Simulation and Meta-heuristic Methods

G. Cornelis van Kooten
REPA Group
University of Victoria

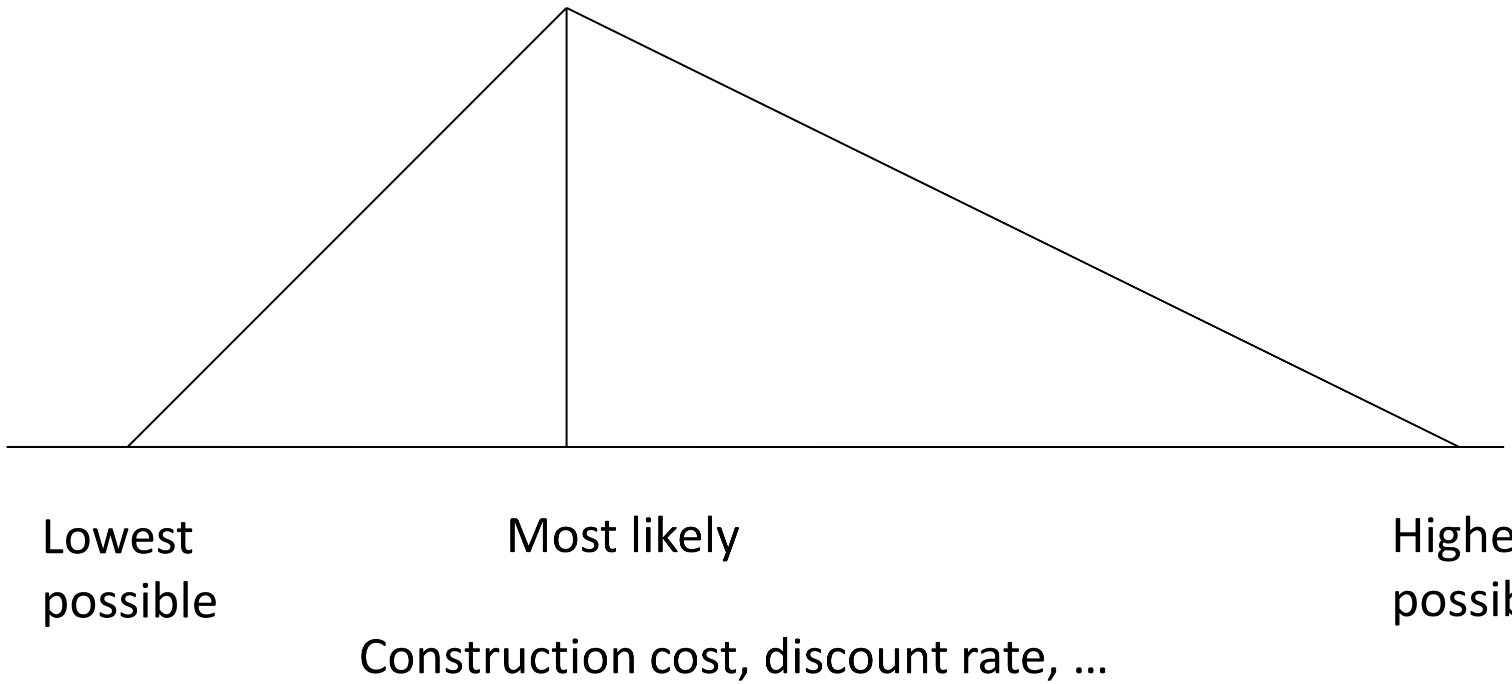
Simulation

- Monte Carlo simulation (e.g., cost-benefit analysis)
- Within a constrained optimization or optimal control model, simulation is done by changing parameter values
- Developing heuristics where optimization is not possible

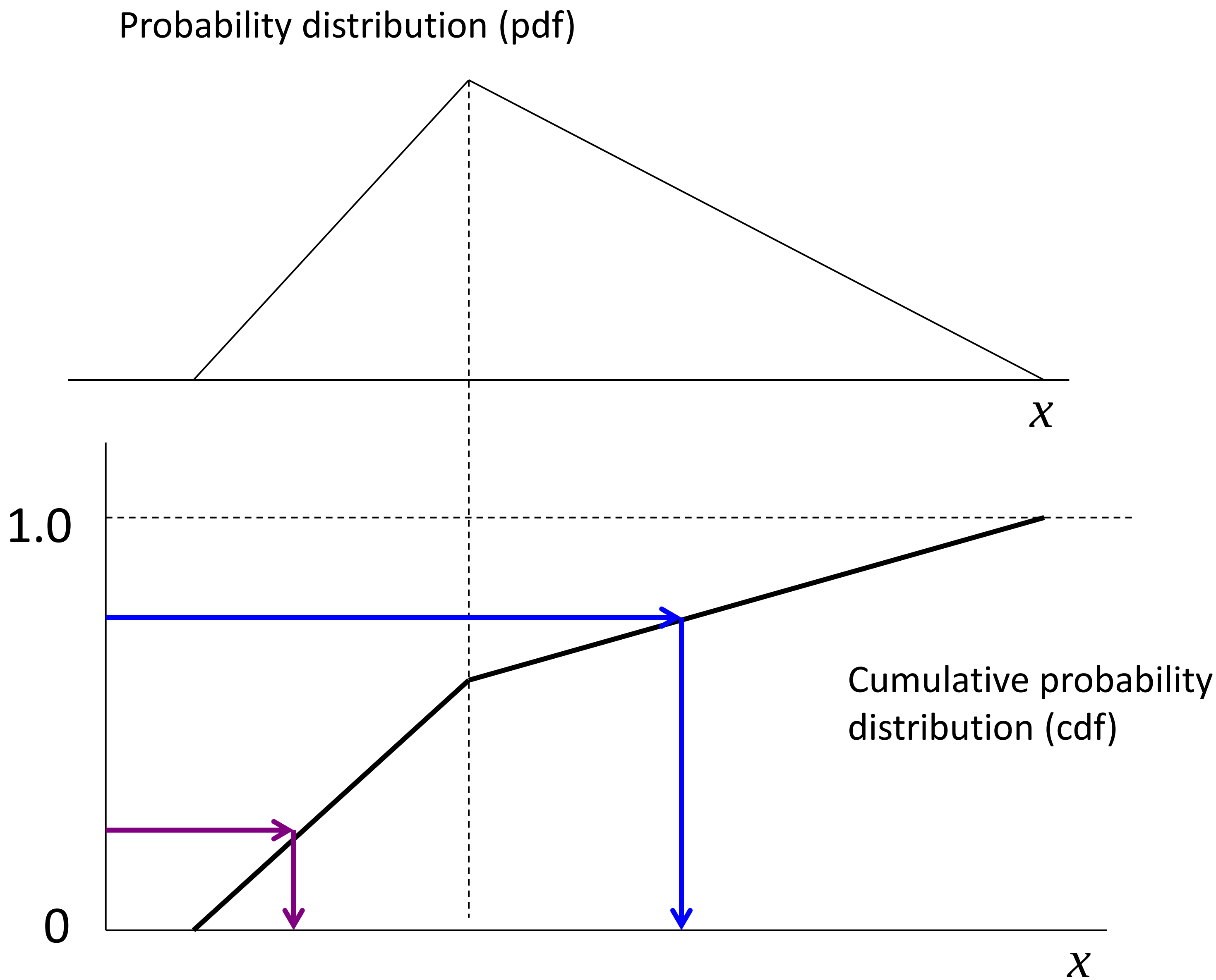
Stochastic Cost-Benefit Analysis: Monte Carlo Simulation

- Monte Carlo simulation involves sampling distributions, calculating the ‘number’ of interest (e.g., NPV, cost-benefit ratio), and getting the mean and standard deviation of that number
- Project evaluation is good example: We do not always know the costs, future benefits, etc.
- Sample unknowns from a triangular distribution

Triangular Distribution



Elicit above three numbers from ‘experts’ or from relevant literature



Choice of random number between 0 & 1 gives random values of x

Procedure

- Elicit information to construct triangular distributions for each variable that might be considered random or uncertain
- Iterations:

  1. For each distribution, obtain a random number in [0 1], find value of variable from cdf

  2. Calculate NPV and/or B-C ratio (retain result)

  3. Go to 1 and repeat loop n times
- Calculate a mean and standard deviation for NPV and B-C ratio
- Determine probability that NPV < 0 or B-C ratio < 1

Problem: If variables are correlated a joint probability distribution is required, and triangular distributions are not well suited to joint probabilities.

Problem with Optimization Models? Rational Expectations?

- How can we better deal with extremely large problems, complex dynamic processes, spatial considerations, and a substantial lack of information about the evolution of a system and expected future returns?
- How can we incorporate adaptive management (learning) into decision models?

Motivation:

- The need to explore and compare properties and results of alternative decision-making models
- Meta-heuristics are most common alternative to optimization

There are many, many alternative approaches.

Meta-heuristic Models

- There are times when it is impossible to find a solution to a constrained optimization problem
- It is possible to employ heuristics where optimization is not possible
- There is philosophical resistance to heuristic models among many, especially economists

Three types of heuristics

1. Tabu search (TS)

   - Employs memory of past solutions, but strategically

2. Randomized methods such as Monte Carlo simulation

   - Includes simulated annealing (SA) – ignores past memory & uses random searching

3. Genetic algorithms (GA)

   - Evolutionary with randomization

Branch and bound methods rely on rigid memory in contrast to TS: “A significant leap is required to conclude that randomization is preferred to intelligent design” (Glover).

Motivating Example of Tabu Search

- Example due to Glover & Laguna (1993)
- We want to arrange components in some order to maximize the insulation value of an object
- Begin by examining a starting point and iterating towards a solution

Rule: Cannot swap (x,y) or (y,x) pair for three iterations once a swap is made.

Iteration 0 (starting point)

Current Solution

2	5	7	3	4	6	1

Tabu structure

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Top 5 Candidates

Swap		Value
5	4	6
7	4	4
3	6	2
2	3	0
4	1	-1

*Swap 5 and 4 to increase insulation value by 6 units.

Objective Value = 10

Iteration 1

Current Solution

2	4	7	3	5	6	1

Top 5 Candidates

Swap		Value
3	1	2
2	3	1
3	6	-1
7	1	-2
6	1	-4

Tabu structure

	2	3	4	5	6	7
1						
2						
3						
4				3		
5						
6						

*Swap 3 and 1 to increase insulation value by 2 units. Cannot swap 4 and 5 for 3 iterations.

Objective Value = 16

Iteration 2

Current Solution

2	4	7	1	5	6	3

Top 5 Candidates

Swap		Value	
1	3	-2	T
2	4	-4	
7	6	-6	
4	5	-7	T
5	3	-9	

* Swaps 1&3 and 5&4 are tabu; choose 2&4 despite negative.

Tabu structure

	2	3	4	5	6	7
1		3				
2						
3						
4				2		
5						
6						

Objective Value = 18

Iteration 3

Current Solution

4	2	7	1	5	6	3

Tabu structure

	2	3	4	5	6	7
1		2				
2			3			
3						
4				1		
5						
6						

Top 5 Candidates

Swap		Value	
4	5	6	T*
5	3	2	
7	1	0	
1	3	-3	T
2	6	-6	

*Large value, so over-ride Tabu swap 5&4 by **aspiration criterion**

Objective Value = 14

Iteration 4

Current Solution

5	2	7	1	4	6	3

Tabu structure

	2	3	4	5	6	7
1		1				
2			2			
3						
4				3		
5						
6						

Top 5 Candidates

Swap		Value	
7	1	0	
4	3	-3	
6	3	-5	
5	4	-6	T
2	6	-8	

* Want to make this swap to keep search going.

Objective Value = 20

- So far we have only kept track of **recency** – how long since the last swap (we assumed a swap to be tabu for three iterations)
- Now introduce **frequency**, perhaps using penalties to discourage swaps/moves that occur with greater frequency in the past.
- Need to balance **intensification** (moves that appear good because they occur frequently) and **diversification** (encourage choices/ moves not made in the past)
- Notice that memory is selective and not rigid

Iteration xx

Current Solution

1	3	6	2	7	5	4

Tabu structure

	1	2	3	4	5	6	7
1				3			
2							
3	3						
4	1	5				2	
5		4		4			1
6					2		
7	2			3			

Recency

Frequency

Top 5 Candidates

Swap		Value	Penalized Value
1	4	3	3 T
2	4	-1	-6
3	7	-3	-3 *
1	6	-5	-5
6	5	-4	-6

Objective Value = 12

Tabu Search Problem Setup

Minimize	$c(x)$
Subject to	$x \in \boldsymbol{X}$

The objective function can be linear or nonlinear, as may the constraint set. The constraint set may contain logical conditions and interconnections that can best be specified verbally (a bit like fuzzy in that sense).

How does it work?

- Let's see how tabu search fits with other algorithms given the above discussion.
  - Neighborhood search: begin with a feasible solution and then we search in the neighborhood for a solution that yields a better value (see "pattern search" below*). In tabu search, neighborhoods are normally assumed to be symmetric
  - Descent method
  - Monte Carlo method (similar to earlier method)
- How does tabu search differ from these algorithms? History!

* Matlab's Genetic Algorithm & Direct Search toolbox has a 'patternsearch' function. (See below)

Neighborhood Search Method

Step 1 (Initialization)

(A) Select a starting solution $x_0 \in \boldsymbol{X}$ and set $x_{now} = x_0$

(B) Record the current best known solution by setting $x^* = x_{now}$ and define $c^* = c(x^*)$, where * refers to 'best'.

Step 2 (Choice and termination)

Choose a solution $x_{next} \in \boldsymbol{N}(x_{now})$. If the choice criteria employed cannot be satisfied by any member of $\boldsymbol{N}(x_{now})$ (hence no solution qualified to be x_{next}), or if other termination criteria apply (such as a limit on the total number of iterations), then method stops.

Step 3 (Update)

Re-set $x_{now} = x_{next}$, and if $c(x_{now}) < c^*$, perform Step 1(B). Then go to step 2.

Descent Method

Step 2 (Choice and termination)

Choose $x_{next} \in \boldsymbol{N}(x_{now})$ to satisfy $c(x_{next}) < c(x_{now})$ and terminate if no such x_{next} can be found.

Monte Carlo Method

Step 2 (Choice and termination)

(A) Randomly select x_{next} from $\boldsymbol{N}(x_{now})$.

(B) If $c(x_{next}) \leq c(x_{now})$ accept x_{next} (and proceed to the Update Step)

(C) If $c(x_{next}) > c(x_{now})$ accept x_{next} with a probability that decreases with increases in the difference $c(x_{next}) - c(x_{now})$. If x_{next} is not accepted on the current trial by this criterion, return to Step 2(A).

(D) Terminate by a chosen cutoff rule.

Tabu Search Method

Step 1 (Initialization)

Start with the same initialization used by Neighborhood Search and with the history record H empty.

Step 2 (Choice and termination)

Determine the $CandidateN(x_{now})$ as a subset of $\boldsymbol{N}(H, x_{now})$. Select x_{next} from $CandidateN(x_{now})$ to minimize $c(H, x_{now})$ over this set. (x_{next} is called a highest evaluation element of $CandidateN(x_{now})$.) Terminate by a chosen iteration cutoff rule.

Step 3 (Update)

Perform the update for the Neighborhood Search Method, and additionally update the history record H.

Traveling Salesman Problem: TS and GA

Traveling Salesman Problem (TSP): Starting from a node, the salesman is required to visit every other node only once in a way that the total distance covered is minimized. Mathematically:

Thanks are due to Sachin Jayaswal, Management Science, University of Waterloo. Material here is from a paper in Applied Optimization MSCI 703. Viewed 18 March 2008 at:
http://www.eng.uwaterloo.ca/~sjayaswa/projects/MSCI703_project.pdf

$$\text{Min} \sum_{ij} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i} x_{ij} = 1, \; \forall j \neq i$$

$$\sum_{j} x_{ij} = 1, \; \forall i \neq j$$

$$u_1 = 1$$

$$2 \leq u_i \leq n, \; \forall i \neq 1$$

$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}), \; \forall i \neq 1, \forall j \neq 1$$

$$u_i \geq 0, \; \forall i$$

$$x_{ii} \in \{0,1\}, \; \forall i, j$$

The 3rd, 4th, 5th and 6th constraints together are called MTZ constraints and are used to eliminate any sub tour in the solution. BUT they add to the number of variables that need to be solved.

Tabu Search Solution

1. **Solution Representation**: A feasible solution is represented as a sequence of nodes, each node appearing only once and in the order it is visited. The first and the last visited nodes are fixed to 1. The starting node is not specified in the solution representation and is always understood to be node 1.

3	5	3	4	7	6	8	1

Solution Representation

Tabu Search Solution (cont)

2. **Initial Solution**: A good feasible, yet not-optimal, solution to the TSP can be found quickly using a greedy approach. Starting with the first node in the tour, find the nearest node. Each time find the nearest unvisited node from the current node until all the nodes are visited.

Tabu Search Solution (cont)

3. **Neighborhood**: Any other solution obtained by a pairwise exchange of any two nodes in the solution. Guarantees that any neighbor-hood to a feasible solution is always feasible (i.e, no sub-tour). If we fix node 1 as the start and the end node, for a problem of N nodes, there are such ${}^{N-1}C_2$ neighborhoods to a given solution. At each iteration, the neighborhood with the best objective value (minimum distance) is selected.

Tabu Search Solution (cont)

4. **Neighborhood solution obtained by swapping the order of visit of cities 5 and 6**

3	5	3	4	7	6	8	1

Tabu Search Solution (cont)

5. **Tabu List**: To prevent the process from cycling in a small set of solutions, some attribute of recently visited solutions is stored in a Tabu List, which prevents their occurrence for a limited period. Attribute used is a pair of nodes that have been exchanged recently. A Tabu structure stores the number of iterations for which a given pair of nodes is prohibited from exchange as illustrated in the next Figure.

Tabu structure

	1	2	3	4	5
1			5		4
2				3	
3		1			2
4		5			
5		2		4	

Recency

Frequency

Tabu Search Solution (cont)

6. **Aspiration criterion**: A tabu may be too powerful, prohibiting attractive moves even when there is no danger of cycling, or they may lead to an overall stagnation of the search process. Thus, it may become necessary to revoke a tabu at times. The criterion used here is to allow a tabu move if it results in a solution with an objective value better than that of the current best-known solution.

7. **Diversification**: Quite often the process gets trapped in a local optimum. To search other parts of the solution space (to look for the global optimum), it is necessary to diversify the search into new regions. Frequency information is used to penalize non-improving moves by assigning a larger penalty (frequency count adjusted by a suitable factor) to swaps with greater frequency counts. This diversifying influence is allowed to operate only on occasions when no improving moves exist.

Additionally, if there is no improvement in the solution for a pre-determined number of iterations, frequency information can be used for a pairwise exchange of nodes that have been explored for the least number of times in the search space, thus driving the search process to areas that are largely unexplored so far.

8. **Termination criteria**: The algorithm terminates if a pre-specified number of iterations is reached

Simulated Annealing Solution

Starts the same as TS

1. **Neighborhood**: At each step, a neighborhood solution is selected by an exchange of a randomly selected pair of nodes. The randomly generated neighbor solution is selected if it improves the solution, else it is selected with a probability that depends on the extent to which it deteriorates from the current solution.

2. **Termination criteria**: The algorithm terminates if it meets any one of the following criteria:

   a. It reaches a pre-specified number of iterations.

   b. There is no improvement in the solution for last pre-specified number of iterations.

   c. Fraction of neighbor solutions tried that is accepted at any time reaches a pre-specified minimum.

The maximum number of iterations is kept large enough to allow the process to terminate either using criterion b or c.

Genetic Algorithms

- A good tutorial can be found at:

http://www.geneticprogramming.com/Tutorial/index.html

Matlab has a ‘Genetic Algorithm and Direct Search Toolbox’ that explains GA and provides a method of solving a function using GA

The same toolbox has a pattern search method.

Pattern Search Method

Matlab’s ‘Genetic Algorithm and Direct Search’ toolbox enables minimizing any function (written as a .m file) subject to linear inequality and equality constraints

[xm fval, exitflag, output] = patternsearch(@fun, x0, A, b, Aeq, beq, lb, ub, options)

Among others, the genetic algorithm is one option for solving the pattern search problem.

COIN-OR

- Operations research initiative to provide public, open-source software for anyone to use (http://www.coin-or.org)
- Written in C++
- Link with GAMS is available:

https://projects.coin-or.org/GAMSlinks

- Check it out!!

Weighted learning model: A type of TS

- Used in Game Theory
- Method uses frequency, but not recency
- Example compares SDP with a weighted learning model
- EWA refers to ‘experienced-weighted attraction’

Eiswerth, M.E. & G.C. van Kooten, 2007. Dynamic Programming and Learning Models for Management of a Nonnative Species, Can J of Agric Econ 55: 485-98.

Objective Function

$$Max\sum_{t=0}^{T-1}\rho^t\left(R(x_t)-c(k_{t)}\right)+\rho^T S(x_T)$$

$where:$

$R =$ net returns function (assumed fixed)

$x_t =$ extent of invasive species infestation at time t

$c =$ cost function

$k_t =$ choice of technology for invasive species control

$S(x_T) =$ value of land in period T

$\rho =$ discount factor

Equation of Motion

$$x_{t+1} = g(x_t, k_t) + \varepsilon_{t,k}$$

where:

$x =$ extent of invasive species infestation

$k =$ choice of technology for controlling invasive

$\varepsilon =$ a random variable with normal distribution

Fundamental SDP Equation

Bellman's recursive equation :

$$V_t(x_t, k_t) = \underset{k_1, k_2, \ldots, k_{T-1}}{\text{Max}} \left\{ \mathrm{E}\left[R(x_t, k_t) - c(k_t)\right] + \rho \sum_{j=1}^{M} p(i, j, k_t) V_{t+1}(x_{t+1}) \right\}$$

where

R = per-acre net revenue exclusive of control costs $c(k_t)$

$p(i, j, k_t)$ = probability that an invasion of state i in period t will evolve to state j by period (t + 1), given choice of option k in period t

M = number of discrete states

Learning Models: Payoffs and Attractions

The average payoffs are termed the "attractions" to strategy *s* by time period τ (denoted as $A_{\tau,s}$) and are calculated according to:

$$A_{\tau,s} = \frac{\sum_{t=1}^{\tau} NR_{t,s}}{\sum_{t=1}^{\tau} d_{t,s}} \text{ if } \sum_{t=1}^{\tau} d_{t,s} \neq 0$$

$$A_{\tau,s} = 0 \qquad \text{otherwise}$$

where:

$NR_{t,s}$ = net returns in period *t* from selecting strategy *s*, and

$d_{t,s}$ = a binary indicator variable equal to one if strategy *s* is chosen in period *t*; otherwise zero.

Probability of Strategy Selection

The probability of selecting strategy s in time period t depends on the attractions as follows:

$$p_s(t) = \frac{e^{\lambda A_s(t)}}{\sum_{k \in S} e^{\lambda A_k(t)}}$$

where A_s is the attraction to strategy s and the parameter $\lambda \geq 0$ represents the extent to which strategies with higher attractions are favored in strategy choice. When λ=0, all strategies are equally likely to be selected. As λ increases, strategies with higher attractions increasingly have a greater probability of being selected for decreasing differences in attractions between strategies.

Enhanced EWA: forage growth

The enhanced EWA model introduces more information via a forage growth equation:

$$F_t - F_{t-1} = \gamma \times PR_t \times \left(1 - \frac{F_{t-1}}{K_t (1 - \eta \, x_t)} \right)$$

where:

PR = precipitation in period t relative to historical mean precipitation,

K_t = maximum forage carrying capacity or animal unit months that can be grazed in period t in the absence of invasive species infestation,

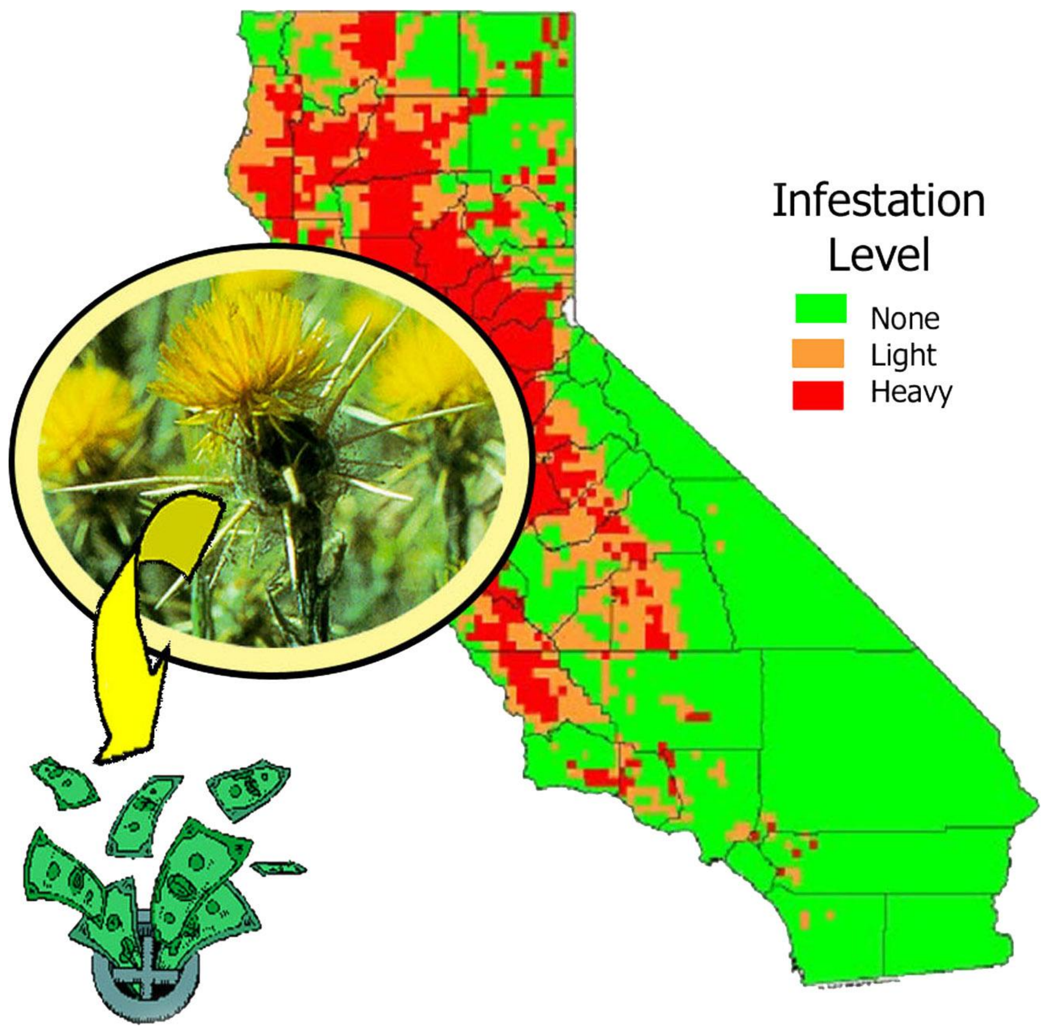
γ = intrinsic growth rate of the forage stock, and

η ($0 \leq \eta < 1$) is an adjustment parameter describing the reduction in carrying capacity due to the presence of x (invasive species).

Penalty functions

- To reflect the ecological benefits of a diversified control strategy, we introduce penalties when repeated applications of burning or herbicide controls are implemented
- The penalty increases in value with the number of times a specific strategy is used over a specific interval, so that the decision maker will learn not to repeat the same control too often

Yellow Starthistle (*Centaurea solstitialis*) in California: Over 14 Million Acres



YST Agricultural Producer Survey: Data Collected

- Ranch characteristics, baseline net revenue, etc.
- YST occurrence, cover rates
- YST control costs
- YST impacts on grazing and crop yields
- Other impacts, actions taken in response to YST, opinions, etc.

Survey Findings: Prevalence and Percent Cover

- 93% of respondents reported that their land currently is, or at some point has been, infested with YST
- The average rancher reported a mean percent cover of YST equal to 25%. (On those lands infested with YST, this species accounts for an estimated 25% of total vegetative cover on average.)

<u>Background on Grazing Impacts.</u> Selected statistics from 2003 survey of California ranchers: baseline grazing productivity and impacts of YST (Eiswerth and van Kooten, unpubl. data 2004).

Characteristic/parameter	Type of grazing land	
	Native range	Improved pasture
Mean net revenue of grazing land not infested with YST or other invasive weeds (baseline net revenue)	\$6.11/acre/yr	\$16.75/acre/yr
Mean percent decrease in forage yield attributable to YST	15.3%	12.8%
Mean decrease in net revenue attributable to YST	\$0.93/acre/yr	\$2.14/acre/yr

<u>More background</u>: Preliminary YST annual loss and cost estimates for Calaveras, Mariposa, and Tehama counties (Yr 2003), based on our 2003 survey of California ranchers.

Category of loss/cost	Estimated YST Losses and Costs, 2003	
	Lower estimate	Higher estimate
Losses due to reduced forage for livestock	$1.1 million	$2.3 million
Losses in alfalfa/meadow hay/cereal grains	$0.07 million	$0.1 million
Rancher out-of-pocket costs for YST control (*excluding* time cost of labor)	$0.7 million	$1.3 million
Subtotal losses/costs	$1.9 million/yr (+)	$3.7 million/yr (+)

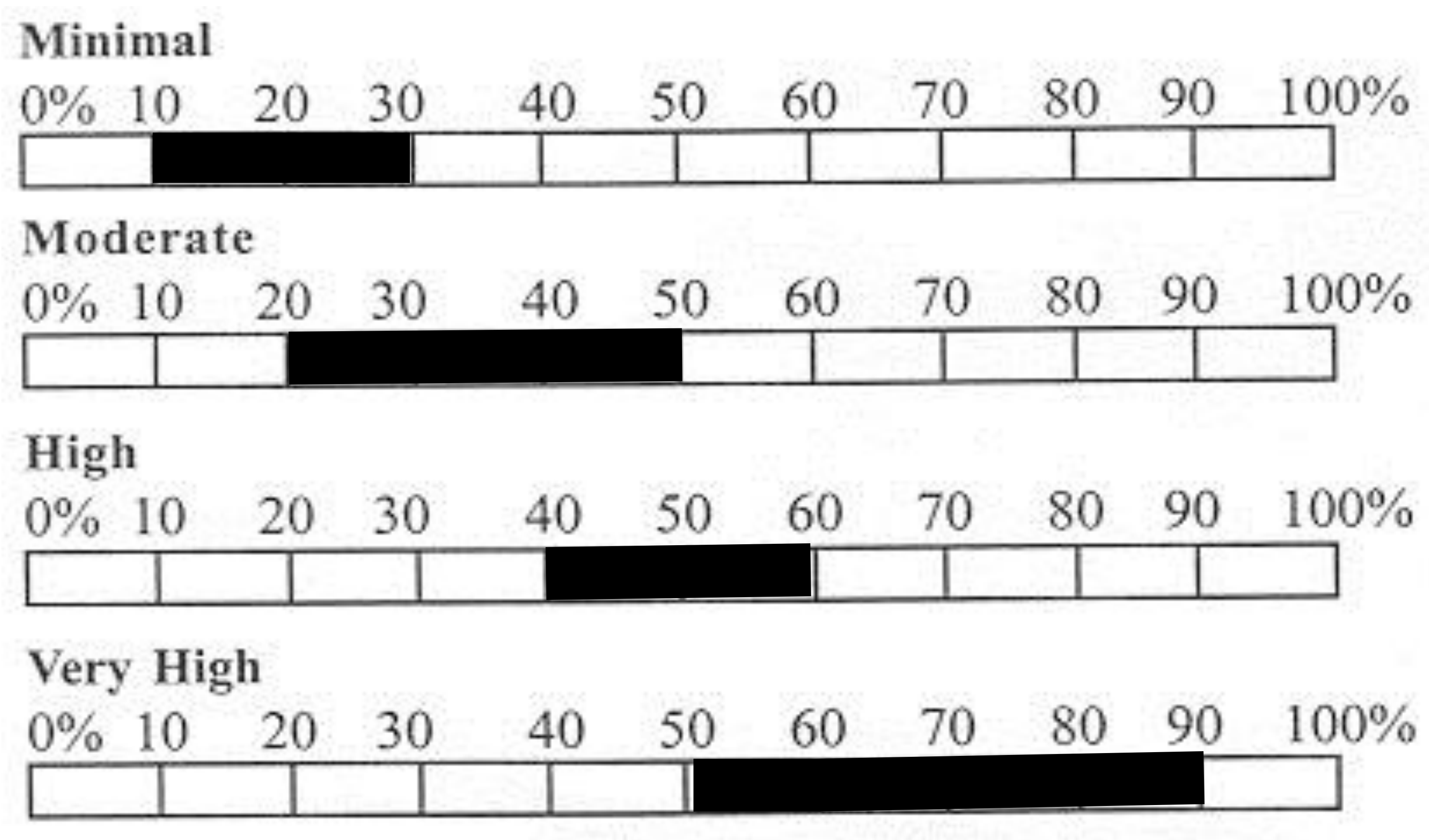
Uncertainty

- Impacts and damages are high, but quite uncertain
- The magnitudes of nonnative species stocks (state variables) are uncertain
- Growth rates and response to management (equations of motion) are uncertain

YST Expert Judgment Survey

- Elicits expert judgments on:
  - Severity of an invasion state?
  - Effectiveness of various control strategies?
  - Likelihood of transitions across states?
  - Impacts of YST on selected agricultural activities?
- Survey sample frame:
  - weed and range scientists
  - county farm advisors
  - public land managers
  - other specialists

Eliciting Expert Judgments on the Severity of Biological Invasions



Policy Options

1. Do nothing, or no control (*NC*)
2. One-time chemical control without follow-up treatment (*CH*)
3. Any combination of strategies that results in "successful management" [best practice], but without follow-up treatment (*BP*)
4. Same as 3, but with follow-up treatment in subsequent years (*BP+F*)
5. Same as 3, plus a program of site revegetation (*BP+R*)

Subjective Transition Probability Matrices (1 for Each Control Strategy)

	Future State			
Current State	Minimal	Moderate	High	Very High
Minimal				
Moderate				
High				
Very High				

Example Data: Transition Probability Matrix for No Control (NC)

	Future State			
Current State	Minimal	Moderate	High	Very High
Minimal	.1769	.3543	.2699	.1987
Moderate	.0412	.3033	.3914	.2639
High	.0317	.0708	.3936	.5037
Very High	.0308	.0434	.1083	.8173

Optimal YST Strategies Selected by SDP Model

Parameters/ States	Scenarios					
	1	2	3	4	5	6
Productivity	2.0	2.0	5.0	5.0	10.0	10.0
Discount Rate	0	5	0	5	0	5
YST States						
Minimal	CH	CH	CH	CH	BP+F	BP+F
Moderate	CH	CH	BP+F	BP+F	BP+F	BP+F
High	CH	CH	BP+R	BP+R	BP+R	BP+R
Very High	BP+F	BP+F	BP+F	BP+F	BP+F	BP+F

Strategy Proportions Resulting from Learning Models (5% discount rate)

Model	AUM/ac/yr	Mean Strategy Choice Proportions (n=30)				
		NC	CH	BP	BP+F	BP+R
EWA-enhanced	2.0	0.696	0.283	0.017	0.003	0.001
EWA	2.0	0.565	0.384	0.044	0.005	0.002
EWA-enhanced	5.0	0.691	0.283	0.021	0.003	0.002
EWA	5.0	0.438	0.484	0.059	0.014	0.004
EWA-enhanced	10.0	0.724	0.248	0.021	0.003	0.003
EWA	10.0	0.170	0.660	0.094	0.073	0.003

Summary of Model Results

Model	Max AUMs	Discount rate	Years	Mean NPV	Std. Dev.
Enhanced EWA	2	5	75	309.49	70.96
	5	5	75	866.69	105.29
	10	5	75	1,672.60	291.52
EWA	2	5	75	572.48	106.24
	5	5	75	1,445.50	285.04
	10	5	75	3,408.50	367.48
SDP	2	5	75	605.70	53.66
	5	5	75	1,758.85	101.64
	10	5	75	3,834.98	174.21

***WARNING*: THE MODEL RESULTS ARE NOT DIRECTLY COMPARABLE BECAUSE OF UNDERLYING ASSUMPTIONS.**

Big Question: Restated

- How can economic models better augment adaptive management frameworks (learning processes) in a context where benefits are large but surprisingly little hard data are available?