Local Search

Introduction to
Artificial Intelligence

COS302

Michael L. Littman

Fall 2001

Administration

Questions?

Reading: Boyan thesis for more information on local search.

http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/jab/mosaic/thesis/

chap3.ps.gz, appen2.ps.gz

Search

Recall the search algorithms for constraint problems (CSP, SAT).

- Systematically check all states.
- Look for satisfied state.
- In finite spaces, can determine if no state satisfies constraints (complete).

Local Search

Alternative approach:

- **Sometimes more interested in positive answer than negative.**
- **Once we find it, we're done.**
- **Guided random search approach.**
- **Can't determine unsatisfiability.**

Optimization Problems

- Framework: States, neighbors (symmetric). Add notion of *score* for states: V(s)
- Goal: Find state w/ highest (sometimes lowest) score.
- Generalization of constraint framework because...

SAT as Optimization

- States are complete assignments.
- Score is number of satisfied clauses.
- Know the maximum (if formula is truly satisfiable).

Local Search Idea

- In contrast to CSP/SAT search, work with complete states.
- Move from state to state remembering very little (in general).

Hill Climbing

Most basic approach

- Pick a starting state, s.
- If $V(s) > \max_{s' \text{ in } N(s)} V(s')$, stop
- Else, let s = any s' s.t.

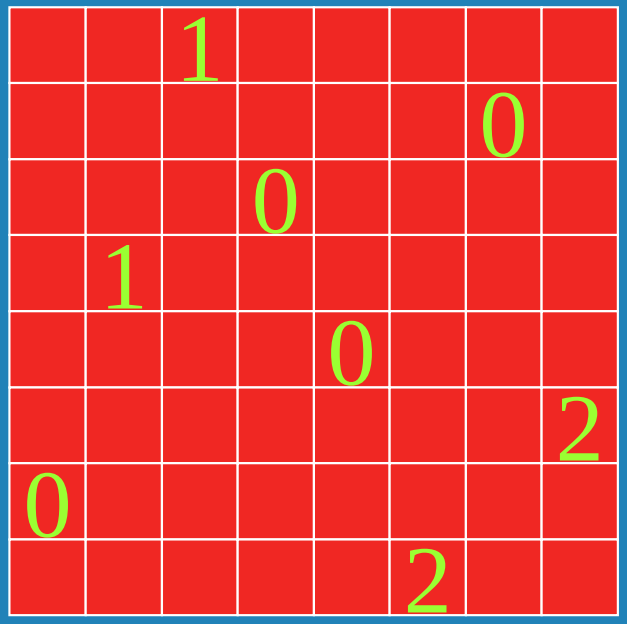
  $V(s') = \max_{s'' \text{ in } N(s)} V(s'')$
- Repeat

Like DFS. Stops at local maximum.

N-queens Optimization

Minimize violated constraints.

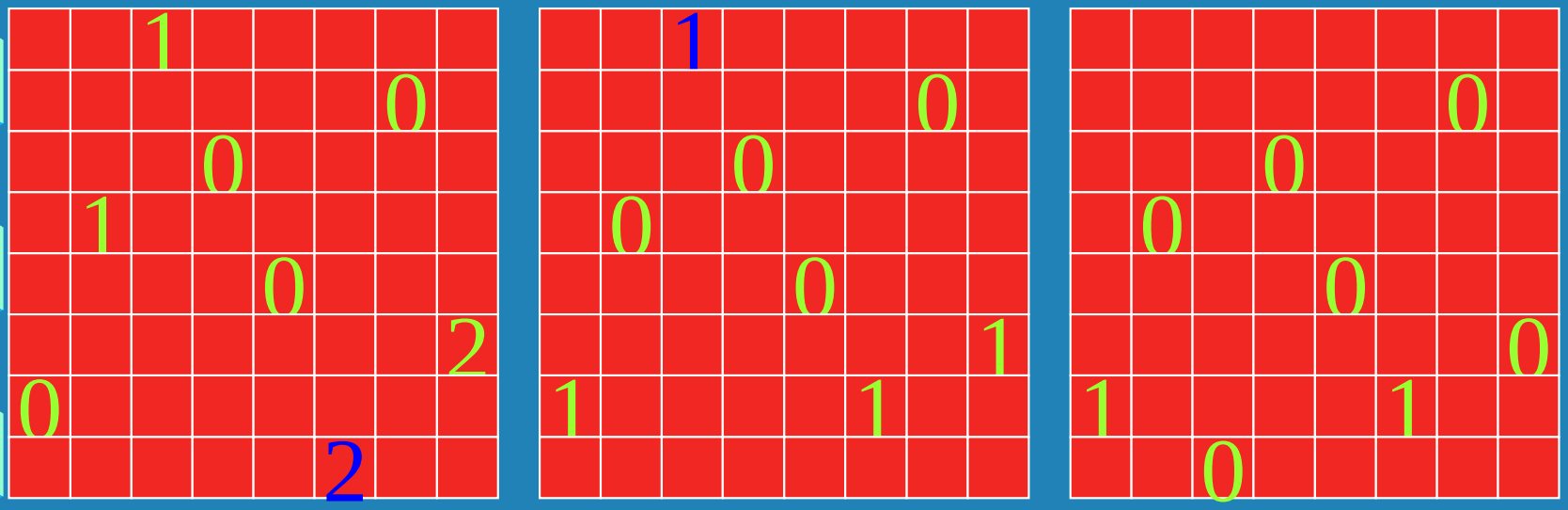
Neighbors: Slide in column.



N-queens Search Space

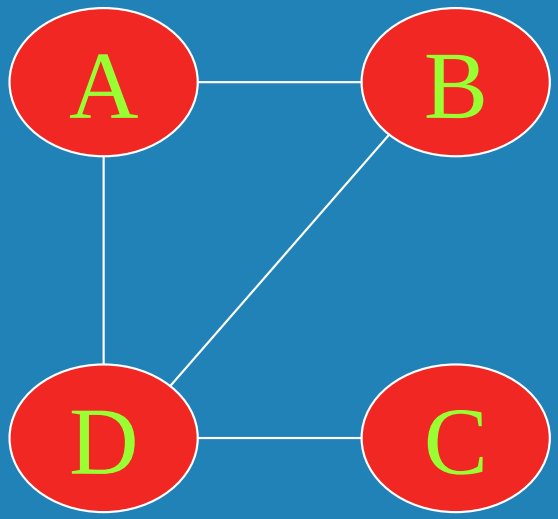
- How far, at most, from minimum configuration?
- Never more than n slides from any configuration.
- How many steps, at most, before local minimum reached?
- Maximum score is n(n-1).

Reaching a Local Min



Graph Partitioning

Separate nodes into 2 groups to minimize edges between them



Objective Function

Penalty for imbalance

$V(x) = cutsize(x) + (L-R)^2$

State space? Neighbors?

State Space

0000: 16+0

0001: 4+3

0010: 4+1

0011: 0+2

0100: 4+2

0101: 0+3

0110: 0+3

0111: 4+2

1000: 4+2

1001: 0+3

1010: 0+3

1011: 4+2

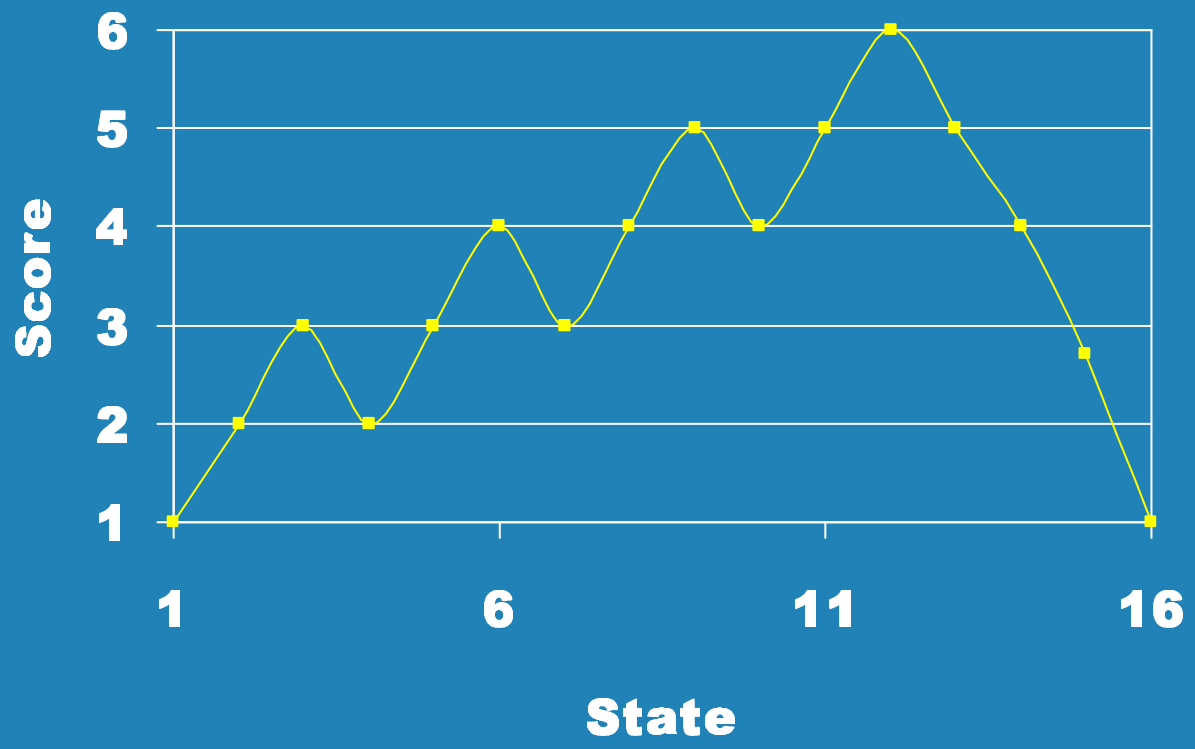
1100: 0+2

1101: 4+1

1110: 4+3

1111: 16+0

Bumpy Landscape



Simulated Annealing

Jitter out of local minima.

Let T be non-negative.

Loop:

- Pick random neighbor s' in N(s)
- Let D = V(s) – V(s') (improve)
- If D > 0, (better), s = s'
- Else with prob $e^{D/T}$, s = s'

Temperature

- As T goes to zero, hill climbing
- As T goes to infinity, random walk
- In general, start with large T and decrease it slowly.
- If decrease *infinitely* slowly, will find optimal solution.
- Analogy is with metal cooling.

Aside: Analogies

Natural phenomena inspire algs

- **Metal cooling**
- **Evolution**
- **Thermodynamics**
- **Societal markets**
- **Ant colonies**
- **Immune systems**
- **Neural networks, ...**

Annealing Schedule

After each step, update T

- Logarithmic: $T_i = \gamma/\log(i+2)$
  provable properties, slow
- Geometric: $T_i = T_0 \gamma^{\text{int}(i/L)}$
  lots of parameters to tune
- Adaptive
  change T to hit target accept rate

Lam Schedule

Ad hoc, but impressive (see Boyan, pg. 190, 191). Parameter is run time.

- Start target accept rate at 100%
- Decrease exponentially to 44% at 15% through run
- Continue exponential decline to 0% after 65% through run.

Locality Assumption

- Why does local search work? Why not jump around at random?
- We believe scores change slowly from neighbor to neighbor.
- Perhaps keep a bunch of states and search around the good ones...

Genetic Algorithms

Search : Population genetics ::

State : individual ::

State set : population ::

Score : fitness ::

Neighbors : offspring / mutation

? : sexual reproduction, crossover

Algorithmic Options

- Fitness function
- Representation of individuals
- Who reproduces? *
- How alter individuals (mutation)?
- How combine individuals (crossover)? *

Representation

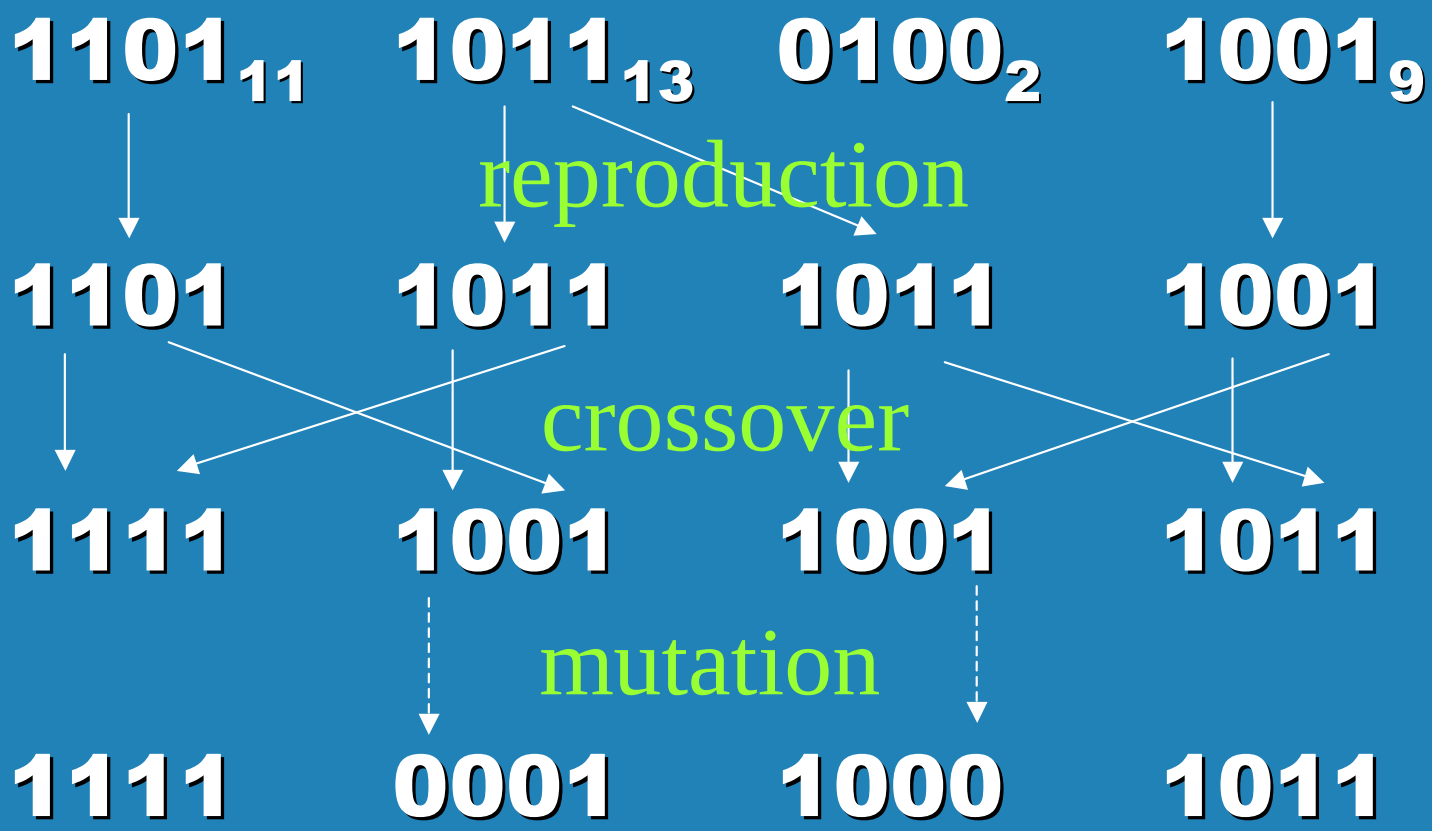
"Classic" approach: states are fixed-length bit sequences

Can be more complex objects: genetic programs, for example.

Crossover:

- One point, two point, uniform

GA Example


1101_{11} 1011_{13} 0100_{2} 1001_{9}
reproduction
1101 1011 1011 1001
crossover
1111 1001 1001 1011
mutation
1111 0001 1000 1011

Generation to Next

G is a set of N (even) states

Let p(s) = V(s)/sum $_{s'}$ V(s'), G' = { }

For k = 1 to N/2

choose x, y with prob. p(x), p(y)

randomly swap bits to get x', y'

rarely, randomly flip bits in x', y'

G' = G' U {x', y'}

GSAT

Score: number of unsat clauses.

- Start with random assignment.
- While not satisfied...
  - Flip variable value to satisfy greatest number of clauses.
  - Repeat (until done or bored)
- Repeat (until done or bored)

GSAT Analysis

- What kind of local search is this most like? Hill climbing? Simulated annealing? GA?
- Known to perform quite well for coloring problems.

WALKSAT

- On each step, with probability p, do a greedy GSAT move.
- With probability 1-p, “walk” (flip a variable in an unsatisfied clause).
- Even faster for many problems (planning, random CNF).

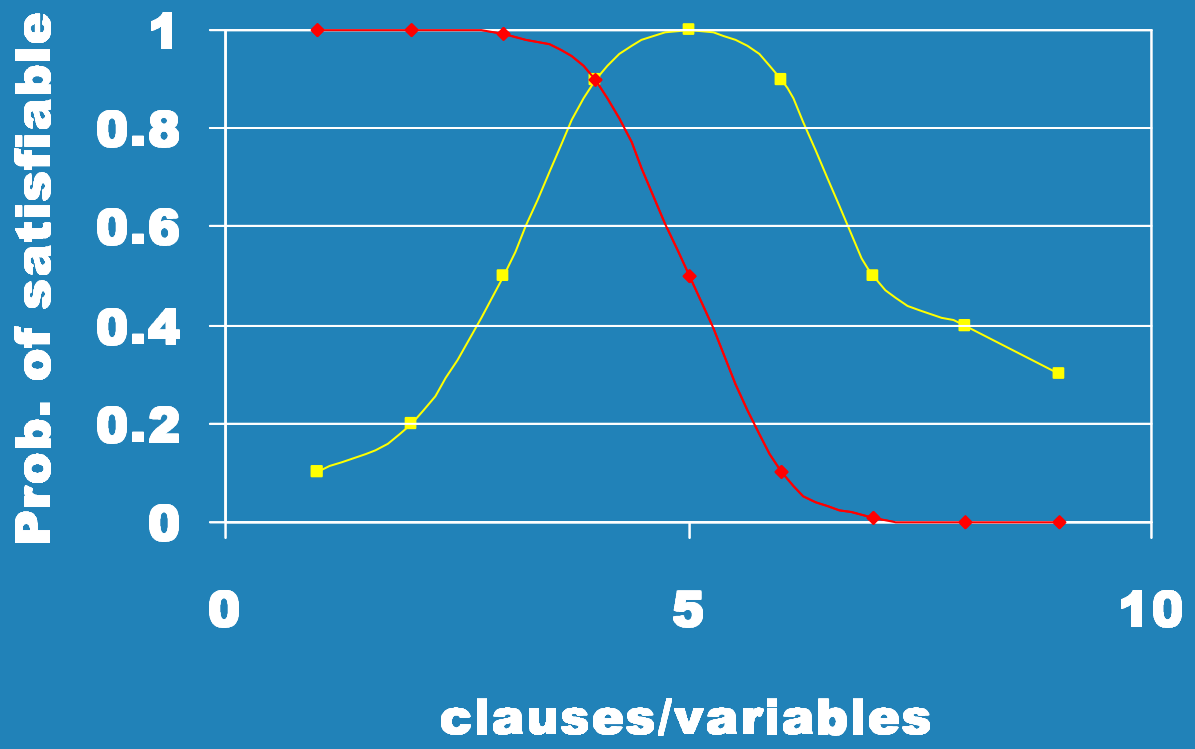
WALKSAT Analysis

- What kind of local search is this most like? Hill climbing? Simulated annealing? GA?
- Known to perform quite well for hard random CNF problems.

Random k-CNF

- Variables: n
- Clauses: m
- Variables per clause: k
- Randomly generate all m clauses by choosing k of the n variables for each clause at random. Randomly negate.

Satisfiable Formulae



Explaining Results

Three sections

- m/n < 4.2, under constrained: nearly all satisfiable
- m/n > 4.3, over constrained: nearly all unsatisfiable
- m/n , under constrainted: nearly all satisfiable

Phase Transition

- Under-constrained problems are easy, just guess an assignment.
- Over-constrained problems aren't too bad, just say "unsatisfiable" and try to verify via DPLL.
- Transition region sharpens as n increases.

Local Search Discussion

- Often the *second* best way to solve any problem.
- Relatively easy to implement.
- So, good first attempt. If good enough, stop.

What to Learn

- Definition of hill climbing, simulated annealing, genetic algorithm.
- Definition of GSAT, WALKSAT.
- Relative advantages and disadvantages of local search.

Programming Project 1 (due 10/22)

- Solve Rush Hour problems.
- Implement BFS, A* with simple blocking heuristic, A* with more advanced heuristic.
- Input and output examples available on course web site
- Speed and accuracy count.

Homework 4 (due 10/17)

1. In graph partitioning, call a *balanced* state one in which both sets contain the same number of nodes. (a) How can we choose an initial state at random that is balanced? (b) How can we define the neighbor set N so that every neighbor of a balanced state is balanced? (c) Show that standard GA crossover (uniform) between two balanced states need not be balanced. (d) Suggest an alternative crossover operator that preserves balance.
2. More soon...