

These results show that the model *CBI-RWS-OPC-SBM-FSR* is clearly superior to the other models. This result is unsurprising, as the baselines do not actively search for an optimal solution. Furthermore, we see that most evolutionary models surpass their baselines by a wide margin.

The difference in computation time is most likely due to the similarity and sparsity measures. The computation of the Damerau-Levenshtein distance is quadratic. As we also apply an additional custom cost function these computation times increase heavily, the longer the sequence. The evolutionary algorithm as described in ?? is a sequential operation that also increases with the sequence length. However, we can deduce that the time difference between the *CBI-ES-UC3-SBM-RR* stems from either the *Ranking-Recombination* or the *Uniform-Crossing* operation. As those two are the only discernible operators.

In contrast, the baselines have been implemented in ways that vectorize most operations using numpy. Meaning, they can vastly decrease their computation time. The evolutionary algorithms, on the other hand, are subject to python’s notorious [CITE s](#) low-looping operations. However, this is not a vital issue for two reasons. First, it is possible to run evolutionary algorithms in parallel manner [CITE](#). Second, we have not explored more optimized implementations, of either the Semi-structured Damerau-Levenshtein distance (SSDLD) or the evolutionary algorithm. However, we are certain, there are better and fast implementations available.

Knowing these results, a couple of questions remain. Namely, whether the results remain consistent for longer sequences and for other datasets? Furthermore, how does this procedure compare to other methods in the literature? The remaining experiments will address these questions.