Three Concepts: Utility
Spring 2007

Tei Laine, PhD.
Department of Computer Science
University of Helsinki

Outline for Jan 16

1. Introduction to problem solving and optimization.
2. Course policies
    1. Meetings
    2. Assignments
    3. Grading
3. Reinforcement learning
4. Introduction to Project I: k-armed bandit

1. Introduction

- Goal: to introduce methods for real-world problem solving!
- Several real-world problems in design, scheduling, or routing can be conceived as optimization problems.
- Some (interesting) real-world problems are difficult:
  - ← The search space is large and multi-dimensional.
  - ← Problems are too complicated.
  - ← Problems may not be well-defined; it is not obvious where the optimal solution is located.
  - ← Real world is noisy and uncertain.
  - ← No efficient (polynomial time) algorithm is known.

1.1 Optimization

- Minimize (maximize) an objective function f of decision variables $\mathbf{x}$, subject to constraints $g_j(\mathbf{x}) \geq b_j$, $i = 1, ..., m$.
  - If $\mathbf{x}$ are continuous (and f and g linear), the problem is a linear programming problem.
  - If $\mathbf{x}$ are discrete the problem is combinatorial optimization problem.
- Finding an optimal solution potentially takes a long time $\rightarrow$ use heuristic search to find an approximate solution.

1.2 Central concepts in optimization

- *Search space*: set of possible solutions.
- *Objective*: description of the desired solution.
- *Evaluation function*: mapping from search space to numerical values characterizing the goodness of the solutions w.r.t. the objective.
- *Constraints* divide the search space into feasible and infeasible regions.

1.3 Central concepts in search

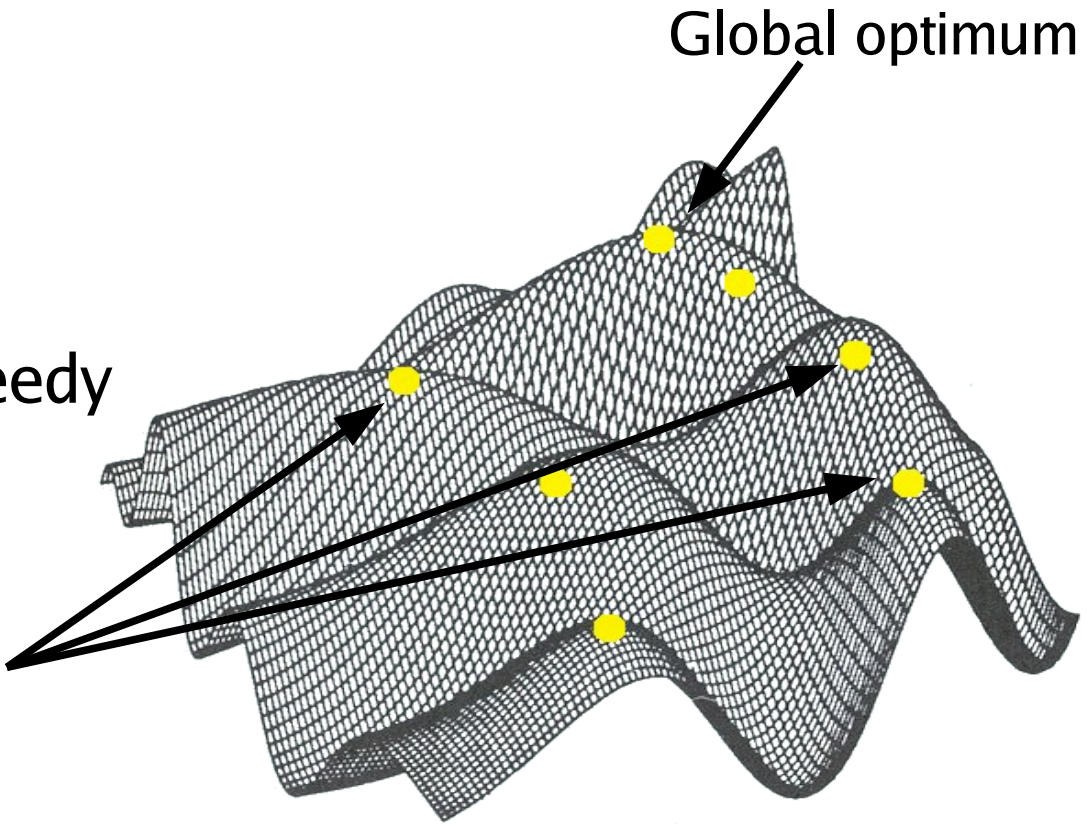
- Given a search space S, and its feasible region F, find $x \in F$ such that $\text{eval}(x) \leq \text{eval}(y), \forall y \in F$.
- *Distance* function specifies "n earness" (of potential solutions)
- *Neighborhood* of solution x is the set of solutions that can be reached from x by a simple operation.
- *Local optimum* is a solution that is better than any of its neighboring solutions.
- *Global optimum* is better than any solution.

1.4 Search methods

- Given the real-world problem sizes exhaustive search is often out of question.
- Domain or task specific methods require knowledge of the search space, and only apply to few cases.
- General heuristics do not necessarily guarantee optimality.
- Metaheuristics, nature inspired methods, have proven to produce empirically good results.

1.5 Local search

- Many heuristics are based on local neighborhood search:
  - Random
  - Iterative improvement
    - Hill-climbing
    - Steepest ascent or greedy
    - Stochastic



1.6 Metaheuristics

- Stochastic
- Inspired by nature: physics, evolution, genetics, etc.
- Solve the problem of getting trapped in a local optimum:
  - Accept worse solutions every now and then (simulated annealing, tabu search)
  - Maintain a population of candidate solutions (genetic algorithms, ant colony optimization)
- Shortcomings:
  - Involve parameter adjustment
  - Time consuming
  - Sensitive to representational format

1.7 Tentative list of topics

- Simulated annealing
  - Temperature
- Tabu search
  - Memory
- Evolutionary algorithms
  - Selection and recombination
- Ant colony optimization
  - Collaboration, division of labor
- Multi-criteria decision making

2. Course details and policies

- Instructors and office hours
  - Tei Laine: Mon, Wed 12-12:30 @A213
  - Teemu Roos: Mon, Thu 10:30-11:30 @D118
- Meetings
  - Third period: Tue 9-12 @C222
  - Fourth period: Tue 10-12 @C222
- Material made available in the course folder in the copyroom C127.
- Course home page: http://www.cs.helsinki.fi/group/cosco/Teaching/Utility/2007/

2.1 Assignments

- Mix of seminar work and projects, no exams
- Individual effort
  - Five (5) homework assignments on lecture topics
    - Completed before the respective lecture.
    - Application of a given optimization technology to a problem.
  - Term paper
    - Topics TBA
- Group effort
  - Three (3) programming projects
    - Project I during the third period in several phases
    - Projects II and III in the fourth period
  - Poster presentation (individually or in pairs)

2.2 Grading

- Project I 15%
- Project II 20%
- Project III 25%
- Poster presentation 15%
- Term paper 25%
- Assignments and classroom participation scale grades up or down, and are used in borderline cases to determine the final grade.

3. Reinforcement learning

- Not characterized by learning method, but by a learning problem (Sutton & Barto, 1998):
  - Map situations to actions
  - Supervised learning not adequate for real-world learning ← impractical to give correct and representative samples of situations that an agent may encounter.
- Learning to do (from experience)
  - Interaction with the environment to achieve one's goals.
  - Perceive the state of environment, and act to change the state.
  - Uncertainty involved.

- Trial and error learning
  - *Exploration*: try actions never tried before.
  - *Exploitation*: retry actions that have been beneficial before.
- Delayed reward
  - Credit assignment problem

3.1 Framework

- *State* — whatever information is available for the learner
- *Policy* — mapping from perceived state to action available in that state.
- *Reward* function — mapping from state-action pair to a single number
  - Immediate desirability of a state
  - Goal is to maximize the total reward in the long run.
  - Cannot be changed by the learner; considered part of the environment
- *Value* function — the total amount of reward accumulated starting from a certain state
  - Long-term desirability of a state
  - Value estimation the most central issue
- *Model* of the environment
  - Used for planning
  - Learner's control defines the learner-environment boundary.

3.2 Key concepts

- At each time point $t = 0,1,...(,T)$ learner perceives the state $s_t \in S$ (set of possible states), selects an action $a_t \in A(s_t)$ (all actions available in s_t), receives a numerical reward $r_{t+1} \in$ R, and moves to state s_{t+1}.
- Policy $\pi_t(s,a)$ maps state s to the probability of choosing action a at time t in that state.
- Learning method determines how policy changes with experience.

- Sequence of rewards received after time t is denoted

  $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$

- Tasks that repeat time steps from 1 through T are called episodic.
- The learner wants to maximize the *expectation of the total return:* $R_t = r_{t+1} + r_{t+2} + r_{t+3} + \ldots + r_T$
- Sometimes $T = \infty$.
- Then the learner wants to maximize the expectation of the *discounted return*:

  $R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \Sigma_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad 0 \leq \gamma < 1,$

  where γ is a discount rate.

4. Project I: k-armed bandit

- The problem
  - Choice of k actions.
  - Numerical reward (-1 or 1) from stationary probability distribution associated to each action.
  - Objective is to maximize the expected total reward.
- Task
  - Write a program that plays k-armed bandit for fixed time.
  - Objective is to maximizes the expected reward, i.e., to solve the exploration and exploitation dilemma when the number of trials is unknown, and depends on the exploration policy.

- The project I is completed in three phases and in teams of 2 or 3 students.
- At each phase
  - The reward distribution changes.
  - The rewards accumulate.
  - The teams can improve their policy.
- Grade depends on
  - Participation: phases 1, 2 and 3 give 1, 2 and 3 points, respectively (max 6pts.)
  - Cumulative reward, so that the team earns $(r/maxR)*9$ points, where r is the team's cumulative reward and $maxR$ is the cumulative reward of the best team.