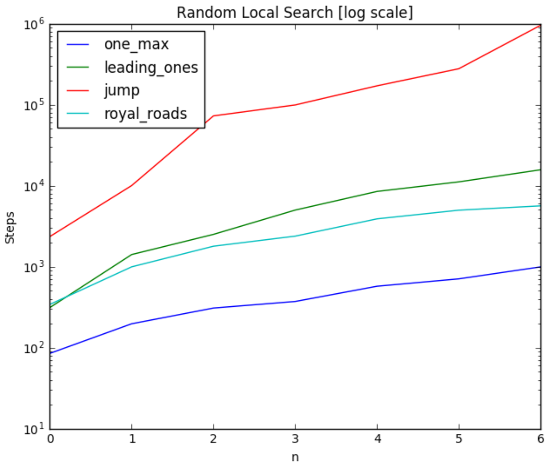
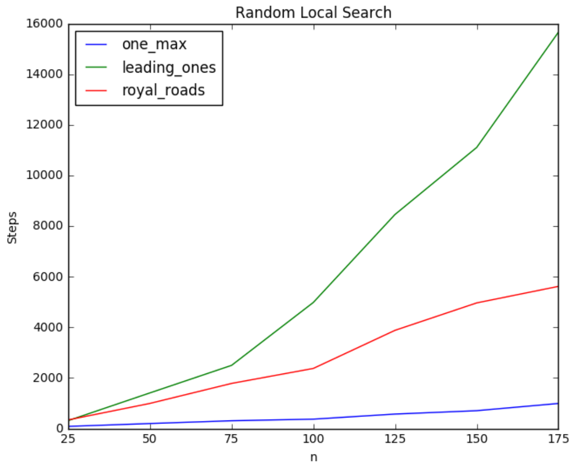
Analysis





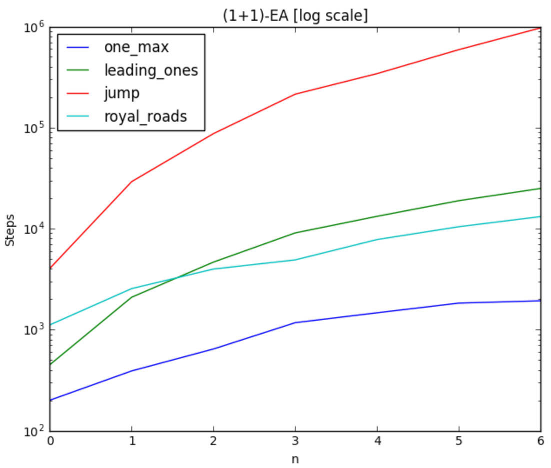
1.) Random Local Search

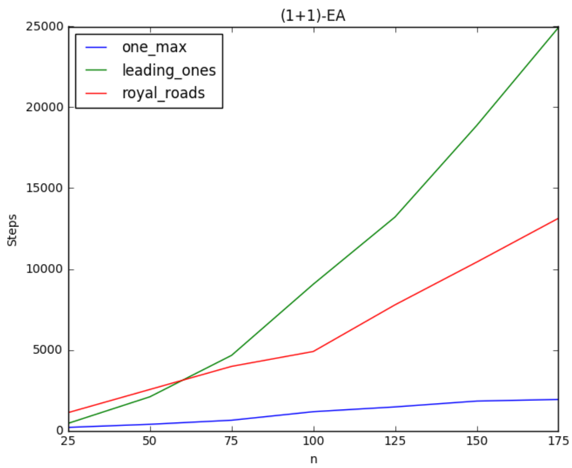
The algorithm takes much more time for "jump" than for each other problem.

In comparison to the other algorithms it always needs longer for "leading_ones" than "royal_roads".

If you look at the plot without logarithmic scale you can see that all problems except for "jump" seem to increase almost linearly to n, but there are some jumps which seem to be random except for "leading_ones".

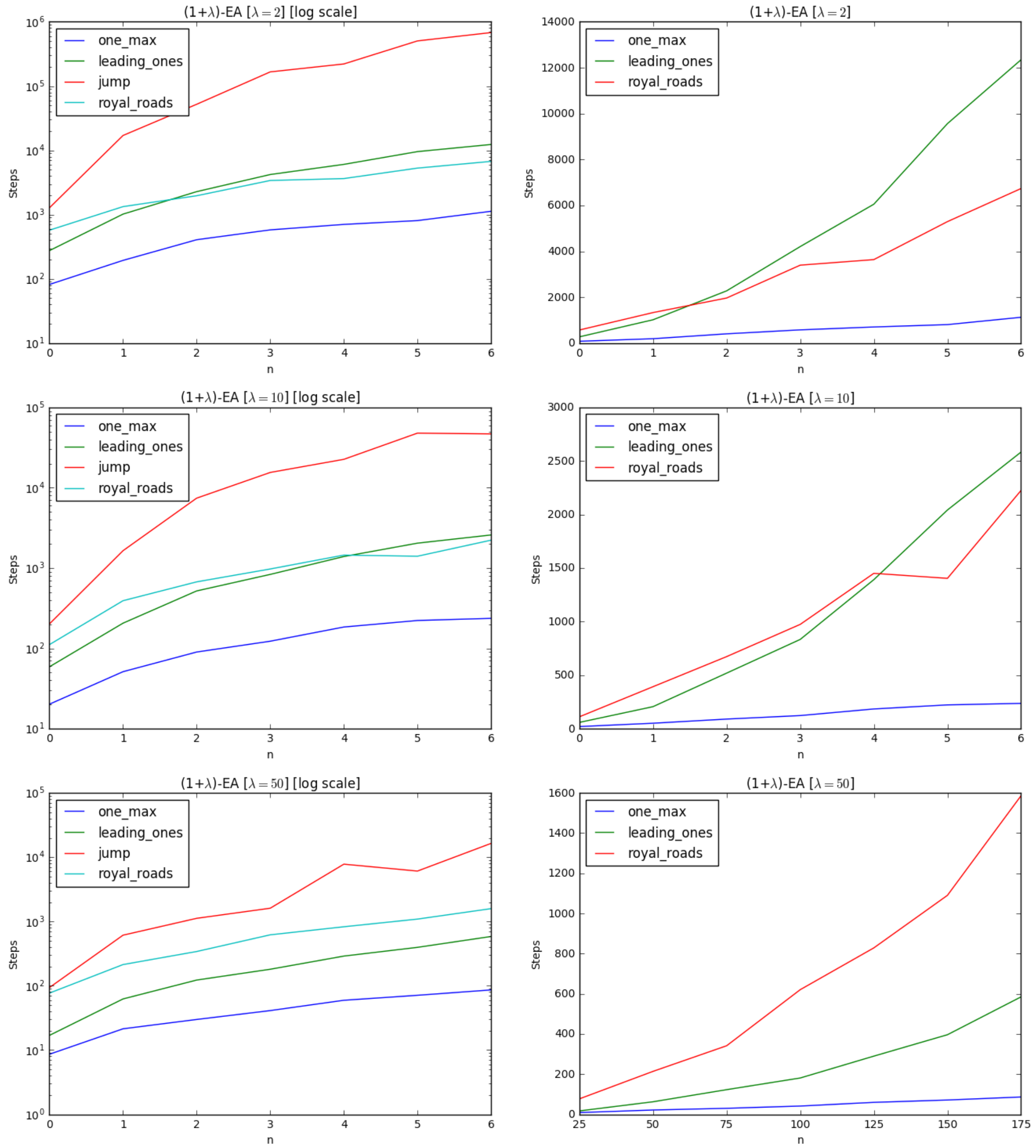
"one_max" is always executed as the fastest one. This might be because there is only one local (and also global) maximum and the steps are smaller. In contrast to that "jump" might return the same value for much more combinations because the bits are splitted into groups.





2.) Evolutionary Algorithm - (1+1)-EA

This algorithm takes just as long as RLS for all problems.



3.) Evolutionary Algorithm - (1+λ)-EA

Again the algorithm takes much more time for "jump" than for each other problem - independent of λ.

Interestingly the costs sometimes decrease between two values n_i and n_{i+1} This might be due to the randomness of finding the solution.

With an increasing λ the algorithm becomes much faster (by factor ~10) for all problems. Furthermore the costs of "leading_ones" shrink faster than the costs of "royal_roads" with a increasing λ.

In comparison to the other algorithms this algorithm with $\lambda = 50$ is the only one with a relative good performance for "jump".