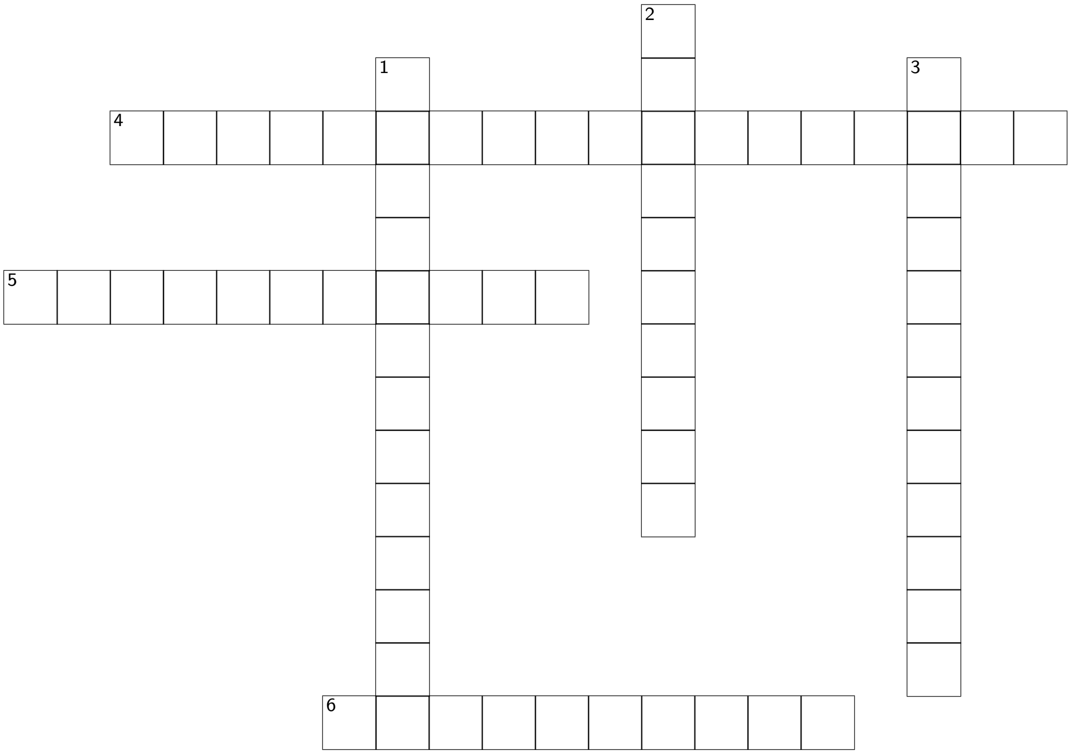
15-281: AI: Representation and Problem Solving — Spring 2024

Recitation 4 — February 9

1 Missing in the Mountains

The 15-281 Course Staff decided to climb the Rocky Mountains together over Winter Break. On their way back down from the mountains, they realize they left Simrit at the top of the tallest mountain! They have no idea where on the mountain they are or which mountain they are on, and are worried about how they will find Simrit before lecture. Help the 281 Staff remember all of the local search algorithms they have learned so they can save Simrit!



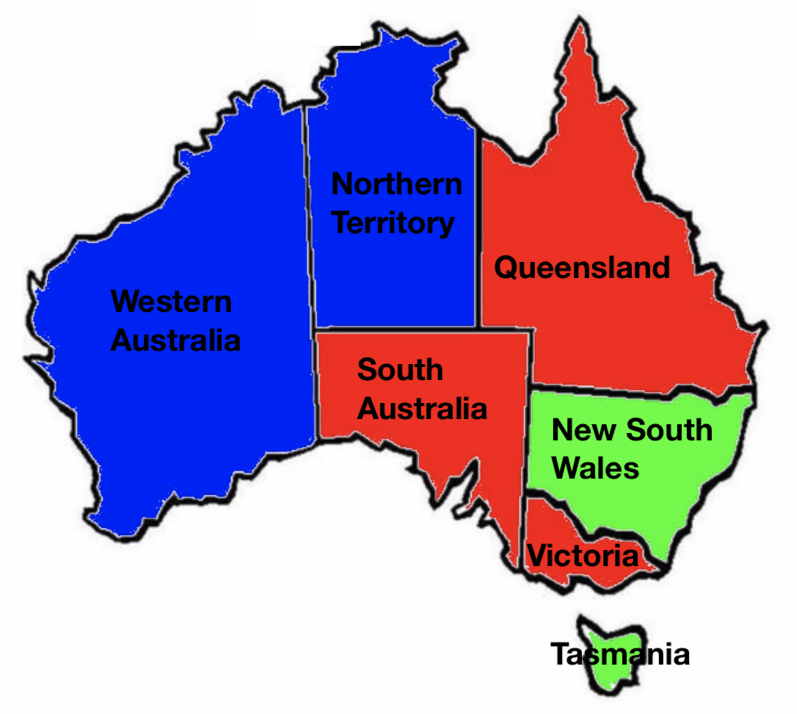
Down

1. A variant of hill-climbing where you conduct a series of searches from randomly generated starting states until the goal is found.
2. A local search technique where you uniformly randomly choose a neighbor to move to.
3. A type of greedy local search where you move uphill to local maxima.

Across

4. A local search technique where you allow for downhill moves but make them rarer as time goes on.
5. A variant of hill-climbing where you generate successors randomly (one by one) until a better one is found.
6. A variant of hill climbing in which you choose a move randomly from the uphill moves, with the probability of a move being chosen dependent on the "steepness" (amount of improvement from making that move).

2 Map Coloring with Local Search



Recall the various local search algorithms presented in lecture. Local search differs from previously discussed search methods in that it begins with a complete, potentially conflicting state and iteratively improves it by reassigning values. We will consider a simple map coloring problem, and will attempt to solve it with hill climbing.

(a) How is the map coloring problem defined (In other words, what are variables, domain and constraints of the problem)? How do you define states in this coloring problem?

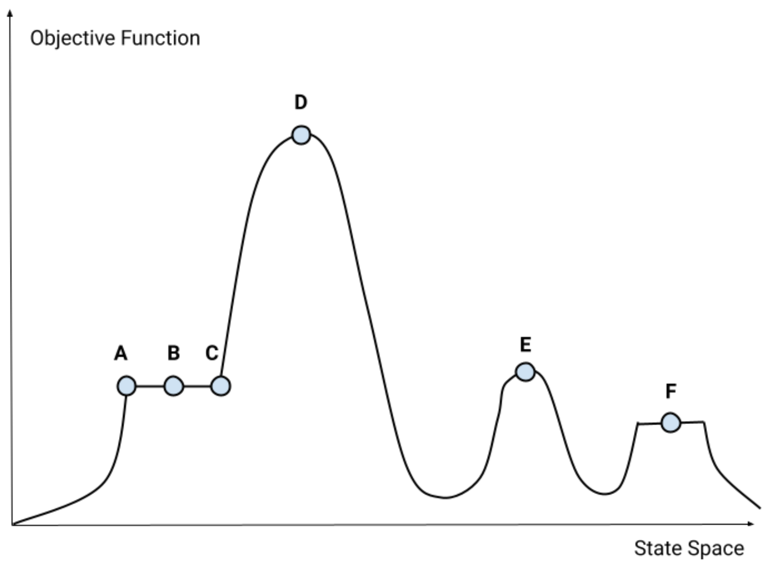
(b) Given a complete state (coloring), how could we define a neighboring state?

(c) What could be a good heuristic be in this problem for local search? What is the initial value of this heuristic?

(d) Use hill climbing to find a solution based on the coloring provided in the graph.

(e) How is local search different from tree search?

3 Local Search Discussion Questions



Consider the state space above in the context of local search. Recall that our goal is to find the state that maximizes the objective.

1. Consider the points A, B, C, D, E, and F on the graph.

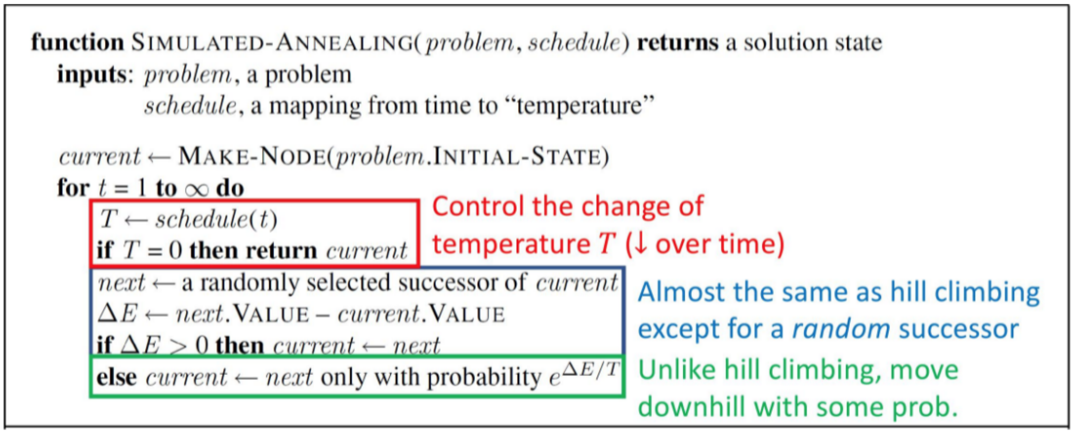
   (a) Which of the points on the graph are on a shoulder? Which of those points are local maximums?

   (b) Which of the points on the graph are a ”flat” local maximum?

   (c) What is the difference between a shoulder and a ”flat” local maximum?

2. Let’s take a look at simulated annealing. Simulated Annealing is quite similar to hill climbing.

   - Instead of picking the best move, it picks a random move.
   - If the move improves the situation, the move is always accepted.
   - Otherwise, it accepts the move with some probability less than 1



```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  inputs: problem, a problem
          schedule, a mapping from time to "temperature"

  current ← MAKE-NODE(problem.INITIAL-STATE)
  for t = 1 to ∞ do
      T ← schedule(t)
      if T = 0 then return current
      next ← a randomly selected successor of current
      ΔE ← next.VALUE – current.VALUE
      if ΔE > 0 then current ← next
      else current ← next only with probability $e^{\Delta E/T}$
```

Control the change of temperature *T* (↓ over time)

Almost the same as hill climbing except for a *random* successor

Unlike hill climbing, move downhill with some prob.

(a) How does the sign of ΔE reflect the "badness" of a move?

(b) In simulated annealing, we control the temperature T. How does the value of T impact the probability with which we choose a "bad" move?

3. Mark True or False for each of the following statements.

(a) Regular hill climbing is optimal (i.e., will always find the global maximum)

(b) Random restart hill climbing is optimal when given an infinite amount of time.

(c) Simulated annealing allows for downward moves according to some fixed constant temperature T.

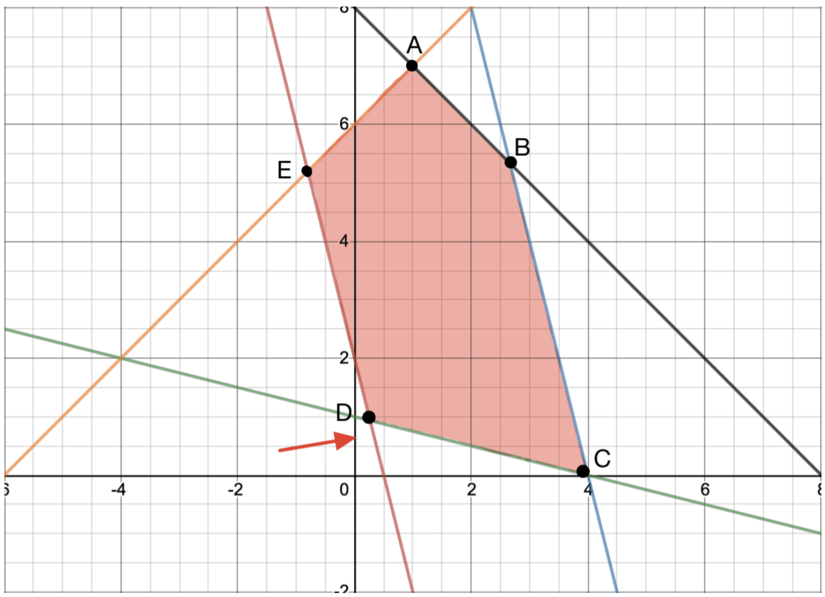
(d) Simulated annealing is generally less time efficient than random walk.

(e) A random walk algorithm is more likely to choose a better neighbor than a worse one.

4 Algorithms for Solving Linear Programming

In lecture, we went through two algorithms for solving linear programming programs - vertex enumeration and the hill climbing algorithm.

Consider this linear programming problem. The goal is to minimize the cost, and the cost vector (red) is perpendicular to the blue and red lines.



1. Briefly describe both algorithms and explain how they differ. (hint: use terms such as vertices, intersections and neighbors).

2. Run the hill climbing algorithm starting from point B. Now try running the algorithm starting from point C. How do their solutions differ?

5 Cargo Plane: Linear Programming Formulation

A cargo plane has three compartments for storing cargo: front, center and rear. These compartments have the following limits on both weight and space:

Compartment	**Weight capacity (tons)**	**Space capacity (cubic metres)**
Front	10	6800
Centre	16	8700
Rear	8	5300

The following four cargoes are available for shipment on the next flight:

Cargo	**Weight (tons)**	**Volume (cubic metres/ton)**	**Profit ($/ton)**
C1	18	480	310
C2	15	650	380
C3	23	580	350
C4	12	390	285

Any proportion of these cargos can be accepted. The objective is to determine how much of each cargo C1, C2, C3 and C4 should be accepted and how to distribute each among the compartments so that the total profit for the flight is maximised. **Formulate** the above problem as a linear program (what is the objective and the constraints?). Think about the assumptions you are making when formulating this problem as a linear program.

If you were to put this linear program into standard form, what would be the dimensions of $A, \mathbf{b}, \mathbf{c}, \mathbf{x}$?

Now consider a simpler problem. There is a cargo plane with a single compartment with limit on 20 tons weight and 2400 cubic meters limit on space. You want to use this cargo plane to transport boxes of oranges and pineapples to sell in a market overseas.

Your goal is to maximize the number of gold pieces under following constraints:

- The market only allows each person to sell 14 boxes.
- 1 box of oranges has weight 1 ton and volume of 100 cubic meters.
- 1 box of pineapples has weight 2 tons and volume of 300 cubic meters.
- You earn 5 gold pieces for 1 box of oranges.
- You earn 12 gold pieces for 1 box of pineapples.

We will now formulate and solve the LP.

1. Write the LP in inequality form.
2. Graph the constraints, cost vector, and at least 3 cost contours. Indicate the feasible region.
3. What is the **optimal number** of boxes of oranges and pineapples? How much gold does this earn?