# Report for Advanced Applied Statistics 2019

Benjamin Heuser, Frank de Morrée, and Iestyn Watkin
*Niels Bohr Institute - Advanced Applied Statistics*
(Dated: March 6, 2019)

For the course Advanced Applied Statistics an article summary is written. The chosen article is "Planning chemical syntheses with deep neural networks and symbolic AI", by Marwin H. S. Segler, Mike Preuss and Mark P. Waller.

## I. INTRODUCTION

To form organic molecules chemists use retrosynthesis, a technique that transforms a target molecule into increasingly simpler constituents. In this way, chemists can backwardly engineer the molecule with the available constituent building blocks. Computers should prove useful to find viable synthetic routes fast by searching through many options more efficiently than any chemist can. However, computer-aided retrosynthesis proves slow and results unsatisfactory. In the article, the use of Monte Carlo tree search (MCTS) and symbolic artificial intelligence (AI) to discover retrosynthetic routes is discussed.

## II. METHOD

### A. Data selection

The possible transformations are called rules and were taken from the Reaxys chemistry database, encompassing 12.4 million single-step reactions which basically covers all known reactions. Two subsets were formed, one for the rollout neural network and the other for the expansion policy defined later on. For the first, rules were kept that occurred at least 50 times in published reactions before 2015. For the latter, rules occurring at least three times after 2015 were used. This resulted in a sample size of 17,134 and 301,671 rules respectively. The neural networks are trained by reactions published before 2015, whereas for testing and validation, data after 2015 are used.

### B. Monte Carlo tree search

In order to explore the large amount of molecule transformations, a Monte Carlo tree search (MCTS) method was chosen. The main algorithm presented in the paper consists of three different neural networks performing the individual MCTS steps, resulting in the 3N-MCTS algorithm. The first being the 'expansion phase' puts possible transformations of the molecule into a tree data shape. Second, these are passed to the 'in-scope network' evaluating their feasibility (i.e. if the reaction will result in the wanted products). To rate the quality of an explored route, the 'rollout phase' is used which rates the nodes of a tree based on the availability of the building blocks passed to the method. A schematic overview of the method is provided in Figure 1 in Appendix A.

### C. Selection phase

Starting at the root node (i.e. target molecule) of the MCTS tree, the algorithm eq. (2.1) selects the most promising position of children nodes sequentially until a leaf node is reached. If a leaf node is visited for the first time, it is evaluated by the rollout network; otherwise, it is expanded by use of the expansion network. At each step $t$ the next node $a_t$ is selected from all available children nodes $\mathcal{A}(s_t)$ in $s_t$ using eq. (2.1). Here $N(a_{t-1}, s_{t-1})$ is the state-action pair (node-edge pair) that lead to the current node, $c$ is an exploration constant, $Q(s_t, a)$ is the action value (value of the current edge) and $N(s_t, a)$ is the visit count to the current node. The prior probability term $P(s, a)$ allows exploration of the most promising lines of analysis first. This term decays when visited repeatedly, which allows exploration of other options.

$$a_t = \underset{a \in \mathcal{A}(s_t)}{\operatorname{argmax}} \left( \frac{Q(s_t, a)}{N(s_t, a)} + cP(s_t, a) \frac{N(s_{t-1}, a_{t-1})}{1 + N(s_t, a)} \right) \tag{2.1}$$

### D. Expansion phase

The expansion phase is triggered as soon as a leaf node is visited for the second time. In this phase the trained expansion network identifies the 50 most probable transformations of the leaf node. The expansion network is a $1 + 5$ layer highway network, meaning it allows direct connections between different layers of the tree. The activation functions are exponential linear unit (ELU) functions for all layers expect the final one, which uses the softmax function. The last layer returns a probability distribution normalised over all transformations for all the possible reactions.

### E. Rollout phase

The rollout network is used to determine the position value of leaf nodes. If a leaf node is visited for the first time and the node is neither already a solution nor terminal, the network starts evaluating its position. This network starts to recursively sample the top ten actions until solved states or a recursion depth is reached. It then casts a reward function, which returns a value of 1 if a state is proven, a value $z \in [0, 1]$ depending on the ratio of solved to unsolved molecules in the child nodes, and a reward of $-1$ in case a child state is unproven or terminal. The neural network has been trained in the same fashion as the expansion network with a relatively high drop out ratio of 0.4 to increase generalisation potential. The activation function used was an ELU function.

### F. In-scope filter

The in-scope filter removes reactions that are not likely to result in the target molecule. This task is performed by a third neural network, which has been trained as a binary classifier based on reaction data published before 2015. Since data is usually only reported in case of successful synthesis, negative data (i.e. failed reactions) was created by applying the rules from the expansion policy and taking the reactions that differ from those reported. The quality of the in-scope filter was tested by comparing the lowest unoccupied molecular orbital (LUMO) properties of Diels-Adler reactions to the results of the filter and looking at the correlation of these two results. This provides a method to rate the quality of a neural network without any physical knowledge based on an empiric set of rules.

### G. Update phase

Here the values of the nodes, actions from eq. (2.2), and visit counts of the edges from the current node to the root are updated. In eq. (2.2), the indicator function $I_i(s, a)$ equals 1 if the edge was visited during the $i^{th}$ iteration and $z_i$ is the reward assigned during rollout. $W^{L_{max}}(b_i)$ from eq. (2.4) is an objective function used to weight shorter paths as being preferential, with use of the $\xi(b_i)$ function from eq. (2.3), and eliminates routes with a length longer than the maximum.

$$Q(s, a) = \frac{1}{N(s, a} \sum_{i=1}^{n} I_i(s, a), z_i W(b_i) \tag{2.2}$$

$$\xi(b_i) = \text{length}(b_i) - \sum kP(s_j, a_j) \tag{2.3}$$

$$W^{L_{\max}}(b_i) = \max\left(0, \frac{L_{\max} - \xi(b_i)}{L_{\max}}\right) \tag{2.4}$$

## III. RESULTS

The neural network for the expansion policy predicts the correct solution with an accuracy of 31% out of 301,671 reaction transformations, which the authors note to be reasonable. Accuracies of 63.3% and 72.5% for the top 10 and top 50 results were reported respectively, as is shown in Table II.

The in-scope filter network achieved an area under the ROC curve of 0.99 on the test set, and 0.94 under the precision-recall curve. This indicates good performance, as the ROC curve shows the False Positive Rate (incorrect reactions passing the filter) versus the True Positive Rate (correct reactions passing the filter) despite the highly artificial nature of the negative training data.

Performance of the 3N-MCTS was evaluated by comparing it to other state-of-the-art search methods, being MC, UCT and BFS. In Table I, the percentage of routes solved and time taken are listed. This was determined for finding routes for 497 molecules. On both fronts it was found that the 3N-MCTS method provides superior performance.

To assess the quality the resulting routes the method provides, the authors conducted two Wilcoxon signed-rank tests in which 45 graduate-level organic chemists had to choose one of two routes leading to the same molecule. It was found that the experts did not identify the machine routes as inferior to those found in literature: $P = 0.26$, with 43% and 57% preference respectively as shown in Figure 2.

## IV. DISCUSSION AND CONCLUSIONS

Several difficulties arise in the application of the MCTS method to the chemical synthesis problem. For one, the sparsity of the training data is a challenge as the performance of deep neural networks is directly correlated to it. The authors mention stronger but slower algorithms, that take more context of chemical reactions into account, are key to finding routes for natural product synthesis which is impossible in the current setup. In the future, more advanced versions of the method should become a valuable assistant in retrosynthesis and the planning thereof.

For computer-aided synthesis, MCTS combined with neural networks was shown to be effective in performing retrosynthesis planning. The approach is thirty times faster faster and and solves for twice as many problems than other canonical methods. A double-blind AB test also confirmed that expert chemists find the machine results of sufficient quality in comparison to literature routes.

## Appendix A: Figures and Tables

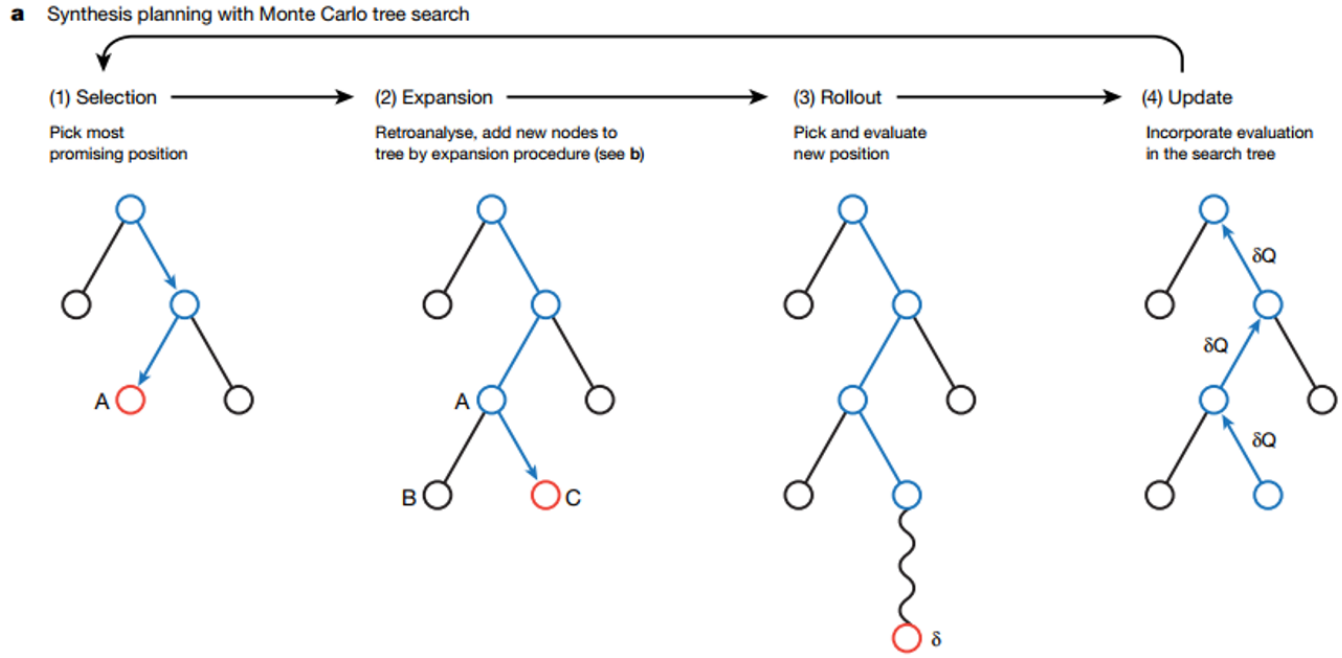


**Figure 1: Schematic of MCTS methodology**. **a**, MCTS searches by iterating over four phases. In the selection phase (1), the most urgent node for analysis is chosen on the basis of the current position values. In phase (2) this node may be expanded by processing the molecules of the position A with the expansion procedure, which leads to new positions B and C, which are added to the tree. Then, the most promising new position is chosen, and a rollout phase (3) is performed by randomly sampling transformations from the rollout policy until all molecules are solved or a certain depth is exceeded. In the update phase (4), the position values are updated in the current branch to reflect the result of the rollout.

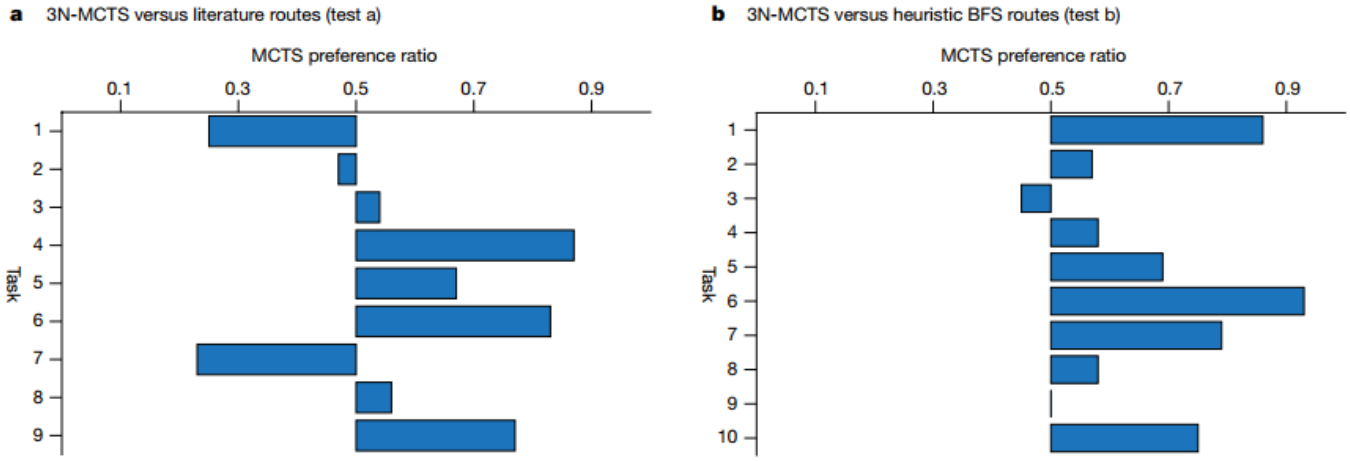


**Figure 2: Double-blind AB testing of MCTS-derived routes against literature and BFS routes. a**, Chemists did not significantly prefer literature routes over routes found by MCTS (Wilcoxon signed-rank test, $P = 0.26$). **b**, Chemists significantly prefer routes found by 3N-MCTS over routes generated by heuristic BFS without a policy network and an in-scope filter (Wilcoxon signed-rank test, P=0.01). A ratio above 0.5 indicates that more than 50% of participants preferred the MCTS solution.

**Table 1 | Experimental results**

| Entry | Search method | Policy* | Percentage solved | Time (seconds per molecule) |
|---|---|---|---|---|
| 1 | MCTS | Neural | 95.24±0.09 | 13.0 |
| 2 | MC | Neural | 89.54±0.59 | 275.7 |
| 3 | UCT | Neural | 87.12±0.29 | 30.4 |
| 4 | BFS | Neural | 84.24±0.09 | 39.1 |
| 5 | BFS | SMILES$^{3/2}$ | 55.53±2.02 | 422.1 |

The time budget was 300 s and 100,000 iterations for MCTS or 300 s and 100,000 expansions for BFS, per molecule. Three restarts were carried out.
*In the BFS, this is the cost function.

**Table I**

**Extended Data Table 1 | Metrics for the supervised neural network policies**

| Policy | # rules | Coverage | Matching rules/mol[b] | Accuracy[a] | top10Acc[a] | top50Acc[a] |
|---|---|---|---|---|---|---|
| Expansion | 301,671 | 0.79 | 46,175 | 0.310 | 0.633 | 0.725 |
| Rollout | 17,134 | 0.52 | 321 | 0.501 | 0.891 | 0.964 |

Top10Acc/top50Acc is the ratio of correct/incorrect predictions if we allow the system to make 10 or 50 predictions.
[a]Accuracy is calculated on the molecules covered by the respective rulebase.
[b]Matching rules/mol corresponds to the branching factor.

**Table II**