03-512/02-512/03-712/02-712	Computational Methods for Biological Modeling and Simulation
Wean Hall 5310	Tuesdays and Thursdays 3:00pm-4:20pm
Lecturer: Russell Schwartz	

Problem Set 1

Assigned 9/4/14, Due 9/18/14

Problem 1 (15 points): The goal of this problem is for you to take informal descriptions of biological questions and cast them as known optimization problems. For each description, identify a classic computational problem we have studied that would solve the informal biological problem and describe how to convert inputs and outputs between the formal and informal problems.

To give an example, here is a similar problem from a prior year:

> Suppose we want to study differentiation of cell types during development of an organism. We examine expression of a large set of cell surface proteins and determine which ones are present in each cell type. Assume that each cell type expresses a different subset of the available cell surface proteins. We would then like to then create antibodies for a few of these proteins, attach fluorescent labels to these antibodies, and use them to label cells in an embryo. We would like to choose the antibody targets so that we can identify the type of each cell based on the labels attached to it. How can we find a minimum number of surface proteins sufficient to label every cell type?

A reasonable solution to this problem would be:

> We can cast this as a minimum test set problem. Our set S is the set of cell types and our collection of subsets C is the set of subsets of S corresponding to cells possessing each surface protein. The minimum test set will then identify a set of cell surface proteins that allows us to distinguish every cell type.

The systems to address are the following:

a. Suppose we are trying to predict how a flexible molecule might bind to a protein. We assume that the protein has a set of binding sites and the molecule has a set of chemical groups that can dock in those sites. Each chemical group has a distinct docking energy for each possible binding site. We would like to know how to position chemical groups into binding sites, with at most one group per binding site, so as to find the best possible way to dock the chemical groups into the sites. Assume the molecule is flexible enough that is can place any chemical groups into binding sites in any combination with equal ease.

b. Suppose we are doing assays to test the effects of many possible drugs on samples of bacteria to try to discover new antibiotics. We have a lot of tests to run, so we buy a robotic assay machine that lets us run many tests in parallel. Some pairs of potential drugs react with one another, though, and cannot be used on the machine at the same time. We would like to test all of the possible compounds while running as few rounds of assays as possible.

c. Some of the recent technologies for DNA sequencing are able to produce much longer reads than prior sequencers but have proven difficult to use in practice because they have a high rate of deletion errors (missing bases), which are difficult to handle computationally. Suppose we would like to use such a sequencer to assemble viral genomes, assuming that the benefits of long reads

will outweigh the disadvantage of having to handle deletions. We want to assemble a genome from a set of such reads, assuming for simplicity that reads can have many missing bases but have no extra insertions or misreads.

d. Suppose we are studying the transport of a molecule through a set of tissues. We know that there are many possible tissues through which the molecule might transit and have an idea of the probability of the molecule passing from any one tissue to any other. Assume we know it is injected in one tissue and is eventually delivered to a different target tissue. We would like to figure out the most likely series of steps it takes to get from the injection site to the target tissue.

e. Suppose we are studying a DNA-binding protein that we believe binds to a specific DNA sequence that we would like to identify. We use the protein to affinity purify a pool of random strands of DNA to identify those that bind to the protein and then sequence those strands. We would like to find the specific DNA sequence that is actually bound by the protein.

Problem 2 (20 points): In this problem, we will consider some potential ways we could model a hypothetical problem drawn from image analysis. Let us suppose that we are neuroscientists and have come up with a way of placing a set of fluorescently labeled proteins into an individual neuron and taken a series of microscopy images of slices of a piece of brain containing that neuron. We can think of the slices as forming a three-dimensional grid of pixels in which a random subset of pixels from the neuron are lit up. We would like to construct a 3D model of the labeled neuron by indentifying the path its structure takes through the labeled pixels.

a. As a first pass to solving this problem, we might propose to link the labeled pixels into a graph that models the shape of the neuron, trying to minimize the total amount of Euclidean distance we need to travel between labeled pixels to link them together. Write up a formal specification of this problem in terms of inputs, outputs, and objective function. (Note: you may find it helpful to assume some preprocessing to simplify the input.)

b. The model you have developed should be solvable as a special case of a classic discrete optimization problem. Identify the problem and use that to suggest an algorithm to solve it.

c. We might decide that the model from parts a-b is not really so good, since it could miss important branch points in the actual neuron. For example, if we have three fluorescently labeled pixels A, B, and C, it might be that the shortest way to connect them would be to propose that the neuron passes from A to an unlabeled pixel D and then from D to B and D to C. The model from parts a-b would not be able to find such a solution. Repose your problem to allow for the possibility of introducing unlabeled pixels into the neural model as a way of reducing the objective value.

d. The model you developed in part c should also be a special case of a classic discrete optimization problem. Identify that problem. This problem should unfortunately be intractable and thus not suggest any particular efficient optimal algorithm. Instead, suggest in a sentence or two an appropriate algorithmic strategy we could use for this model if we assume that we generally have large problem instances but are willing to tolerate suboptimal solutions.

e. If we are presented with some real image data and have to choose between using the model of parts a-b or the model of parts c-d, how would we decide? Identify properties of the data that should lead us to favor the first model versus those that would favor the second model.

Problem 3 (20 points): Suppose we want to sequence a piece of DNA. We have identified the following fragments of the full DNA molecule:

1. AATGTGCGCT
2. CGTTGTAATGT
3. GTACGTTG
4. CGCTAATG

a. Model the assembly of this set of fragments as a traveling salesman problem. Show the graph and edge weights and label the nodes according to the sequences they represent.

b. Find the shortest common superstring of all of the input fragments.

c. Now model the problem using the Eulerian path shotgun (de Bruijn graph) method with k-mer length 4. Show the graph induced by the 4-mers, with nodes and edges labeled by the sequences they represent. Does the graph have a unique solution? If not, why not?

d. Now model the problem using the Eulerian path shotgun method with k-mer length 5. Show the graph induced by the 5-mers, with nodes and edges labeled by the sequences they represent. Does the graph have a unique possible solution? If not, why not?

e. What is the shortest possible string consistent with the graph of part d? Is it the same as the shortest common superstring of the fragments? If not, why not?

Problem 4 (25 points): This is a programming problem in which we will address a hypothetical issue in modern cancer treatment. One of the big advances in cancer treatment has been the development of what are called "targeted therapeutics," which are a class of drugs that are each custom-designed to treat cancers with a particular kind of mutation. The idea is that if we can figure out which mutations a particular patient's tumor has, then we can prescribe that patient a drug designed for their particular mutations. One problem with that approach, though, is that tumors are highly heterogeneous: they actually contain many genetically distinct cells each of which might have its own mutations. A drug that works against some cells in the tumor may be useless against others, so prescribing a drug that works against most of the tumor may just cause it to temporarily shrink before recovering in a drug-resistant form.

We might propose instead to try to find all of the mutations in all of the cells of a tumor and prescribe all of the targeted therapeutics that seem relevant at once. That is not likely to be very good for the patient, either, though, because some of these drugs may interact and prove excessively toxic to the patient. That is, maybe a patient can tolerate drug A or tolerate drug B, but will be poisoned by taking drugs A and B together. We propose to try to handle this problem by identifying all of the toxic interactions and then prescribing a cocktail of drugs that contains as many therapeutics as possible without containing any toxic pair of drugs. We will assume for the moment that only pairs of drugs are toxic and there are no higher-order effects (e.g., we will never have a case where drugs A, B, and C together are toxic but any two of the three are safe).

a. Provide a formal statement for the computational problem we wish to solve. You can assume that we are given a set of drugs that are individually likely to help the patient and a set of pairs of drugs that are toxic when taken together.

b. What well known computational problem that we have seen before provides a model for the problem we wish to solve here?

c. You might notice that the problem you identified in part b is intractable. We might decide that this is a case in which we would really like to have an optimal solution to the problem anyway, since we would normally not have too many drugs to consider and a better outcome for the patient is worth waiting a while for a compute job to finish. Provide pseudocode for a brute-force solver that will guarantee an optimal (maximum-size) cocktail of therapeutics for a given set of input constraints. (Hint: You may find this easiest to write as a recursive program.)

d. Write code implementing your algorithm from part c. Your code should take as input a number of drugs, a number of pairwise constraints, and a set of lines each listing a pair of drugs that cannot be delivered together. For example

```
5
3
0 1
2 3
2 4
```

would describe a problem instance with five drugs and three constraints: drugs 0 and 1, 2 and 3, and 2 and 4 cannot be delivered together.

Your code should output a list of drugs for a maximum-size cocktail given the constraints. For example "0 3 4" would be a correct output for the input given above.

e. Run your code on the problem instance above and on the following harder test case

```
6
7
0 1
0 2
1 2
2 3
3 4
4 5
5 1
```

and return the results for the two cases. Feel free to supply other test cases that you feel show the effectiveness of your code.

***Problem 5 (20 points):** In this problem, we will continue working with the drug cocktail model from problem 4. The number of targeted cancer therapeutics is constantly growing, so we will suppose that we might need a faster algorithm than the brute-force approach you coded for problem 4. We will consider below a few possibilities.

a. A friend of yours suggests that you can use an approximation algorithm to turn your brute-force solver into a faster branch-and-bound solver. He has an idea for a simple approximation algorithm:

1. Pick any drug D at random and throw it into the cocktail
2. Remove all drugs that interact with D
3. If any drugs are left the return to step 1, otherwise return the current cocktail.

Your friend suggests that this must be an approximation algorithm for some value of α. Prove your friend is mistaken by finding a counterexample for any possible value of α. That is, imagine that you are given some arbitrarily small α and show an example for which your friend's algorithm could return a solution that is worse than an α fraction of the best possible solution.

b. Since the approximation algorithm approach has not worked out, our friend suggests that maybe we can come up with an integer linear program (ILP) to solve the problem. We can propose a program in which we define a $\{0, 1\}$ variable x_i for each drug v_i. Setting $x_i = 1$ will mean that the drug is in the cocktail and $x_i = 0$ that it is not. Given this definition, show how to express as a linear inequality the constraint that a pair of disallowed drugs (v_i, v_j) are not both in the cocktail.

c. Use your answer in part b to define a complete ILP that will solve the drug cocktail problem, specifying the full set of variables, constraints, and objective function.

d. Suppose we want to move to a more realistic model of the problem, in which we allow that there may be arbitrary groups of drugs that are collectively toxic rather than just toxic pairs. For example, it may be that drugs A, B, and C together are toxic even though A+B, B+C, and A+C are all safe. Modify your formal problem statement and ILP to handle this more general version of the problem.

*Recall that the starred problems are only for the 03-712/02-712 students.