CS/CSYS/Biol 352 Homework 1
Assigned: Tuesday 9/4/12
Hardcopy of hand-drawn predictions due in class: 9/11/12 (I encourage you to complete these earlier)
Softcopy Code and Write-up due in BB: Thursday 9/20/12 by 9:00 am
Hardcopy Code and Write-up due: 9/20/12

Use the GA toolbox (write a batch driver, don't use the GUI) to tackle the N-queens (which is interesting for $N \geq 8$) problem roughly as described in section 2.4.1; you will need to create a custom representation for the permutation vector (hint: see the Matlab function randperm for initialization) and operators. A few things will need to be different than described in the text. For example, the GA toolbox only returns 1 offspring from each recombination, so when you implement the cut and crossfill crossover (Fig 2.3) you will only need to make child 1. Note that the type of parent selection they're describing is called tournament selection; in the GA toolbox each parent is selected independently, however, so instead of picking best 2 out of 5 just indicate a tournament size of 3 (so each parent is picked independently as the best 1 out of 3). The GA toolbox is a 'generational' GA, meaning that all the children replace all the parents, so there really is no survivor selection (assuming elitism is set to zero, which you should do for this experiment). Also, for the termination condition, let's say you'll stop only if you find a complete solution or if you hit MaxGens generations. Finally, the GA toolbox applies mutually exclusive recombination and mutation, so you can't set these probabilities independently; rather, set the crossover fraction to 0.5, so we get half and half (this is not the recommended setting, but is for the purposes of the experiment you will conduct). For your final set of experiments, use PopSize = 100 and MaxGens = 500, although I suggest setting these values much lower for debugging during code development. To make the problem more difficult, lets try N=16 instead of N=8. I doubt this will be (reliably) solved with this popsize and maxgens, but we should be able to make good progress.

Our goal is not so much to solve the N-queens problem as it is to explore the impacts of the different evolutionary operators. Thus, what I want you to do is to keep track of how crossover and mutation are each impacting fitness, and see how (if at all) this is related to the mean and/or standard deviation of the population fitness (one measure of population diversity) at the *start* of the generation (i.e., before mutation and crossover) in which the change occurs. In order to do this, you'll have to add some accounting code into to your evolutionary operators. For every mutation, record the amount by which fitness improved or didn't improve. For every crossover, record this amount relative to the most fit of the two parents. Unfortunately, the GA toolbox assumes only 1 output argument from the mutation and crossover operators. Thus, for the sake of not having to get in and muck with their code, just use global variables for recording the accounting data so you can access it after the run is complete.

I suggest you store this info in two PopSize/2 $\times$ MaxGens element matrices, one for mutations (Mchange) and one for crossovers (Cchange); each row will correspond to a child and each column correspond to a generation. Preallocate these matrices using NaN(PopSize/2,MaxGens), so unused spaces will be marked. You can keep track of the population statistics using a custom output function, similar to what is shown in the lab tutorial, although you will have to modify this code to start the record with the initial population. Also, I suggest you preallocate any fields you use in the history structure with vectors of NaNs to speed things up (avoids a lot of dynamic memory reallocation). Feel free to simply edit my gaoutput function to suit your needs, but don't leave stuff in there you're NOT using for this experiment.

Make a batch program to do 10 repetitions of this, using different seeds 1 to 10 for the RNG. E.g.,

```
for seed=1:10
     rand('seed',seed); % set the seed used by randperm and rand
     % do the run
end
```

I want to you make several plots of the results, as follows:

1) Make a plot of mean population fitness (y-axis) as a function of generations (x-axis), using different lines for the 10 different runs
2) Make a plot of std of population fitness (y) vs generations (x), using different lines for the 10 different runs
3) Make a plot of the mean fitness change caused by mutations (y-axis) as a function of generations (x-axis), using different lines for the 10 different runs
4) Make a plot of the mean fitness change caused by crossover (y) vs generations (x), using different lines for the 10 different runs
5) Make a scatter plot of all fitness changes (both bad and good, from all 10 runs lumped together) caused by mutation (y) vs the corresponding standard deviation of the population before the mutation (x)
6) Make a scatter plot of all fitness changes (both bad and good, from all 10 runs lumped together) caused by crossover (y) vs the corresponding standard deviation of the population before the crossover (x)

For those of you less familiar with Matlab, I've posted tricks.m to show you a few handy vectorized operations you may find useful for plots 5 and 6.

However, BEFORE running any code, I want you to sketch roughly what you expect you might see in each of these plots. Briefly write down your guiding rationale/intuition in making these predictions.

Once you've completed the experiments and have the real plots, prepare a short write up that includes your (well-labeled with legibly sized fonts!) plots and discusses the results, addressing the following questions/points. IF you were not able to get a working code to make your own plots, or if you think your code is buggy and your results look different from those of other students, then please ask someone else for their results and do your write up based on their results (just make it clear you did this).

1) How big is the search space for this problem, and how does this compare to the number of function evaluations needed by your GA (not counting any extra function evaluations you have done in order to assess fitness changes)?
2) Were your plots created by your own code (if not, then whose). If not, then please describe how far you got with your own code, describe any bugs you know or suspect your code has, and tell me how many points out of 10 you think your code is worth.
3) Try to explain the observations in the plots. Did you see what you expected to see? If not, do you think this was because of any or all of the following:
    a. flaws in the experimental design? If so, suggest a better experimental design that you would try next in an attempt to resolve the issue, and explain why you think this might help.
    b. your original intuition and rationale was simply wrong? If so, see if you can come up with an alternative hypothesis that is consistent with what you observed.
4) Based on these observations, can you get any intuition about what makes sense for initializing the crossover fraction, and whether this should change dynamically or be static throughout the run?
5) Briefly outline an experiment to test to see if your idea for setting crossover fraction is a good one. Please specific about what you would compare and give some idea of how you would analyze the comparison.

HAND IN:

1) Upload all m-files and a pdf of your write-up into BB. If you worked in a pair, also upload your pair statement (see below).
2) Stapled hardcopy of your write-up and code. Only include Matlab functions or scripts *you* wrote or modified, even if the code doesn't work and even if you based your write up on someone else's results; you don't have to document your code for this assignment excessively, but DO include your name in all files and please insert comments to clarify particularly confusing pieces of code. IF you used someone else's plots in the write up, then include printouts here of what plots (if any) your code makes.

RULES REGARDING COLLABORATION ON THIS ASSIGNMENT:

1) You must sketch the predictions on your own, without discussing this with others or showing them your predictions.

2) Each student must write all of their own code. However, you are allowed to discuss aspects of the programming with each other and help other students over humps if you have the time and the will. You MAY ELECT to find someone to pair up with for the programming part.

   If you want to work in a pair, send me an email by Wed noon and I'll help pair people up; feel free to suggest who you would like to pair with if you know, and please tell me where your office is so I can try to pair you with someone nearby.

   Pairs may discuss ways to get started in tackling things, and otherwise give each other pointers and suggestions, but you still need to be writing your own code and doing (most of) your own debugging. You should not be working so closely together that the codes are virtually identical.

   IF you work in a pair, you must each confidentially and independently outline the nature and degree of help you received from and gave to your partner, and indicate to me how well you felt this type of pairing worked for you. Do not discuss or show this to your partner. Upload this to BB as a pdf, but do not hand in hardcopy.

3) All students are encouraged to compare their plots to those of others in the class – if your plots look really different from others, you may have a bug.

4) If you do not have what you think are correct plots by Tuesday Sept 18, you may get a copy of what you think are correct plots from another student and do the write-up based on these. You must still hand in your buggy or incomplete code and clearly state on the write-up whose plots you are using.

5) You must do the interpretation and write-up on your own, without discussing this with others or showing them your writeup.

Grading:

1) Thoughtful predictions: 5
2) Code: 10
3) Writeup: 10