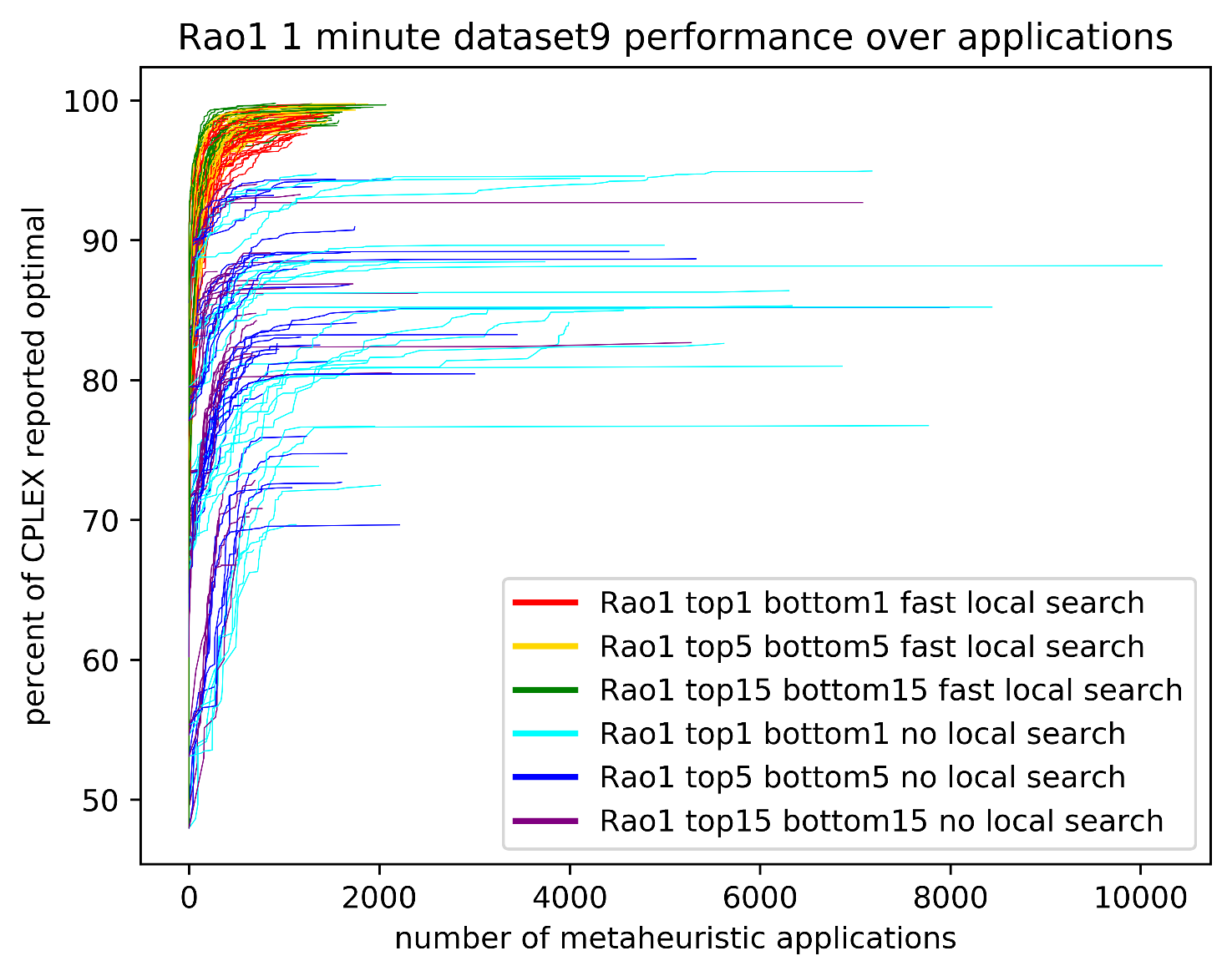
Various Rao1 algorithms were applied to dataset 9 for 60 seconds, and the generation of each application that yielded an improved best-found score were recorded. These improvement generations were used to make the following graphs.

For each of the 90 problems in each algorithm’s results, graph how the performance changes with the amount of applications. This shows the relative spread of different algorithms’ results:



This is the same graph of the performance of each algorithm, but averaged into one line. The lines stop once there are no more results for that application, so the tips of the lines are averaging less than 90 problems. I am surprised they seem so smooth. The different line lengths all represent the same amount of time. For the algorithms with no local search, it appears as if a 20 second run time will not have much worse performance than the 60 second, but for the algorithms with a local search, improvement doesn’t stop until 40 or 50 seconds.

A screenshot of a cell phone

Description automatically generated

This graph shows how using a max-failed-applications stopping criteria affects the performance. The lines stop when increasing stopping criteria any more won’t make any change to the performance of the line, so even the long flat sections of line must have a small positive slope. It seems like a stopping criterion of 200 is a good compromise for the NLS algorithms, and 100 is good enough for the FLS algorithms.

A close up of a map

Description automatically generated