Population Search Methods

Contents

Introduction

In this set of experiments, we compared the performance of several population-based search methods: simple mutation, uniform crossover, and 2-point crossover. A full description of the methods can be found here.

Experimental setup

We ran the following experiments, each 10 times with a population size of 100 over 1000 generations:

- `simple_mutation_random`: A simple mutation strategy, flipping one bit at random, with a randomly generated starting `knapsack`.
- `simple_mutation_greedy`: A simple mutation strategy, flipping one bit at random, with a starting `knapsack` (greedily) filled to capacity or slightly under.
- `uniform_xo_random`: Uniform crossover with a randomly generated starting `knapsack`.
- `uniform_xo_greedy`: Uniform crossover with a starting `knapsack` (greedily) filled to capacity or slightly under.
- `2pt_xo_random`: 2-point crossover with a randomly generated starting `knapsack`.
- `2pt_xo_greedy`: 2-point crossover with a starting `knapsack` (greedily) filled to capacity or slightly under.
- `3pt_xo_random`: 3-point crossover with a randomly generated starting `knapsack`.
- `3pt_xo_greedy`: 3-point crossover with a starting `knapsack` (greedily) filled to capacity or slightly under.

All of the experiments were run on the following problems:

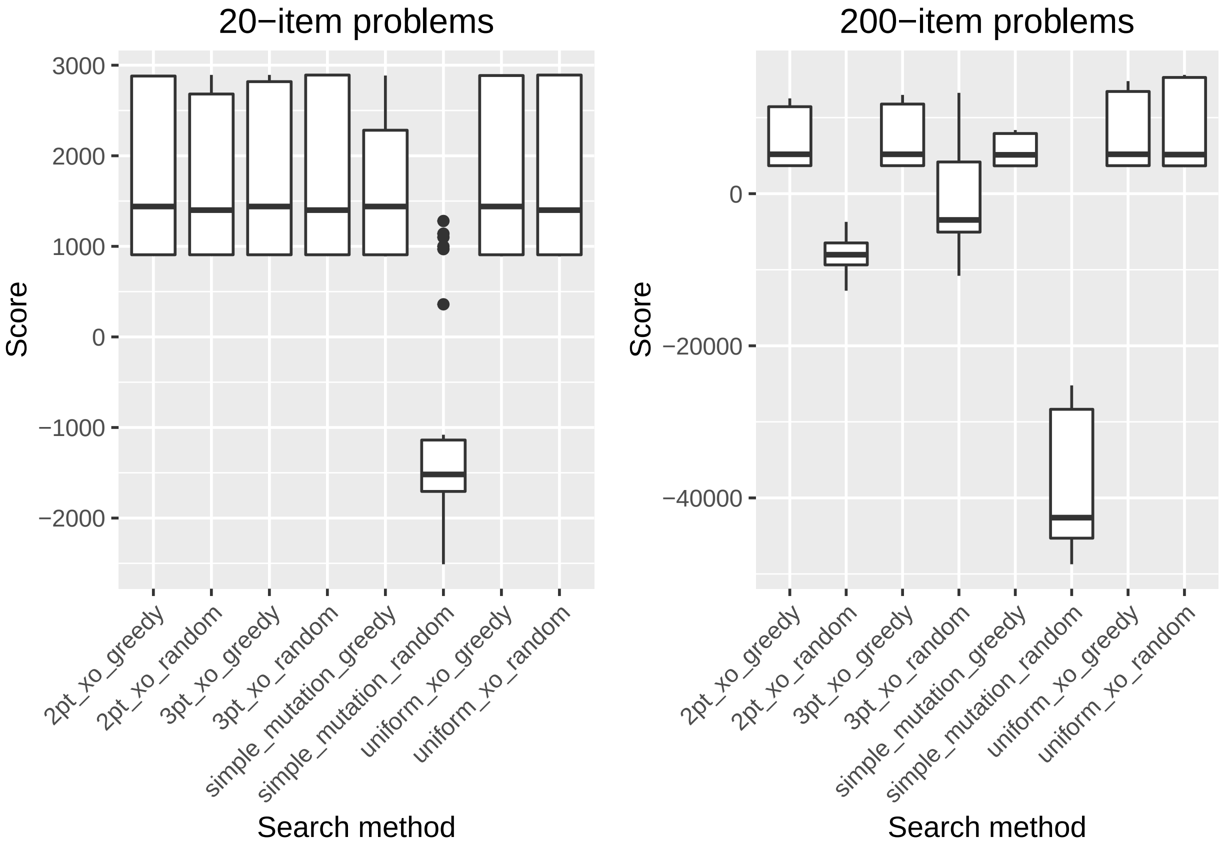
- `knapPI_11_20_1000_4`
- `knapPI_13_20_1000_4`
- `knapPI_16_20_1000_4`
- `knapPI_11_200_1000_4`
- `knapPI_13_200_1000_4`

- `knapPI_16_200_1000_4`
- `knapPI_16_1000_1000_3`

Note that problems in this report are represented without the leading `knapPI_`.

Results

To get a better idea of how each method performed on a specific type of problem, we separate the plots by item count:

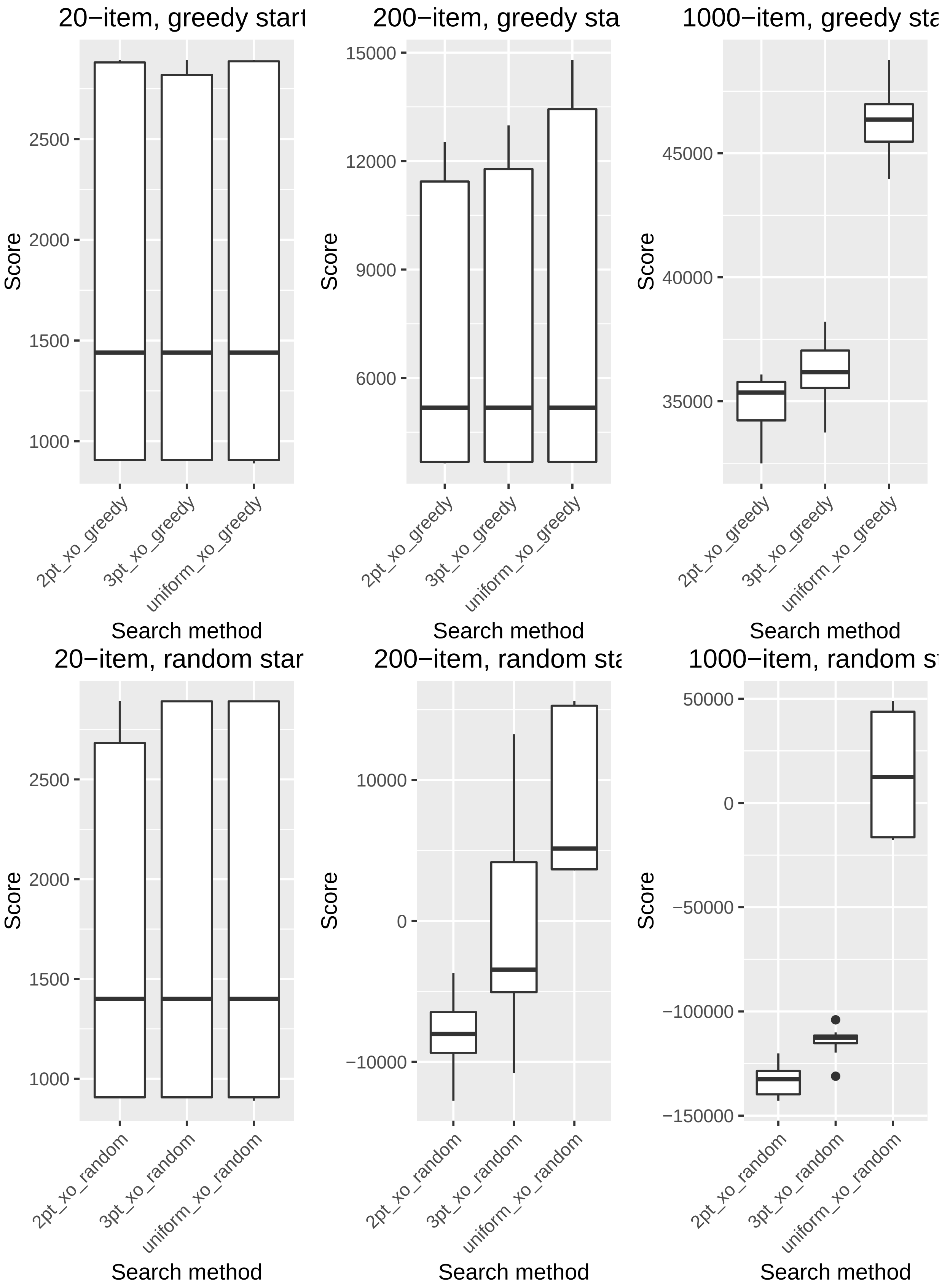


With the exception of the simple mutation strategy, the distribution of results for every search in the 20-item problems performed similarly.

For the larger problems, the results were more varied. We can clearly see that the simple mutation strategy with a random starting knapsack suffered, failing to produce positive answers. In fact, the random starting knapsack performed poorly overall, though when used with uniform crossover the results were surprisingly good.

Comparison of crossovers

To get a better idea of how our different crossover methods compare to each other, we separate them by item count and starting knapsack type:



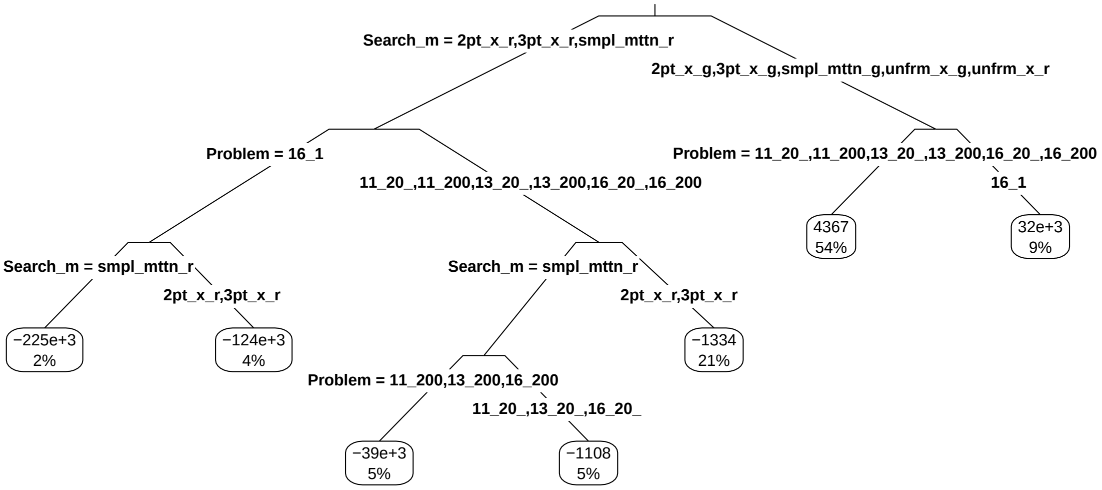
With larger datasets, the greedy starting knapsack significantly reduces the variance in score. For example, on the 1000-item problem, scores fall within a range of 20,000 while the random knapsack increases that range by an order of magnitude.

Uniform crossover outperforms the *n*-point crossovers in all cases, though its advantage is less pronounced in the 200-item problem with the greedy starting knapsack.

The plots for the 1000-item problems look strikingly similar if the scales are ignored. (We aren't at all sure what that means. It may be an anomaly given that we only ran one 1000-item problem, and it's known to be weird.)

Recursive partitioning

To help determine which of our experimental variables were significant, we ran a recursive partition:



The first split is on greedy versus random, with the exception to uniform crossover. This means that a crucial difference resulted from how the knapsacks were initialized. When the knapsacks were filled in a clever (ie greedy manner), we saw our results differ immensely. We found it interesing that the uniform crossover was lumped in here, even when we generated inital knapsacks randomly.

From there, the biggest difference was the scale of the given knapsack problems. The 1000 item problem, in particular behaved differently than the other problems.

Final thoughts

We were intruged to find that uniform crossover worked so well, regardless of the method used when generating the initial knapsacks. We hypothesize that this is because there isn't any strong relationship among the values of the choices string. Since uniform crossover doesn't maintain stucture in the same way as the n-point crossovers, we surmise that uniform crossover was able to do more exploration.

Concerning the problem sets, we think that running the experiments on more 1000 item knapsacks would give us a clearer understanding of our search methods' behaviors on a larger scale.

It may also be interesting to start with a completely empty knapsack in additon to our random and greedy knapsack initilization methods.