CAS CS 440
Lec 6

Local Search (cont. II)

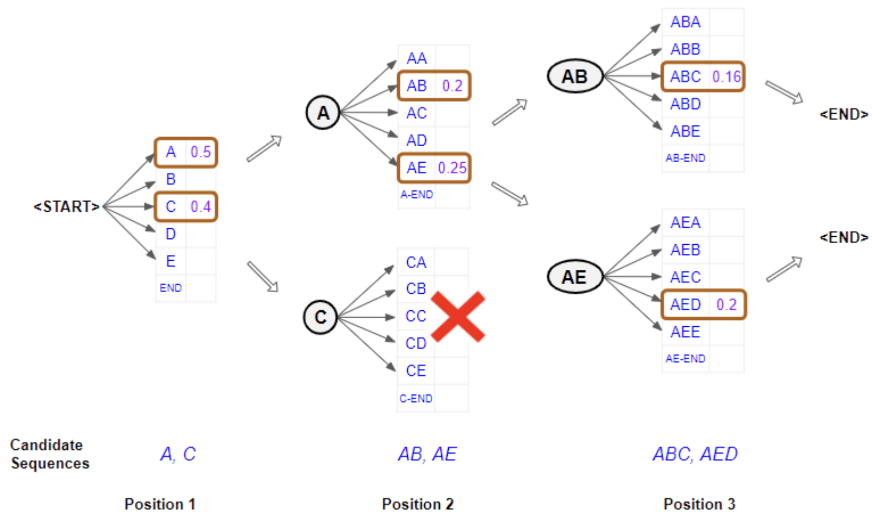
1. Local Search in Continuous Spaces
    a. So far, algorithms proposed bad for continuous spaces
        i. (GA for model parameters works on continuous spaces)
    b. Problem: enumerating child states
        i. Potential infinite!
    c. Can still measure “good” and “bad” states
        i. Calculus not enumeration
2. Derivatives
    a. If we have an objective   which is continuous (or piecewise continuous)
        i. Can still optimize!
        ii. For instance:

$$f(x) = (x - 2)^2$$

    b. Optima can be found using 1st (and 2nd derivatives)
        i. 1st derivative can tell us where optima is
        ii. 2nd derivative can tell us what kind of optima it is
        iii. Unless there is a saddle point
3. Derivatives in Higher Dimensions
    a. Called a gradient
    b. Algorithm:
        i. For every variable in :
            1. Calculate the partial derivative (differentiate with respect to that single variable)
            2. Collect partial derivatives into a vector
    c. Ex) → partial derivatives
        i.

$$\nabla f(x, y, z) = z^3 e^{(x+y)^2}$$
$$= \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)^T$$
$$= \left(z^3 e^{(x+y)^2} 2(x+y), z^3 e^{(x+y)^2} 2(x+y),\ 3z^2 e^{(x+y)^2}\right)^T$$

4. Derivative & Gradient Practice
    a. What is the derivative of $f(x) = 2\sin(x)e^{\cos(x)}$
        i. $$\frac{df}{dx} = -2\big(\sin(x)\big)^2 e^{\cos(x)} + 2\cos(x)e^{\cos(x)}$$
    b. What is the gradient of $f(x, \alpha) = \frac{\cos(\alpha x)}{x^2} + 4\alpha$
        i. $$\nabla f = \left(\frac{-\alpha x^2 \sin(\alpha x) - \cos(\alpha x)2x}{x^4},\ 4 - \frac{\sin(xy)}{x^2}\right)^T$$
5. Derivatives & Gradients
    a. Sometimes:
        i. We can set the derivative to 0 and solve (rare!)
        ii. Formula we get is called a closed form solution
        iii. Super super super super super rare since most equations cannot be driven or if you have them, you cannot derive them
        iv. We are excited this because we havea formula for the answer (if I have a formula, there is no need to search since the formula provides them → known as closed form solution)
        v. If you have an equation, you know the position (x,y,z) when you are at a state → substitute them to the equation and measure the gradient
        vi. Gradient will always point to the local maxima/minima (for minima, multiply -1 to the calculated gradient)
        vii. We are also generally given with step_size → how many steps to take to that direction given by gradient
    b. What if we can't get a closed form solution?
        i. Derivative + gradient always points towards local maximums
        ii. Continuous hill climbing?
            1. Called gradient descent/ascent
6. Optimization with Local Optimizers
    a. Path on Objective Surface = trajectory
    b. Local optimization only looks at "local" region to decide where to go next
    c. Can get stuck!
        i. Local optima (bad)
        ii. How to get unstuck? → beam search
7. Local Beam Search
    a. Keep k states in memory (the beam): (run k climbers at the same time)
        i. For every state in the beam, generate all possible child states
        ii. Recompute the beam: keep the top k best states (according to objective)
            1. If k = 4, choose 4 climbers

            2. The 4 climbers each move to 4 different states each
            3. Gives total of 16 states
            4. Out of those 16 states, keep k = 4 best states
            5. If any child is a goal state, stop!
            6. You also stop whenever the new beams that are created are better than the initial states
            7. Each climbers may have different algorithm in climbing (diversify the algorithms – simulated annealing, vanilla, or stochastic)
    b. Information is "shared" between states in the beam! (hill climbers talk to each other)
        i. Not directly: sorting is a form of communication!
    c. 

    d. How to encourage diversity in the beam?
        i. Difficult: lots of risk for imposing our own beliefs on states!
            1. One of them is that we are still "greedy" since we are choosing k best states and lead to the idea that the best path always leads to the goal state, which is not true
        ii. Idea:
            1. Rebuild beam probabilistically (stochastic beam search)
            2. Instead of top k children: choose k at random
                a. Probability of choosing a child ~ objective value
                b. "natural selection"-ish
8. Genetic Algorithms
    a. Form of stochastic beam search
        i. Beam = population
        ii. Requires a "fitness function" (objective function)
            1. Larger values are better
        iii. Child state is product of two parent states (not one)
            1. Parents can be chosen deterministically

        2. Parents can be chosen probabilistically
        3. Mutation probability (once two parents form a child, do operations with that child)
   b. GAs can operate directly on states OR directly on agents
      i. We can calculate the hyperparameter of plateau threshold based on this method (run two different agents and check their efficiency)
      ii. After running two different agents, we can choose the one for children to run according to how the two agents performed
      iii. This also works for states (get two states and generate a state and continue until we find a goal state)
9. Theory of GAs
   a. Schema = Fixed part of the subject type (part of the query that matches parts of the individuals)
      i. ex) find all population that has threshold of 4
      ii. ex) find all x-coordinates at the upper left corner
      iii. Some properties fixed, other properties left free
   b. Individuals that match the schema = instances of schema
      i. Let S = {all possible schema individuals}
      ii. Let I = {all possible individuals}
         1. S is the subset of I
      iii. Let P^t = {population of individuals at time t}
         1. S is a bigger set than P^t

$$\mathop{\mathbb{E}}_{x\sim S}\left[obj(x)\right] > \mathop{\mathbb{E}}_{x\sim P^t}\left[obj(x)\right] \quad \uparrow$$

$$\forall t' > t \frac{\left|x \in P^{t'}\right| x \in S\big|}{|P^{t'}|} \geq \frac{\left|x \in P^{t}\right| x \in S|}{|P^{t}|}$$

   c.
      i. The average value of S > the average value of all current values
      ii. The objectivity value increases as time
      iii. Individuals that are better than the schema will appear gradually in my population (natural selection)
      iv. If threshold greater than 4 is good, it will start to appear in population than previously
10. GA Applications
   a. Good sampling algorithm
   b. Can have surprisingly large exploration of subject space!