

Monday, April 5

Lecture #29

Announcements

→ HW 5 assigned, due Wed, Apr 14

Topic #14 - Tabu Search

"Tabu" = "Taboo"

Think about H-C. You walk up a hill, and get stuck on the top. What do you do?

(1) Random Restarts

(2) Sometimes go downhill with a probability (S.A.)

(3) Like steepest ascent - go to the best location nearby that we haven't already been to, even if that's downhill (Tabu Search)

Main idea: * Keep a list of solutions we've seen.

* Do steepest-ascent hill climbing:

move to the best neighbor
that we haven't already seen.

Note: Only makes sense for discrete
problems (or discretizations of
continuous problems).

Small problem: if you store a lot of solutions,
it will be slow to check if
a new solution has been
seen or not

Big problem: it would take way too much
memory to store every full
solution you've seen

Fix #1: When you see a solution, you
add it to the tabu list for some
fixed # of iterations, called the tabu
tenure (L). Problem: cycling.

In code: `d = dict()` keys = solutions
 values = first time
 allowed again

Keep track of how many iterations you've done, and when you're leaving a solution S at iteration N , set $d[S] = N + L$

Whenever you want to go to a new state T , check $d[T]$. If T is not even a key in d , we're good. If it is a key, if the value is \leq current iteration, then T is allowed.

Problems

- cycling, you'll eventually repeat the same L solutions over and over
- still not ideal to store entire solutions

Fix # 2: Instead of remembering whole solutions to avoid, we'll remember moves to avoid.

Ex: Knapsack, $N=6$
 $\{1, 4, 5\} \rightarrow \{1, 3, 4, 5\}$ (add item 3)

Could remember:

* don't remove 3 for the next 20 moves

* don't add 3 again

* or some combination

Vague Pseudocode:

generation = 0

taboo = dict() (the first time a move is allowed)

taboo_tenure = 20

x = random elt of search space

while True:

generation += 1

neighbors = nbhd(x) # each neighbor is a pair (s,m), where s is the new solution, and m is the move that turned x → s

new_x, move = the pair (s,m) in neighbors such that m is allowed, and

the score is as high
as possible

$$\text{taboo}[\text{move}] = \text{generation} + \text{taboo-tenure}$$

S.A. 100: 8.881307