### Simulation-optimization, stochastic search and dynamic programming Is it all the same?

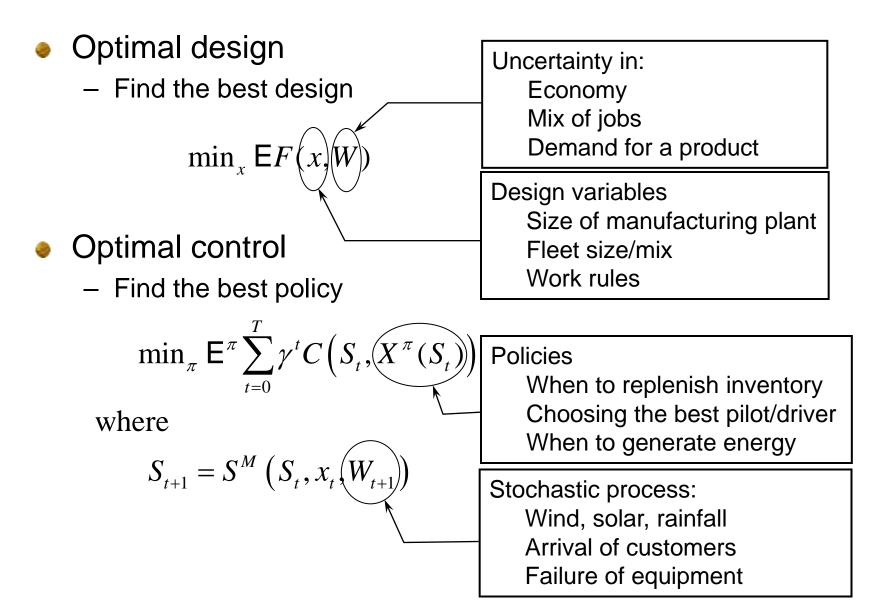
Warren B. Powell Simulation-optimization workshop Baltimore December, 2010

## From design to control

#### Design problems

- Where do I locate fire houses?
- How large should my buffers be in my job shop?
- What is the best mix of aircraft for my fleet?
- What is the cost of new driver/pilot work rules?
- Control problems
  - Which driver should be assigned to move a load of freight?
  - How many spare parts should I order? Where should they be stored?
  - Which power generating plants should I turn on, and when, in the presence of variable wind, solar, rainfall, prices and weather.
- What is the difference?
  - When is the design of a physical system different from finding a policy to manage a system?

### Stochastic optimization



## Algorithms

How do we solve stochastic design problems?

- Stochastic search
  - Stochastic approximation methods (with and without gradients)
- Metaheuristics
  - Simulated annealing, genetic algorithms
- Simulation optimization
  - OCBA, LL(B), Compass, response surface methods, knowledge gradient, ...
- How do we solve stochastic control problems
  - Dynamic programming via Bellman's equation

 $V(S_t) = \max_{x} \left( C(S_t, x) + \gamma \mathsf{E}\left\{ V(S_{t+1}) \mid S_t \right\} \right)$ 

- Approximate dynamic programming, reinforcement learning

## Algorithms

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

$$\hat{v}^n = \max_x \left( C(S_t^n, x) + \gamma \sum_f \theta_f^{n-1} \phi(S_t^x(S_t^n, x)) \right)$$
 a.k.a. TD(0)

or

$$\hat{v}^n = \sum_{t=0}^T \gamma^t C\left(S_t, X^{\pi}(S_t \mid \theta^{n-1})\right)$$
 a.k.a. TD(1)

where

$$X^{\pi}(S_t \mid \theta^{n-1}) = \arg\max_x \left( C(S_t, x) + \gamma \sum_f \theta_f^{n-1} \phi(S_t^x(S_t, x)) \right)$$

- Now we might use recursive least squares to update  $\theta^{n-1}$ .
- But what if we simply view θ as a static design parameter?

$$\max_{\theta} \mathsf{E} F(\theta, W) = \mathsf{E} \sum_{t=0}^{T} \gamma^{t} C(S_{t}, X^{\pi}(S_{t}))$$

## Policies

- Policies come in four fundamental flavors:
  - Myopic policies

 $X^{\pi}(S_t) = \arg\max_{x_t} C(S_t, x_t)$ 

- Look-ahead policies (tree search, rolling horizon procedures)

$$X^{\pi}(S_{t}) = \arg\max_{x_{t}, x_{t+1}, \dots, x_{t+T}} \sum_{t'=t}^{T} C(S_{t'}, x_{t'})$$

Policy function approximations

$$X^{\pi}(S_t) = \begin{cases} Q - q & \text{If } S_t < q \\ 0 & \text{Otherwise} \end{cases}$$

or

$$X^{\pi}(S_t) = \theta_0 + \theta_1 S_t + \theta_2 S_t^2$$

- Policies based on value function approximations:

$$X^{\pi}(S_{t}) = \arg \max_{x_{t}} \left( C(S_{t}, x_{t}) + \gamma E \overline{V}_{t+1}(S_{t+1}) \right)$$

# Thoughts

- Differences between simulation-optimization (and stochastic search) and approximate dynamic programming (sequential optimization) are primarily cosmetic:
  - ADP may use bootstrapping, where value of being in state  $S_t$  depends on approximation  $V(S_{t+1})$  of the downstream state (also known as TD(0) or TD( $\lambda$ )).
  - Simulation-optimization/stochastic search does not approximate the value of being in a state.
  - Simulation-optimization *may* use response surface methods to approximate  $\overline{F}(x) = EF(x,W)$ , which is similar to a value function approximation.
  - Policy search (using policy function approximations) is equivalent to stochastic search.
  - Simulation-optimization *often* assumes discrete alternatives, and expensive, noisy measurements.