Population_Search_Analysis

Maggie Casale, Lenny Scott, & Zach Copic

March 4, 2016

Introduction

In the analysis we discuss outcomes of our population search algorithms. We use the methods: Uniform-XO (U), Two-Point-XO (TP), and Mutation (TW) with four different population-to-selection ratios. The population-to-selection ratios are indicated in the lables. These runs have 100 individuals per generation with 5, 10, 25, and 50 selections. We'll look at a general summary of scores from all of the searchers, and then break these down into individual plots per knapsack problem. In each of these subsections, we'll start with a summary highlighting the perfomance of each of our search methods in the varying sizes (20, 200, 1000) of the knapsack problem. We will then break out into several smaller, more specific graphs, of the performance of our search methods per each varying difficulty (11, 13, 16) of the specific knapsack problem.

Our Main Methods

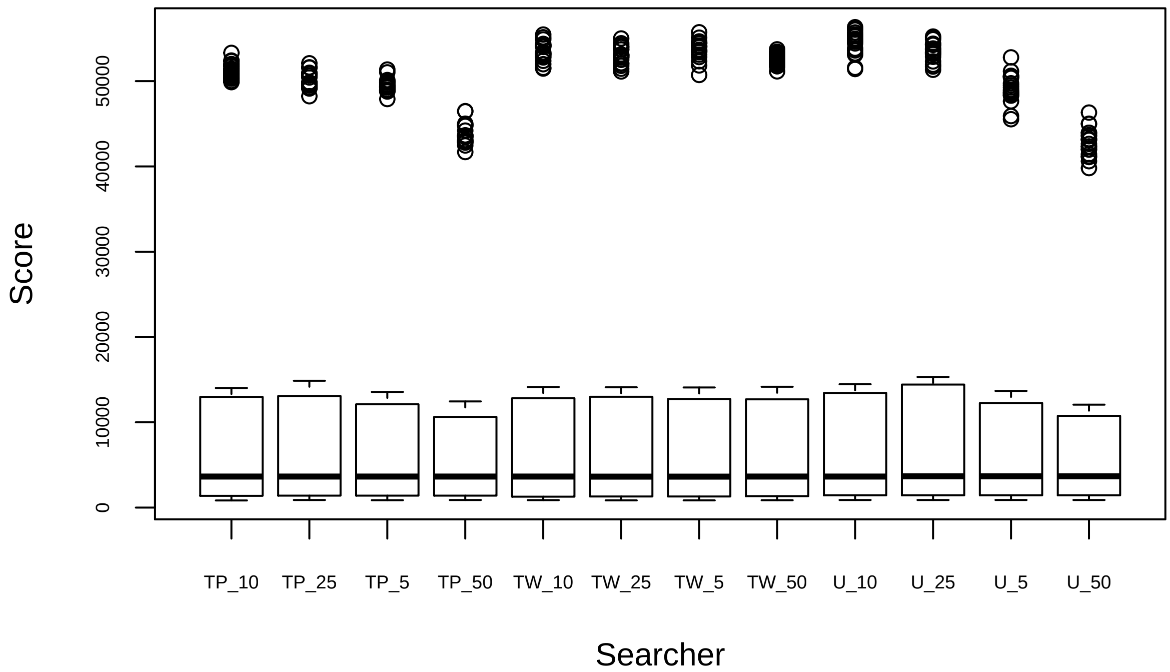
Population-Search: This is the main method of our algorithm. It takes in a mutation function, number of inividuals per generation, number of selected inividuals to select, the instance and max steps. We repeatedly make new generations by using top the top individuals and either mutating them or recombining them with other top and worst individuals. In these generations we score each individual and assign the highest scored individual as the `best`.

Make-Next-Gen: This method is used to make each new generation from the selected individuals. We use the method `make-children` to make n children for the new generation. When we make children, we get them by recombining parents or mutating a parent. Child is muatated again after this by either adding or removing an item depending if it's overweight or not.

General Summary of Scores

```
pop_data_20_runs <- read.csv("/home/casal033/ECAI/new-simple-search-the-second/population_search_everyt

plot(pop_data_20_runs$Score ~ pop_data_20_runs$Search,
     xlab="Searcher", ylab="Score", cex.axis=.55)
```

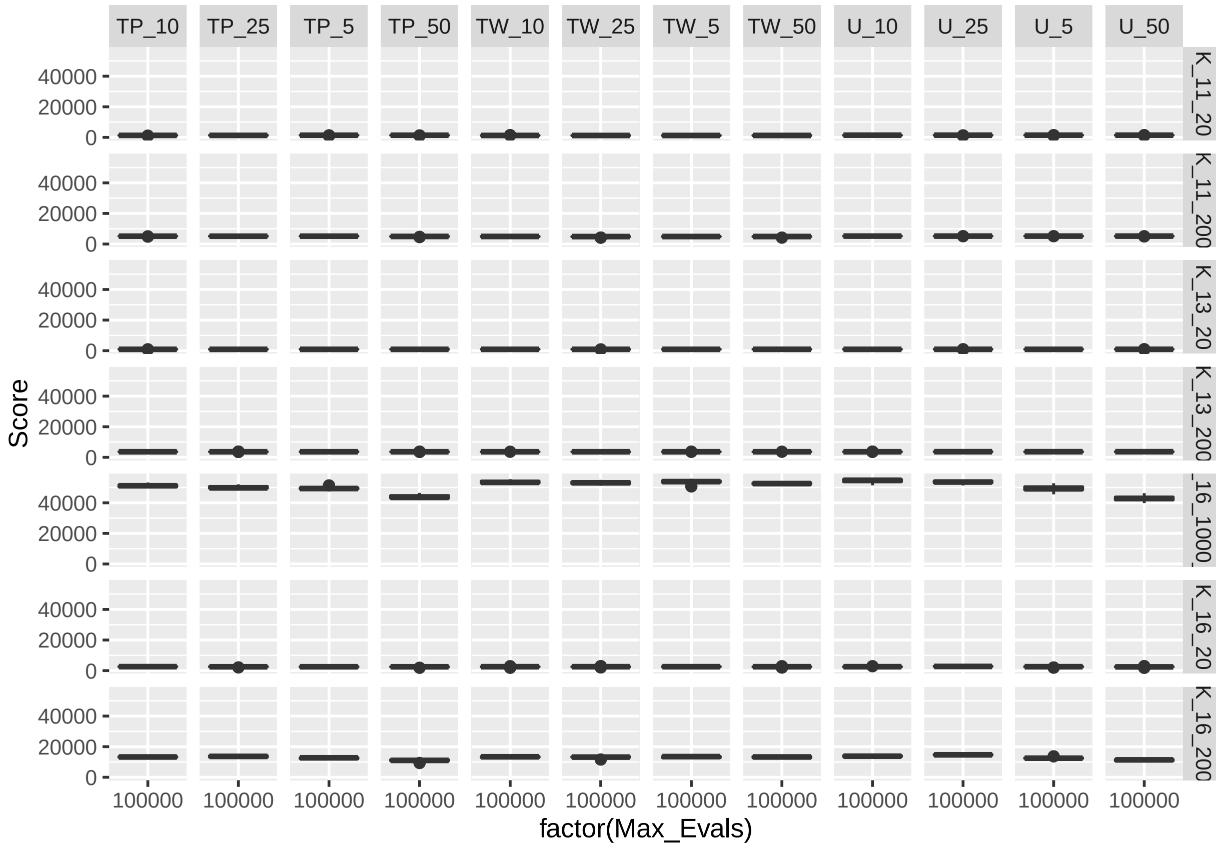


This plot has a lot of data and it's hard to tell what exactly is going on.

Alternative Summary of Scores

```
library("ggplot2")

ggplot(pop_data_20_runs,
      aes(x=factor(Max_Evals), y=Score, group=Max_Evals)) +
 geom_boxplot() + facet_grid(Problem ~ Search)+ theme_grey(base_size = 10)
```

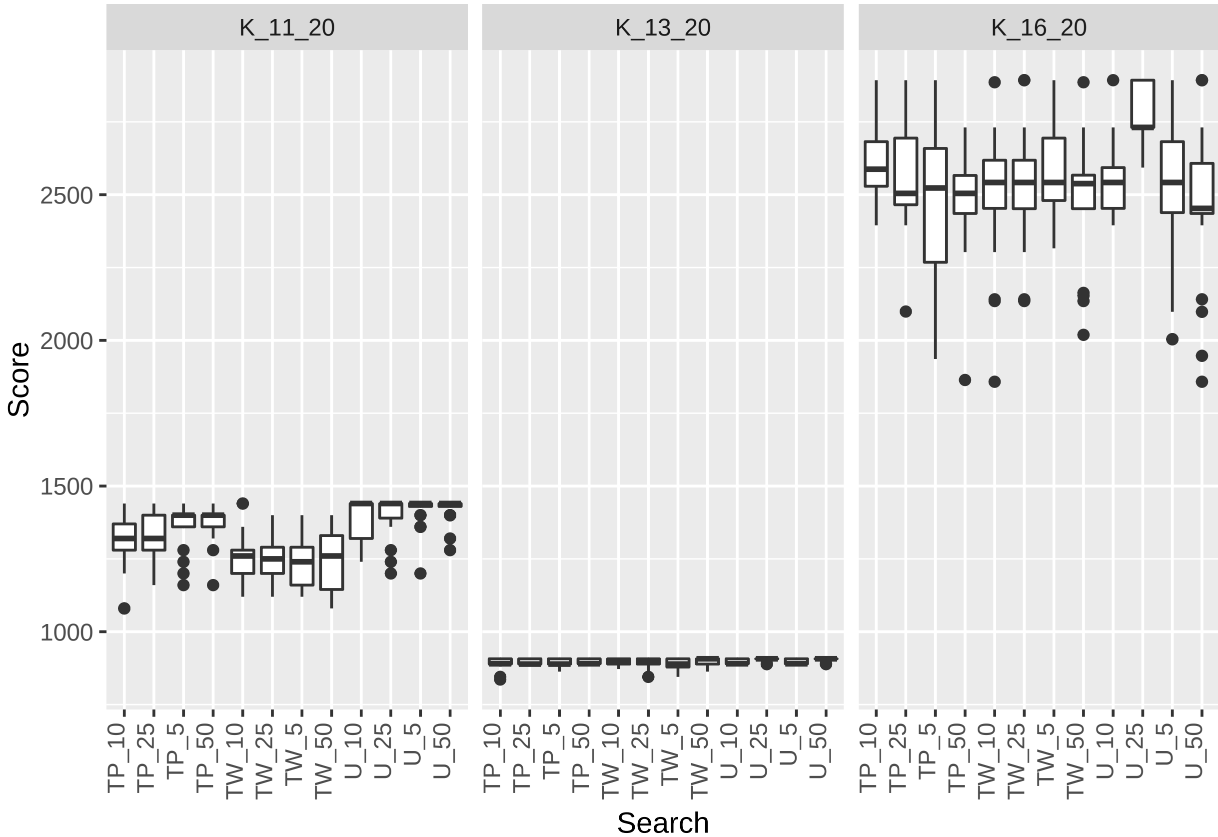


Again, this does not lead us to any obvious correlations between problems and searchers, let's break it down.

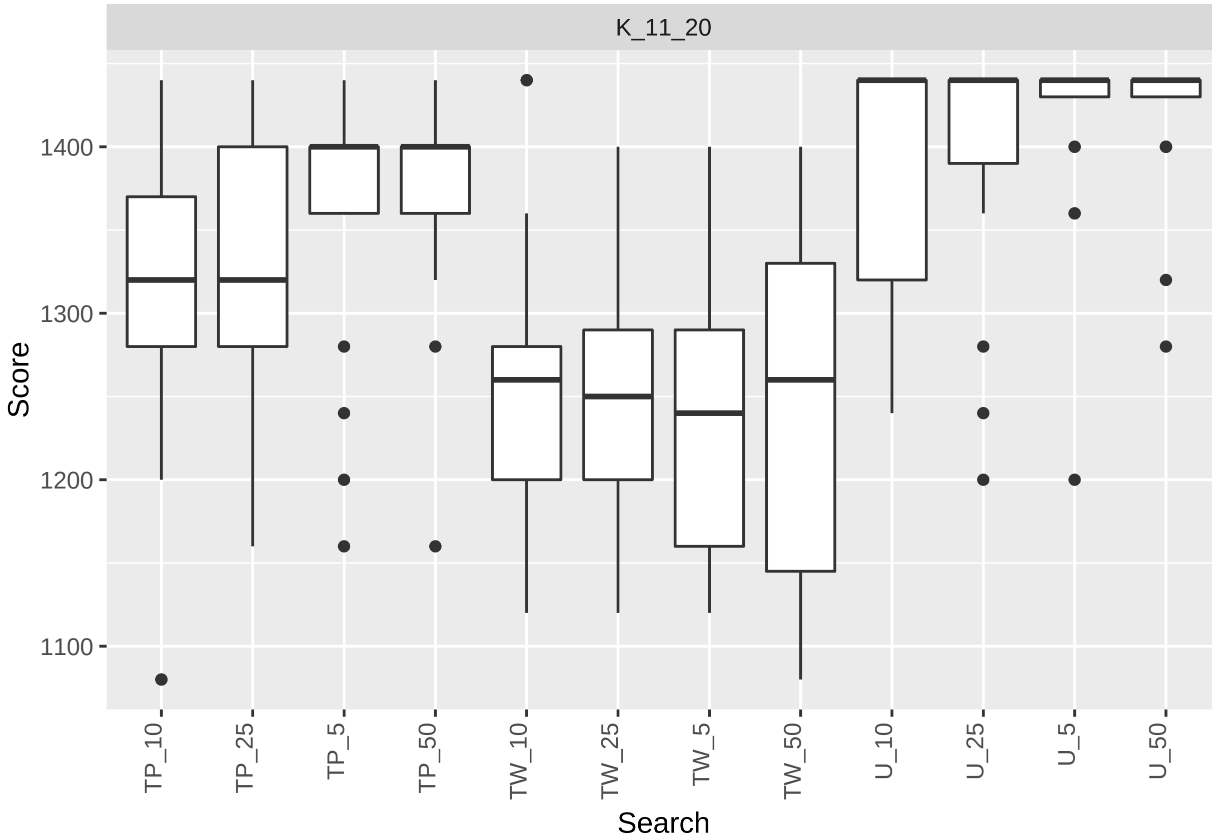
Twenty Item Problems

```
twenty_item_problems = subset(pop_data_20_runs, Problem=="K_11_20" | Problem=="K_13_20" | Problem=="K_1
twenty_item_11 = subset(pop_data_20_runs, Problem=="K_11_20")
twenty_item_13 = subset(pop_data_20_runs, Problem=="K_13_20")
twenty_item_16 = subset(pop_data_20_runs, Problem=="K_16_20")

ggplot(twenty_item_problems, aes(Search, Score)) + geom_boxplot() + facet_grid(. ~ Problem)+ theme(axis
```

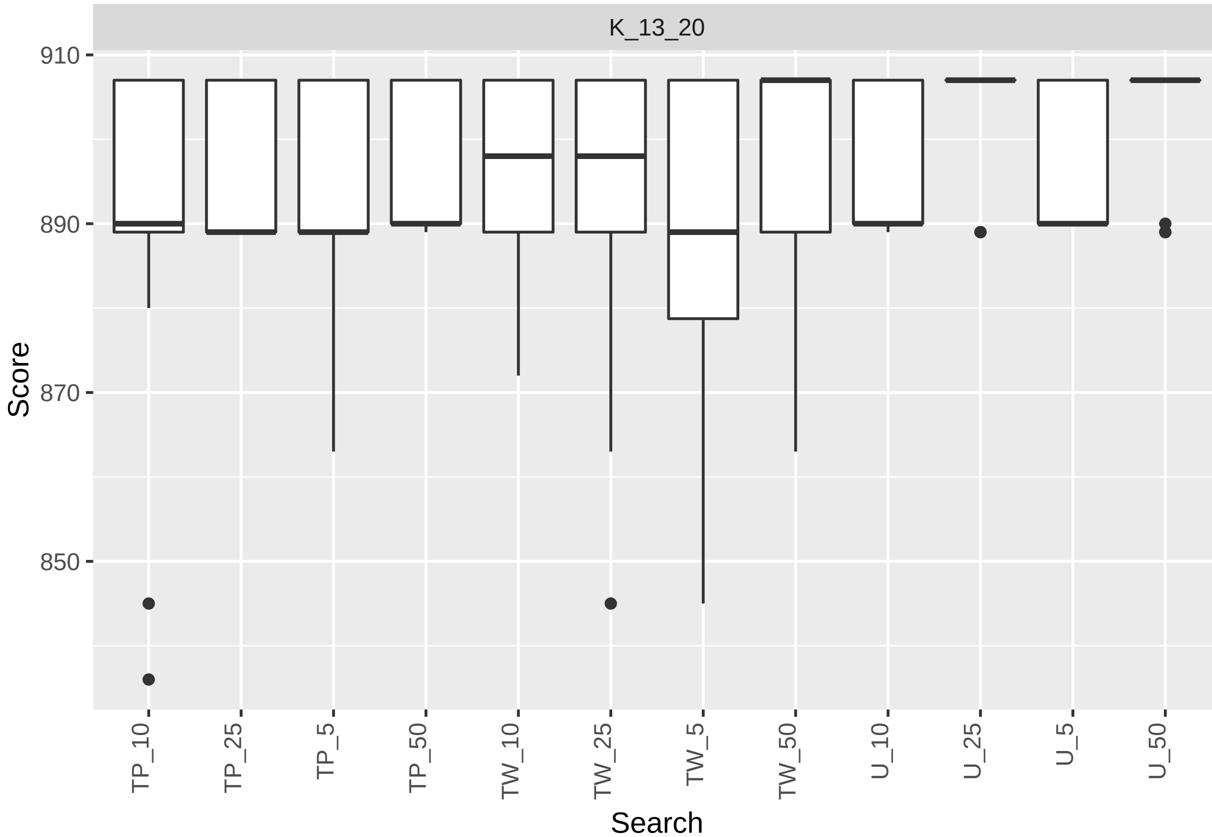



```
ggplot(twenty_item_11, aes(Search, Score)) + geom_boxplot() + facet_grid(. ~ Problem)+ theme(axis.text.:
```



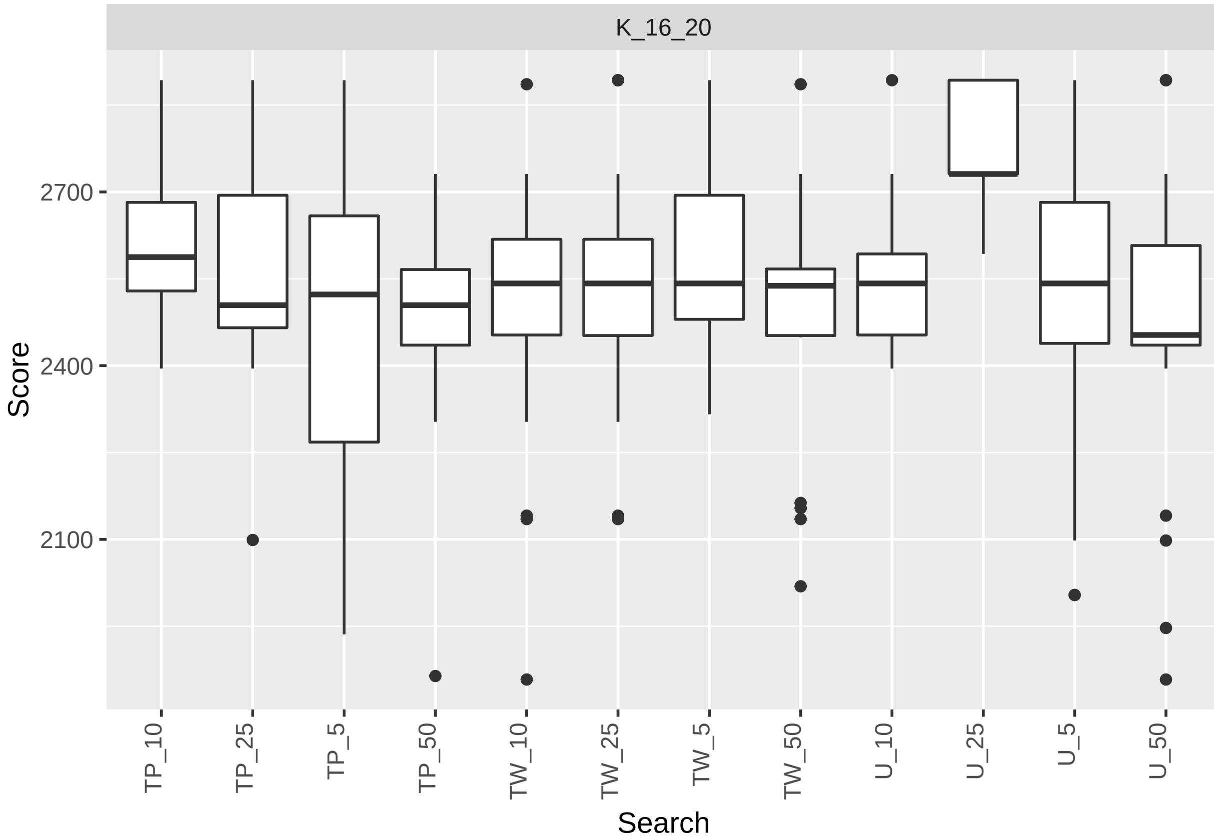
Uniform-XO has the highest (maximum) median scores for this problem. Two-Point-XO consistenly hits the global optima, however the median varied and was below the optima. Our Tweaker method only once touches the global optima (with an outlier). Exluding outliers, `U_5` and `U_50` have very similar distributions of data. `U_25` has a larger distribution of data, followed by `U_10`. When we analyze Two-Point-XO, we notice the same pattern in distribution of data as Uniform-XO (`TP_10` and `TP_50` have similar distributions, followed by `TP_25` and `TP_10`). Another important thing we find is that the medians of `TP_5` and `TP_50` are substantially higher than those of `TP_10` and `TP_25`. When we look at the mutation search method, we notice a higher volitililty in data, we suspect this might happen when a population doesn't find the global optima.

```
ggplot(twenty_item_13, aes(Search, Score)) + geom_boxplot() + facet_grid(. ~ Problem)+ theme(axis.text.:
```



All searchers consistenly hit the global optima. In Uniform-XO we can see it has the smallest range of scores, with no scores below around 890. In Two-Point-XO, the median was consitently at Uniform-XO's lowest scores of 890. Though it did reach the global optima, it had a larger range of scores than Uniform-XO. Mutation however was a bit surprising, All of its medians were either as good or better than Two-Point-XO, with the majority being better. This leads us to conclude that the global optima in this problem was just easier to find than other problems. In contrast to the previous problem, where `U_5` and `U_50` were very similar, now the top two (`U_25`, `U_50`) are just as similar in range. Another contrast in the same variety occurs in Two-Point-XO.

```
ggplot(twenty_item_16, aes(Search, Score)) + geom_boxplot() + facet_grid(. ~ Problem)+ theme(axis.text.:
```

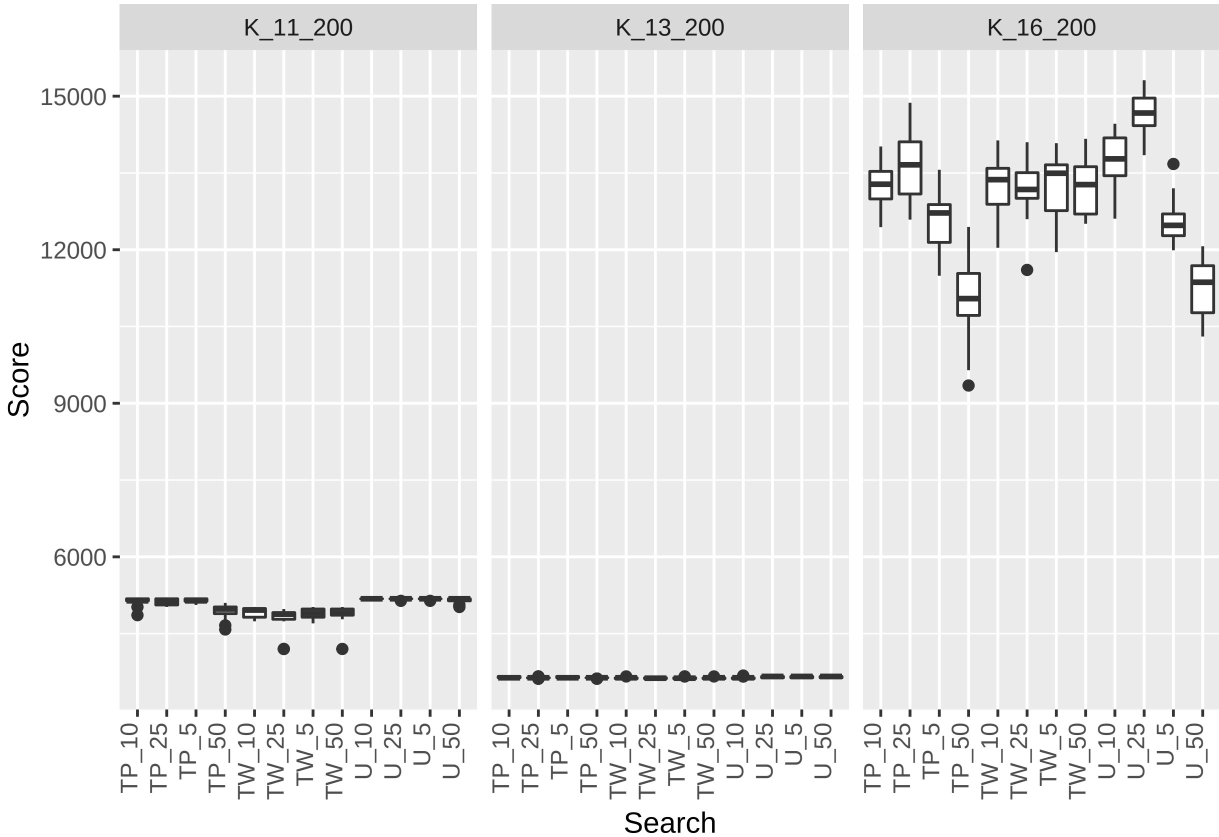


Looking at the data, `U_25` was the best overall searcher, however in general Two-Point_XO did the best overall. The results were very level, this might be due to the fact the searchers are getting stuck on sub-prime local optimas. While search methods selecting 5 individuals appeared to have a wider distribution of data no real conclusions can be drawn.

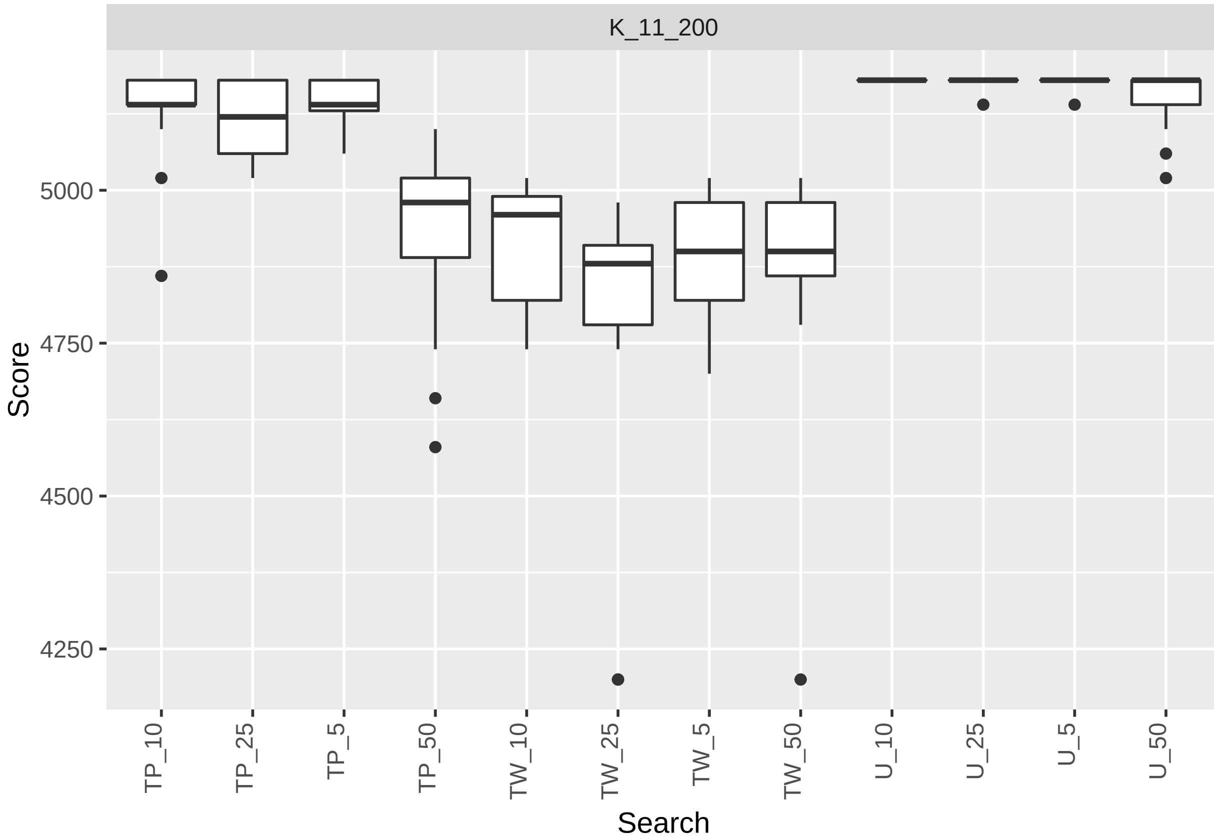
Two-Hundred Item Problems

```
twohundred_item_problems = subset(pop_data_20_runs, Problem=="K_11_200" | Problem=="K_13_200" | Problem=
twohundred_item_11 = subset(pop_data_20_runs, Problem=="K_11_200")
twohundred_item_13 = subset(pop_data_20_runs, Problem=="K_13_200")
twohundred_item_16 = subset(pop_data_20_runs, Problem=="K_16_200")

ggplot(twohundred_item_problems, aes(Search, Score)) + geom_boxplot() + facet_grid(. ~ Problem)+ theme(a
```

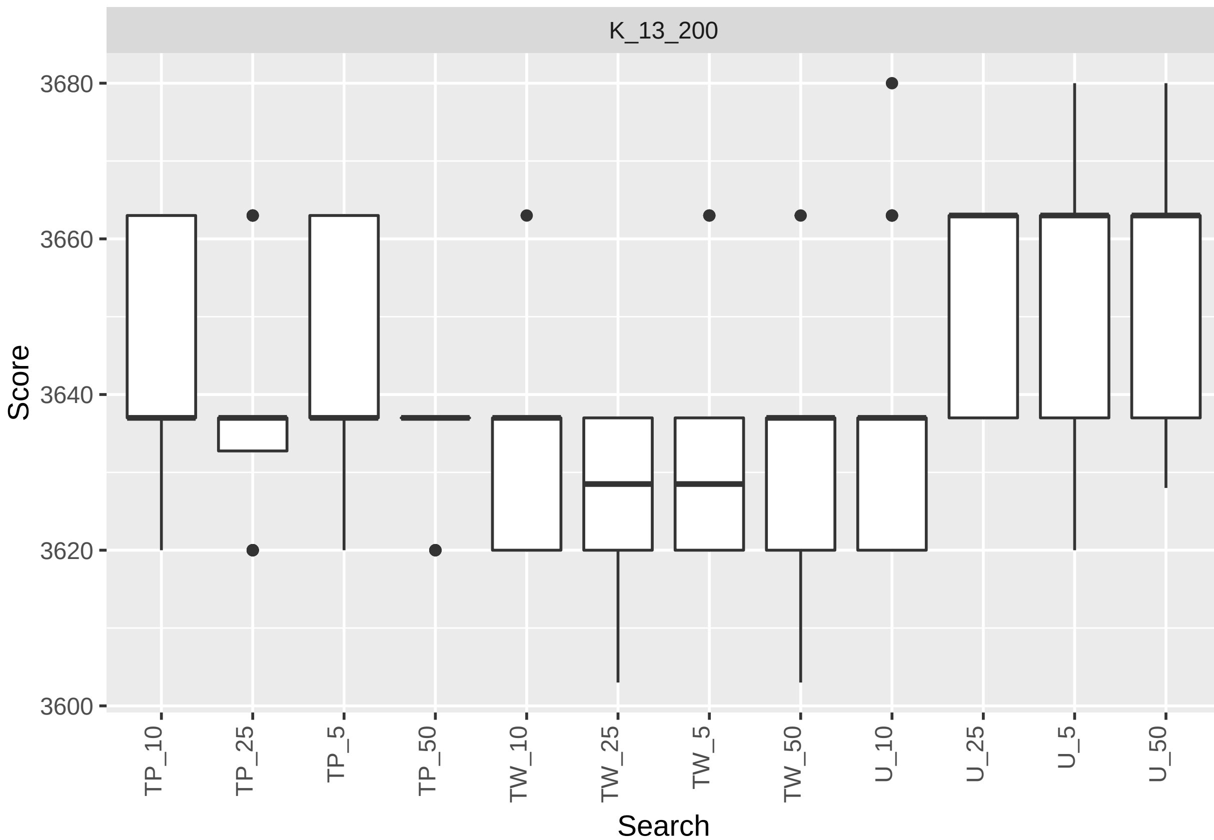



```
ggplot(twohundred_item_11, aes(Search, Score)) + geom_boxplot() + facet_grid(. ~ Problem)+ theme(axis.te
```



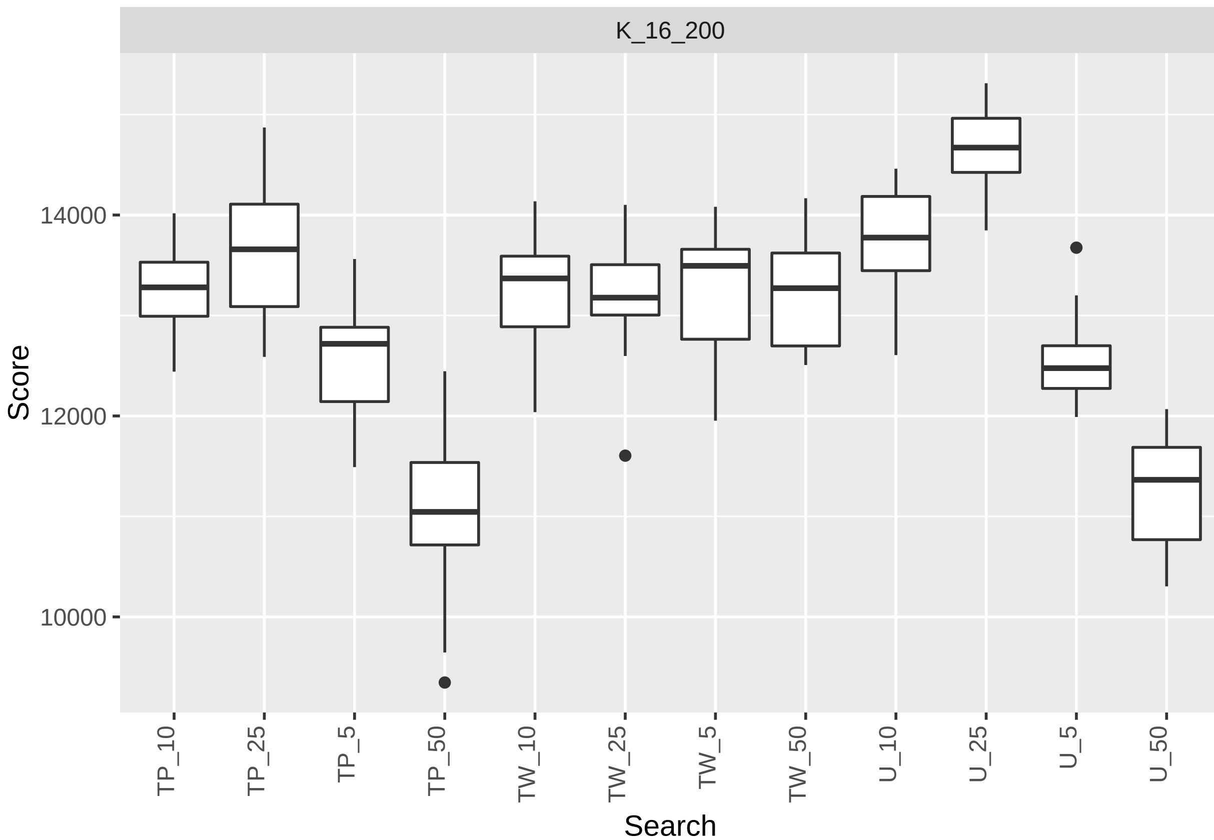
In this problem, the median of Uniform-XO was at the global optima in all four cases. Two-Point-XO hit the global optima frequently. Mutation based search, never hit the global optima. `U_50` suffered due to its explorational tendencies. When observing the Two-Point-XO, it is apparent that higher selection-ratios performed worse due to explorational tendencies. The mutation method never finds the global optima and there is very little correlation between selection-ratio.

```
ggplot(twohundred_item_13, aes(Search, Score)) + geom_boxplot() + facet_grid(. ~ Problem)+ theme(axis.t
```



In the problem, Uniform-XO did better overall with `U_5` and `U_50` hitting the maximum score. However, `U_10` hit the same local optima as Two-Point-XO. Unfortunatly `U_25` was not the best, though it had a tight grouping of scores compared to `U_5` and `U_50`. Two-Point-XO was the next best, with consistant medians across all selection-ratios. This same median was a local optima as seen by the 7 of the 12 search methods. Our mutation method is the worst as expected seen in previous examples.

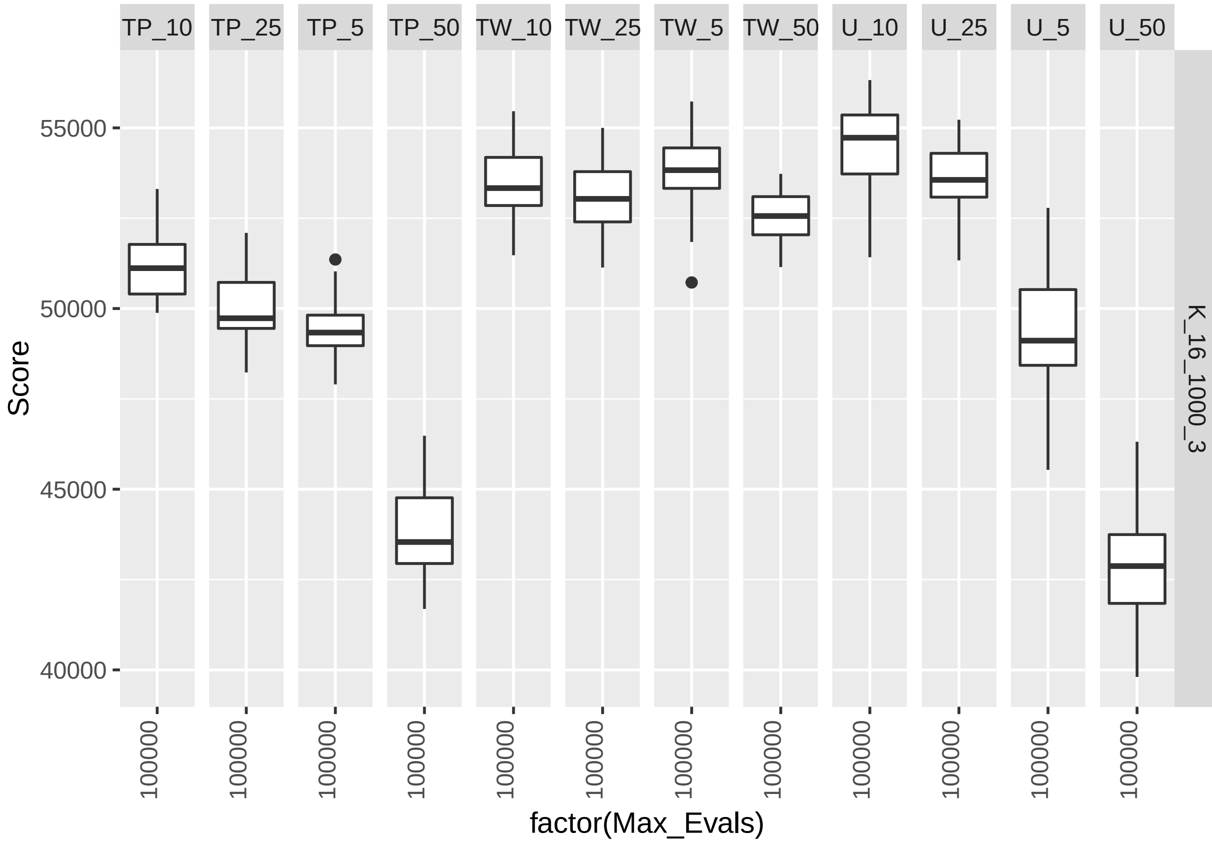
```
ggplot(twohundred_item_16, aes(Search, Score)) + geom_boxplot() + facet_grid(. ~ Problem)+ theme(axis.te
```



As expected, `U_25` did the best with `U_10` in second. However, `U_5` and `U_50` did poorly as the second and third worst in this data set. This same pattern can be seen in Two-Point-XO as well, with `TP_50` being the worst in the entire set. It makes sense for `U_50` and `TP_50` to do so poorly because they are too explorative and similar to random search. Strangely, the mutation method was very consistant with median around 13,500.

The One-Thousand Item Problem

```
thousand_item_problem_16 = subset(pop_data_20_runs, Problem=="K_16_1000_3")
ggplot(thousand_item_problem_16, aes(factor(Max_Evals), Score)) + geom_boxplot() + facet_grid(Problem ~
```



Looking at the data, Uniform-XO has the opportunity to be the best though because of how many items there are (with `U_10` and `U_25` finding that sweet spot), however `U_5` and `U_50` did poorly in finding a decent optima. The same sort of pattern can be seen the Two-Point-XO runs with the 10 and 25 variants being the best of the bunch. Surprisingly, our mutation did the best overall. The challenge of the thousand item knapsack is to get underweight and this is where our mutation method shines by taking items out and getting to that underweight point first and then proceeding to randomly add and remove items.

Conclusions

Looking back on our data, the most conclusive trend we can see is that `U_25` performs best on almost all cases (exceptions include `K_13_200_4` and `K_16_1000_3`). But it is worth noting that even on the problems that `U_25` did not win, it consistantly came close or tied. Uniform-XO regularly outperformed Two-Point-XO and mutation based searching. We suspect that Uniform-XO perserves more diversity than Two-Point-XO thus avoiding premature convergence. Moreover, mutation based search did not seem to respond to varying selection ratio in any statisticaly significant way. We noticed that the optimal selection ratio varied per problem; this makes it hard to draw correlations. Because of this variance, it seems that selection ratio should be adjusted per problem. It is worth noting that we had no negative scores, this is due to tweaking indivuals post-crossover.