CSCI 4560/6560 Evolutionary Computation

Assignment Number 3: Due Tuesday 10/30/2001 **(in class)**

1. [**50 points**]

   The N-queens problem generalizes the 8-queens problem to the goal of placing N queens on an N by N chess-board such that no queens attack each other.

   Use a genetic algorithm to solve the N-queens problem. You may download and use an existing GA or implement your own. You should try to solve the problem for the highest possible value of N that your program can handle in a reasonable time. Turn in a printout of your code and your solutions. If you use an existing package, you should only turn in a printout of the parts you modify/introduce, such as the fitness function.

2. [**50 points**]

   The Subset-sum problem is stated as follows. Given a set of N positive integers $X = \{x_1, x_2, \dots, x_n\}$ and an integer S. Find a subset P of the set X such that the sum of the elements of P is equal to S. For example, if N=5, S=21 and the set $X = \{12, 17, 3, 24, 6\}$, the set $P = \{12, 3, 6\}$ is a valid solution for the Subset-sum problem in this example.

   Use a genetic algorithm to solve the Subset-sum problem. You may download and use an existing GA or implement your own. You should try to solve the problem for the highest possible value of N that your program can handle in a reasonable time. Turn in a printout of your code and your solutions. If you use an existing package, you should only turn in a printout of the parts you modify/introduce, such as the fitness function.

Note: You can use the same program (with some modifications) to solve both the N-queens and the Subset-sum problems. The only part that **needs** to be different is the fitness function. On the other hand, if you choose to write two different programs you are welcomed to do so as long as you finish in time.