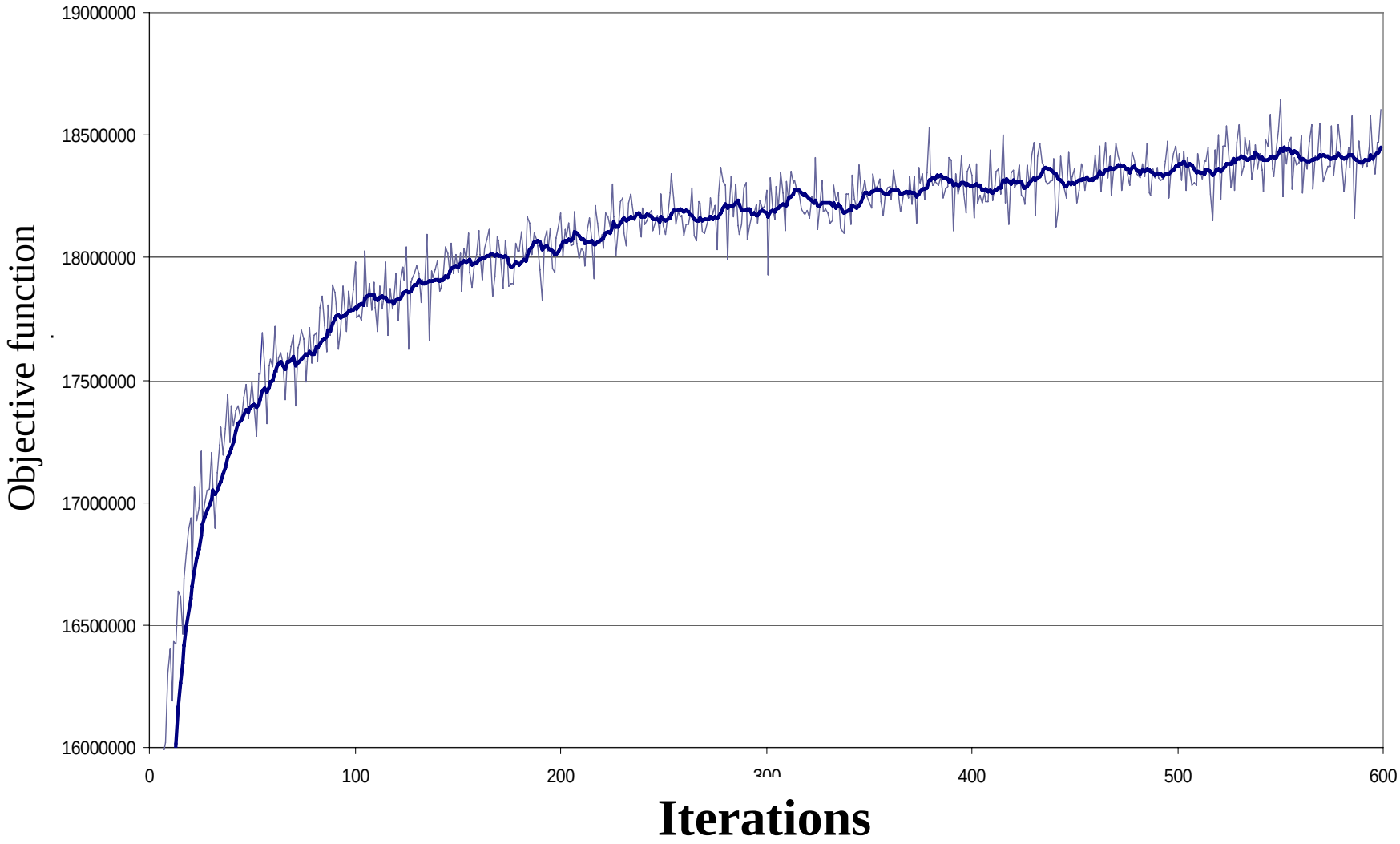
Case study: truckload trucking



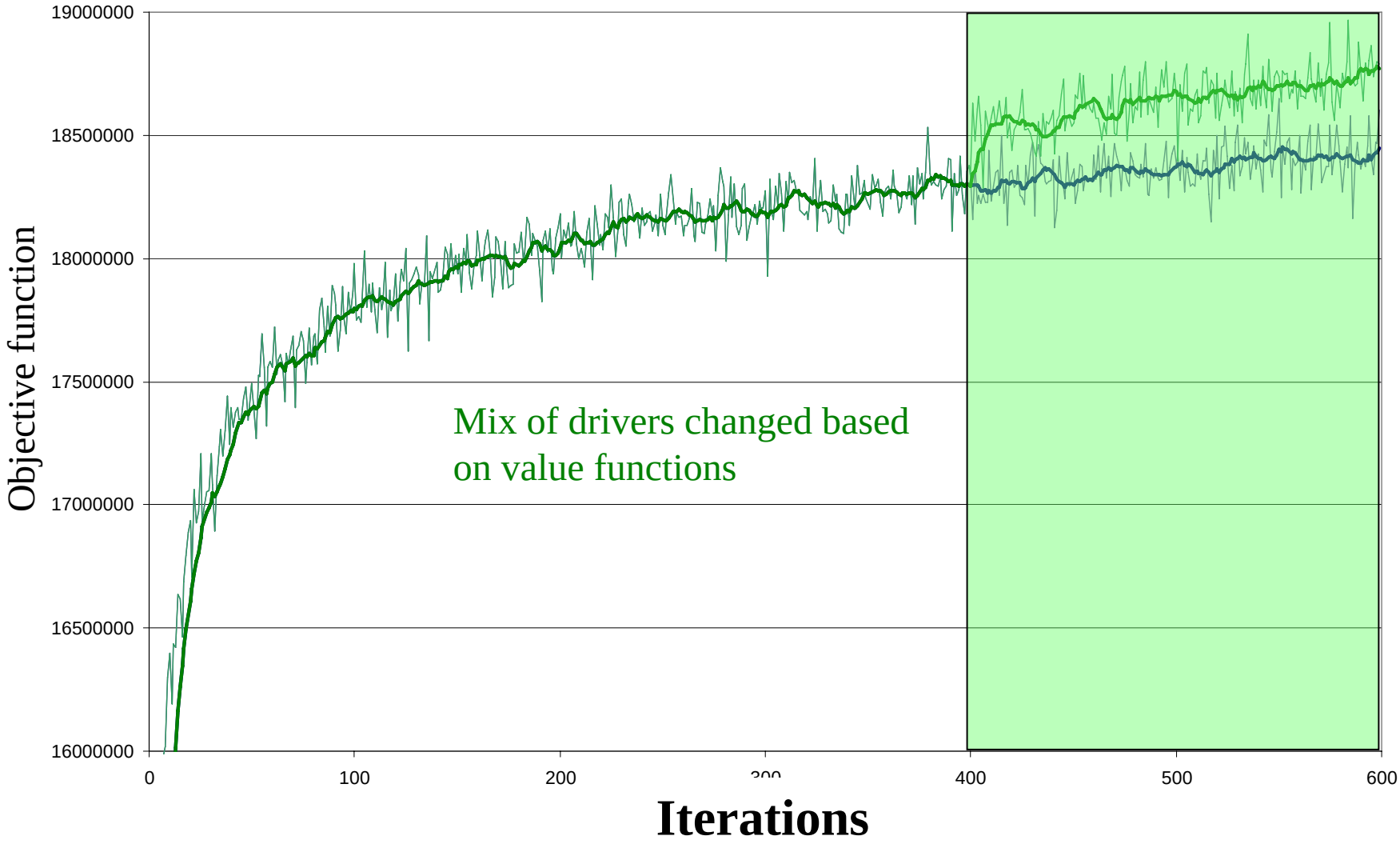
Case study: truckload trucking



Case study: truckload trucking

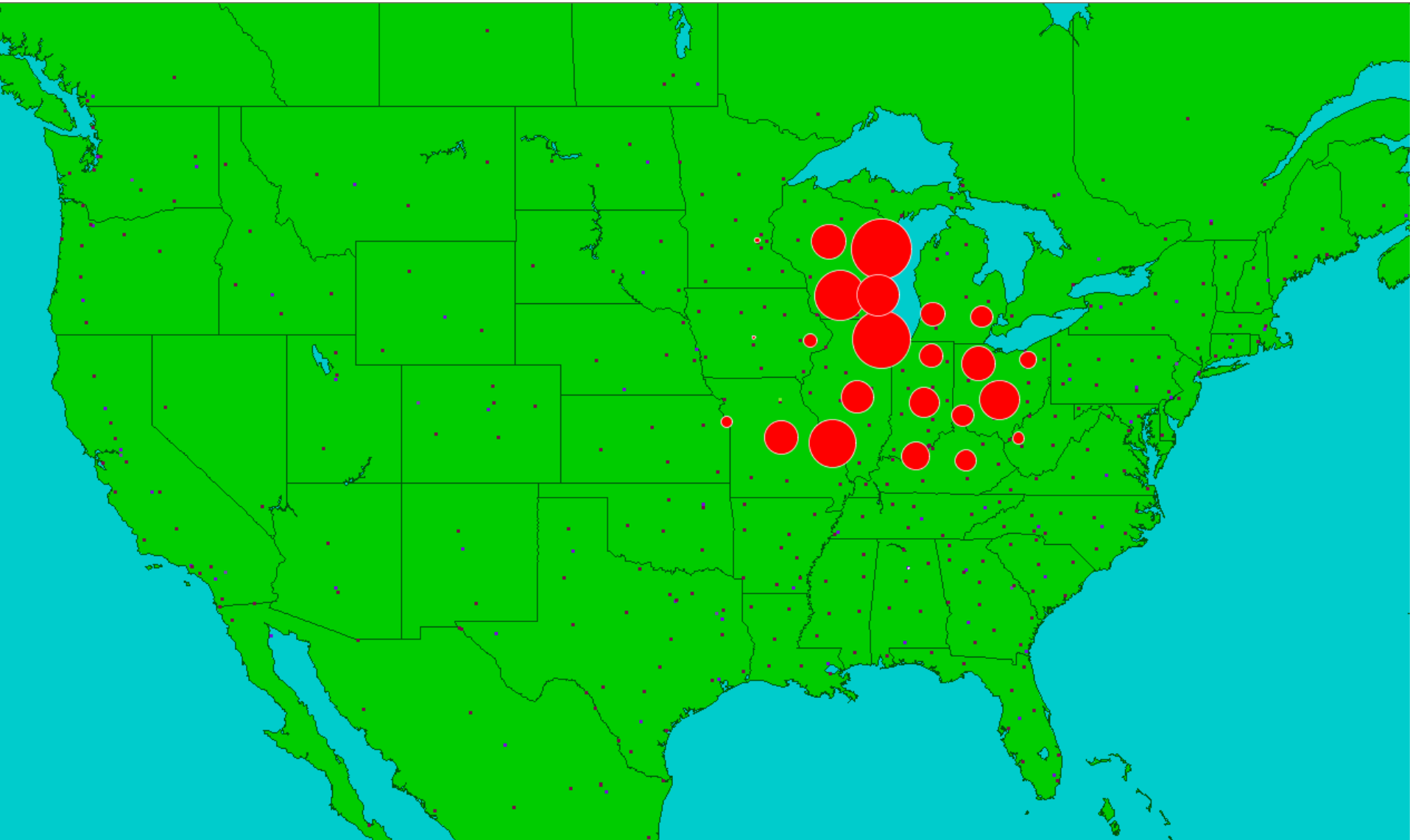


Case study: truckload trucking

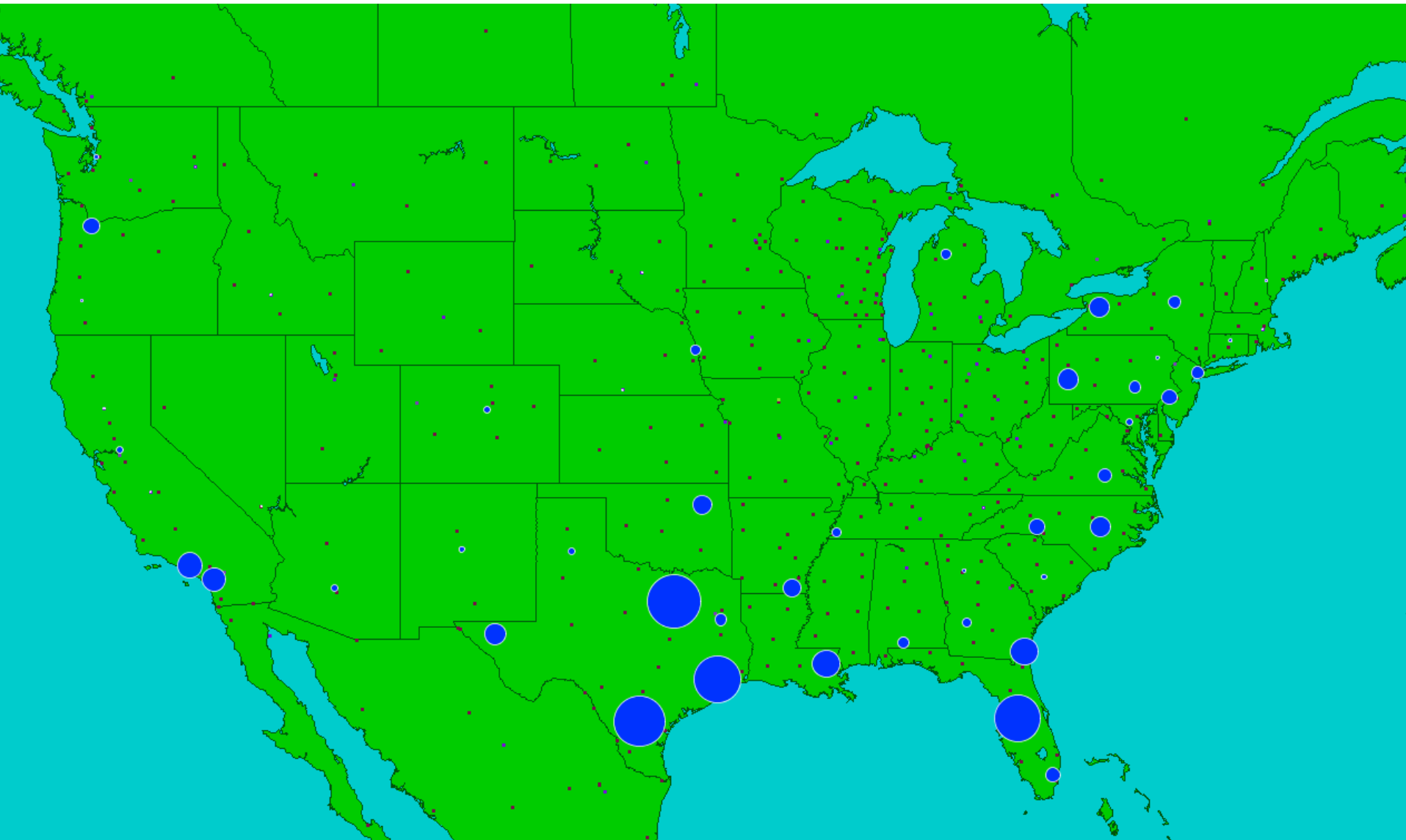


Case study: truckload trucking

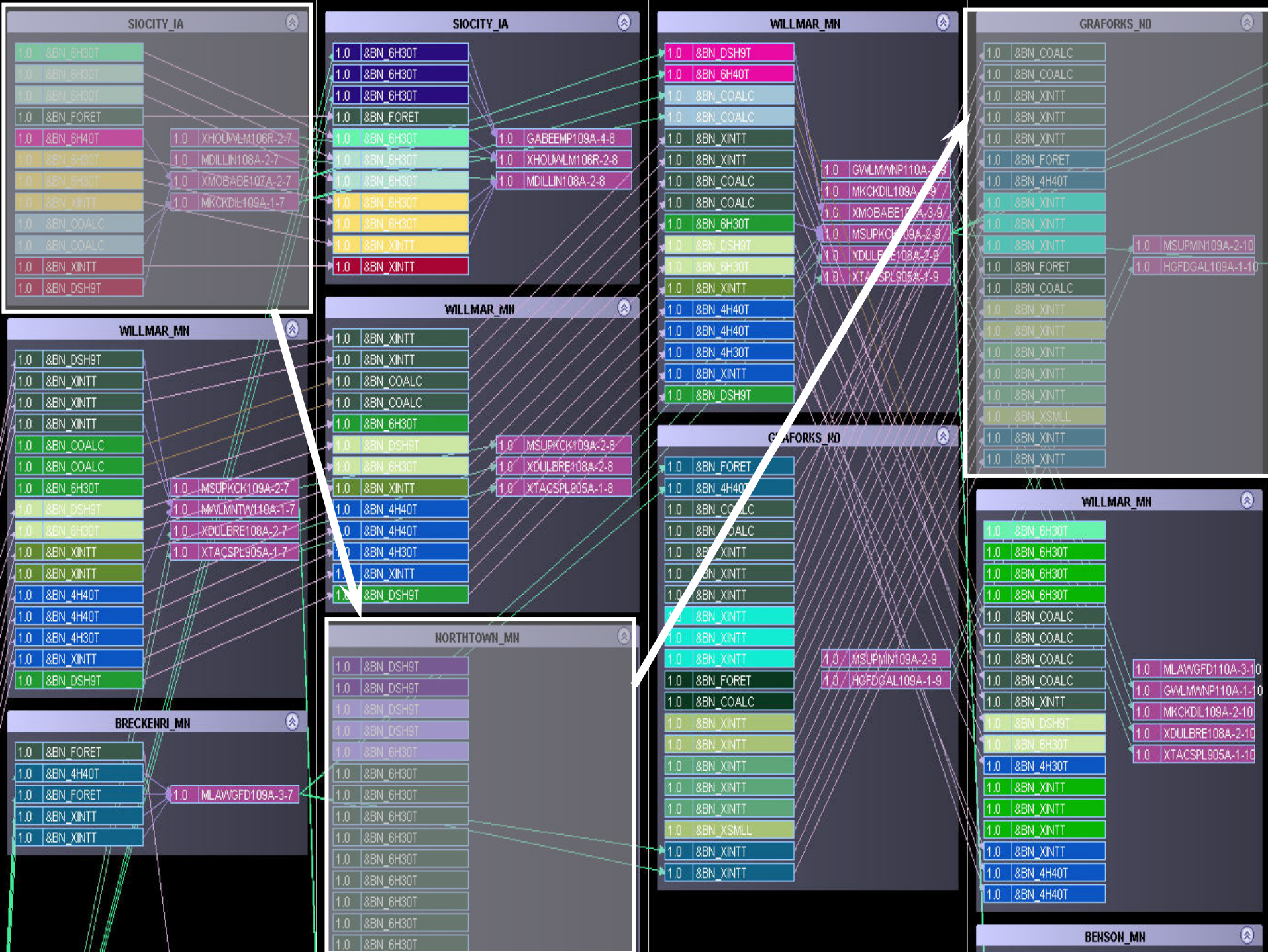
Case study: truckload trucking

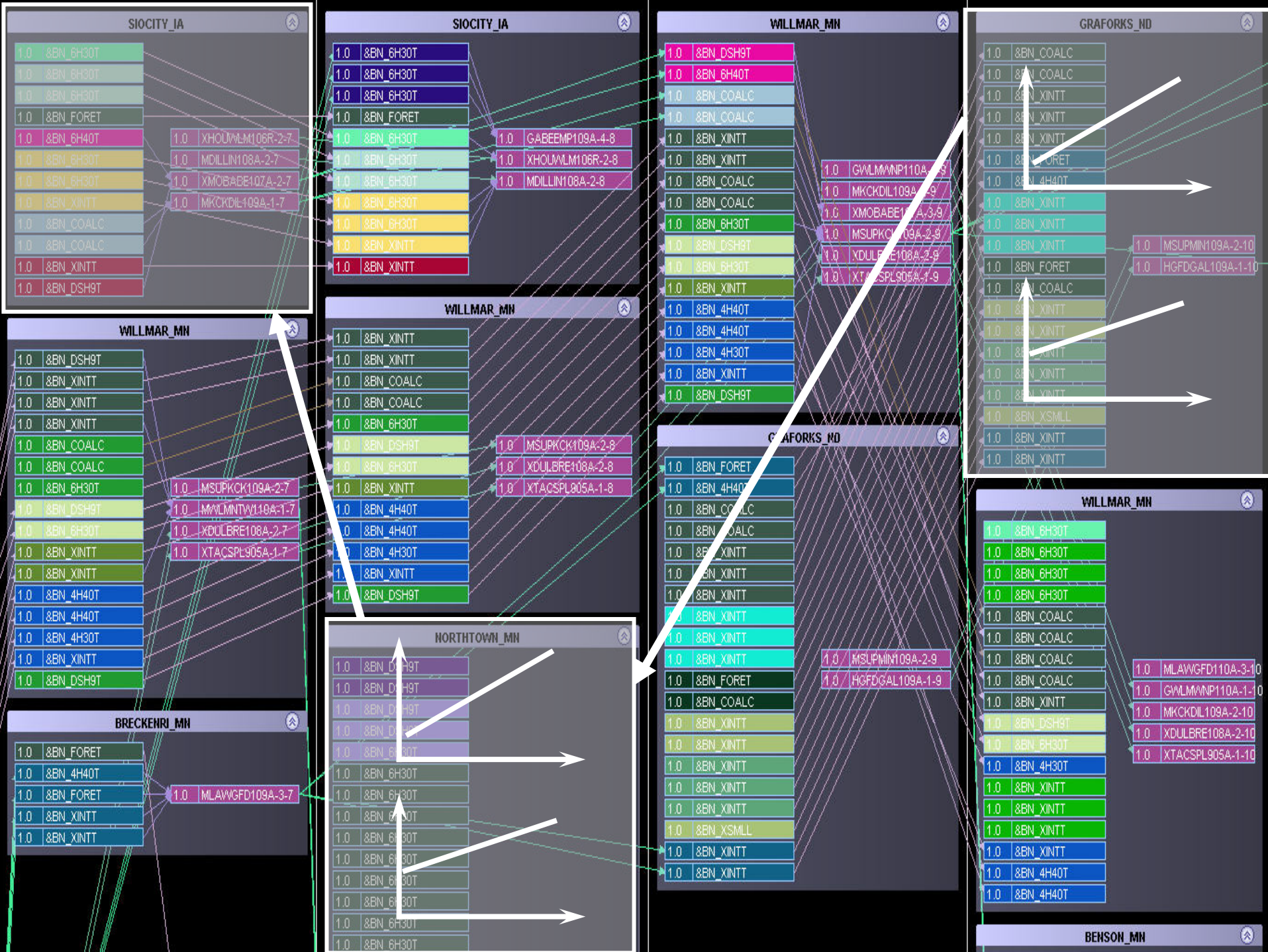


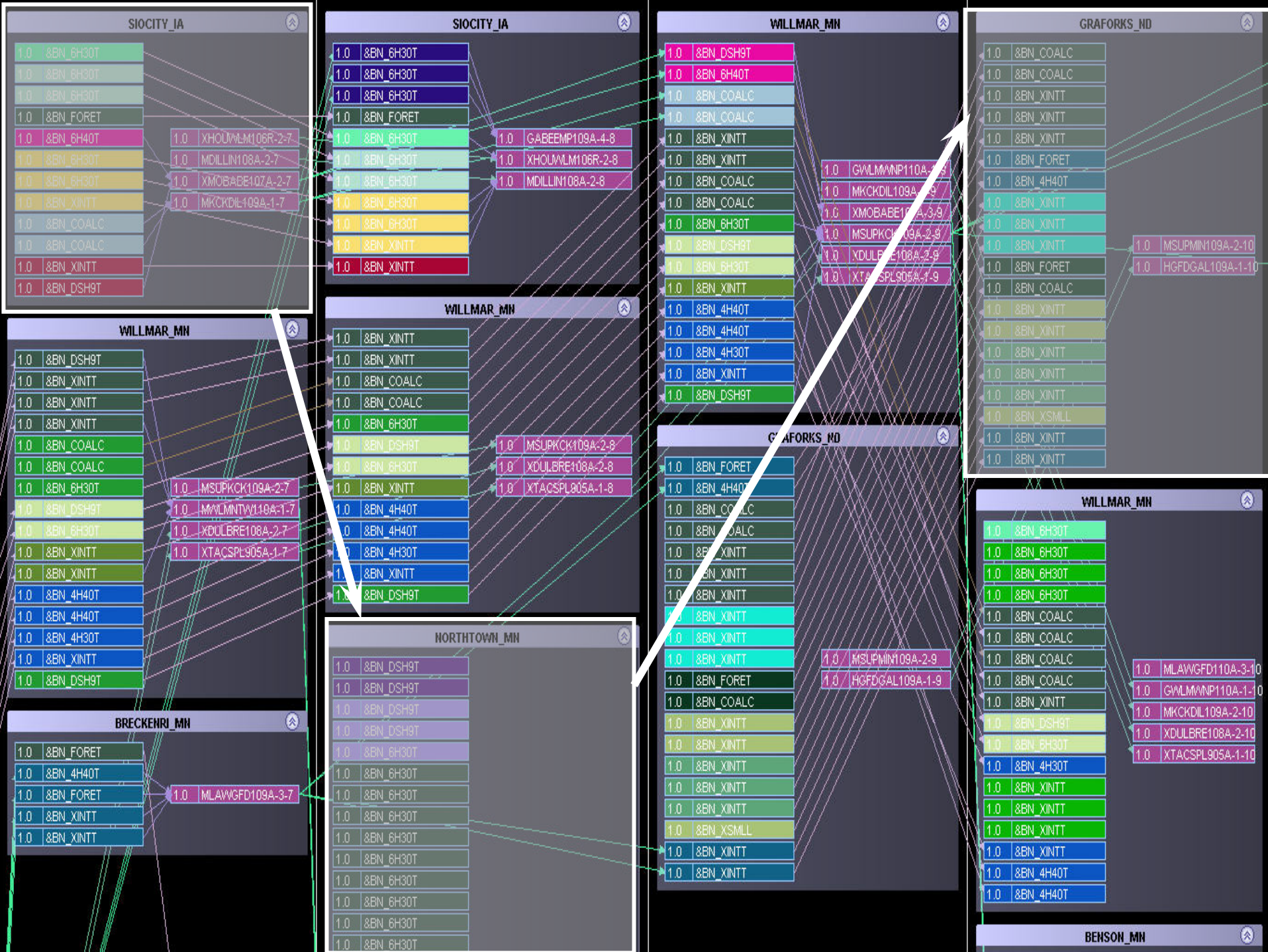
Case study: truckload trucking

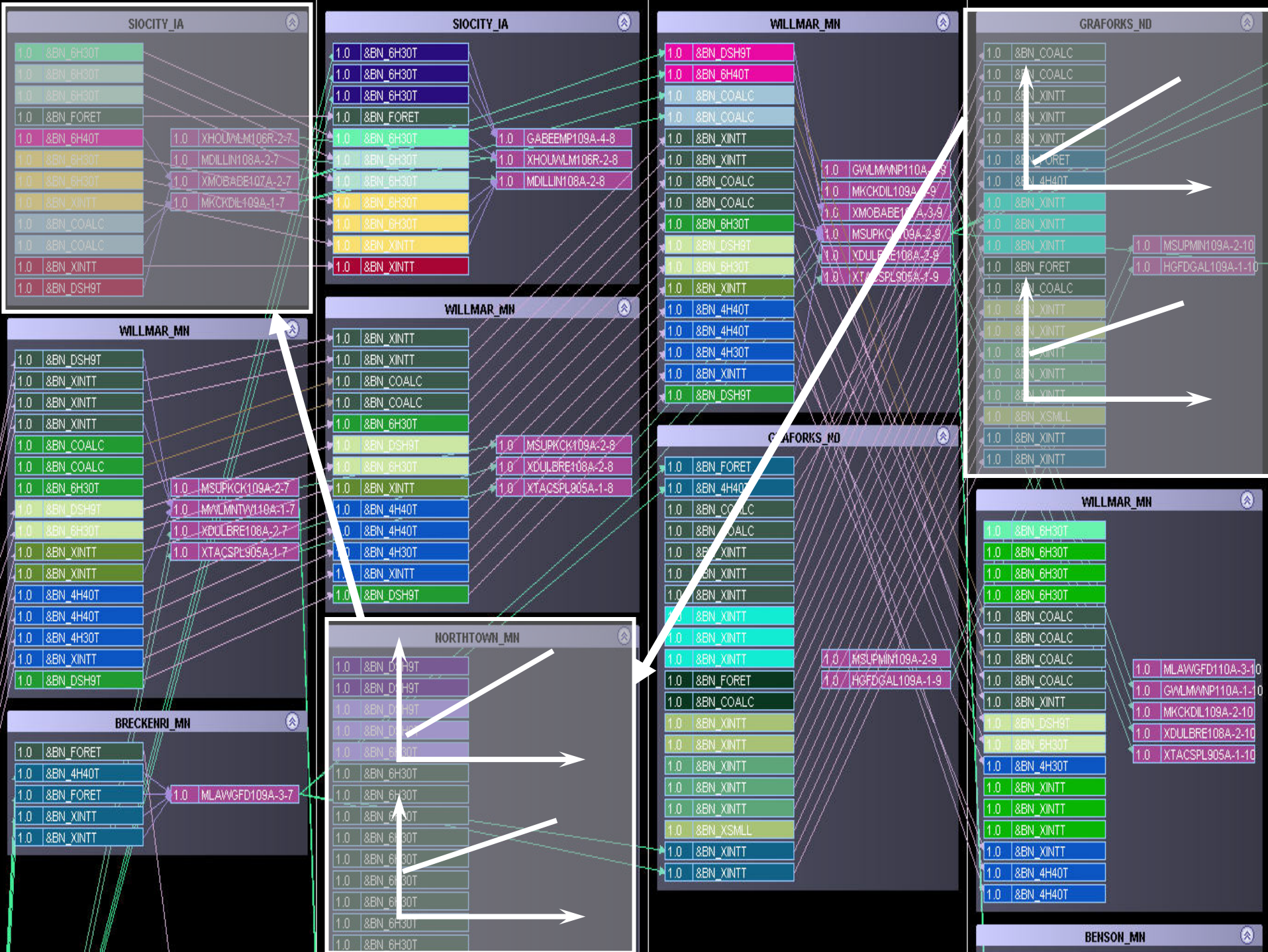



4448
BNSF
4448
BNSF
Burlington Northern
Santa Fe
Railway


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

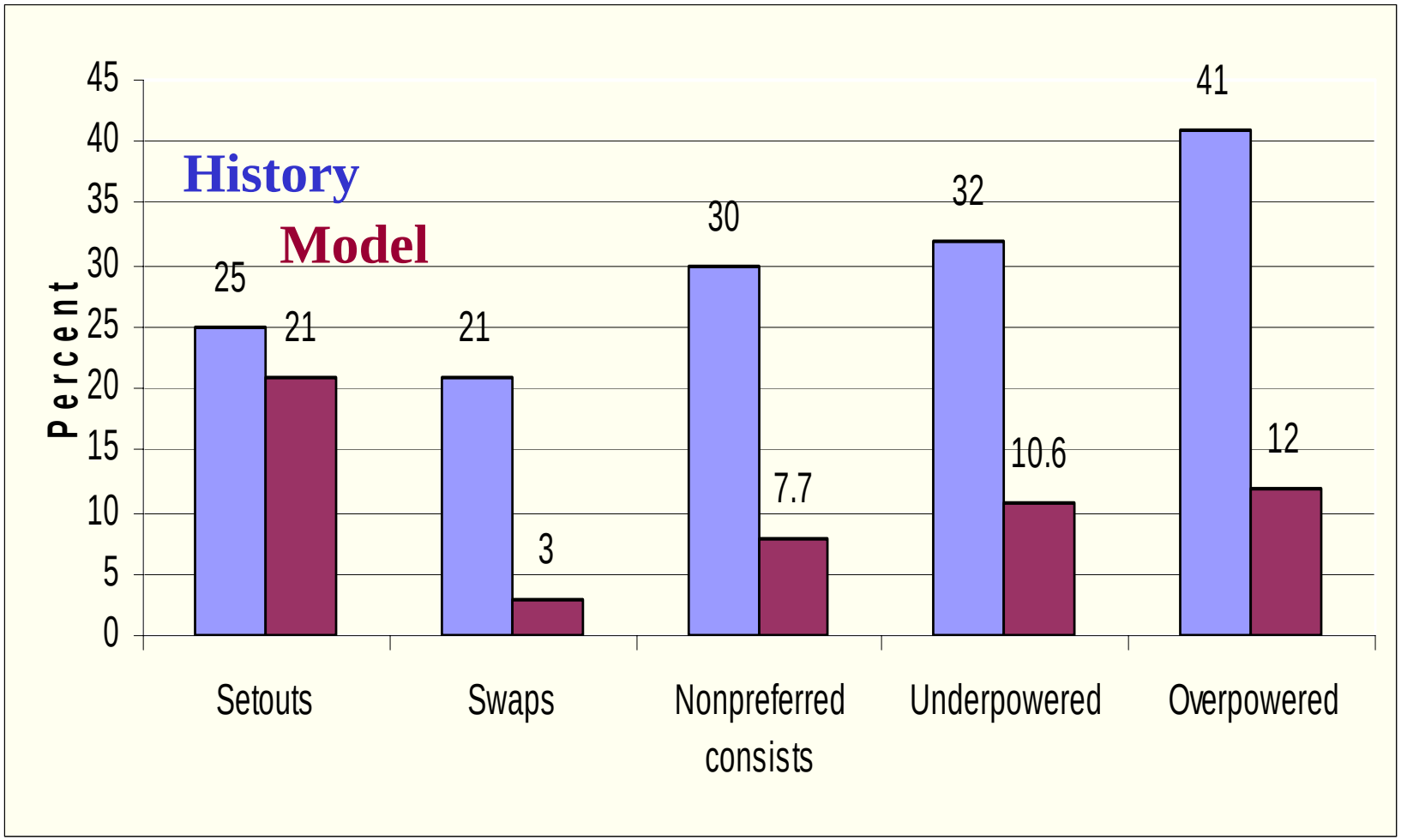

SIOCITY_IA
1.0 &BN_6H30T
1.0 &BN_6H30T
1.0 &BN_6H30T
1.0 &BN_FORET
1.0 &BN_6H40T
1.0 &BN_6H30T
1.0 &BN_6H30T
1.0 &BN_XINTT
1.0 &BN_COALC
1.0 &BN_COALC
1.0 &BN_XINTT
1.0 &BN_DSH9T
1.0 XHOUWLM106R-2-7
1.0 MDILLIN108A-2-7
1.0 XMOBABE107A-2-7
1.0 MKCKDIL109A-1-7
SIOCITY_IA
1.0 &BN_6H30T
1.0 &BN_6H30T
1.0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


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


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

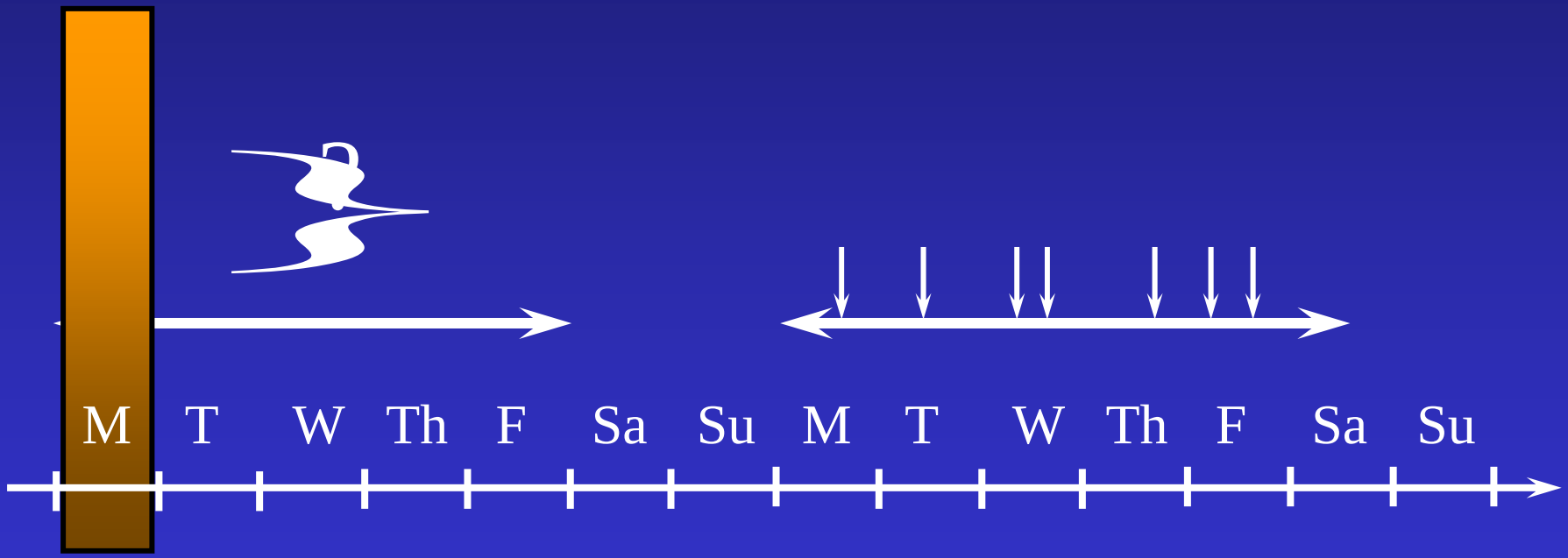
Implementation metrics

- Results from the real world:



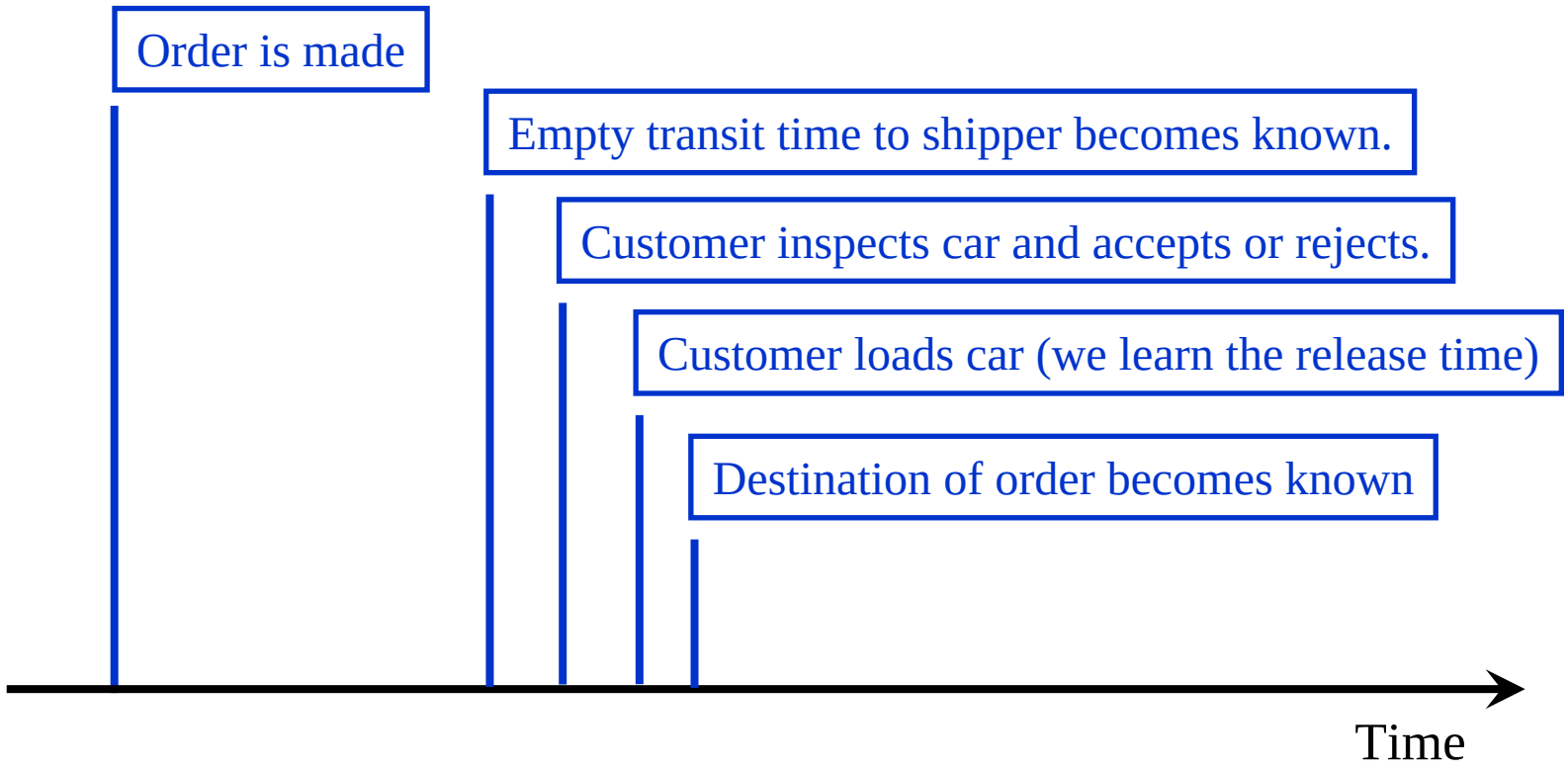

7044 7044

The planning process

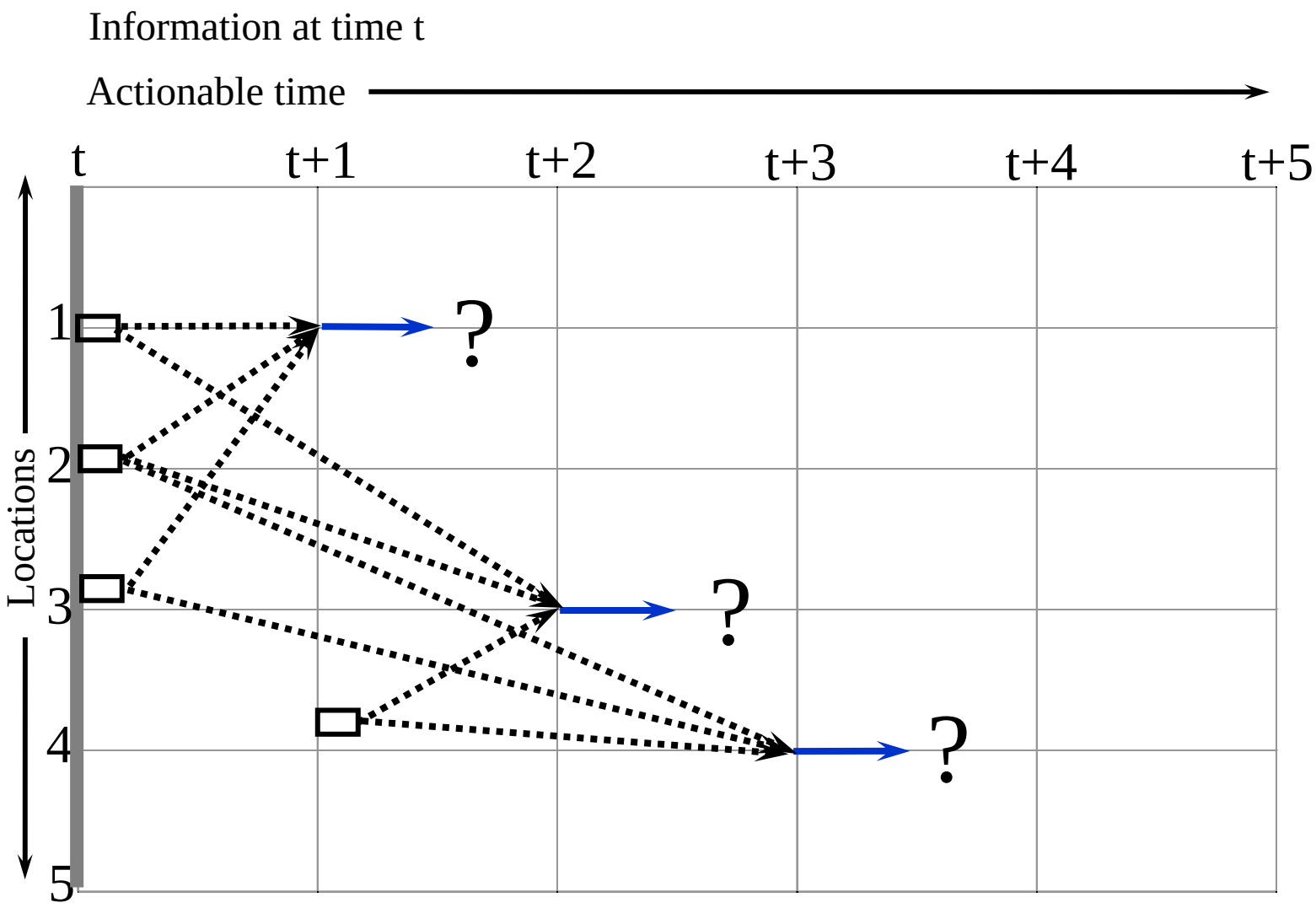


The flow of information

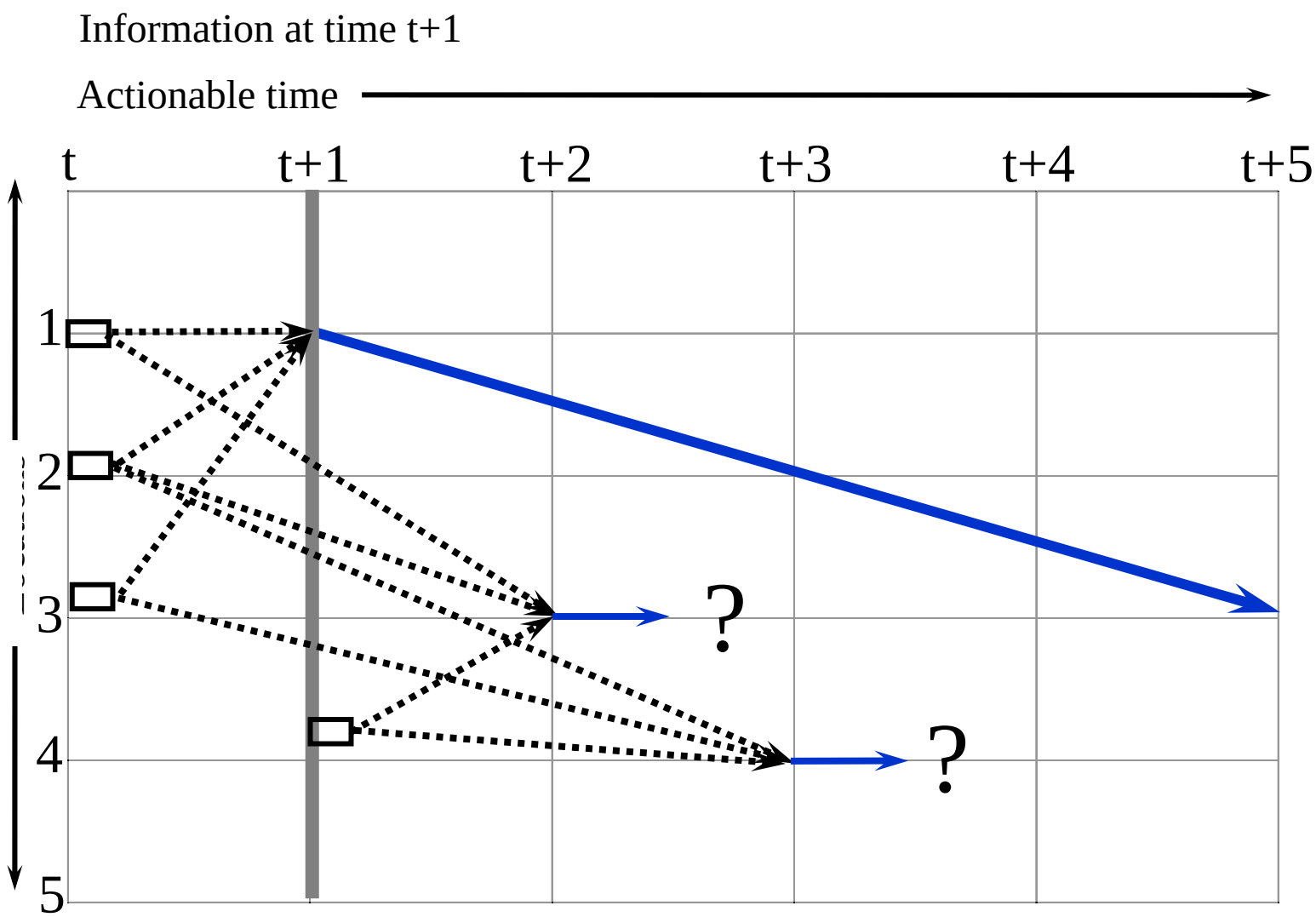
In practice, there are a number of parallel information processes taking place:



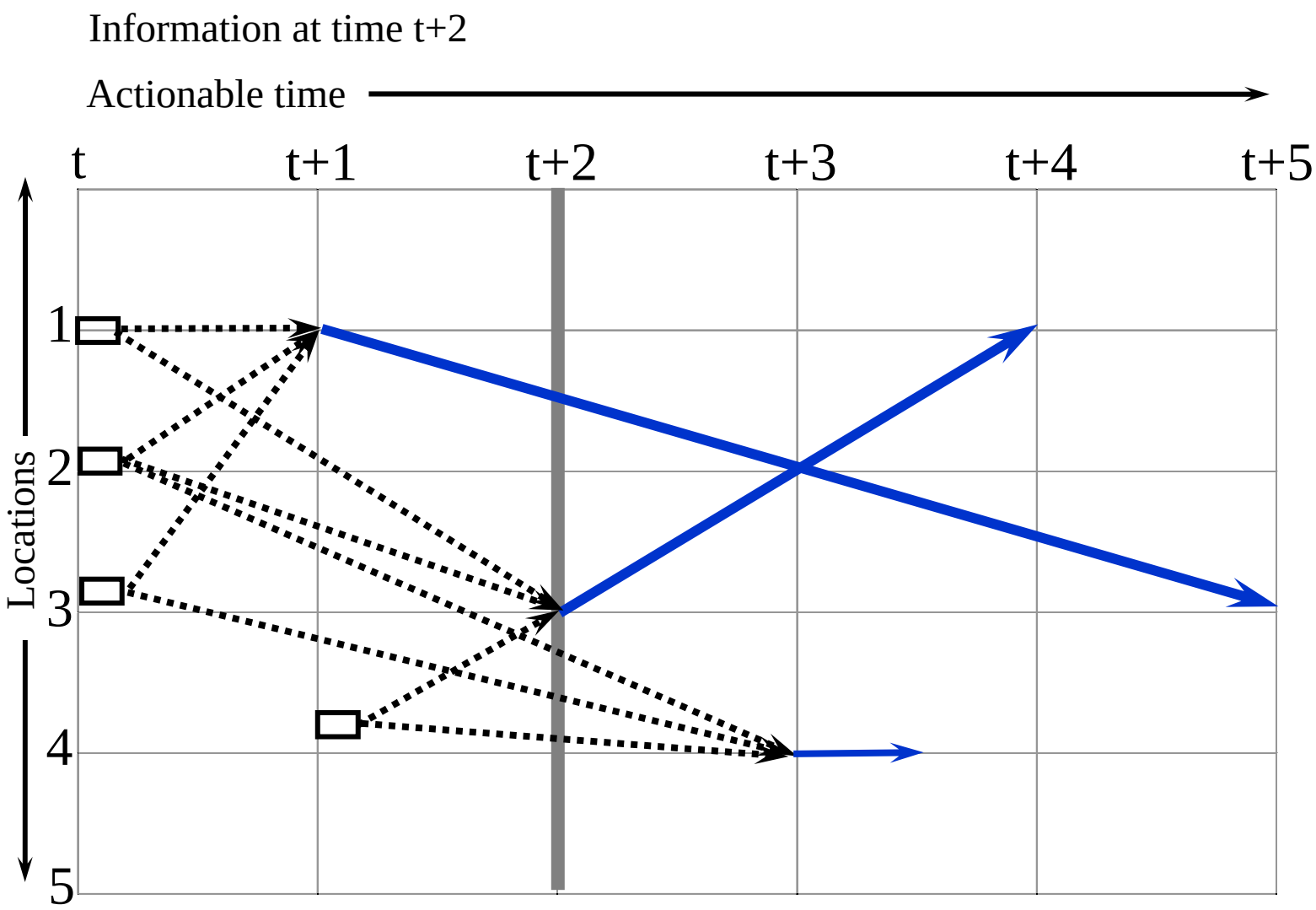
Multiattribute resources



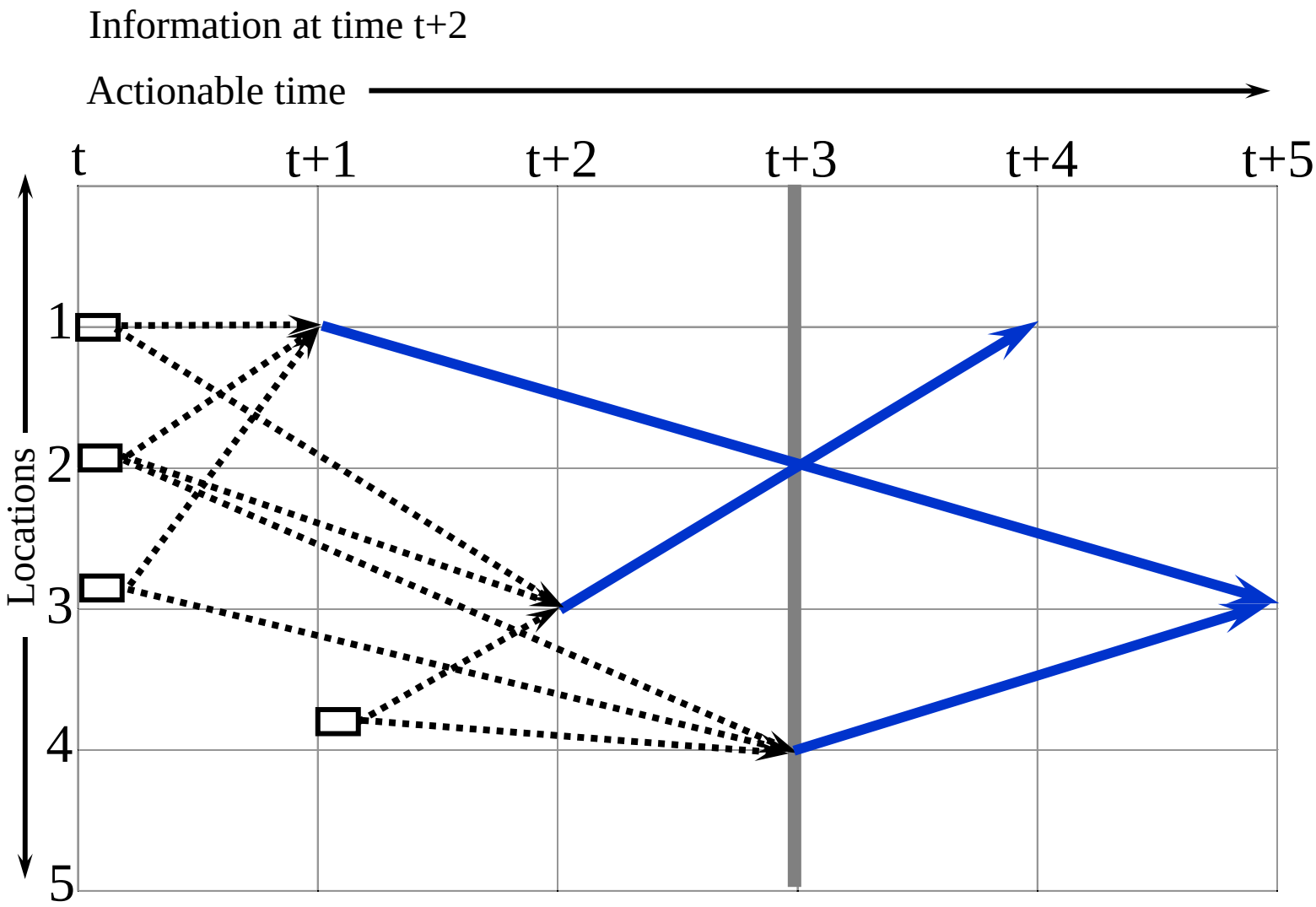
Multiattribute resources



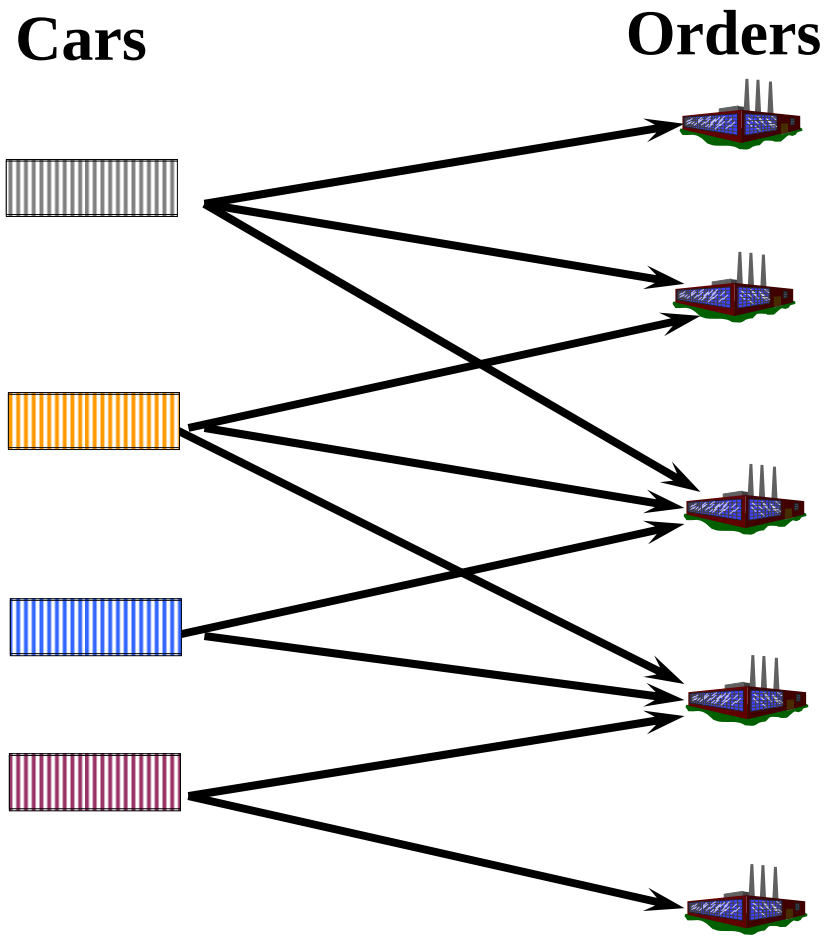
Multiattribute resources



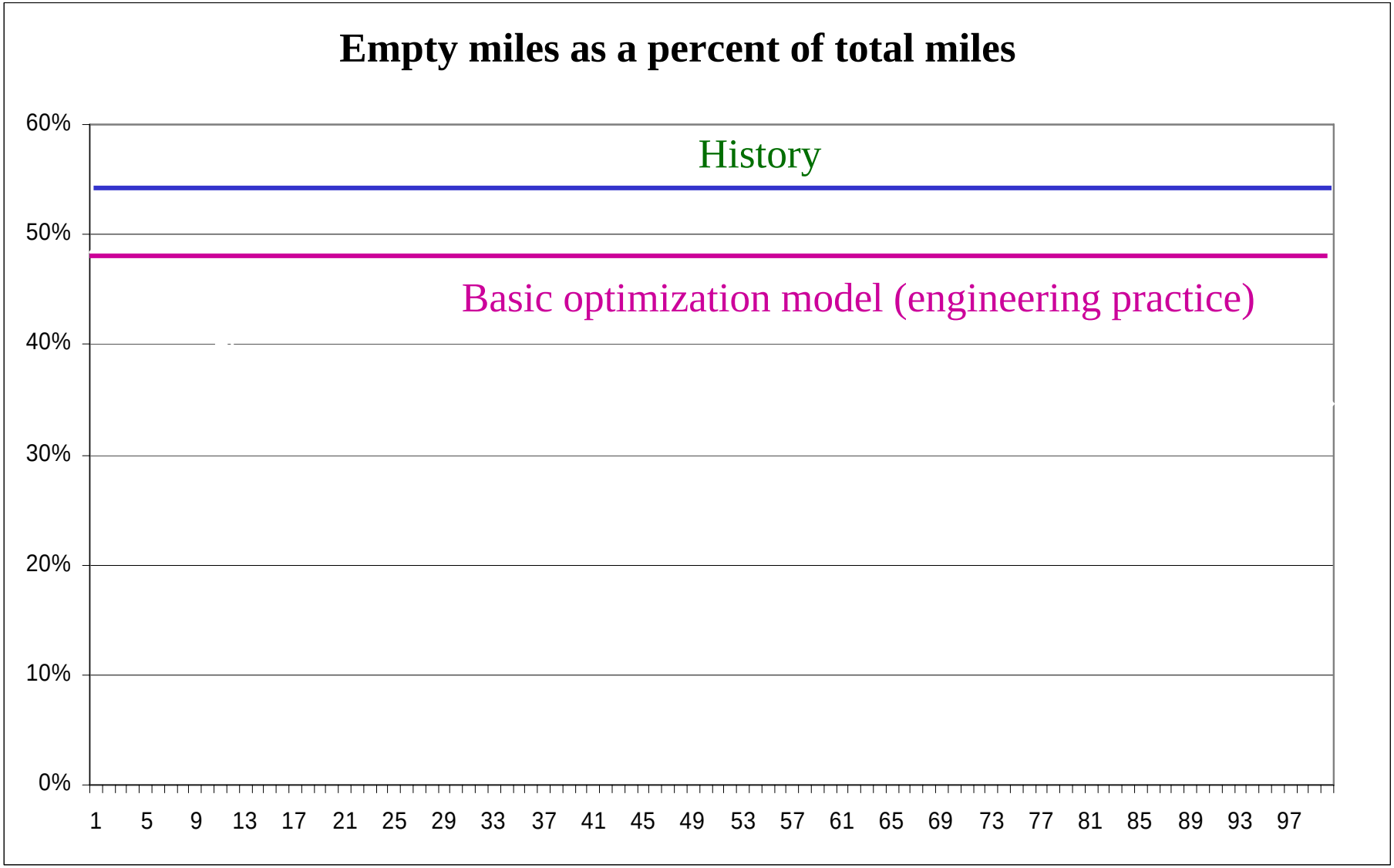
Multiattribute resources



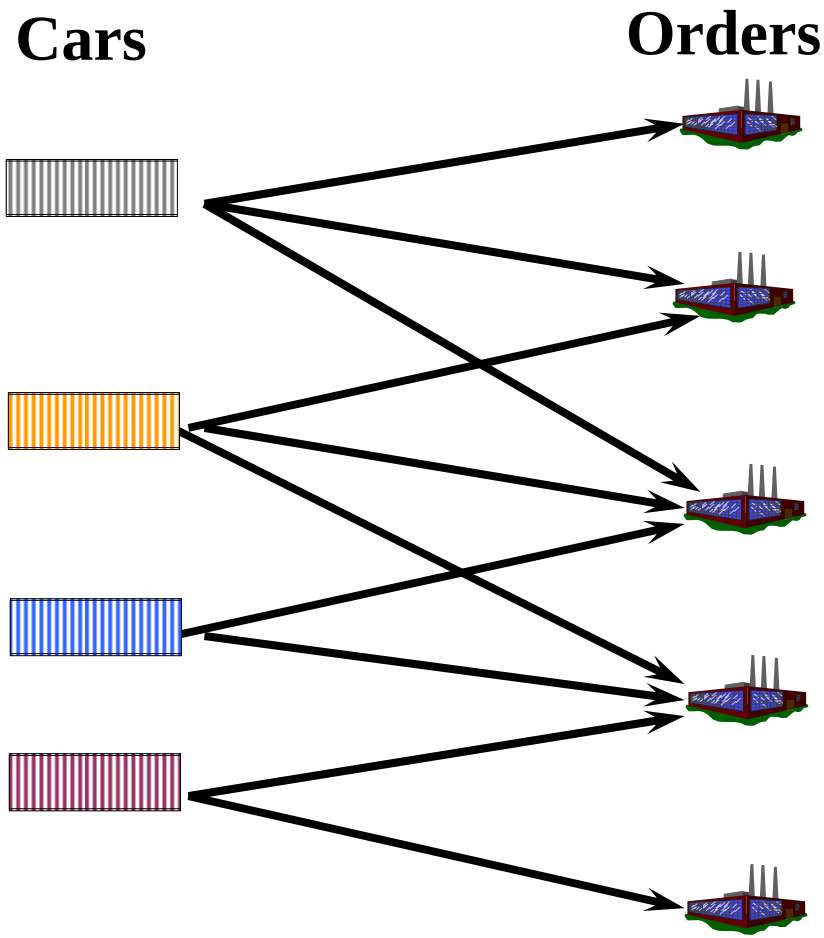
Engineering practice



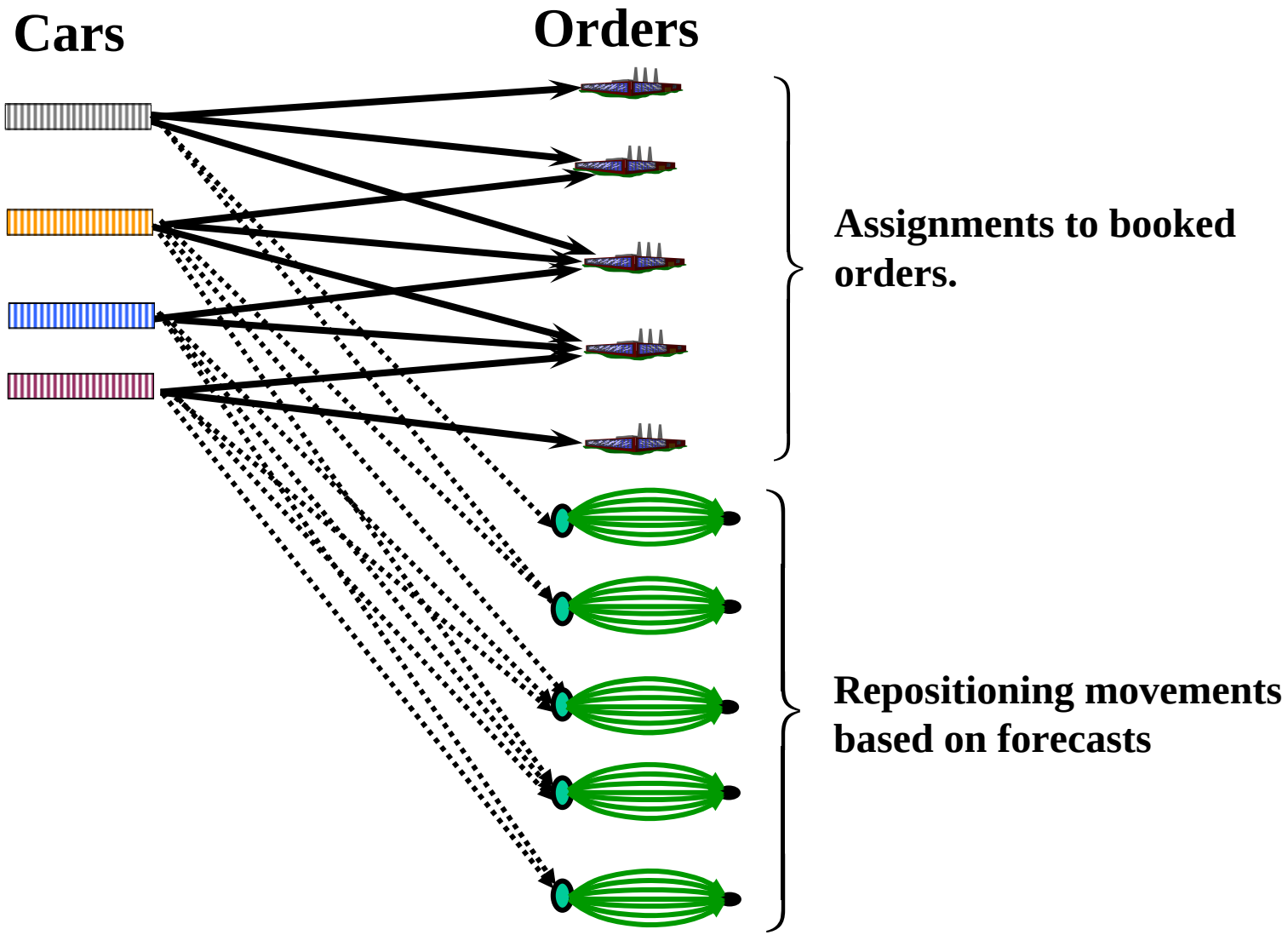
A car distribution problem



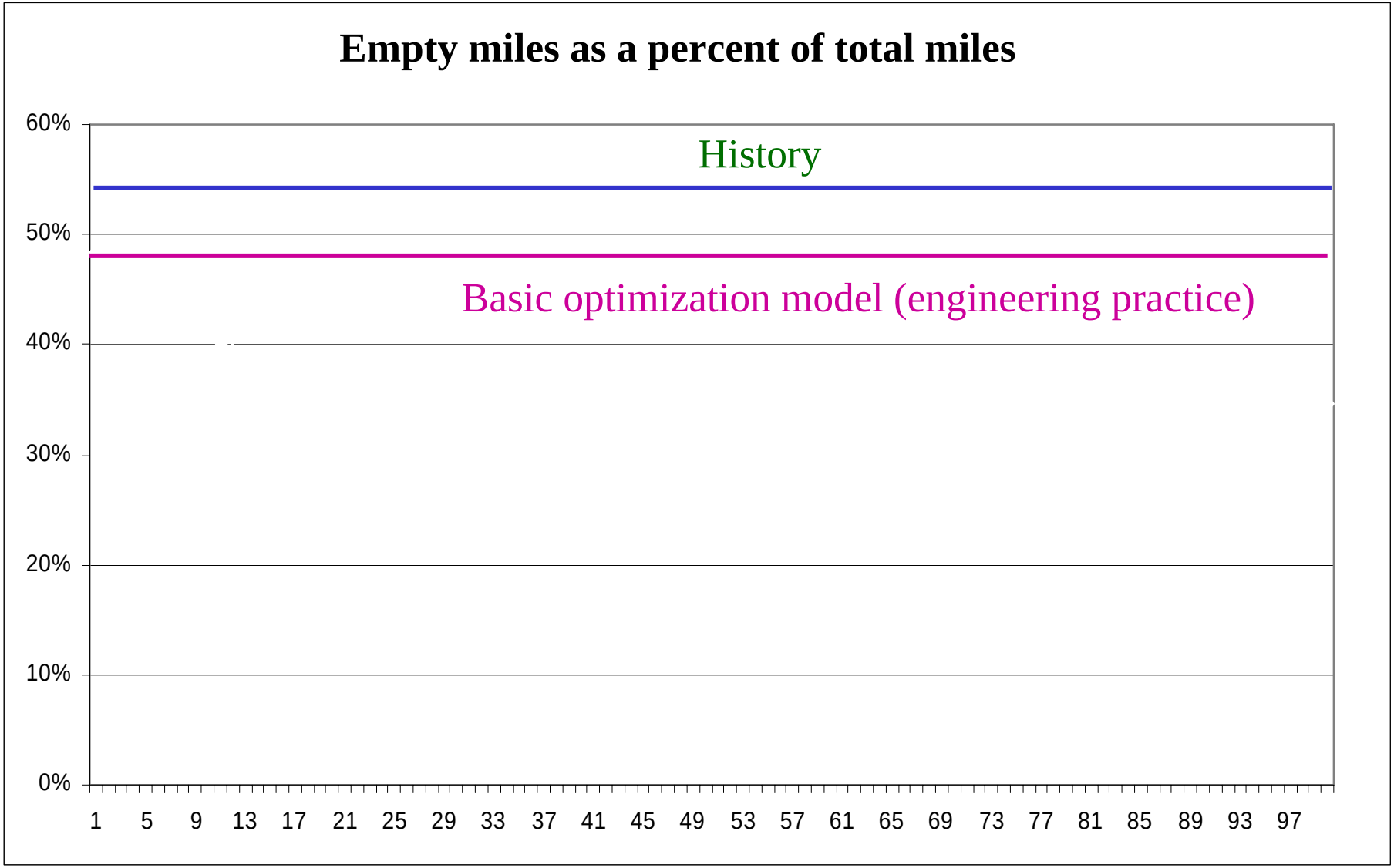
Engineering practice



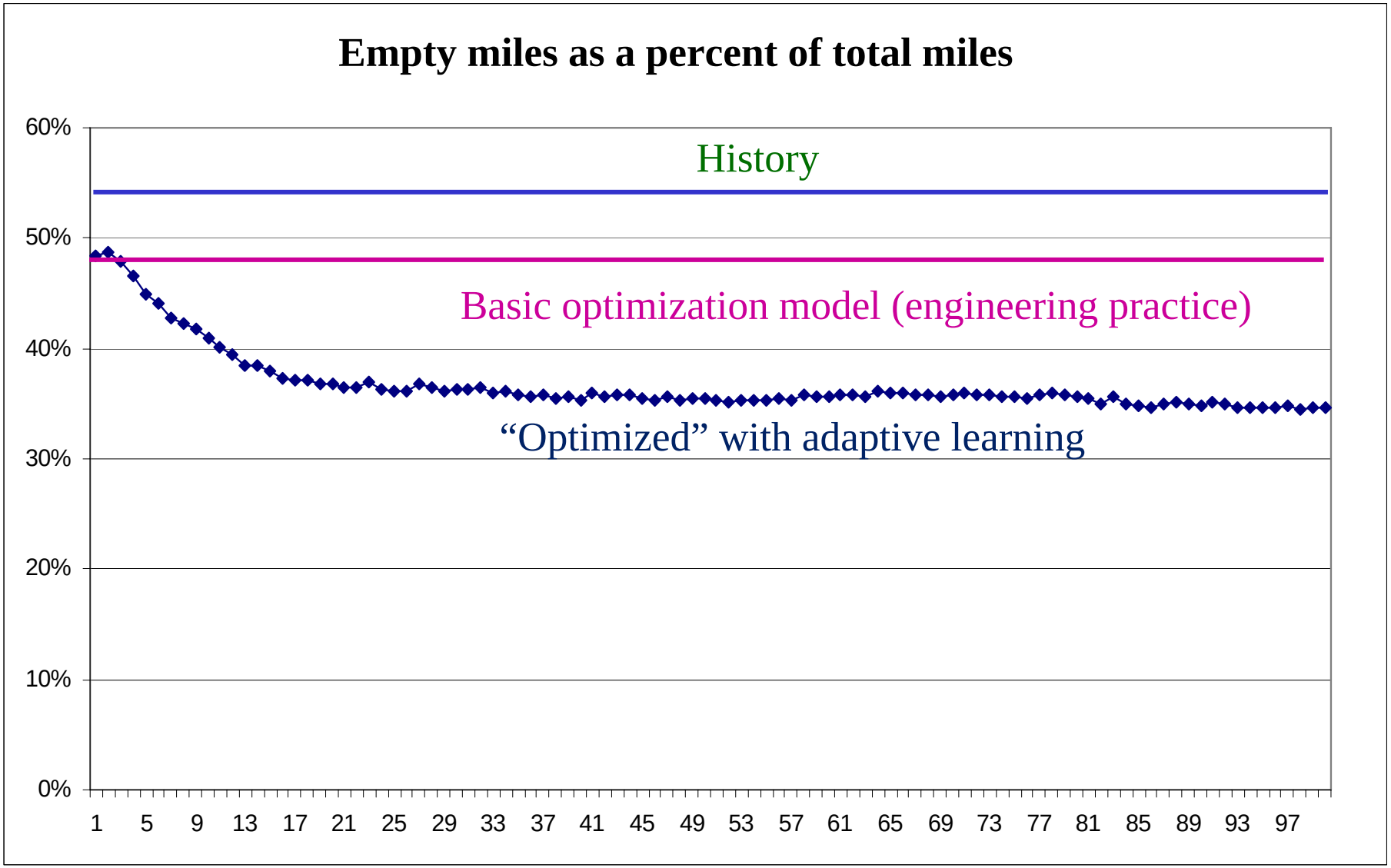
Engineering practice



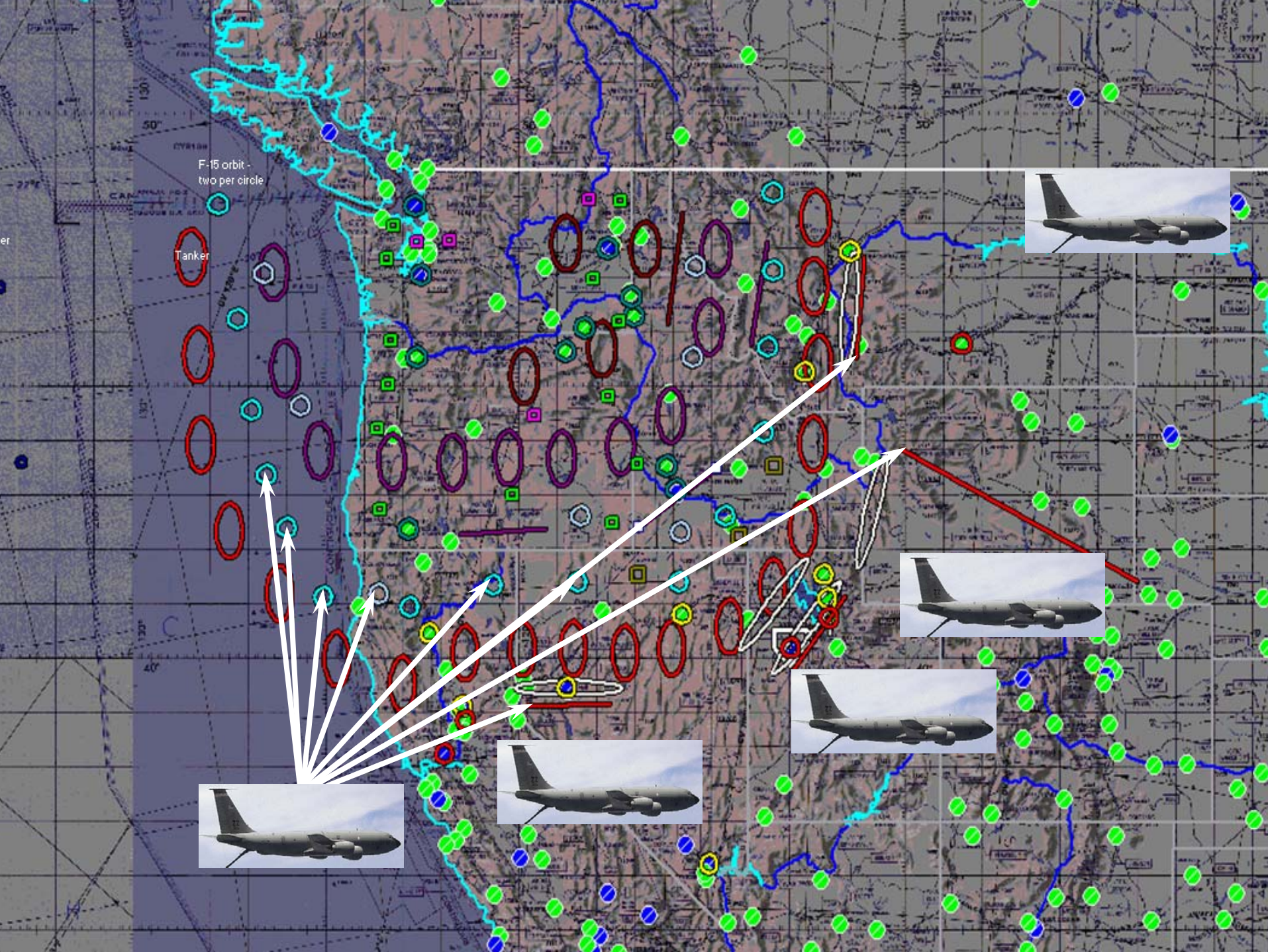
A car distribution problem

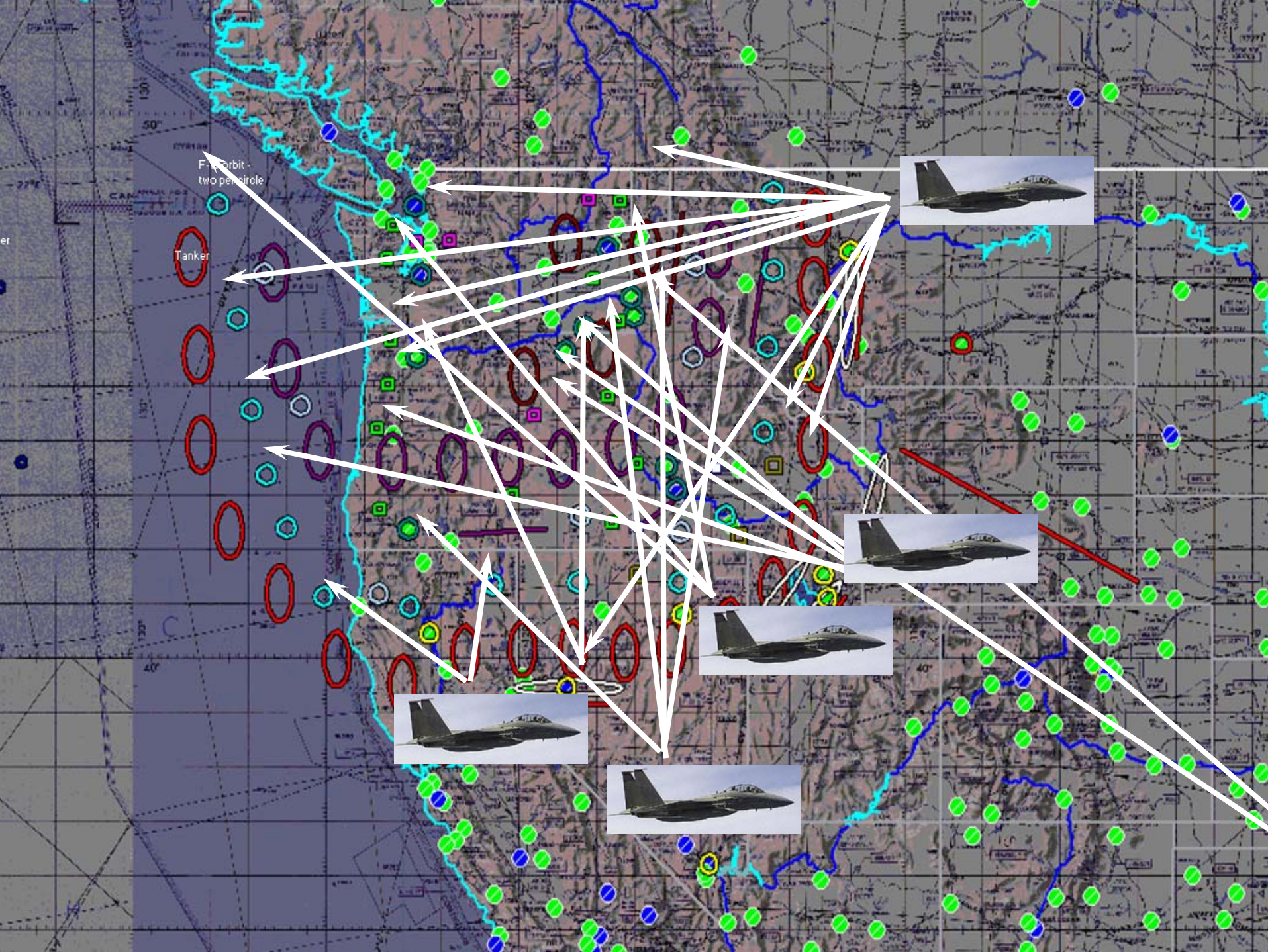


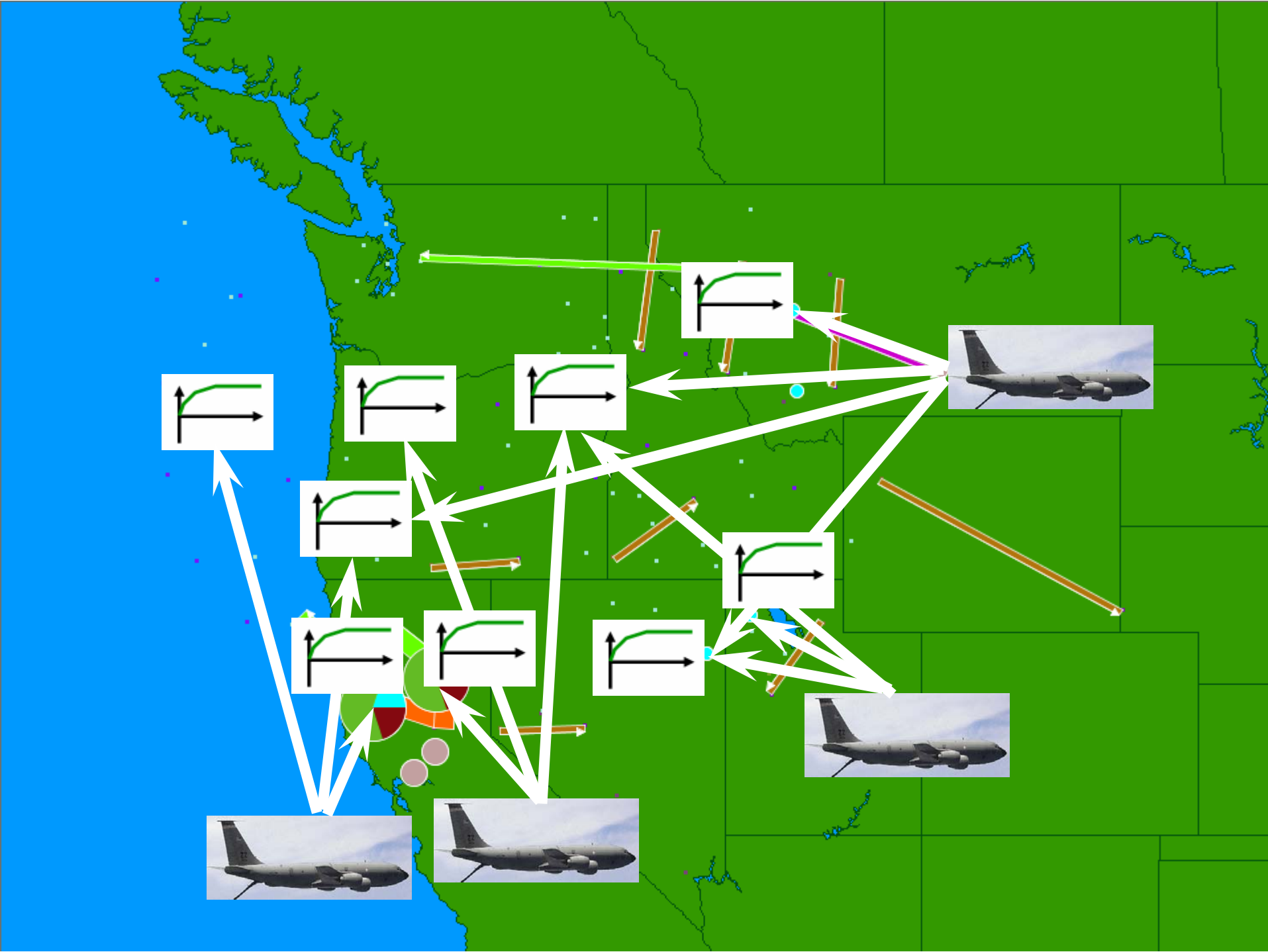
A car distribution problem

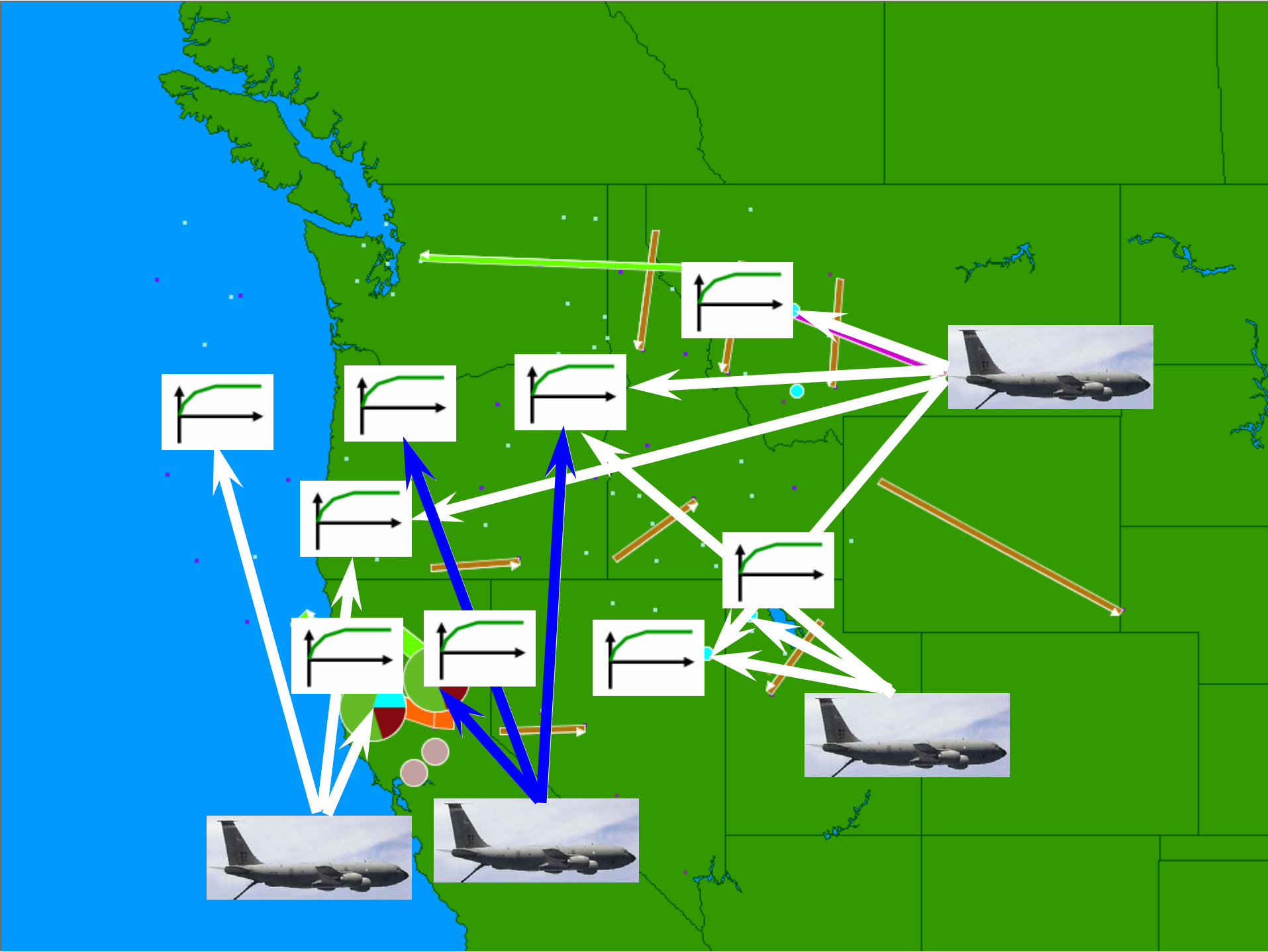


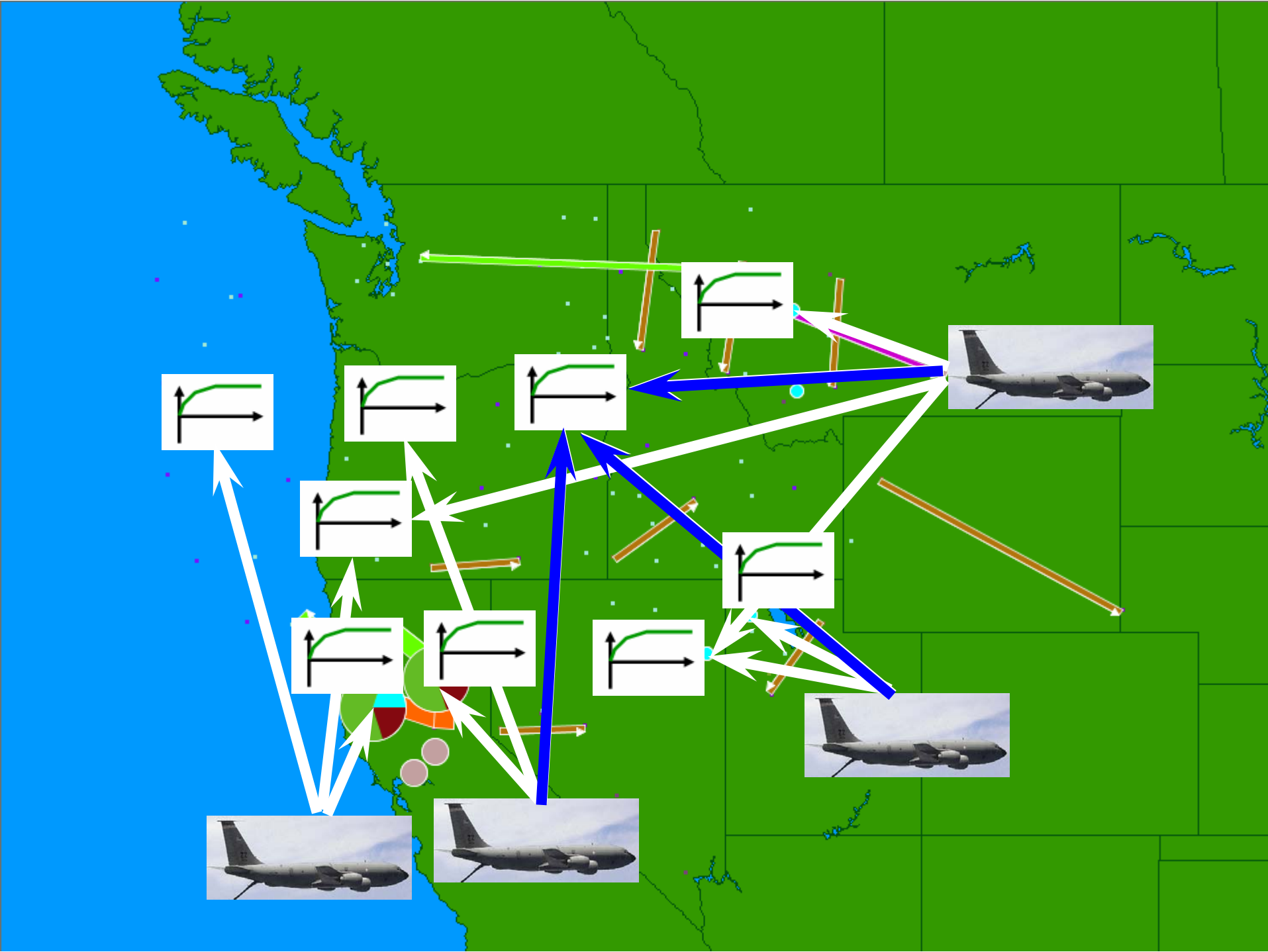

05
22

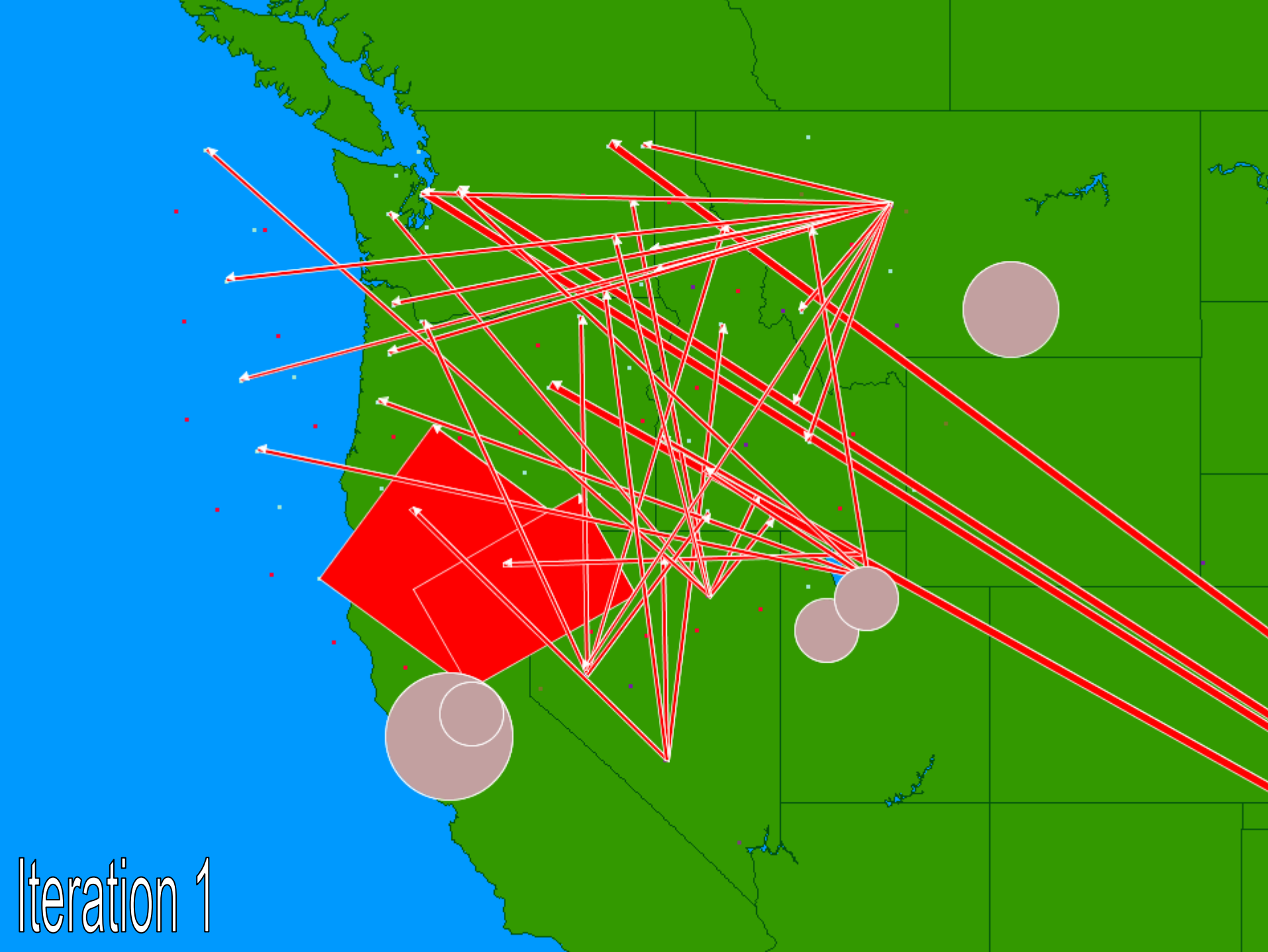

F-15 orbit -
two per circle
Tanker

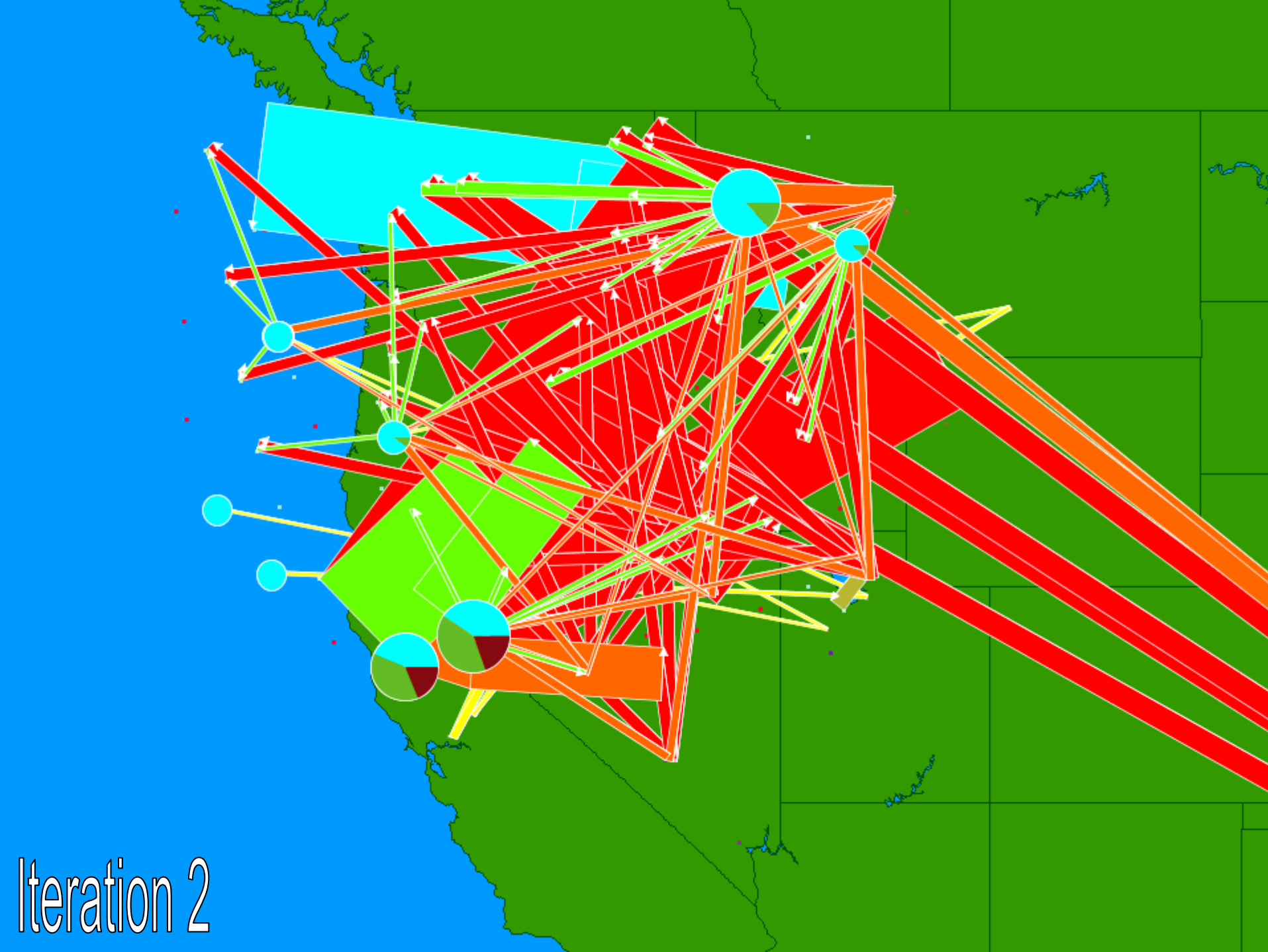

Tanker

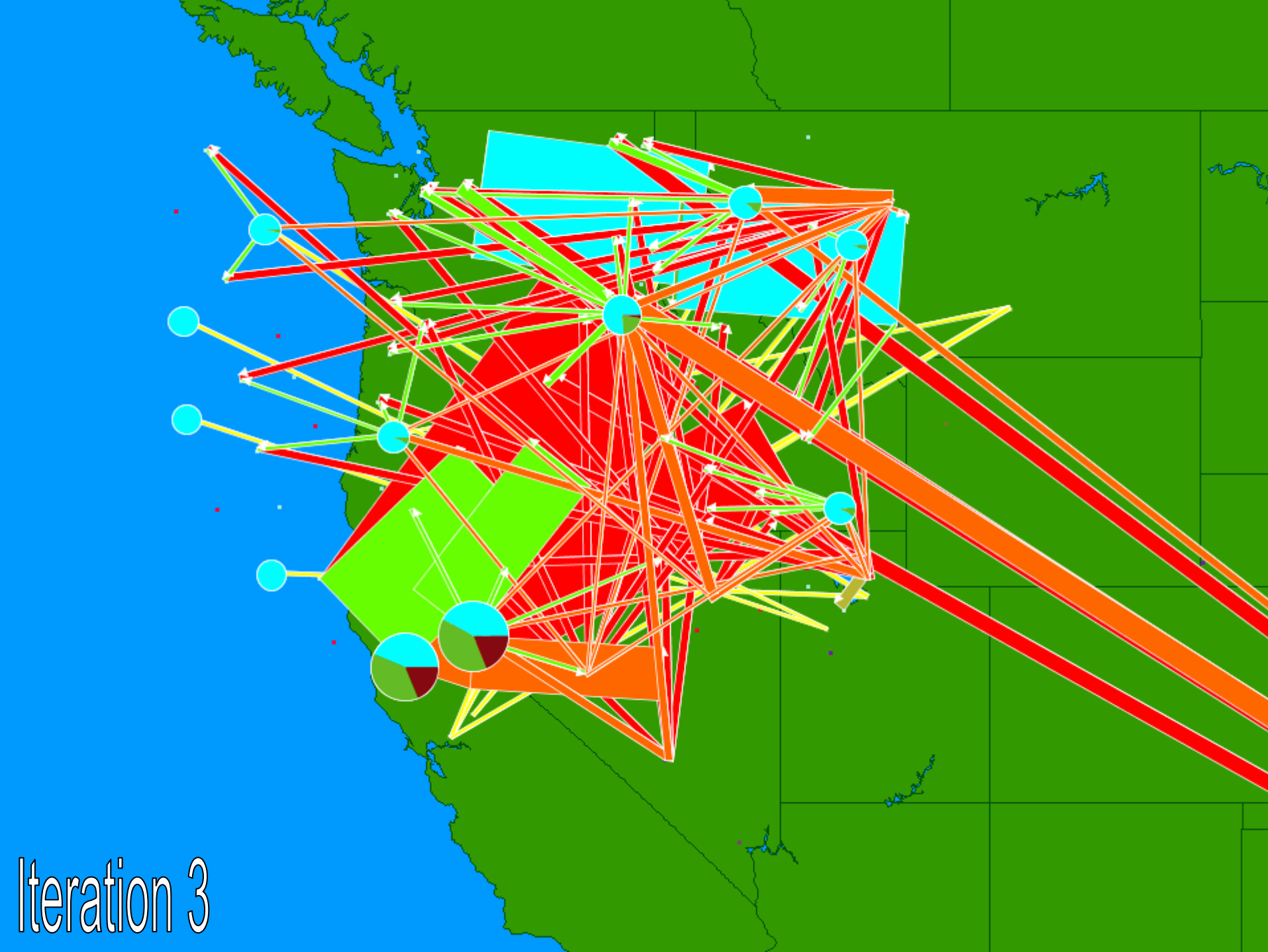


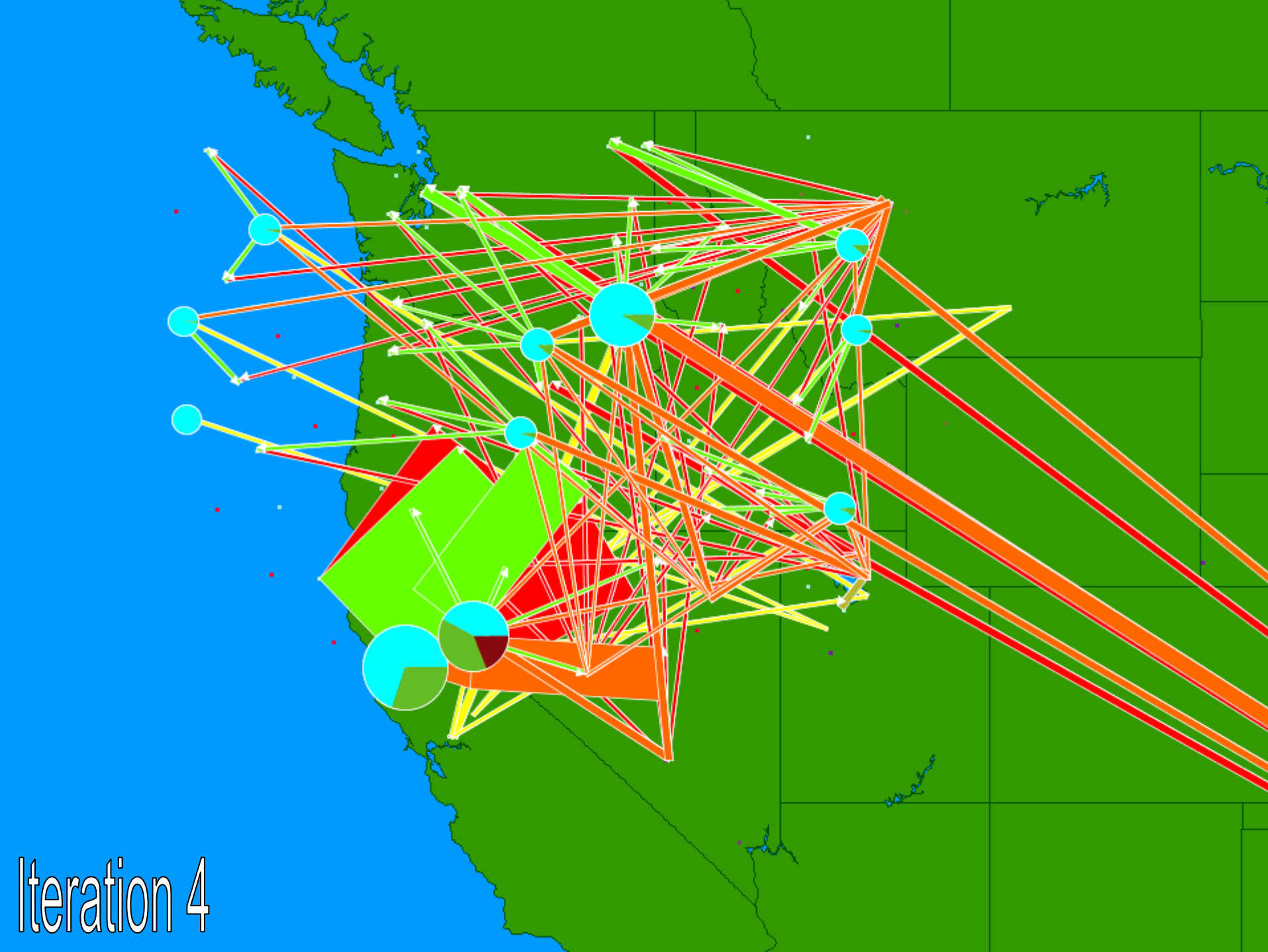


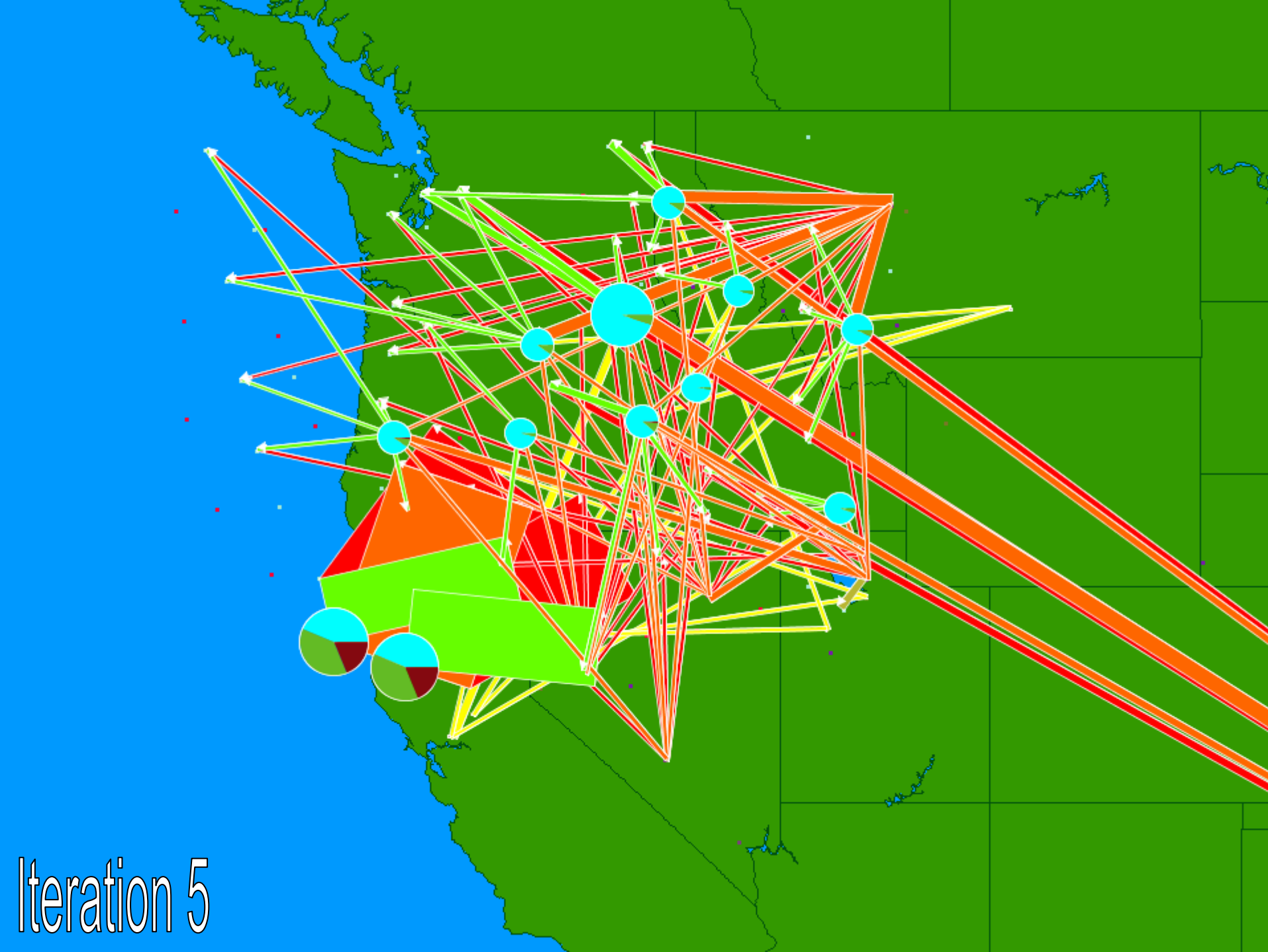


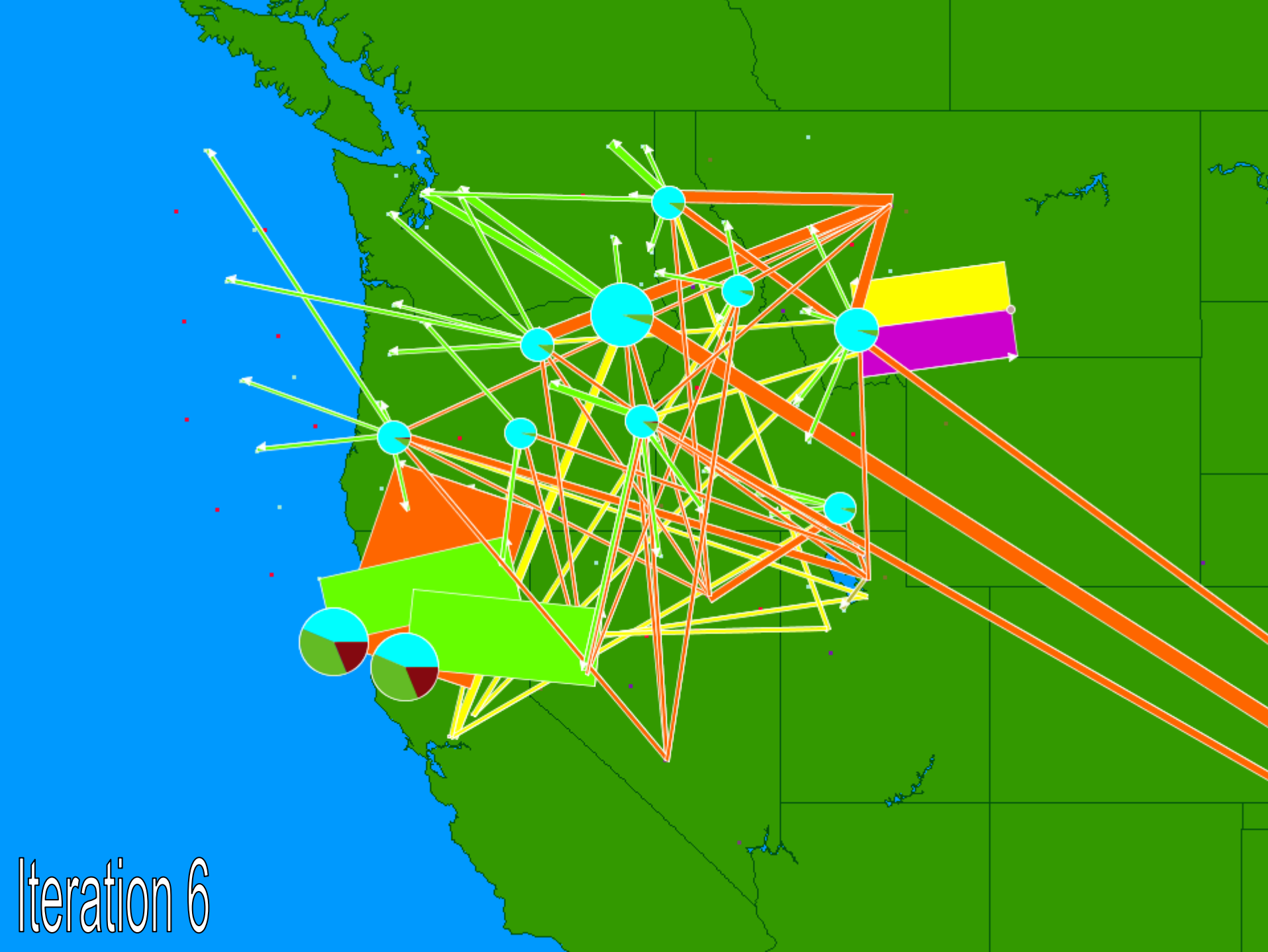

Iteration 1

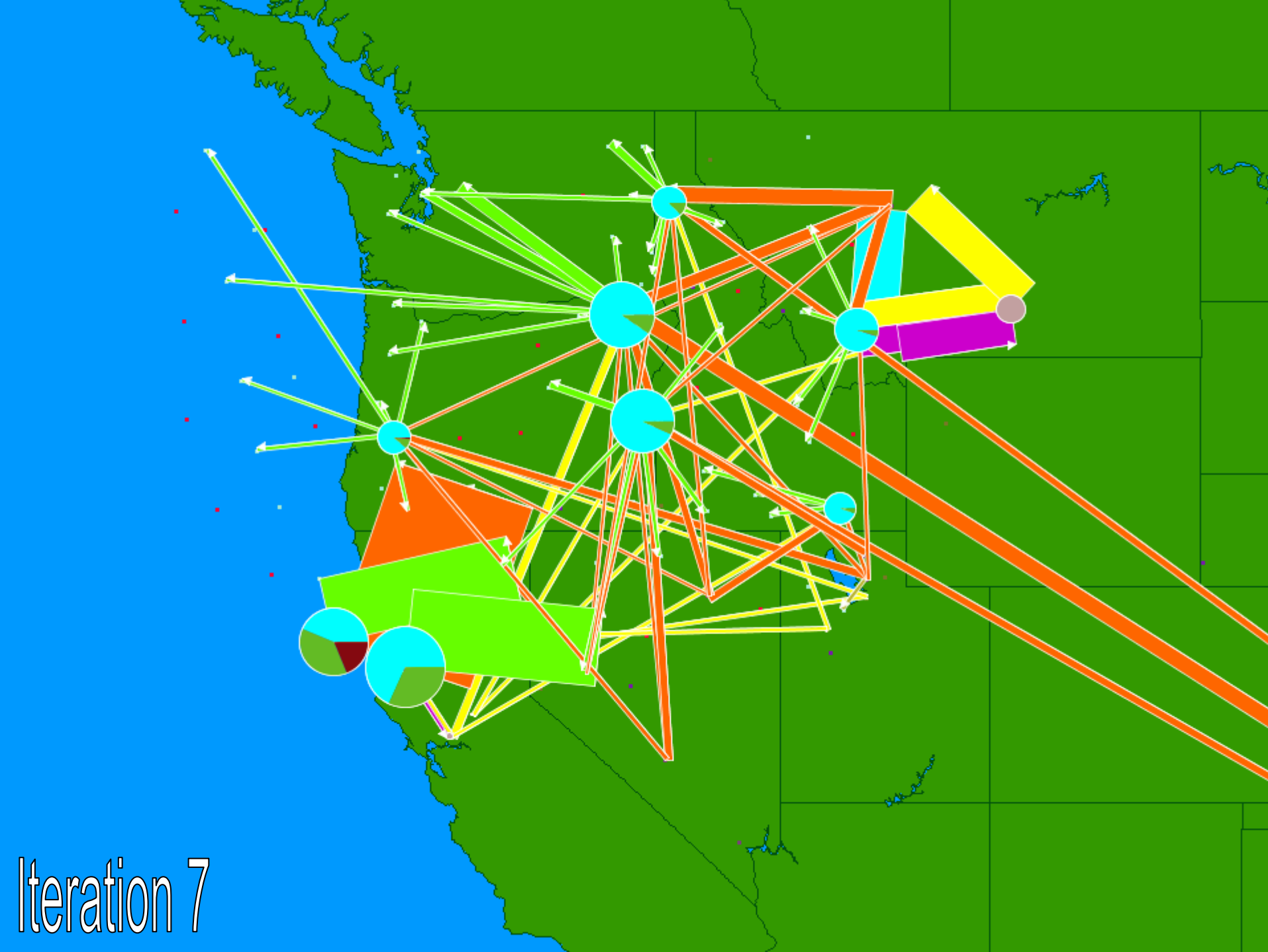

Iteration 2

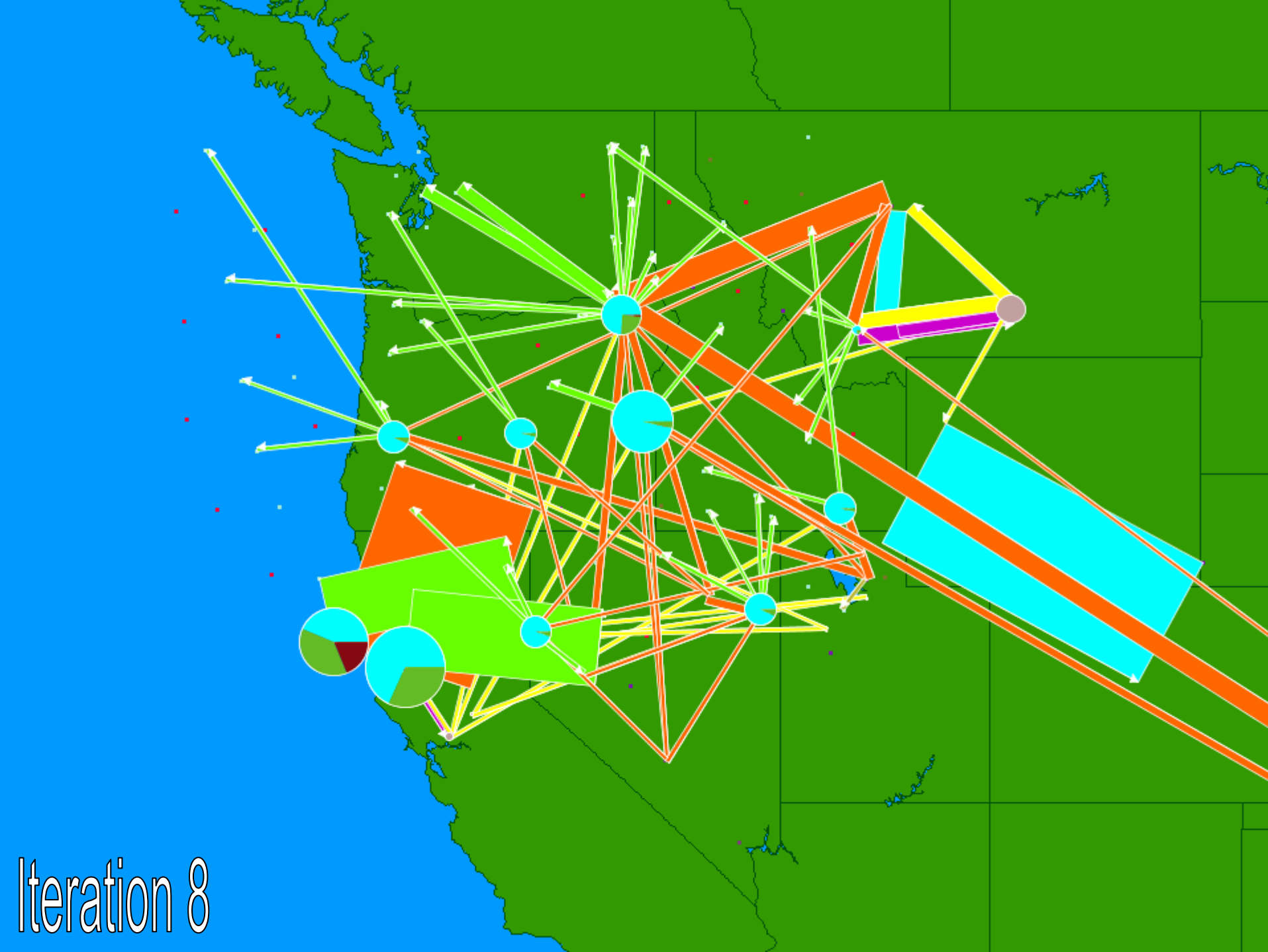

Iteration 3

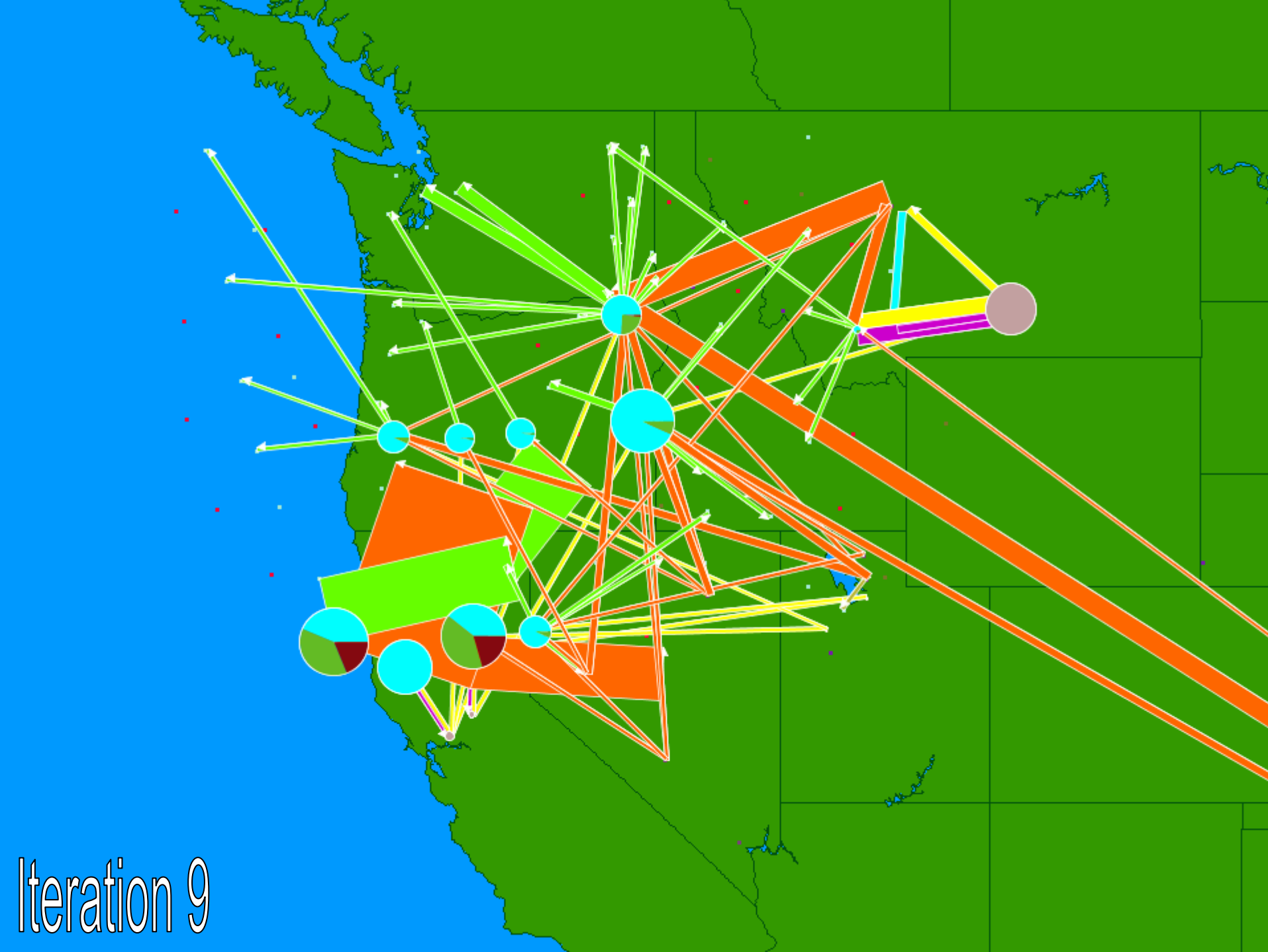

Iteration 4

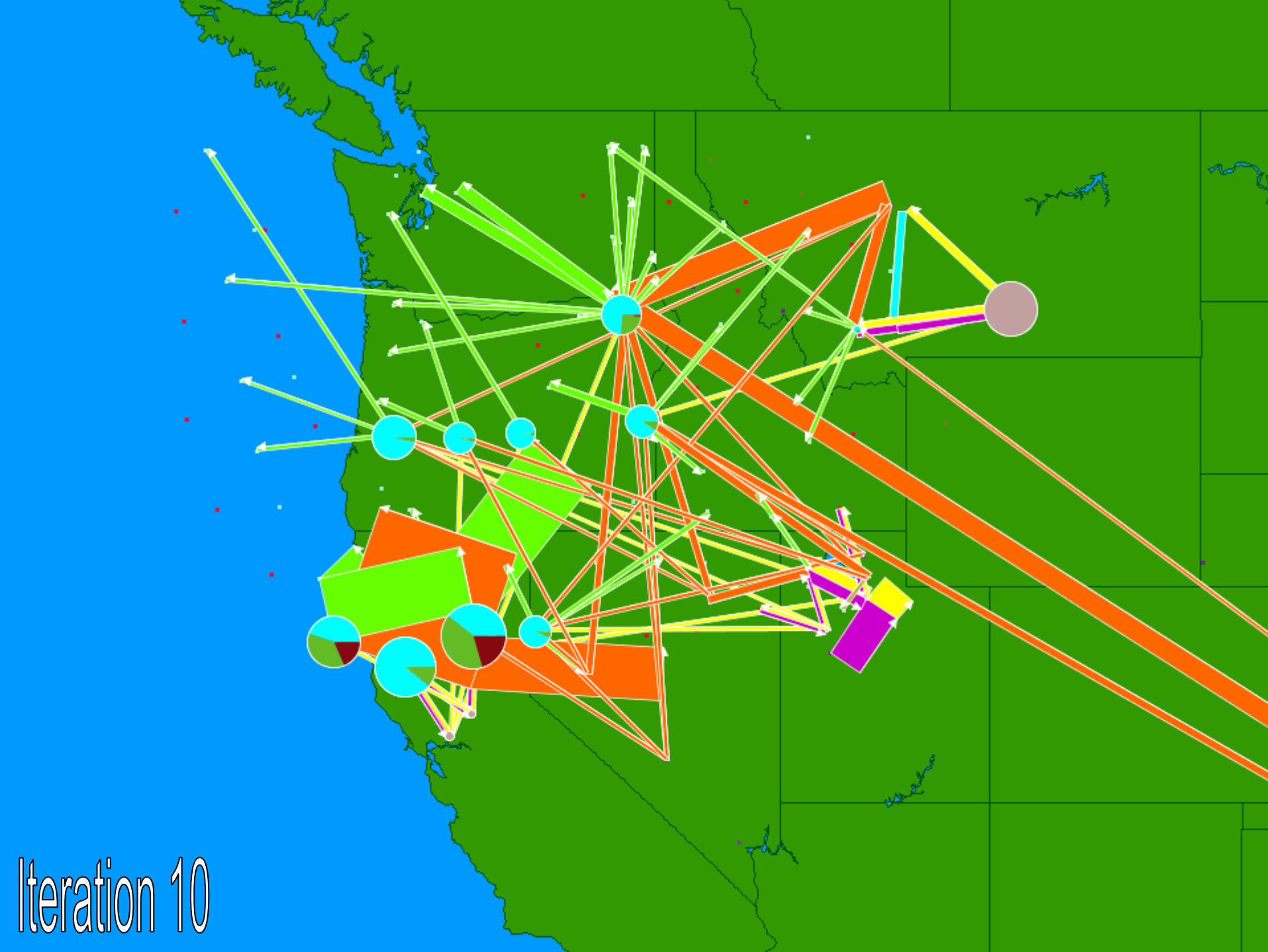

Iteration 5

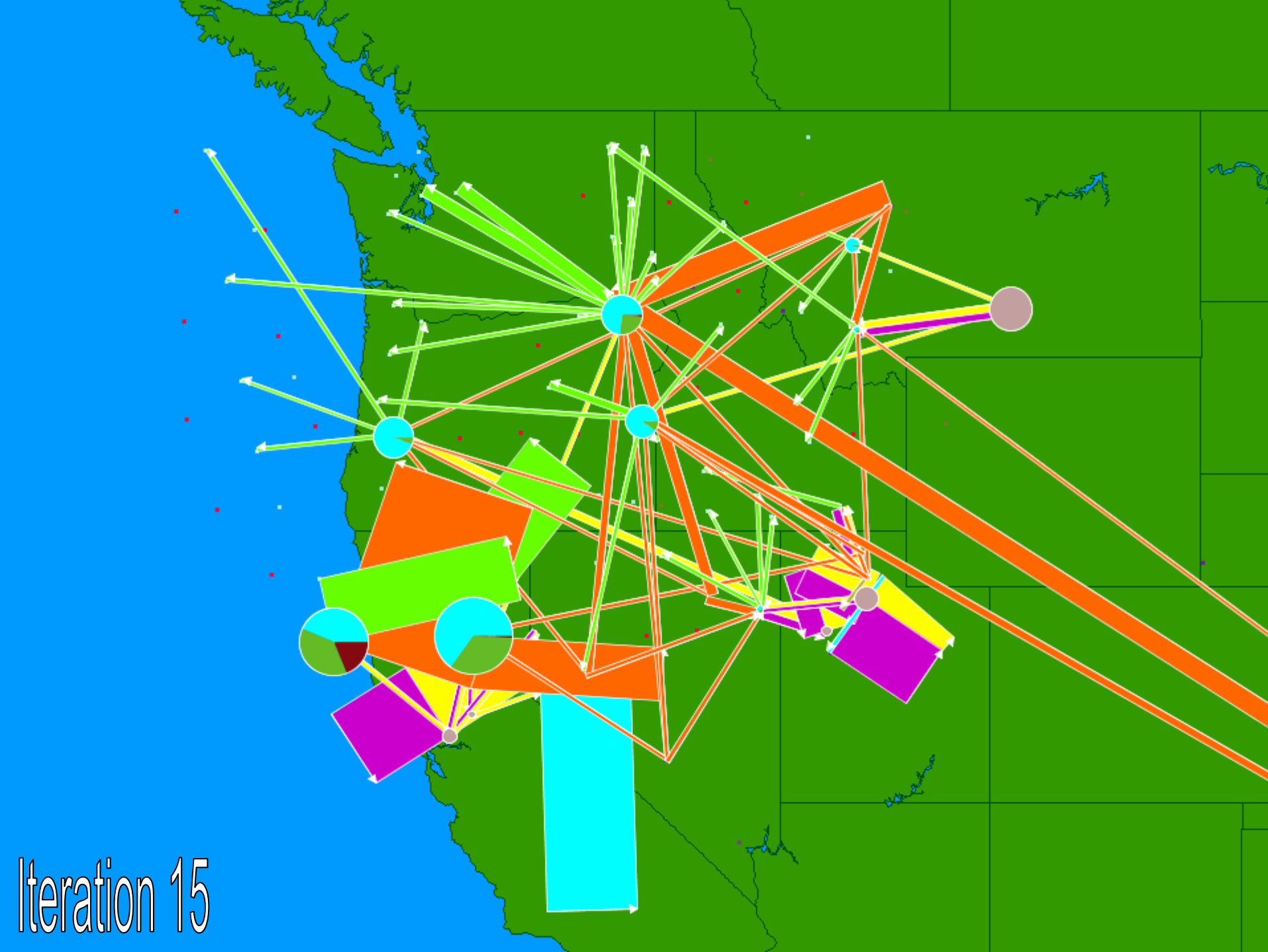

Iteration 6

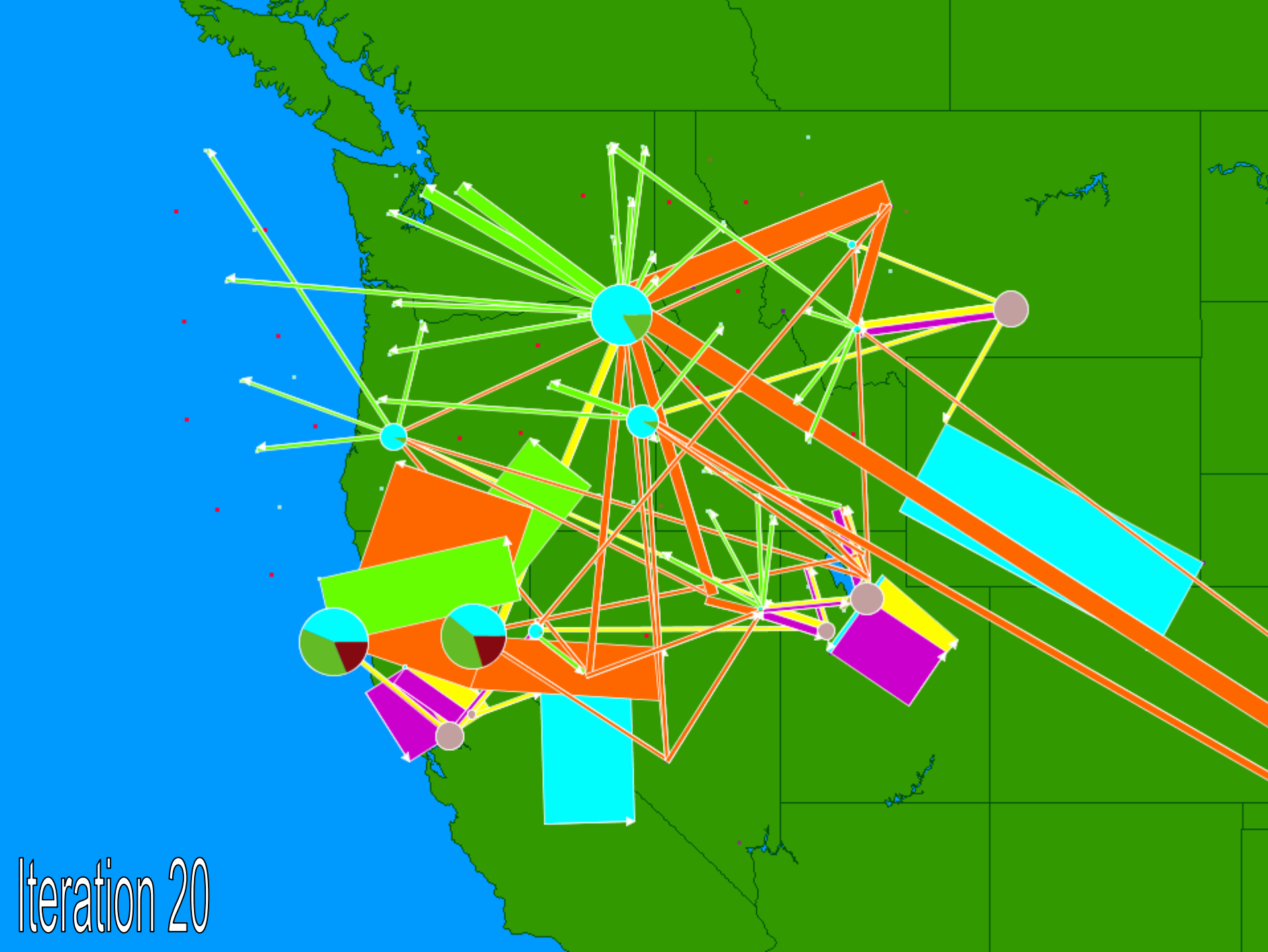

Iteration 7

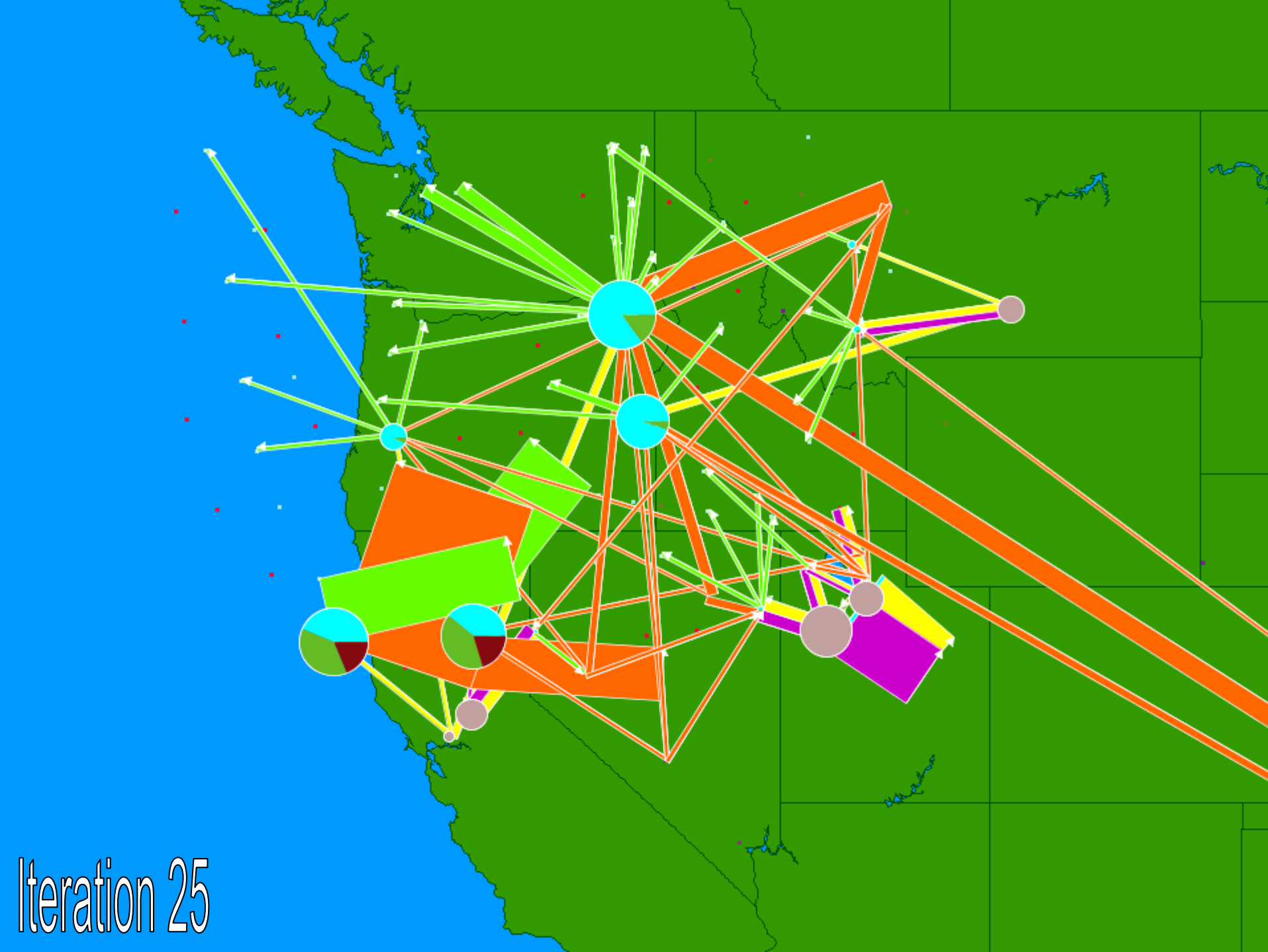

Iteration 8

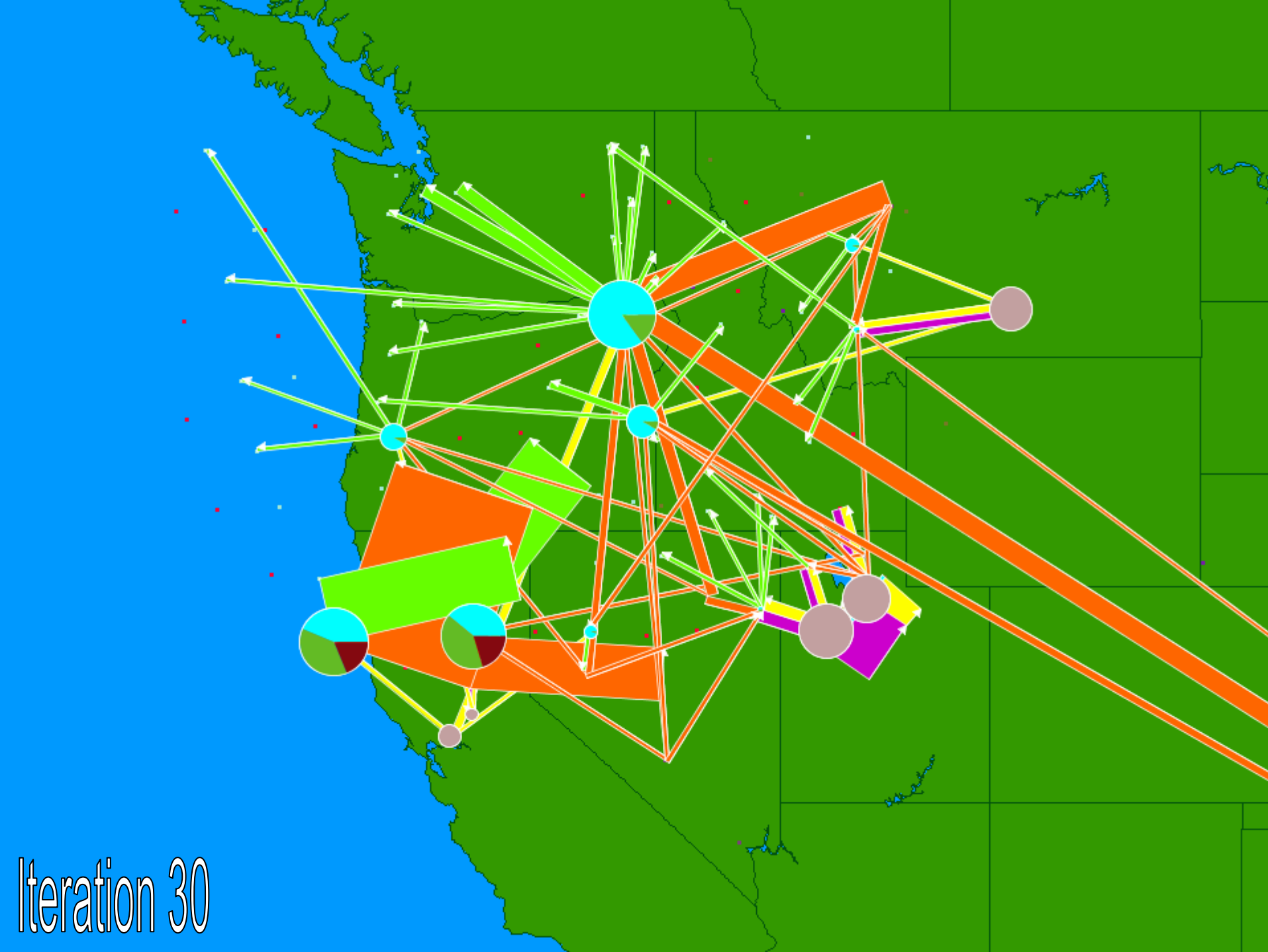

Iteration 9


Iteration 10


Iteration 15


Iteration 20


Iteration 25


Iteration 30

Tanker study