Knapsack experiments

Brennan Gensch, Snuffy Linder

February 22, 2016

5 runs per treatment

First let's load up the data with just 5 runs per treatment.

```
data_5_runs <- read.csv("ec/simple-search/finalTest.txt", sep="")
data_5_runs$Non_negative_score = ifelse(data_5_runs$Score<0, 0, data_5_runs$Score)
```

Introduction

We have three different methods for hillclimbing. The methods are Hillclimbing with random restarts, hillclimbing without random restarts, and just random search. Each method has been tested with the problems containing 20 and 200 items. Each method was given 10000 generations or mutations. There were also a few different variables that we wanted to test. Those variables are starting with either a random answer or a greedy answer before climbing.

Now let's plot the score as a function of the search method.

g = greedy

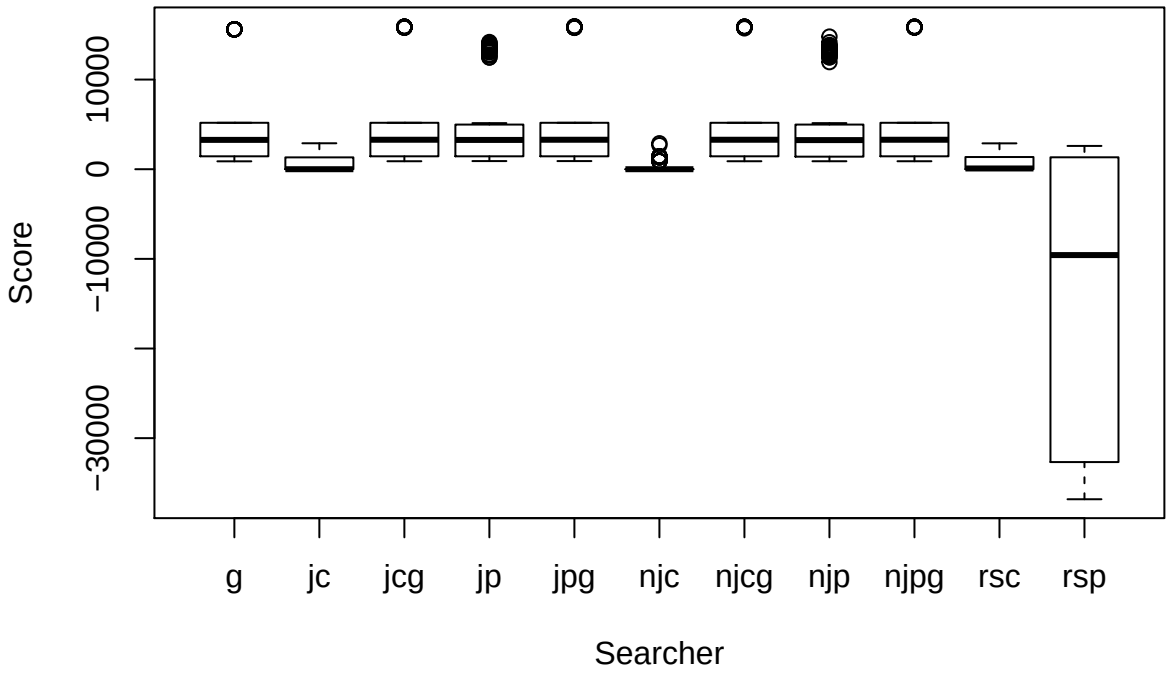
p = penalized score

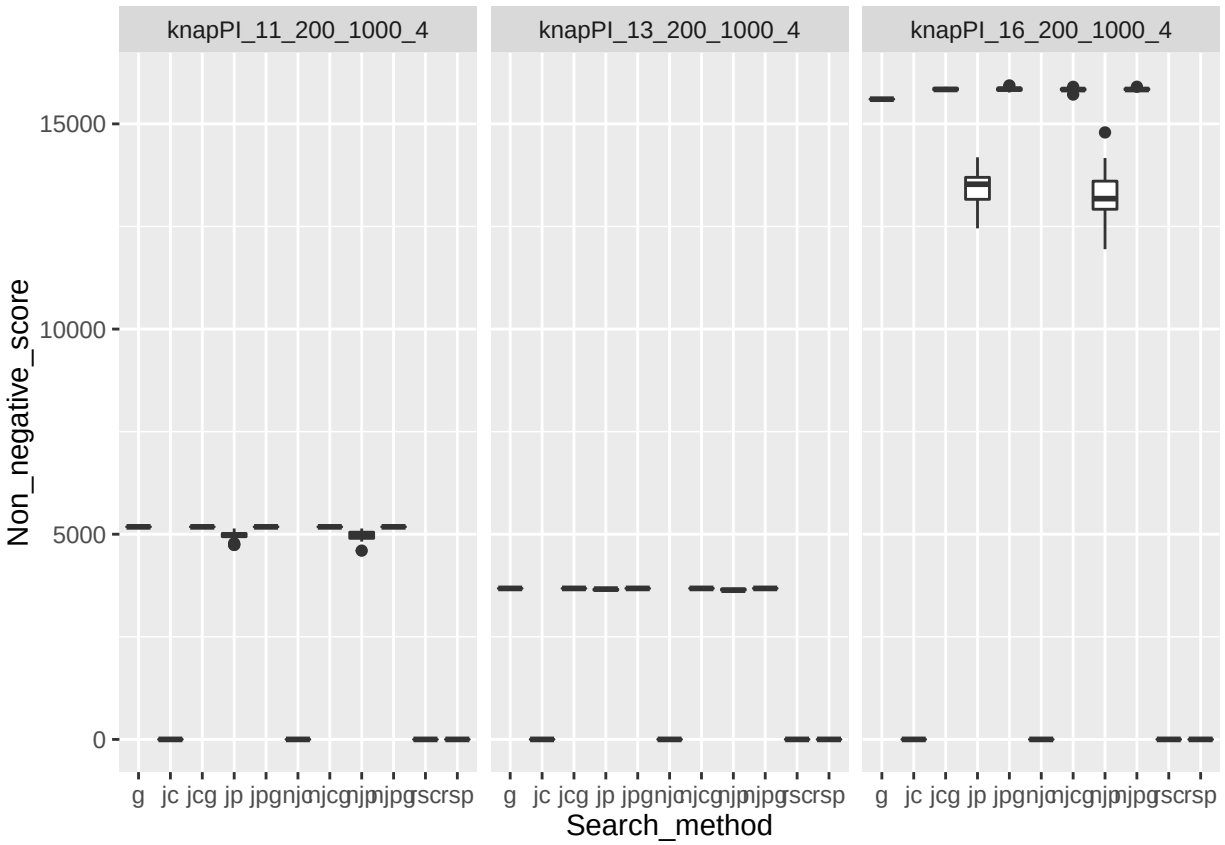
c = cliff score

nj = no jump

j = jump

rs = random search





Interesting results

In the following graphs you can see the greedy algorotihm performs the best. Then the ones with penatilized search closely follow. And the ones with clif search method perform the worst.

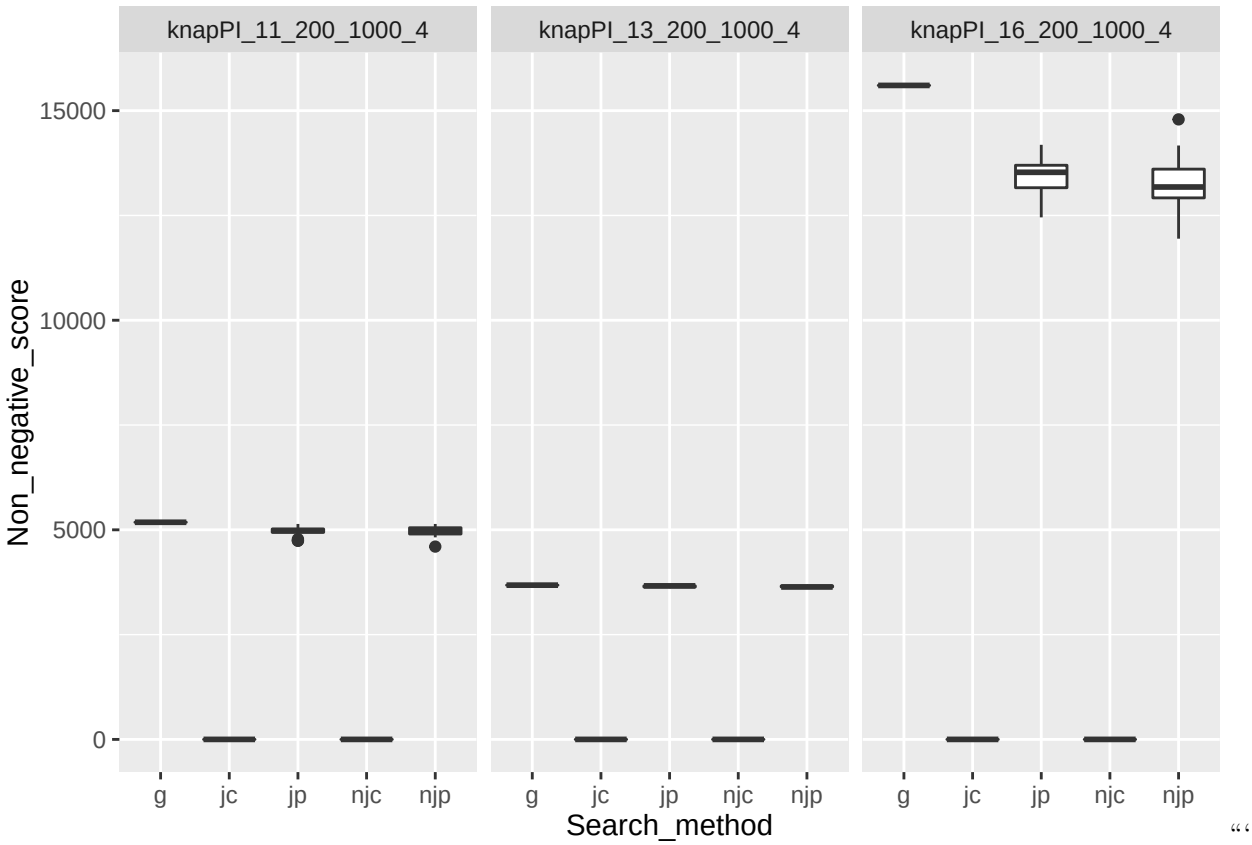
It is possible if we ran more than 10,000 iterations, that the hill climbing methods would eventually have a higher score than the greedy algrotithm, however, we did not have enough time to run that many interations.

```
twenty_item_problems = subset(data_5_runs, Problem=="knapPI_16_20_1000_4")



two_hundren_item_greedy_comparison = subset(data_5_runs, (Problem=="knapPI_11_200_1000_4" & (Search_metl
```

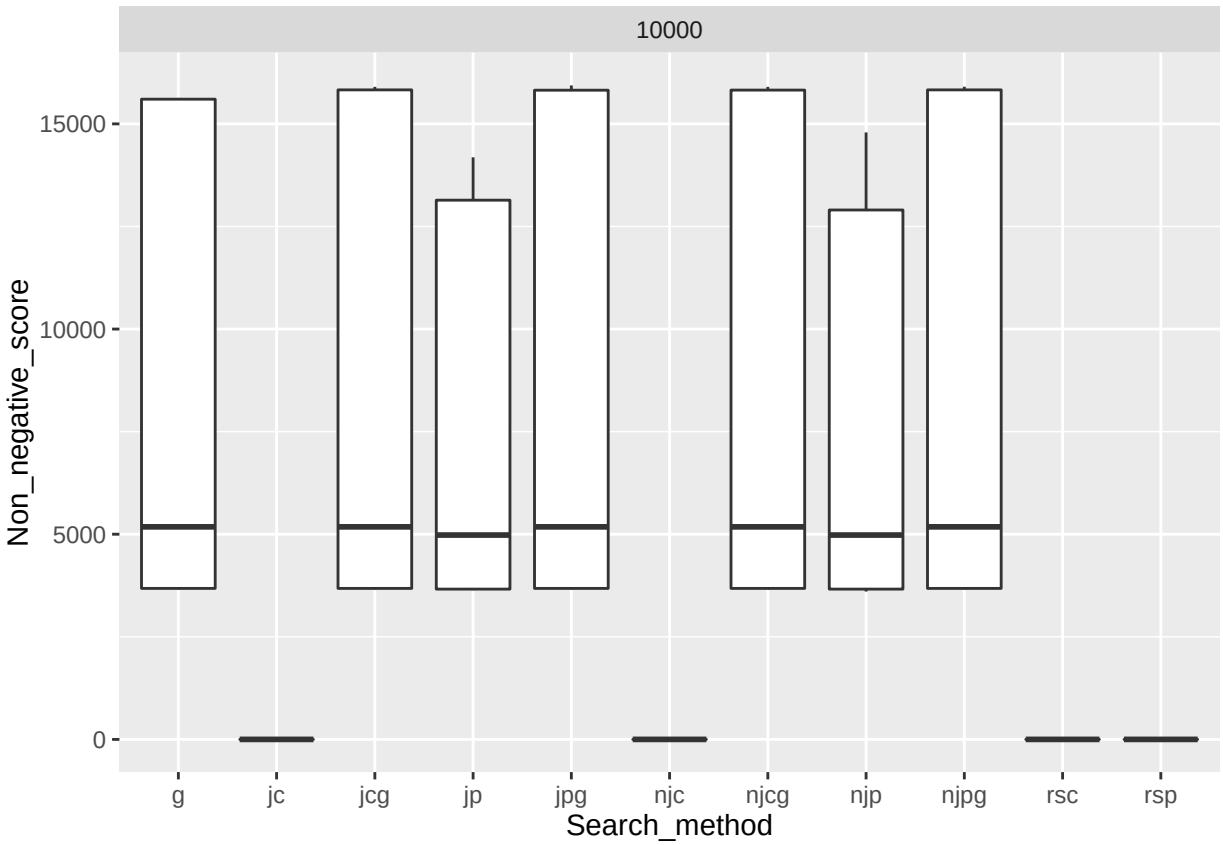
```
ggplot(two_hundren_item_greedy_comparison, aes(Search_method, Non_negative_score)) + geom_boxplot() + fa
```



Other Information

This graph does a good job of showing that our random restart method does not significantly improve on normal hillclimbing when the greedy answer is used, and when it is not it preforms marginally better. Also you can see on this graph that the methods that used the cliff scoring method were always 0. The random searhc is "0" or very negative.

```
ggplot(two_hundren_item_problems, aes(Search_method, Non_negative_score)) + geom_boxplot() + facet_grid
```


10000
15000
10000
5000
0
Non_negative_score
g
jc
jcg
jp
jpg
njc
njcg
njp
njpg
rsc
rsp
Search_method