

Intro to Computer-Aided Digital Design

ECE Department

Out: 4/13/04

Homework 7

Due: 4/27/04

We'll discuss the solutions to these problems at the beginning of class on the due date. Bring your solutions then. Solutions will be handed out, so we can't accept late assignments.

1. Material Science here I come!

40%

Use the simulated annealing program `anneal.c` to investigate the effect of a cooling schedule. For a brief description of the program, see after the other questions here. For the following, use the input file `big.ex`: the run should take a few minutes on most decent sized workstations. Files `anneal.c` and `big.ex` are both in `/afs/ece/class/ee360/anneal`.

- Good values for the problem are `HOT = 20.0`, `COOL = 0.95`, `MOVESPER = 30`, `RANGE = 0.98`. Run the code selecting a value for the random seed. Now plot the **cost** values you got versus **temperature**. You may do this by hand, but if you want to be efficient you can write a Perl or Awk script to yank out the values from the output file and stick it into your favorite plotting package.
- Do part a again with a different random seed. How different are the costs? Plot them together.
- Now change the value of `COOL` to `0.7` and rerun with the random seed of the best of the two above results. Plot the cost versus temperature on the same graph with the data from the best above. Comment on the differences you see in the results between these two.

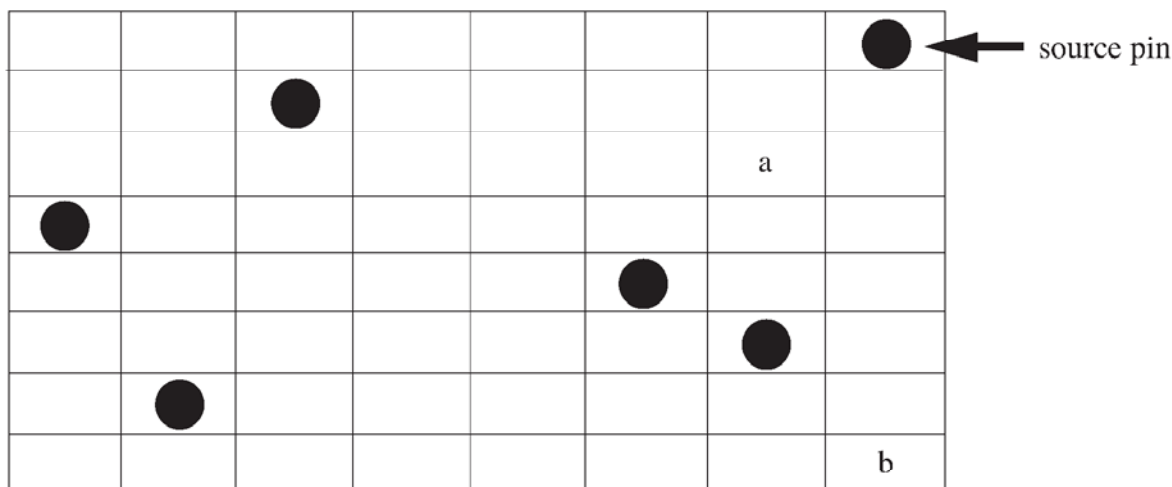
All you need to turn in are the plots of the cost lines (properly labeled and annotated) and your comments regarding parts b and c. I don't want to see pages of output from the annealer.

2. Wired

25%

Below is shown a wiring grid and group of pins that need to be wired together. There is a cost of 1 to run a wire through a single box in the wiring grid. Compute the estimated wire length and draw the estimated wiring that this length is based on for each of the following methods. On the diagram, consider a step across a line to count as one. Thus if "a" and "b" were nodes on a net, they would be a Manhattan distance of 6 (5+1) apart. In the parts below, don't consider a and b to be part of the pins to connect.

- minimum spanning tree - build minimum spanning tree for the nodes
- Steiner tree - build minimum length connections where the sink node just needs to attach to the nearest connection that will lead back to the source
- half perimeter - as shown in the lecture notes
- Explain the some of the good and bad points of each of these 3 estimation methods.



3. My finals will be over early

35%

So, how would you do final exam scheduling through simulated annealing? How do you model the problem? What is the move set? Propose a cost function. Take about a page or so to explain how you would set this problem up.

Description of program `anneal.c`:

Anneal is a simple simulated annealing program targeted towards a placement problem, where there are pads around the perimeter of an area and gates that must be placed inside the area. The goal of the annealer is to minimize the length of the wires connecting the gates. Each gate is assumed to have the same area and must be placed on a rectangular grid in the area.

Compile `anneal.c` on your favorite machine like this:

```
gcc -O anneal.c -lm -o anneal
```

The command line for `anneal` is

```
/anneal input_file SEED HOT COOL MOVESPER RANGE > output_file
```

where:

input_file is the name of the netlist file that you will input. Below is the annotated example input file `ex.tiny`.

SEED is a big random integer (like 1234567), which starts the random number generator in the code. The random number is used in the Metropolis criterion inner loop: first the probability $p = \exp(-\Delta E/kt)$ is calculated. Next a random number between 0 and 1 is generated. If the generated random number is greater than p , the move is accepted; else it is rejected.

HOT is the starting temperature as a floating point number, e.g. 200.0.

COOL is the cooling rate, a fraction less than 1, e.g., 0.9. In the annealing algorithm the temperature cools as

```

7 7
1 p 0 6 1
2 p 0 0 2
3 p 6 6 3
4 p 6 0 4
5 g 3 1 5 9
6 g 3 5 6 10
7 g 3 6 7 11
8 g 3 7 8 12
9 g 3 3 8 13
10 g 3 9 14 18
11 g 4 10 14 15 19
12 g 4 11 15 16 20
13 g 4 12 16 17 21
14 g 3 13 17 22
15 g 3 18 23 27
16 g 4 19 23 24 28
17 g 4 20 24 25 29
18 g 4 21 25 26 30
19 g 3 22 26 31
20 g 3 27 32 36
21 g 4 28 32 33 37
22 g 4 29 33 34 38
23 g 4 30 34 35 39
24 g 3 31 35 40
25 g 3 2 36 41
26 g 3 37 41 42
27 g 3 38 42 43
28 g 3 39 43 44
29 g 3 4 40 44
-1

```

The placement grid is a 7x7 array of slots. The numbering is 0..6 for rows, 0..6 for columns. Rows 0 and 6 are for pads. Same for columns 0 and 6.

There are 4 pads in this example, placed at fixed locations around the edges of the grid. Each object - pad or gate - has an integer ID. This line says object 4 is a pad, it's at x=6, y=0 in the grid, and it connects to electrical net 4.

This is a gate. It has ID 16. It has 4 pins on it. In order, the 4 pins connect to the nets numbered 19, 23, 24 and 28.

In this format, the objects get numbered sequentially starting with 1. Objects include pads (which do not move) and gates (which do). Nets are also numbered starting from 1. You have to read all objects to figure out how many nets there are and what they connect to. Nets can connect to several objects, i.e., it's not just 2-point connections here.

$$T_{\text{new}} = \text{COOL} * T_{\text{old}}$$

MOVESPER is an integer that determines how many moves we do at each temperature. We multiply MOVESPER by the number of objects and we do this many moves at each temperature to get equilibrium. For example, if you set MOVESPER to 100 and you have 100 objects to place, you do 10,000 moves at each temperature.

RANGE is a floating point fraction < 1 which determines how fast we shrink the range limiting window during move generation. Typical example of RANGE is 0.98. The closer RANGE is to 1, the better your result is, but the longer it will take.

output_file is where you want to see the results. The code just writes to the C stdout. The code writes out all the settings you picked, some characteristics of the grid and also a line of info for each temperature. Such a line typically looks like this:

The 8th temperature of the run was 47.8

The cost at the end of this placement temperature was 152. Mean cost over all moves was 155.237

variance seen in cost values at this temp was about 103.2

run 8 T 47.8297 att 2900 acc 2787 accratio 0.961034 cost 152 meancost 155.237 GlobalRangeLim 4 varcost 103.235

We attempted 2900 moves at this temperature, accepted 2787 with an acceptance ratio of 0.96

Range limiter window size was 4 here