Midterm - Optimization Algorithms Solution #1645

Kai Chen STAFF
4 hours ago in Midterm – Optimization

2.1.1.1 - 2.1.1.7

2.1.2

In SA, suppose our temperature schedule is set to a constant value for N iterations and then set to 0. Select all that are True

- We can specify an N that guarantees an optimal state when stopped
- There is a certain amount of randomness while performing N iterations (selected)

2.1.3

In SA, we switch our point of view from (1) _______________ to (2) ________________ and imagine the task of getting a ping-pong ball into the deepest crevice in a very bumpy surface.

- Hill climbing (1)
- Random Walk
- Local beam search
- Genetic Algorithm
- Gradient descent (2)

2.1.4

Rank temperature schedules based on number of iterations to 0. (1, first schedule to approach T=0. 3, last to approach T=0)

1. Fast cooling schedule
2. Exponential cooling schedule
3. Logrithmic cooling schedule

2.1.5

The Logarithmic cooling schedule can be shown to converge to the set of optimal states given *c* be greater than or equal to the largest energy barrier.

Select reasons why you might **not** use the Logarithmic cooling schedule

- Converges to set of optimal states
- Speed of convergence is too slow (selected)
- We should always use the Logarithmic cooling schedule

2.2.1

GA are similar to ____________________, but with the addition of the crossover operation.

- Stochastic beam search (selected)
- Random Walk
- Gradient descent
- Hill climbing search

2.2.2

Select all that are true about Genetic Algorithms

- In GA, collection of individuals are utilized to guide progression towards an optimum (selected)
- GA utilize selection, crossover, and mutation to progress towards a better generation (selected)
- Local search approaches can aid in convergence for GA (selected)
- Without mutations GAs are guaranteed to find global optimum
- In GA, mutation drives exploration through randomness (selected)

2.2.3

Suppose you utilized GA and found a solution but you know that there is a better solution. Would you increase the mutation rate?

- yes (selected)
- no

Matthew Hofstetter 1h

Can we get the math behind the table values? I somehow got the right value for 4-1 and 4-2, but 3-1 is different, and 5-1, thus the acceptance probability is different as well.

1 Reply ···

Jun Zhu 1h

Exponential cooling overtakes fast cooling by the 7th iteration:

https://www.wolframalpha.com/input?i=1%2Fn+%3D+0.75%5En (for illustration, I didn't use this on the exam)

13 Reply ···

Sodiq Akanni Yusuff 36m

I agree. I in fact wrote a python script to compare both and exponential was way faster!

3 Reply ···

Travis J Munyer 30m

Exponential was way faster in my python experiments as well.

3 Reply ···

Chi Yung Patric Wong 2h

I don't understand the answer for 2.2.3. How does increasing mutation rate help find the better solution?

1 Reply ···

Sarabraj Singh 2h

I also don't understand. I thought it's not always a guarantee that increasing the mutation rate leads to better outcomes unless you're explicitly employing some sort of elitism in picking the next generation of parents.

1 Reply ···

Pardeep Singh 2h

I think it's less of that it would guarantee it, but if i have unlimited resources I can force the algorithm to keep randomly walking until it starts getting the known better solution.

In any random walk it's hard to guarantee a solution, but this would be an approach you take to tweaking your algorithm to get to a known better solution

Reply ···

Matthew Hofstetter 2h

Replying to Pardeep Singh

In this case, isn't there a chance of the algorithm never converging if mutation rate is just cranked up without an upper limit?

I think this question was just missing some clarifications.

1 Reply ···

Harrison Takuya Ooi 2h

Replying to Matthew Hofstetter

exactly what i was thinking. ill have to find the ed video but i feel like they said that somewhere too...

Reply ···

Jun Zhu 2h

The key is that you have unlimited computational resources. There's a line in the textbook that states that random walk is theoretically guaranteed to find the global optimum with enough computation and increasing mutation increases the stochastic element making it closer to random walk.

Reply ···

Suvan Satya Paturi 42m

would this idea not go against that in the book (1.3.3) or is this different, because from this I assume it meant despite having unlimited resources and changing mutations doesn't guarantee progress?

> The illusion of unlimited computational power was not confined to problem-solving programs. Early experiments in **machine evolution** (now called **genetic programming**) (Fried-Machine evolutionberg, 1958; Friedberg *et al.*, 1959) were based on the undoubtedly correct belief that by making an appropriate series of small mutations to a machine-code program, one can generate a program with good performance for any particular task. The idea, then, was to try random mutations with a selection process to preserve mutations that seemed useful. Despite thousands of hours of CPU time, almost no progress was demonstrated.

Reply ···

Jun Zhu 34m

Replying to Suvan Satya Paturi

Well those guys were using a Frinkiac 7 back then, not a Ryzen 7950X. But it's a purely theoretical question - you would never actually use brute force like this. The idea is just that pure randomness is immune to being trapped by local optimum and will find the global optima as computational time approaches infinity.

Reply ···

Michael Joseph Matsako 3h

For 2.1.3:

I was certain it should be Hill Climbing and Random Walk. Does anyone have a reference as to why it is not?

Thanks

2 Reply ···

Pardeep Singh 3h

4.1.2 Simulated annealing

A hill-climbing algorithm that *never* makes "downhill" moves toward states with lower value (or higher cost) is guaranteed to be incomplete, because it can get stuck on a local maximum. In contrast, a purely random walk—that is, moving to a successor chosen uniformly at random from the set of successors—is complete but extremely inefficient. Therefore, it seems reasonable to try to combine hill climbing with a random walk in some way that yields both efficiency and completeness. **Simulated annealing** is such an algorithm. In metallurgy, **annealing** is the process used to temper or harden metals and glass by heating them to a high temperature and then gradually cooling them, thus allowing the material to reach a low-energy crystalline state. To explain simulated annealing, we switch our point of view from hill climbing to **gradient descent** (i.e., minimizing cost) and imagine the task of getting a ping-pong ball into the deepest crevice in a bumpy surface. If we just let the ball roll, it will come to rest at a local minimum. If we shake the surface, we can bounce the ball out of the

SIMULATED ANNEALING

GRADIENT DESCENT

It's a direct quote from textbook

9 Reply ···

Matthew Bisson 3h

Oh wow. This is very frustrating. The only way that makes sense is in the context of the textbook because they were talking about hill climbing algorithms beforehand. The simulated annealing algorithm resembles a random restart transitioning into a gradient descent so I picked the same answer as Michael.

6 Reply ···

Sahil Jain 3h

Replying to Matthew Bisson

I also did the same.

2 Reply ···

Matthew Alexander Busch 2h

Replying to Matthew Bisson

I agree with you completely. 2.1.3 makes no sense on an exam. Only if you are talking about random hill climbing before the question

1 Reply ···

Pardeep Singh 2h

Replying to Matthew Alexander Busch

Yeah, I think i got lucky that I searched for it in textbook otherwise would've been in similar boat

Reply ···

Anh Hong Nguyen 2h

The lecture video on SA actually compares the first part (high temperature) to random walk and the second part (low temperature) to normal hill climbing.

2 Reply ···

Steven Candelaria 2h

Replying to Anh Hong Nguyen

Yup, lectures mentioned random walk and then normal hill climbing.

2 Reply ···

Harrison Takuya Ooi 3h

for 2.2.3, i feel that the question might be misleading (maybe its a skillbased issue on my end) but i dont recall them saying that increasning the mutation rate will give you the best answer. i know it can HELP to find a better solution but the way that the question was worded to me felt like it was implying that we need to keep finding the better solution (ergho best soliution) since we have "umlimited resources and contraints". and from what i recall, just because you crank up the mutation rate, it doesnt guarantee a best solution

2 Reply ···

Zhi Yao Tee 3h

Hi,

For 2.1.4, could I check why is Fast cooling faster than Exponential?

I simplified

$$T_{n+1} = yT_n$$

to

$$T_{n+1} = y^n T_1$$

When n -> inf , wouldn't (0.75)^n causes the term to tend to 0 faster?
(Not sure where my understanding went wrong)

34 Reply ···

SeungHui Huh 3h

same..

3 Reply ···

Alexis Martin 3h

Same!

3 Reply ···

Travis Sherwood Wheeler 3h

I plotted this and exponential was faster.

14 Reply ···

Pardeep Singh 3h

Yeah in my plottings Fast cool was initially quicker depending on T but exponential was always quicker after a bit.

1 Reply ···

Xueyuan Li 3h

Replying to Pardeep Singh

Same here.

Reply ···

Michael Ryan Prokopchuk 3h

Me too.

1 Reply ···

Piyush Mishra 3h

same here, I tried with multiple T1 values.

1 Reply ···

John M Hagood 3h

Replying to Piyush Mishra

The clarifications specified T=100, that's what I plotted on and Fast cooling was definitively fastest.

1 Reply ···

Piyush Mishra 3h

Replying to John M Hagood

As per clarifications

> For Optimization Algorithms, 2.1.1.1-2.1.1.7, assume T=100.0

I assumed it was not valid for 2.1.4 and thus plotted for different T values.

1 Reply ···

Michael Joseph Matsako 2h

Replying to John M Hagood

Same, posted answers seem right in this case.

Reply ···

Jun Zhu 1h

Replying to John M Hagood

T_initial is just a scalar in all 3 so it actually doesn't affect the rate. Exponential should be faster and the cross over point can be calculated with algebra 1/n = 0.75^n

https://www.wolframalpha.com/input?i=1%2Fn+%3D+0.75%5En (I didn't use this resource on the exam, just posting it for easy illustration)

2 Reply ···

John M Hagood 7m

Replying to Jun Zhu

Actually noticing now when I go back to my calculations, I did T(n-1)/n instead of T1/n -- it's possible that was the intent and I mistakenly got to the correct answer.

Reply ···

Sahil Jain 3h

Same here, I plotted and even I did 10^6 iteration for see what is the final values.

Reply ···

Gayatri Himthani 3h

Same!

Reply ···

Zanlang Yin 3h

Same. I plot the curves. Exp should be the fastest.

3 Reply ···

Vamsi Krishna Lingamaneni 3h

^^Same

1 Reply ···

Mohamed Aamir 3h

Same

Reply ···

Joel Parker 3h

Same, does anyone know if we need to all submit a regrade request if an answer in the key turns out to be wrong?

2 Reply ···

Piyush Mishra 3h

Guess we should wait for the update?

Reply ···

Thomas James Swope 3h

add one more to this list.

Reply ···

Wai Kuan Quah 2h

Same here. I am pretty sure

$$\left(\frac{3}{4}\right)^n << \frac{1}{n}$$

3 Reply ···

Nicholas Ian Miller 2h

same shows up clearly in plot

1 Reply ···

Jialiang Bai 2h

Same

1 Reply ···

Zachary Fisher Broome 1h

Same here

Reply ···

Anurag Lal 1h

Yeah. At n=100, 0.75^100 = 3.2072022e-13, while fast cooling only reduces by a factor of 0.01.

1 Reply ···

Erin Christine Deye 1h

I plotted exponential as faster than fast as well... this one should be awarded to either or because it's dependent on T

Reply ···

Cameron James Porteous 1h

Yup, with T=100, the exponential schedule will yield lower temperatures than fast cooling from n=9 onwards.

$$fast\ cooling: \ T_9 = \frac{100}{9} \approx 11.11$$

$$\exp: \ T_9 = 100 \cdot \left(\frac{3}{4}\right)^8 \approx 10.01$$

I agree that fast cooling *initially* approaches zero faster than the exponential schedule, but saying that fast cooling is the "first schedule to approach T=0" seems blatantly false.

2 Reply ···

Nicola Joseph Filicetti II 1h

Same

Reply ···

Gabriel Mark Wilson 46m

Adding a voice to the chorus, also same. I put exp above fast cooling because it clearly gets below 1 first with T = 100.

Reply ···

Ming Hang Poon 45m

Oh no.... It is T_n+1 = y * T_n so T_n = T_n+1 / y ....

1 Reply ···

Matthew Bisson 45m

I just noticed that too

Reply ···

Ming Hang Poon 44m

Replying to Matthew Bisson

Yup... sigh...

Reply ···

Yu Wang 40m

Same here.

Reply ···

Xueyuan Li 3h

Why is cell 2-2 not 1? e^(-(-5.070723)/100) = 1.05 > 1.

1 Reply ···

Jongmin Kim 3h

I think you have the delta T flipped, it should be 5.070723 not negative

Reply ···

Xueyuan Li 3h

Is delta t not 302.610378 - 307.68110?

Reply ···

Hyee Sun Chun 3h

Replying to Xueyuan Li

The other way. Delta final - initial.

Reply ···

Xueyuan Li 3h

Replying to Hyee Sun Chun

Thanks, but I'm still a little confused, doesn't it say Value(current) - Value(next) in the textbook on page 115?

Reply ···

Hyee Sun Chun 3h

Replying to Xueyuan Li

In the text book in the last line of the algorithm, e^ΔE/T. If you want to use the Value(current) - Value(next), you need to use e^ΔE/T instead of e^-ΔE/T which is the formula given in the problem.

Reply ···

Xueyuan Li 3h

Replying to Hyee Sun Chun

Section 4.1 Local Search and Optimization Problems 115

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  current ← problem.INITIAL
  for t = 1 to ∞ do
     T ← schedule(t)
     if T = 0 then return current
     next ← a randomly selected successor of current
     ΔE ← VALUE(current) – VALUE(next)
     if ΔE > 0 then current ← next
     else current ← next only with probability e^(–ΔE/T)
```

Figure 4.5 The simulated annealing algorithm, a version of stochastic hill climbing where some downhill moves are allowed. The *schedule* input determines the value of the "temperature" T as a function of time.

I see e^-ΔE/T 😭

Reply ···

Dana Tareq Alnabulsi 3h

Replying to Xueyuan Li

I tried it both ways and tested against given probabilty value in row 3 of 1.000000. Whatever matched that was the method I used

3 Reply ···

Xueyuan Li 3h

Replying to Dana Tareq Alnabulsi

All my other values for this question are the same since the delta T's were positive so I figured the acceptance probability was 1, which I guess was misleading for 1-2. Still not sure why delta t is next - initial though...

Reply ···

Ayesha Usman Ilyas 2h

Replying to Dana Tareq Alnabulsi

Same, I feel the lectures didn't help much for this question.

Reply ···

Hyee Sun Chun 3h

Replying to Xueyuan Li

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**
    $T \leftarrow schedule(t)$
    if $T = 0$ **then return** *current*
    next ← a randomly selected successor of *current*
    $\Delta E \leftarrow next.\text{VALUE} - current.\text{VALUE}$
    if $\Delta E > 0$ **then** *current* ← *next*
    else *current* ← *next* only with probability $e^{\Delta E/T}$

Mine has different version. But I think the question is different here. The algorithm in the book is for a stochastic hill climbing where some downhill moves are allowed. So we are trying to find the peak (maximum). But in our test, we are finding optimal path (minimum).

Reply ···

Pardeep Singh 3h

Replying to Hyee Sun Chun

Thread where this was discussed:
https://edstem.org/us/courses/50169/discussion/4289142?comment=10102876

There's different versions depending on what delta E is defined as.
I asked a clarifying question on this tho, to verify

Reply ···

Xueyuan Li 3h

Replying to Pardeep Singh

AH I see, Thanks! The exam didn't really specify if we were maximizing or minimizing though...

Reply ···

X Xueyuan Li 2h

Replying to Pardeep Singh

An "Optimal Path" based off "Energy" sounds like minimizing the energy however. And probably in the question uses E(-delta T / T) instead of E(delta T / T)

Reply ···

P Pardeep Singh 2h

Replying to Xueyuan Li

Yep and that’s why 2-2 isn’t 1 tho. Since it increases energy from previous path. While all the others are 1 since they decrease energy from previous path

Reply ···

X Xueyuan Li 2h

Replying to Pardeep Singh

That makes sense, but I thought the thread implied we would use Value(current) - Value(next) for minimizing.

Reply ···

P Pardeep Singh 2h

Replying to Xueyuan Li

Yeah I was surprised they didn't post a clarification after I asked since I pointed out that Paul said that.

1 Reply ···

P Pardeep Singh 2h

Replying to Xueyuan Li

I think that was if it was delta E, if it's -delta E, you have to do the other way. That's how both approaches end up being correct cause they evaluate to same thing when the negative is distributed

Reply ···

X Xueyuan Li 2h

Replying to Pardeep Singh

That kind of threw me off since in the lectures and the 3rd edition of the textbook, its maximizing with positive delta and (next - current).

But its minimizing with negative delta and (current - next) in the 4th edition.

And minimizing? with negative delta but with (next - current)? in this problem.

Reply ···

X Xueyuan Li 38m

Replying to Pardeep Singh

The thread implies the opposite I think

Reply ···

X Xueyuan Li 2h

Replying to Pardeep Singh

Regardless, it would have been nice if they publicly clarified this. Especially since they mentioned "We will make this clear if needed on the exam" in that thread. Thanks for the help though.

Reply ···

Sahil Jain 4h

Hi,

Will there be partial marks for 2.1.3 and 2.2.2?

8 Reply ···