Fiona Heaney & Pat Lebold

CS 4341 C16

Assignment 5 – CSP

**Instructions for compilation**

* Our program require Java 8
* Run: java – jar BagsAndWeights.jar <input\_file.txt>

**Our Approach**

* Create list of sub-lists of objects that are casted to constraint objects, weights, and bags
* Create a flag, isValid that if false, terminates the DFS backtracking search for a node. If true, we know that the currently placed items satisfy all constraints. Additionally create a global flag for validity (allFinal) that ensures all bag placements so far are valid.
* Recursively call solve method on the remaining list of unplaced items (weightsToTest).
  + This is the list we modify with the LCV and MRV algorithms, ultimately reordering the list of remaining items to optimize placement
* Begin placing items with DFS

(For each bag in the list of bags)

(For each of the unplaced items)

(If item can be placed in this bag)

Remove from unplaced item list

Add item to this bag

Recursively call solve on remaining items

Remove item from bag and place back into unplaced item list if recursive call returns invalid or no solution.

* Adding heuristics

Add things here for the love of god

**Testing**

* After implementing just our DFS, we ran our program with all 26 sets of test data. We found that it correctly placed the first 24 input files correctly but could not efficiently compute the placement of items for file 25 and 26. The estimated run times can be seen in trace files by uncommenting our code, however they ranged anywhere from XXXXXXXXXXXXXXXXXXXXXX to ZZZZZZZZZZZZZZZZZZZZZZZZZZ We then implemented our heuristics
* Add things here for the love of god
* We found that our program performance for MOAR HERE

**Strengths and Weaknesses**

* NEED ALL CODE FIRST

**Comparison of CSP Algorithms**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *BT* | *BT + MRV* | *BT + LCV* | *BT + Tiebreaking* | *BT + Heuristics + FC* |
| *Avg # of Comparisons Needed for Solution* | FILL | ME | IN | PLZ | !!!!!!! |