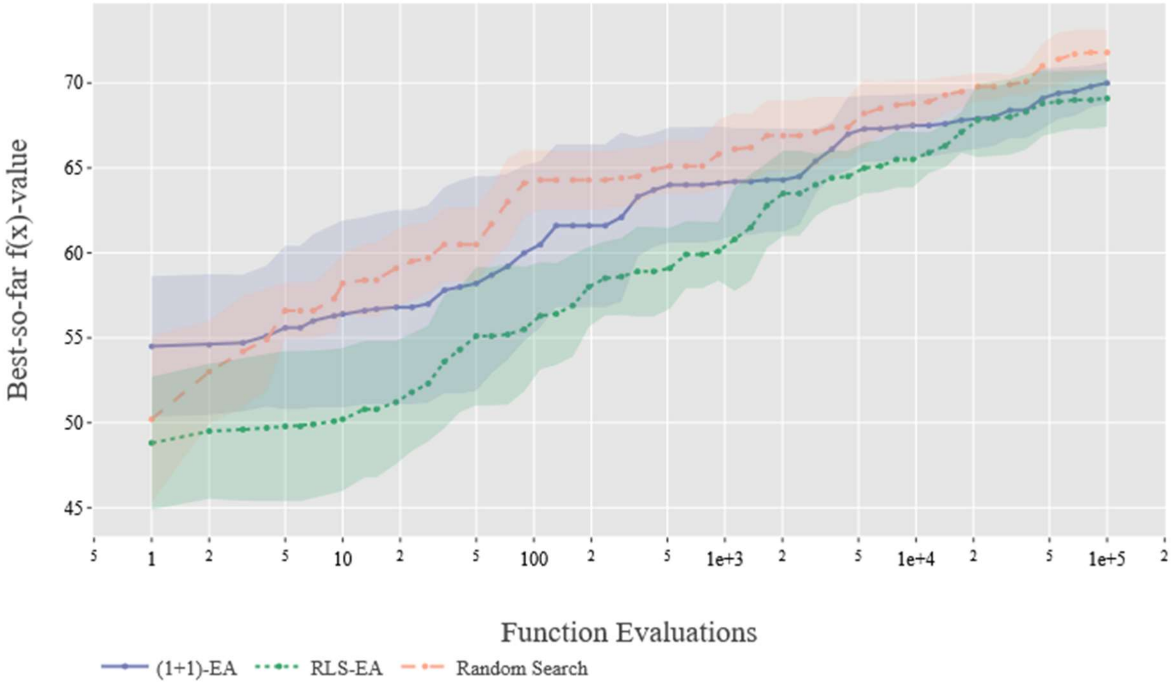
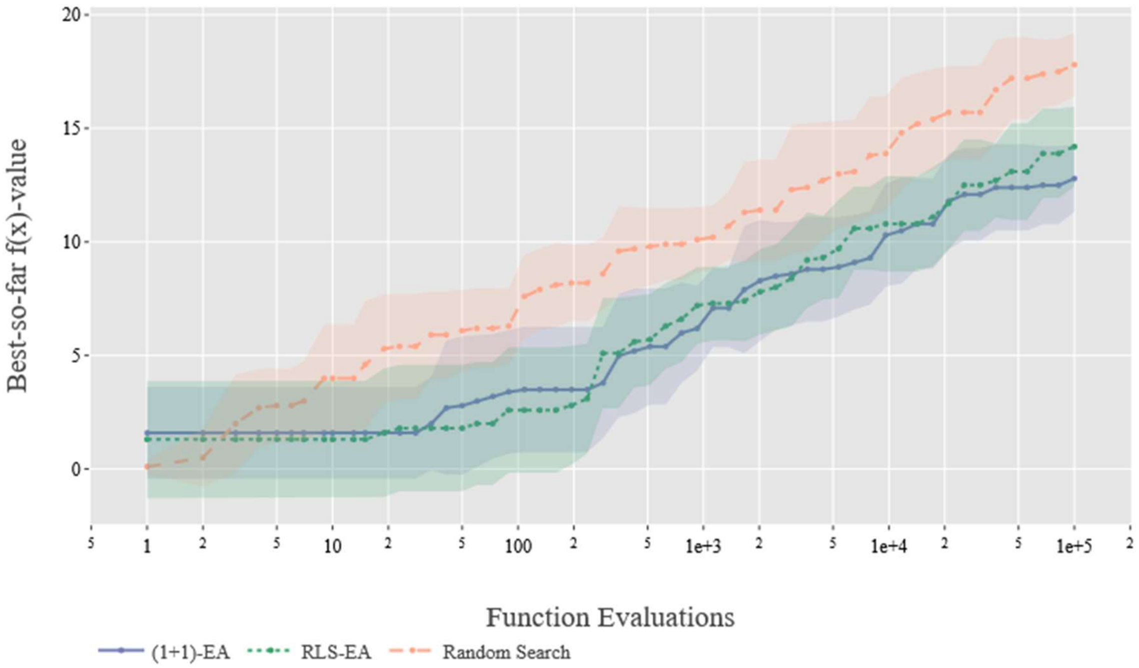
PBO – Exercise 2 Results Discussion

Function 1: OneMax



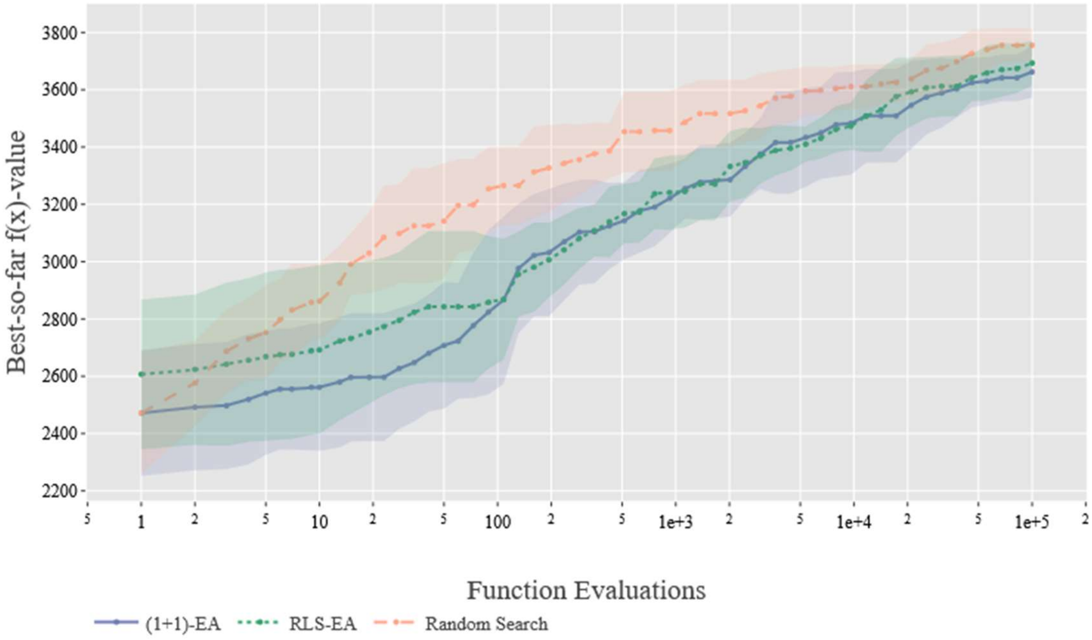
OneMax returns the number of ones present in the bit string. The result of our exercise shows that Random search had finished with the highest mean of the three algorithms. As is shown by the trendline of the mean, the Random search algorithm tends to make jumps in function value then reaching a plateau for some evaluations before continuing. A reason for these rapid spikes would be from two or more changes in the bit string, leading to larger increases in the function value. The (1+1) EA algorithm can also modify more than one bit in the string. It too has occasional spikes, but it tends to plateau more frequently. (1+1) EA had the largest standard deviation compared to the other algorithms, with some runs performing better than the random search. The RLS algorithm performed the worst of three algorithms at the start, but throughout the evaluations, the algorithm reached a similar mean to the (1+1) EA.

Function 2: LeadingOnes



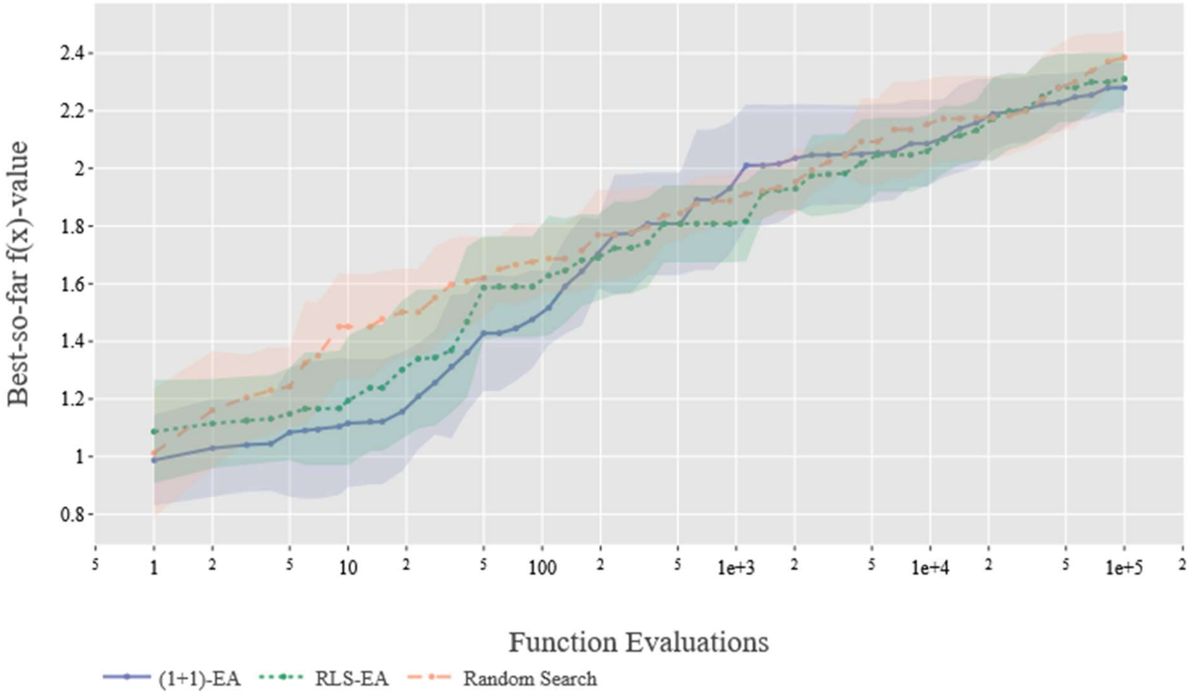
LeadingOnes is a function which counts the number of ones at the start of the bit string, compared to OneMax which is interested in any ones in the bit string. Both the (1+1) and the RLS EA show a similar trend in the data. Both share a jagged linear trendline from around 200 evaluations onwards. Both of the algorithms have very similar standard deviations and even final evaluations, with RLS coming out on top with a higher mean. However, random search eclipses both of the evolutionary algorithms in function value. Random search, on average, outperforms the other two algorithms by a sequence of about five 1s. A possible explanation is that RLS and (1+1) are changing bits into disconnected sequences in the bit string and thus are not increasing the problem function.

Function 3: Linear With Harmonic Weights



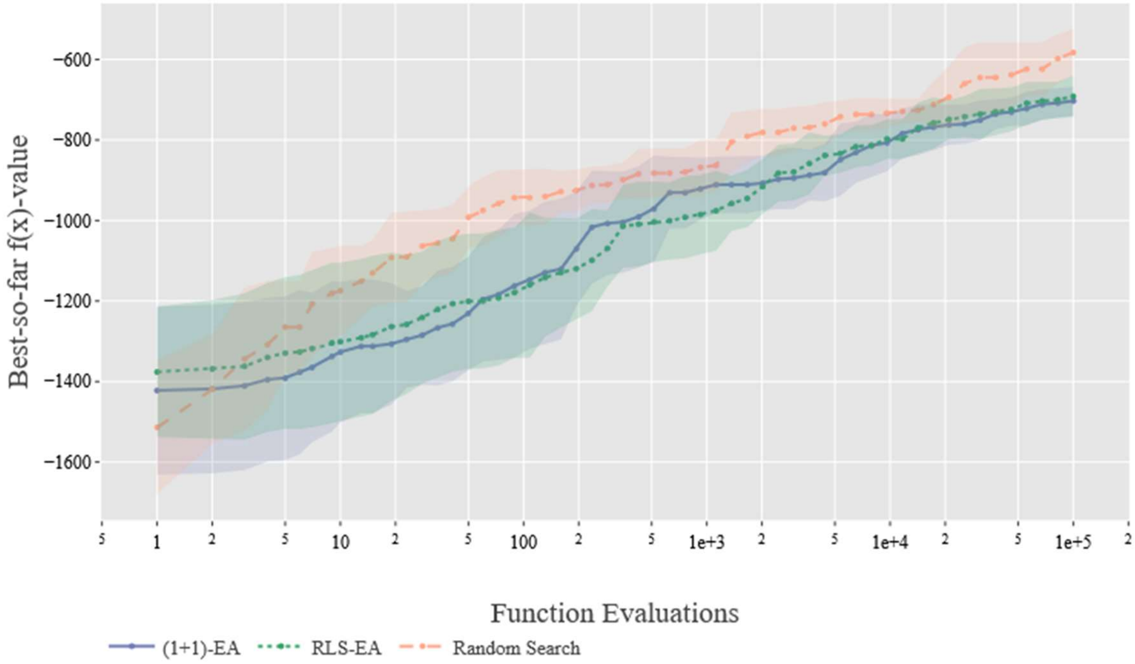
“Linear Function with Harmonic Weights” is the summation of the bits in the bit string by some weight number. This function continues to show the general trend over all of our experiments, with (1+1) EA and RLS showing similar trendlines between evaluation 100 through to 100000. Random Search had the higher mean value through evaluations 100 through to 100000, however all algorithms ended near each other in the end.

Function 18: Low Autocorrelation Binary Sequence



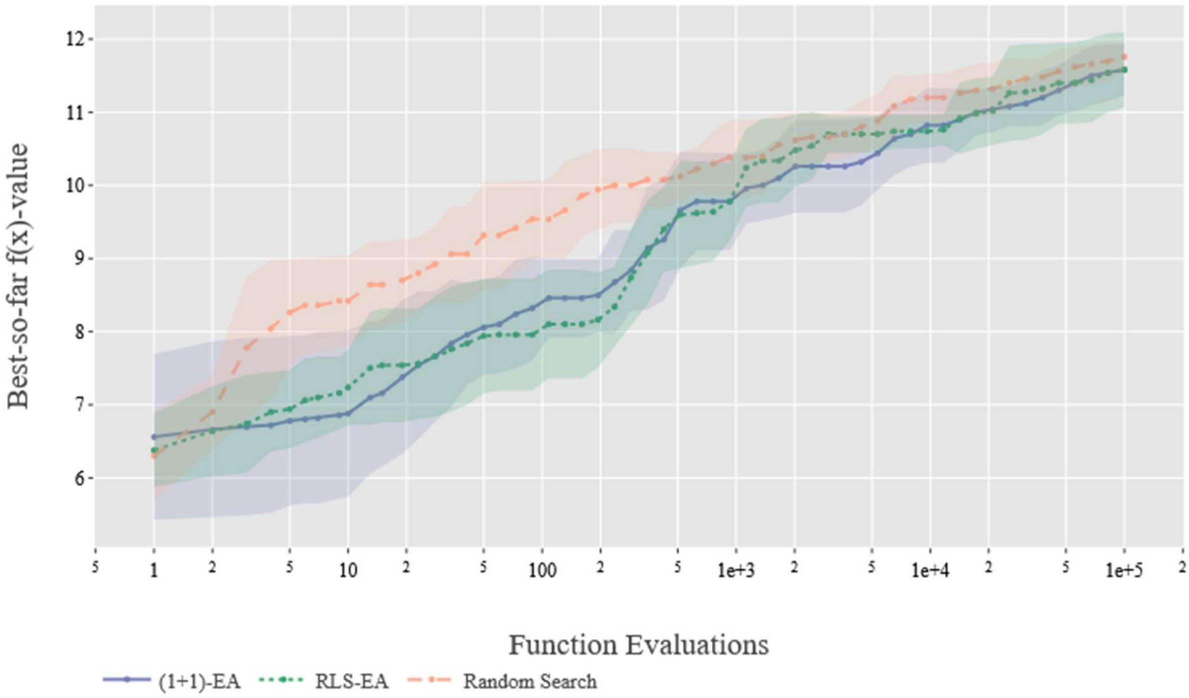
Autocorrelation is the measure of correlation between a piece of data (typically a signal sampled from a random variable) with another delayed piece of data. This function, Low Autocorrelation of Binary Sequences, attempts to maximize the reciprocal of the autocorrelation of the bit string. Interestingly, Random Search did not deviate from any of than the other algorithm as the evaluations continued. All three algorithms were almost identical to each other, and all stayed within the same standard deviation between evaluations 100 to 100000. This function is interesting as it is one of the only functions where the early start Random Search finds quickly diminishes. The standard deviation of the algorithm overlaps for a majority of the evaluations, with random Search deviating early and (1+1) starting to deviate around the thousand iterations.

Function 23: n-Queens



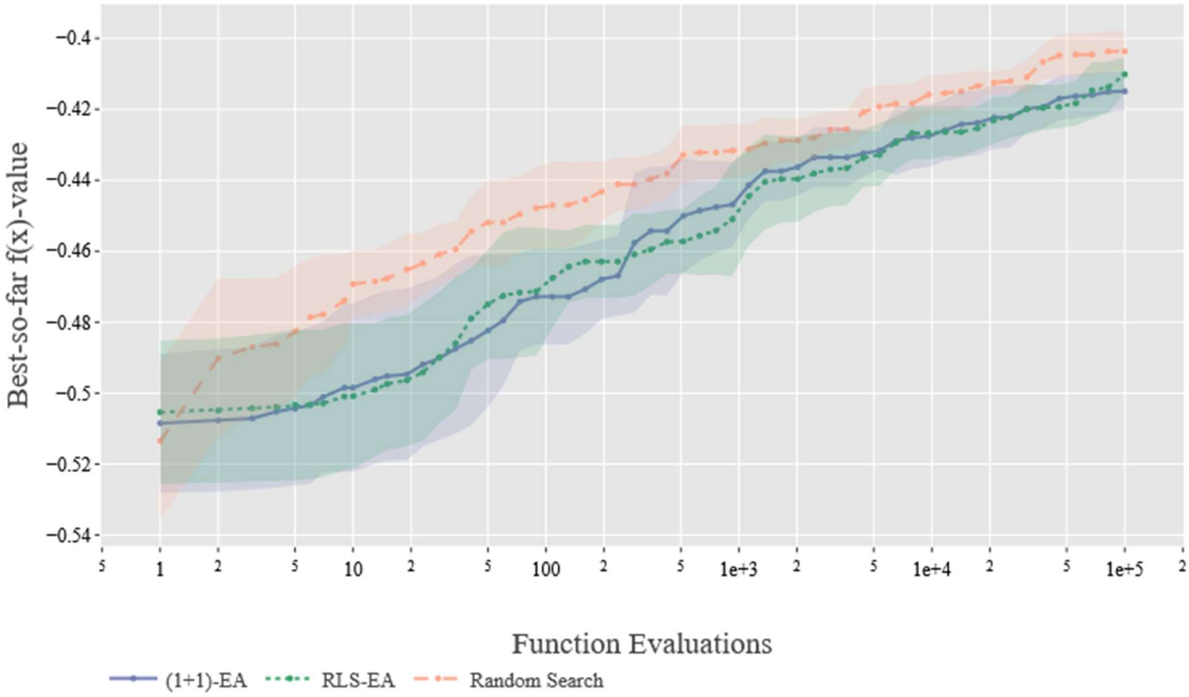
The n-Queens function is based on the classic computer science and combinatorics problem of the same name. Given an n by n chess board, you must find a valid configuration of n queens in which none of them are in “check” (or can attack another). Since the goal of the function is to achieve a valid board, the function in our case penalises a solution based on how many queens are in check. The result from the experiments further continues the pattern seen in the other functions: RLS and (1+1) EA have similar results and Random Search performing slighting better. The standard deviation for both RLS and (1+1) EA overlap for the entirety of this function.

Function 24: Concatenated Trap



Concatenated Traps function takes k length portions from the bit string and uses those portions as inputs to n/k number of trap functions. Random search tended have better function values earlier evaluation, as it had a rapid increase at the start but doesn't find significant improvement throughout the later evaluations. However, towards the end of the runs (1+1) EA and RLS both caught up in value. Both of these algorithms made large improvements in value around 200-500 evaluations. (1+1) EA tended to be more gradual compared to RLS which had sharp increases then flattens out and repeats.

Function 25: NK Landscapes



NK landscapes return the average of n sub functions where each subfunction takes input of k bits specified by an internal neighbourhood matrix. Again, this function continues to show that RLS and the (1+1) EA are very closely related to each other. The algorithms share a similar trend line and have an overlapping standard deviation. Random search performs as expected when compared to the previous functions. It makes small but consistent improvements throughout the evaluations, leading to a lead in function value compared to the others.

Overall comments:

Random search starts off with lower values but consistently makes jumps to reach better optima. RLS EA makes large jumps in function value but with a more gradual increase in early evaluations. (1+1) shares this characteristic with RLS and have very similar mean values for functions 2 through 25.