Homework 5: Genetic Algorithms
The Knapsack Problem

Due Date: Tuesday March 13, 2007.  Handin name: knapsack

Using a genetic algorithm, come up with a solution to the knapsack problem. In this problem one assumes one has a container (knapsack) of capacity $C$. Furthermore, there is a set $N$ of objects in front of you. Each object $i \in N$ has a size $s_i$, and a value $v_i$, where $s_i, v_i \in \mathbb{N}$. A solution is a subset of the $M \subseteq N$ such that

$$\sum_{i \in M} s_i \leq C$$

and that

$$\sum_{i \in M} v_i$$

is maximized.

As with any genetic algorithm one must decide:

- The size of the chromosome pool for the start of each successive generation.
- The number of generations you will run the optimization.
- How mutation occurs, if at all, and which chromosomes get to mutate.
- How crossover occurs, if at all, and which chromosomes get to participate in crossover.
- How selection decisions will be made: standard method, rank method, or the rank-space method.

As with the 8-Tile programming task, late work will be accepted at a 5% per day penalty. Also, you will be responsible for signing up for a one-on-one grading session.

Level of Achievement:

- B level: A working implementation that provides a reasonably decent answer.
- A level: In addition to the C-level work, your implementation should be sufficiently “generic” to allow for the easy modification of any (all?) of the above parameterizations. (i.e. pool size, selection algorithm, etc.) Hence these values would be specified as parameters instead of being hard-coded into your solution.
- A+ level: All of the above along with a correct implementation of the rank-space selection method.