

Ryan Skinner

Project 2 (Homework 4)

CS3310

The branch-and-bound solution doesn't always perform better than backtracking. I think when there are a lot of items with the same weight and value the backtracking solution works better, for example if I have 4 items, all 1 weight and 1 value and a bag can hold up to 3 weight: backtracking solves in 7 steps while branch-and-bound solves it in 13.

I could not find any specific data sets that made branch-and-bound and backtracking slow down to be comparable to brute-forcing. However for small data sets, when the two faster algorithms have to check a higher percentage of the overall items, they do become similar.

It is impossible to say how much more efficient the two algorithms are than brute-forcing. As each data set has a different number of nodes checked. However this is the data I have noticed

brute-force	backtracking	branch-and-bound
$(n+1)^2-1$	$\sim n^2$	$\sim n^2$

Data

Items	W	Brute	Backtrack	B-n-b
$\{(\$40,2)(\$30,2)(\$50,10)(\$10,5)\}$	16	31	13	11
$\{(\$1,1)(\$1,1)(\$1,1)(\$1,1)\}$	3	31	7	13
$\{(\$5,10)(\$6,2)(\$2,4)(\$31,45)(\$22,24)(\$33,45)\}$	66	127	27	27
$\{(\$4,12)(\$2,1)(\$2,2)(\$10,4)(\$1,1)\}$	15	63	25	25
$\{(\$505,23)(\$352,26)(\$458,20)(\$220,18)(\$354,32)(\$414,27)(\$498,29)(\$545,26)(\$473,30)(\$543,27)\}$	67	2047	109	77