CS 5320/4320

Project #4: ConstraintHandling

Points 100, Due: April 11

Note: Different tasks for the undergraduate and graduate sections

In this project you will implement a binary-coded (bit-string) GA to solve the 0-1 knapsack problem of size 20. Create (possibly randomly, within some bounds) your own weights w_i's, values v_i's and knapsack capacity W (being able to create meaningful problem instances is an important learning activity), where i runs from 1 to 20. (Study the discussion in Section 4.5 of Michalewicz on the difficulty levels of different types of knapsack problems.) Implement proportional, stochastic binary tournament, and linear ranking selection in the same program (allow the user to choose one of these three).

- The undergrad section will implement one of the static penalty approaches.
- The grad section will do both: (one) static and (one) dynamic penalty methods.

Explain in the report the penalty methods you used. Include in your results two extreme cases (the one where no single item can be fit in the knapsack and the one where the problem is trivially solved, i.e., the knapsack can hold all the items), in addition to one “regular” case. [This means a total of three problem instances will need to be created and solved.]

The output should include the problem statement (weights, values, capacity of the problem instance) and averages over 10 independent runs of the best-of-run solutions on each of the three instances. Submission and group instructions are as before.