Monte Carlo Tree Search for Chemistry and Biology

AI Chemist

10 March 2026

Tristan Cazenave

LAMSADE, Université Paris Dauphine - PSL

Tristan.Cazenave@dauphine.fr

Outline

- Monte Carlo Tree Search
- Modeling Gene Regulatory Networks
- Nested Monte Carlo Search
- Retrosynthesis
- Drug Discovery
- Nested Rollout Policy Adaptation
- RNA Design

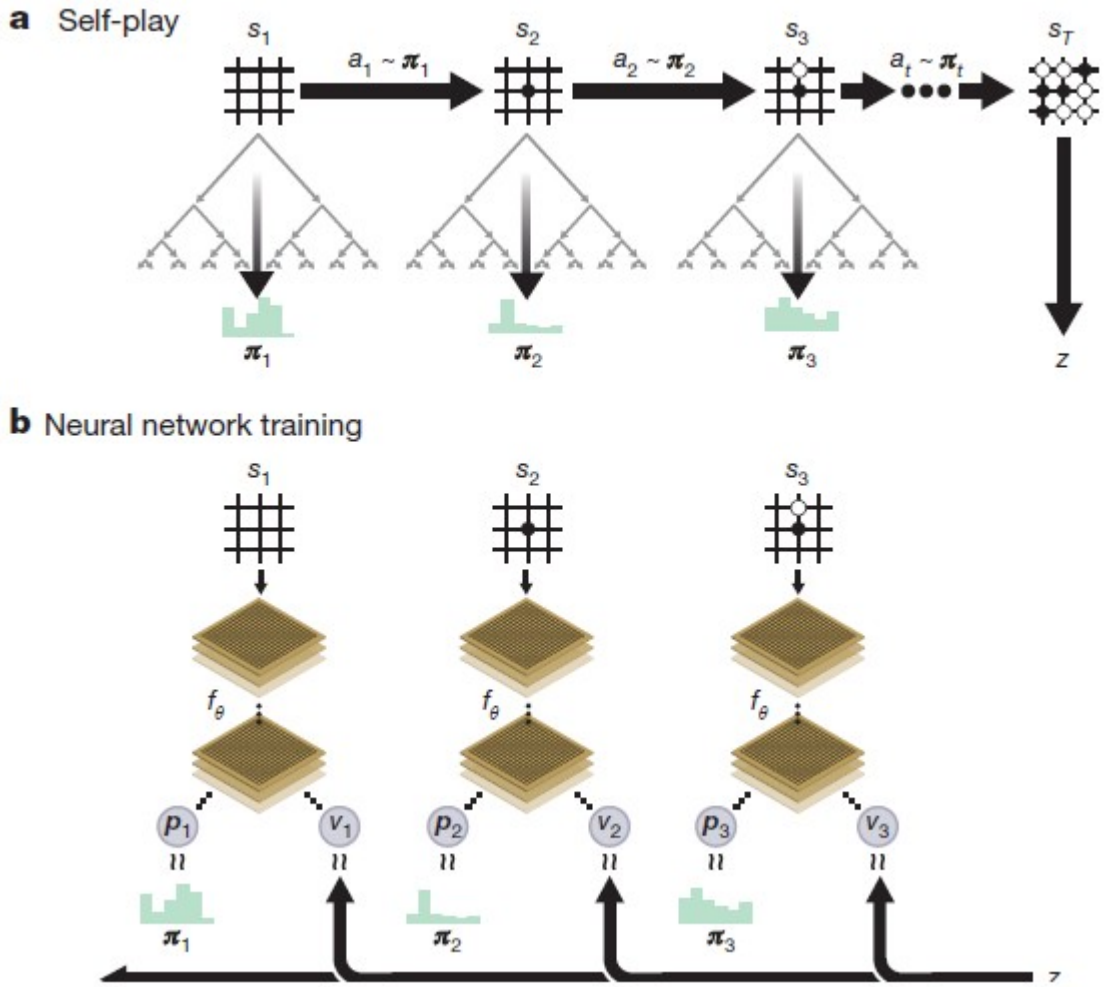
Monte Carlo Tree Search

AlphaGo

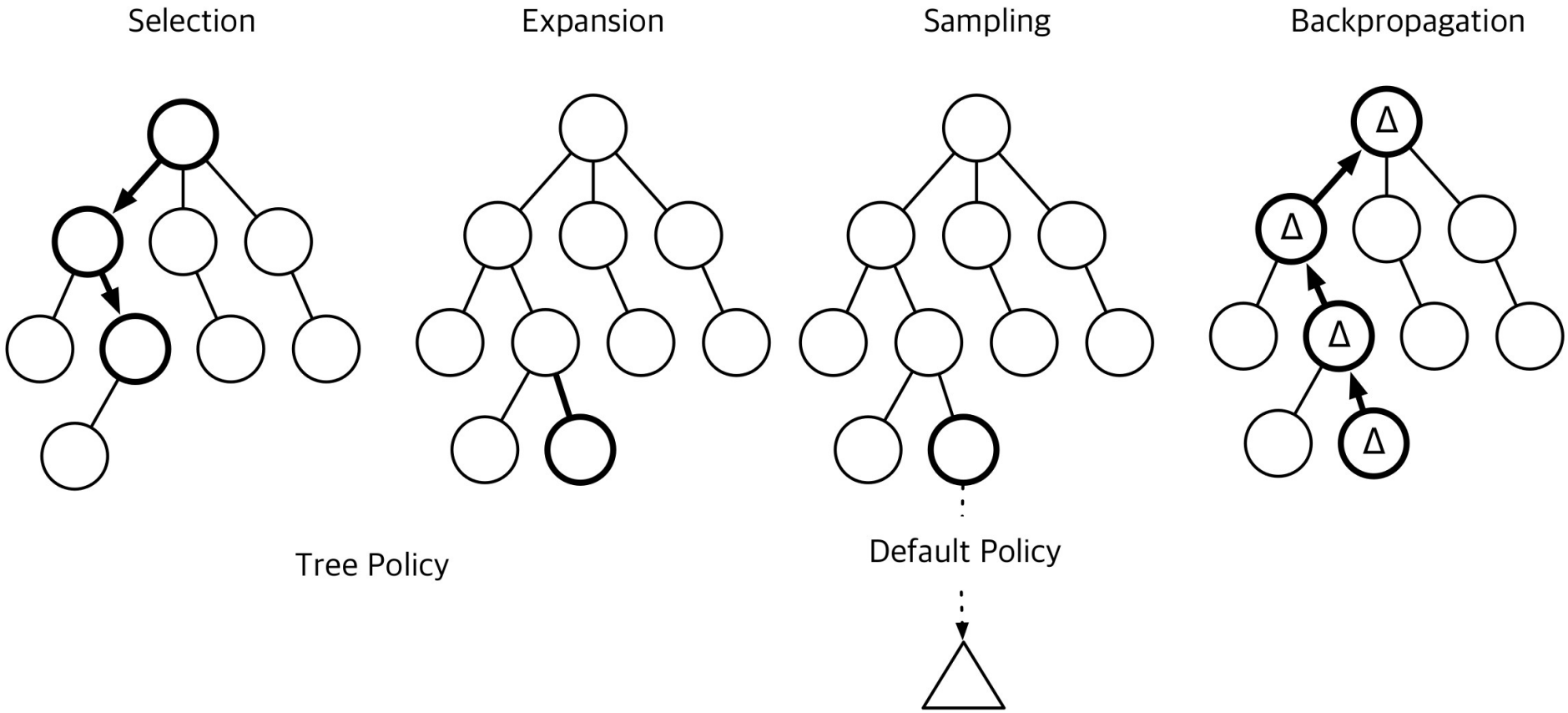
Lee Sedol is among the strongest and the most famous 9p Go player :

AlphaGo Lee won 4-1 against Lee Sedol in march 2016.

AlphaGo Zero



MCTS


Selection
Expansion
Sampling
Backpropagation
Tree Policy
Default Policy

GRAVE

- State of the art in General Game Playing (GGP)
- Best AI of the Ludii system (https://ludii.games/)
- Simple modification of RAVE
- Uses statistics both for Black and White at all nodes
- “In principle it is also possible to incorporate the AMAF values, from ancestor subtrees. However, in our experiments, combining ancestor AMAF values did not appear to confer any advantage.”

Modeling Gene Regulatory Networks

Improving continuous Monte Carlo Tree Search for identifying parameters in hybrid Gene Regulatory Networks

Romain Michelucci[1], Denis Pallez[1], Tristan Cazenave[2], and Jean-Paul Comet[1]

Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France
firstname.name@univ-cotedazur.fr
LAMSADE, Université Paris Dauphine - PSL, CNRS, Paris, France
firstname.name@lamsade.dauphine.fr

Abstract. Monte-Carlo Tree Search (MCTS) is largely responsible for the improvement not only of many computer games, including Go and General Game Playing (GPP), but also of real-world continuous Markov decision process problems. MCTS initially uses the Upper Confidence bounds applied to Trees (UCT), but the Rapid Action Value Estimation (RAVE) heuristic has rapidly taken over in the discrete and continuous domains. Recently, generalized RAVE (GRAVE) outperformed such heuristics in the discrete domain. This paper is concerned with extending the GRAVE heuristic to continuous action and state spaces (cGRAVE). To enhance its performance, we suggest an action decomposition strategy to break down multidimensional actions into multiple unidimensional actions, and we propose a selective policy based on constraints that bias the playouts and select promising actions in the search tree. The approach is experimentally validated on a real-world biological problem: the goal is to identify the continuous parameters of gene regulatory networks (GRNs).

Keywords: MCTS · continuous GRAVE · constraints-based selective policy · action decomposition · chronotherapy · hybrid GRN.

1 Introduction

MCTS is a general decision-time planning algorithm that was initially designed for the improvement of computer Go [13]. The MCTS core idea is to incrementally build a search tree whose nodes represent the states of the environment and edges represent the actions taken from one state to a successor state. MCTS has proved to be effective in a wide variety of settings, including General Game Playing (GGP) [15, 23] but is not limited to games [5, 26]: it can be effective for single-agent sequential decision problems if there is an environment model simple enough for fast multistep simulation. The most popular MCTS algorithm is Upper Confidence bounds applied to Trees (UCT) [19], which addresses the exploration *versus* exploitation trade-off in each state of the tree search using the Upper Confidence Bound [1]. The Rapid Action Value Estimate [16, 17] is a

Hybrid Gene Regulatory Networks

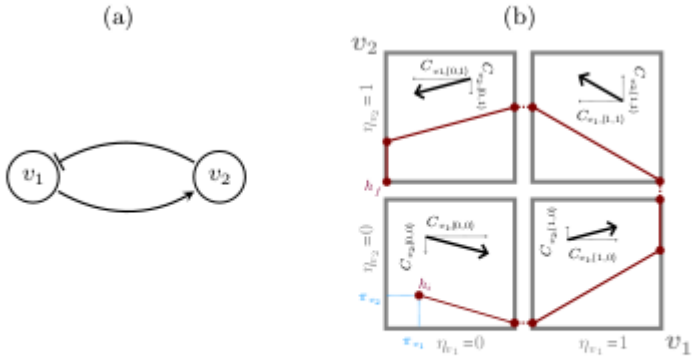


Fig. 2: Example of a hGRN depicted as a directed graph (a), and a possible hybrid state graph (b). The hGRN dynamic parameters are depicted as black arrows.

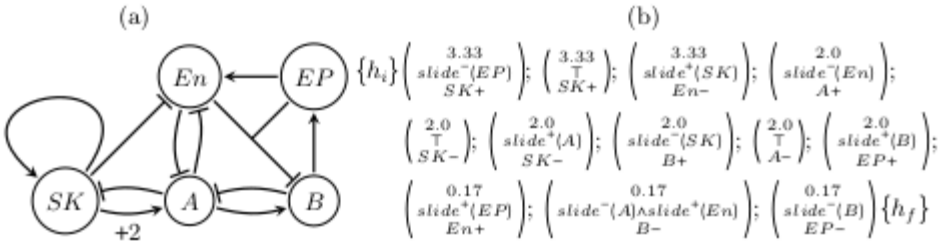


Fig. 3: Interaction graph of the 5-genes hGRN (a) and its corresponding biological knowledge (b).

cGRAVE

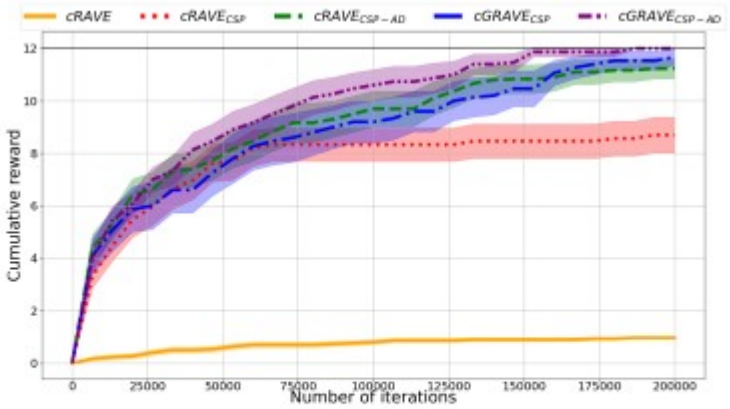
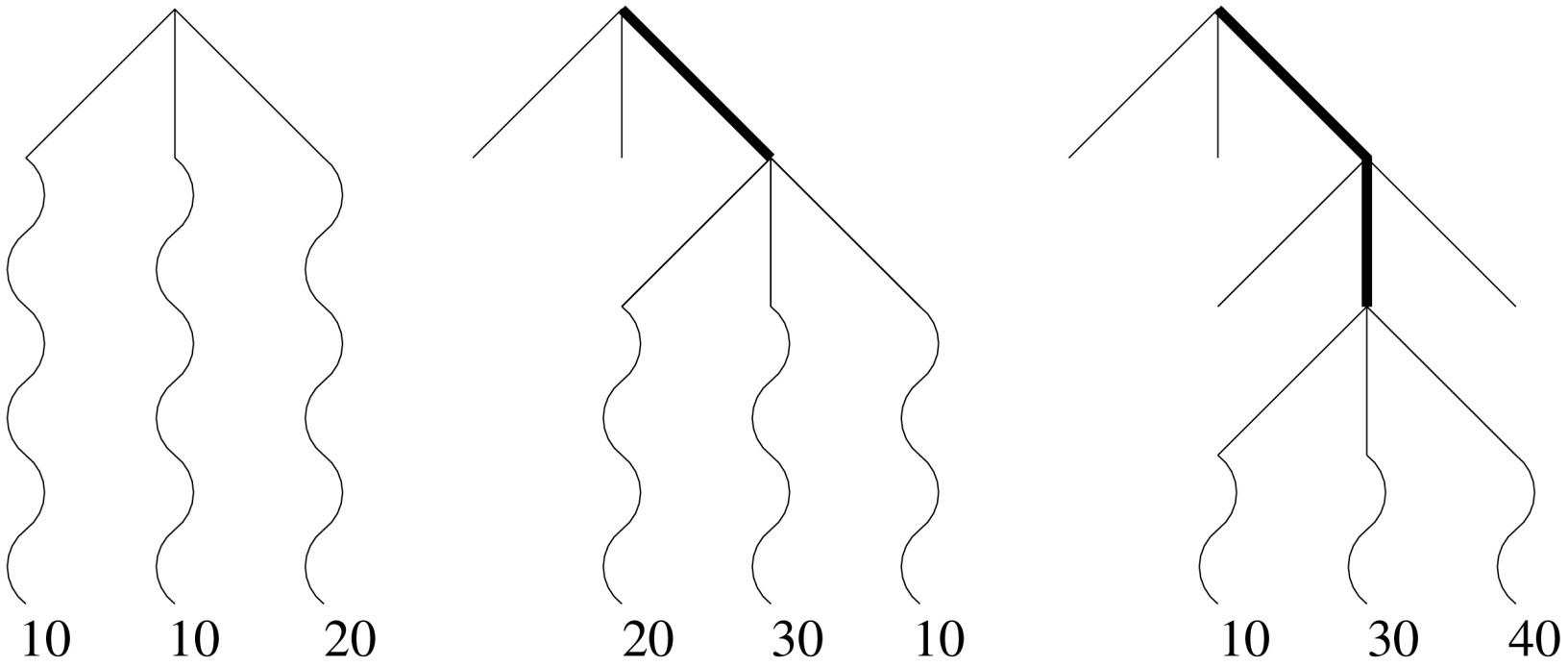


Fig. 4: Comparative performances (cumulative reward) of the different variants on the 5 genes hGRN, versus the computational budget (number of iterations). The upper the better: a reward of 12 means that a solution is found.

Nested Monte Carlo Search

Nested Monte-Carlo Search



Refutation of Spectral Graph Theory Conjectures

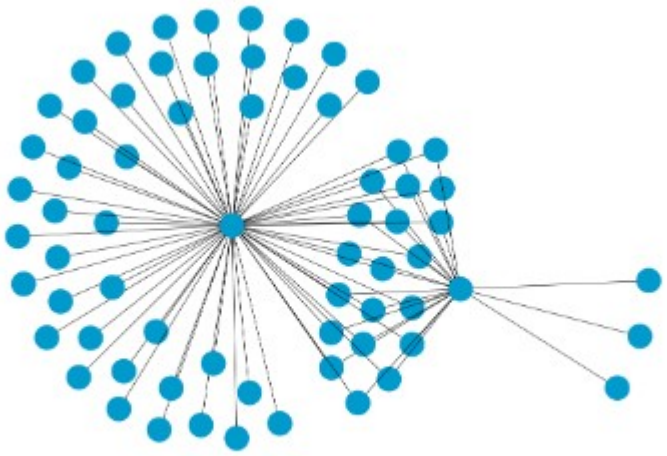


Figure 5. A counter-example of Graffiti 137 of size 67 (second largest eigenvalue $\leq$ harmonic)

- Monte Carlo Search better than Deep RL [Roucairol & Cazenave 2022]

Coalition Structure Generation

- Lazy Nested Monte Carlo Search with clever state space [Roucairol et al. 2024] :

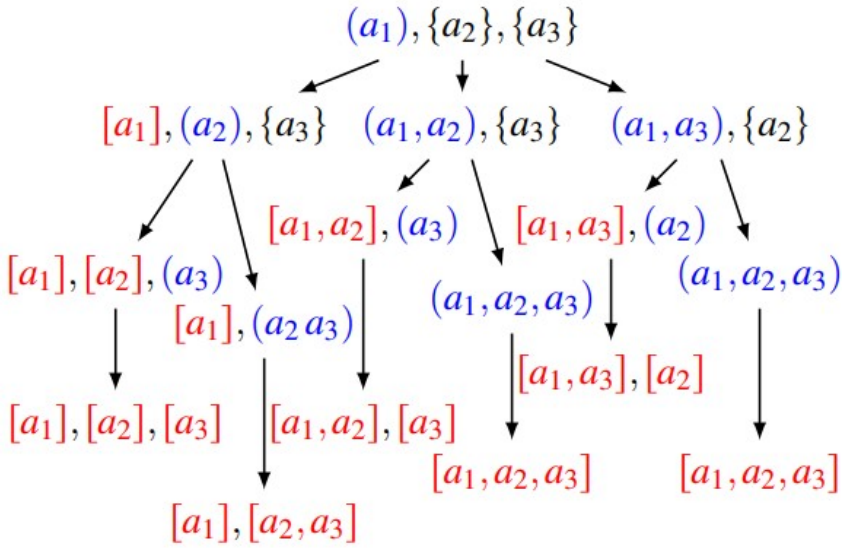


Figure 2: Model B: an example with three agents. We denote {} when the coalition is not locked and not active, () when the coalition is not locked and active, and [] when the coalition is locked.

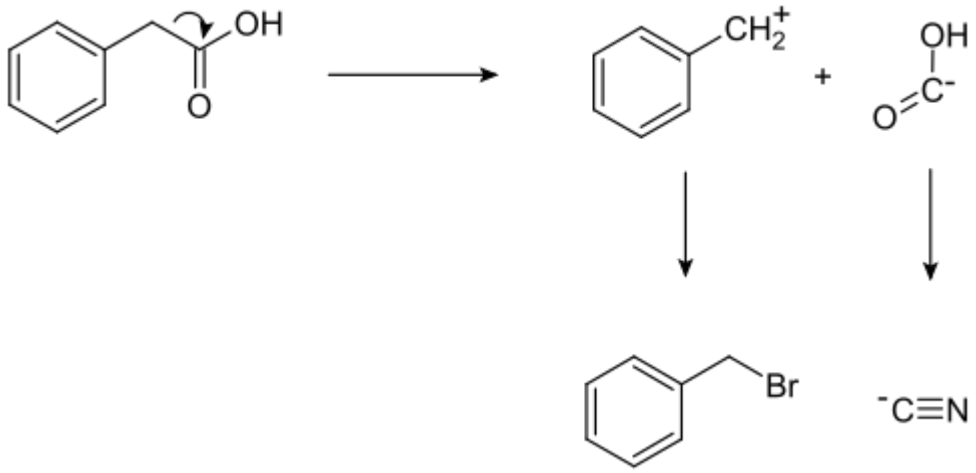
Applications

Nested Monte Carlo Search :

- Morpion Solitaire [Cazenave 2009]
- SameGame [Cazenave 2009]
- Sudoku [Cazenave 2009]
- Expression Discovery [Cazenave 2010]
- The Snake in the Box [Kinny 2012]
- Cooperative Pathfinding [Bouzy 2013]
- Software Testing [Poulding et al. 2014]
- Heuristic Model-Checking [Poulding et al. 2015]
- Pancake problem [Bouzy 2015]
- Games [Cazenave et al. 2016]
- Cryptography [Dwivedi et al. 2018]
- RNA inverse folding problem [Portela 2019]
- Perfect Rectangle Packing [Doux et al. 2022]
- Refutation of Spectral Graph Theory Conjectures [Roucairol et al. 2022]
- Coalition Structure Generation [Roucairol et al. 2024]
- Optimization of Radars [Ardon et al. 2024]
- Neural Architecture Search [Lallouet et al. 2024]
- Retrosynthesis [Roucairol et al. 2024]
- Drug-like molecule generation [Roucairol et al. 2024]
- …

Retrosynthesis

Retrosynthesis



Retrosynthesis

- Find a set of chemical reactions that enable to synthetize a given molecule.
- The state space is an AND/OR tree as in games.
- DF-PN and MCTS have been used to find retrosynthesis pathways.
- Alphachem [Segler et al. 2017].
- AiZynthFinder [Genheden et al. 2020].

Received: 26 September 2023 | Revised: 19 February 2024 | Accepted: 26 March 2024

DOI: 10.1002/minf.202300259

RESEARCH ARTICLE

molecular informatics

Comparing search algorithms on the retrosynthesis problem

Milo Roucairol | **Tristan Cazenave**

LAMSADE, Université Paris Dauphine - PSL, Paris, France

Correspondence
Milo Roucairol, LAMSADE, Université Paris Dauphine - PSL, Paris, France.
Email: milo.roucairol@dauphine.eu

Funding information
French government under the management of Agence Nationale de la Recherche, Grant/Award Number: ANR19-P3IA-0001

Abstract

In this article we try different algorithms, namely Nested Monte Carlo Search and Greedy Best First Search, on AstraZeneca's open source retrosynthetic tool : AiZynthFinder. We compare these algorithms to AiZynthFinder's base Monte Carlo Tree Search on a benchmark selected from the PubChem database and by Bayer's chemists. We show that both Nested Monte Carlo Search and Greedy Best First Search outperform AstraZeneca's Monte Carlo Tree Search, with a slight advantage for Nested Monte Carlo Search while experimenting on a playout heuristic. We also show how the search algorithms are bounded by the quality of the policy network, in order to improve our results the next step is to improve the policy network.

KEYWORDS
MCTS, Monte Carlo Tree Search, retrosynthesis, search algorithm

1 | INTRODUCTION

Retrosynthesis is a domain of organic chemistry that consists of finding a synthetic route (a sequence of reactions) for a given molecule in order to synthesize it from a given set of available precursor molecules [1]. It is an important part of organic chemistry molecule synthesis, and can be used to produce newfound drugs. What we aim for in this paper is to evaluate the strengths and weaknesses of two search algorithms by comparing them to AiZynthFinder's Monte Carlo Tree Search (MCTS) on a small benchmark consisting of curated and complex molecules, covering many reactions encountered by chemists.

The second section presents the retrosynthesis problem, the third section presents the AiZynthFinder retrosynthesis tool, the fourth section describes the search algorithms we compare, the fifth section details the benchmark used to compare the search algorithms, and the sixth section gives experimental results.

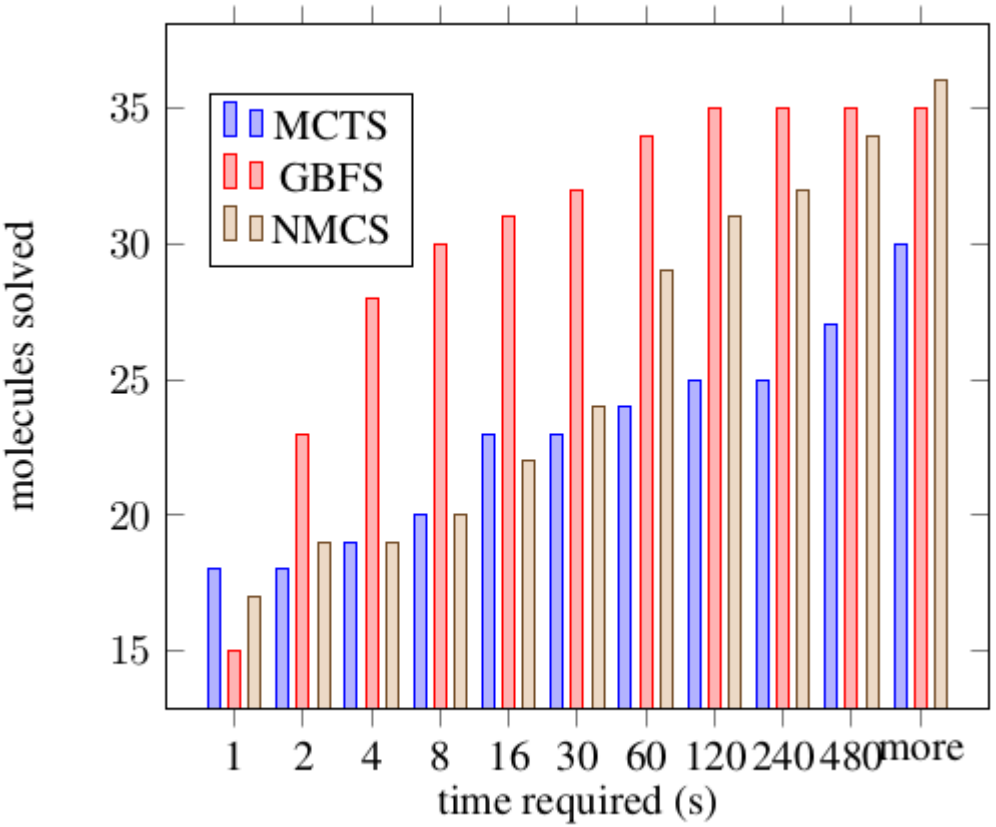
2 | THE RETROSYNTHESIS PROBLEM

Before diving into the details, let's broadly present the retrosynthesis problem.

- precursors: molecules that form one or multiple product molecules when they react together. ZINC [2] is a database of precursors that are available on the market.
- reaction template: a patent predicting the product of the reaction of one or multiple molecules. USPTO is a database of reaction template patents.
- One step retrosynthesis: an important part of retrosynthesis is selecting a few promising reaction templates before applying them as MCTS moves, this step uses a neural network.

As said before: the retrosynthetic analysis of a molecule is trying to find a sequence of reactions from a

Retrosynthesis



Drug Discovery

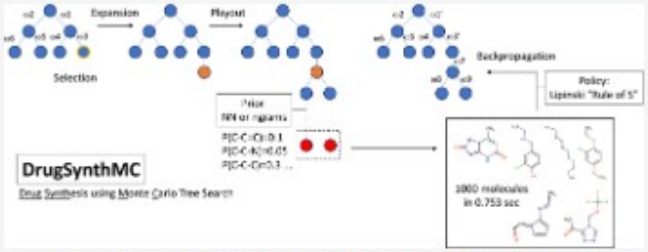
JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim
Article

DrugSynthMC: An Atom-Based Generation of Drug-like Molecules with Monte Carlo Search

Milo Roucairol, Alexios Georgiou, Tristan Cazenave,* Filippo Prischi,* and Olivier E. Pardo*

Cite This: https://doi.org/10.1021/acs.jcim.4c01451
Read Online



ABSTRACT: A growing number of deep learning (DL) methodologies have recently been developed to design novel compounds and expand the chemical space within virtual libraries. Most of these neural network approaches design molecules to specifically bind a target based on its structural information and/or knowledge of previously identified binders. Fewer attempts have been made to develop approaches for *de novo* design of virtual libraries, as synthesizability of generated molecules remains a challenge. In this work, we developed a new Monte Carlo Search (MCS) algorithm, DrugSynthMC (Drug Synthesis using Monte Carlo), in conjunction with DL and statistical-based priors to generate thousands of interpretable chemical structures and novel drug-like molecules per second. DrugSynthMC produces drug-like compounds using an atom-based search model that builds molecules as SMILES, character by character. Designed molecules follow Lipinski's "rule of 5", show a high proportion of highly water-soluble nontoxic predicted-to-be synthesizable compounds, and efficiently expand the chemical space within the libraries, without reliance on training data sets, synthesizability metrics, or enforcing during SMILES generation. Our approach can function with or without an underlying neural network and is thus easily explainable and versatile. This ease in drug-like molecule generation allows for future integration of score functions aimed at different target- or job-oriented goals. Thus, DrugSynthMC is expected to enable the functional assessment of large compound libraries covering an extensive novel chemical space, overcoming the limitations of existing drug collections. The software is available at https://github.com/RoucairolMilo/DrugSynthMC.

INTRODUCTION

Since the 1980s, *in silico* approaches have been extensively and routinely used in drug discovery and have transformed the medicinal chemistry field,[1] with expectation to do so even more in the future. The need for rapid response, highlighted by the emergence of resistant bacteria and, among others, the COVID-19 pandemic, has fueled the development of novel computational tools for drug design and screening.[2] *In silico* virtual-library screening (VS) is usually the first critical step in structure-based drug discovery, where the algorithm aims to predict the best matching binding mode of a ligand to a receptor.[3] Despite the many attempts to improve accuracy of VS methods,[4,5] the relatively limited chemical diversity of compounds in libraries reduces the ability of structure-based VS to identify hits and leads.[6,7] Indeed, it has been estimated that only a small portion (10^{8}–10^{2}) of the 10^{63} drug-like molecules predicted to be synthetically accessible has been explored.[8]

Several studies have shown that screening larger libraries that expand the accessible molecules by several order of magnitude ($\sim 10^{11}$) improves the rate of true high affinity (nM-pM) binders.[9−13] To further expand the chemical space within virtual libraries, generative models based on deep learning (DL) methodologies have been used to produce molecules

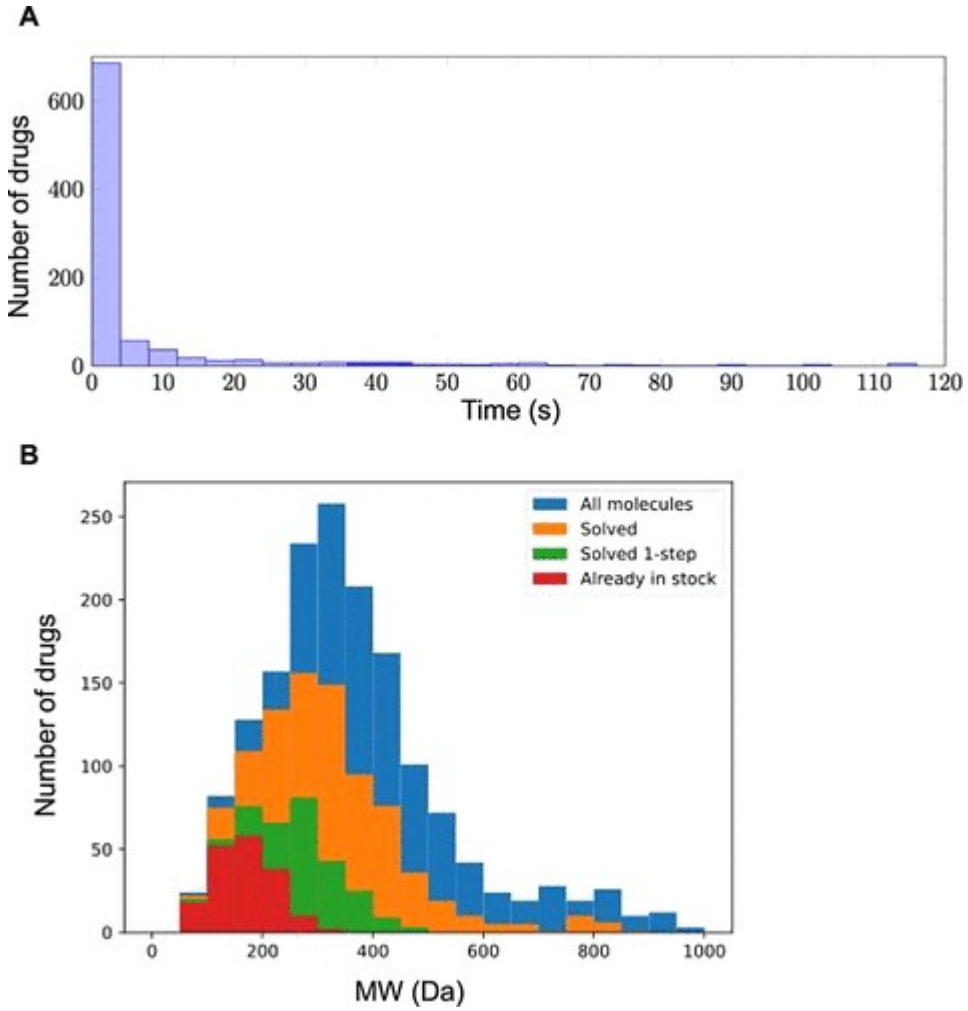
Received: August 12, 2024
Revised: August 16, 2024
Accepted: August 16, 2024

https://doi.org/10.1021/acs.jcim.4c01451
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

DrugSynthMC

- De novo design of virtual libraries
- Statistics on ngrams with the molecules of the FDA
- Lipinski rule of 5
- Synthesizability with AIZynthfinder
- Thousands of novel drug-like molecules per second
- Very small dataset used to train the ngrams (FDA)
- Future work : target oriented evaluation

DrugSynthMC



Nested Rollout Policy Adaptation

Nested Rollout Policy Adaptation

- NRPA is NMCS with policy learning.
- It uses sampling with a softmax of the move weights as a playout policy.
- There are recursive levels of best sequences as in NMCS.
- There is a policy at each level.
- The policy is reinforced on the best sequence.

Nested Rollout Policy Adaptation

- Each move is associated to a weight w_i
- During a playout each move is played with a probability:

$$\exp(w_i) / \Sigma_k \exp(w_k)$$

Nested Rollout Policy Adaptation

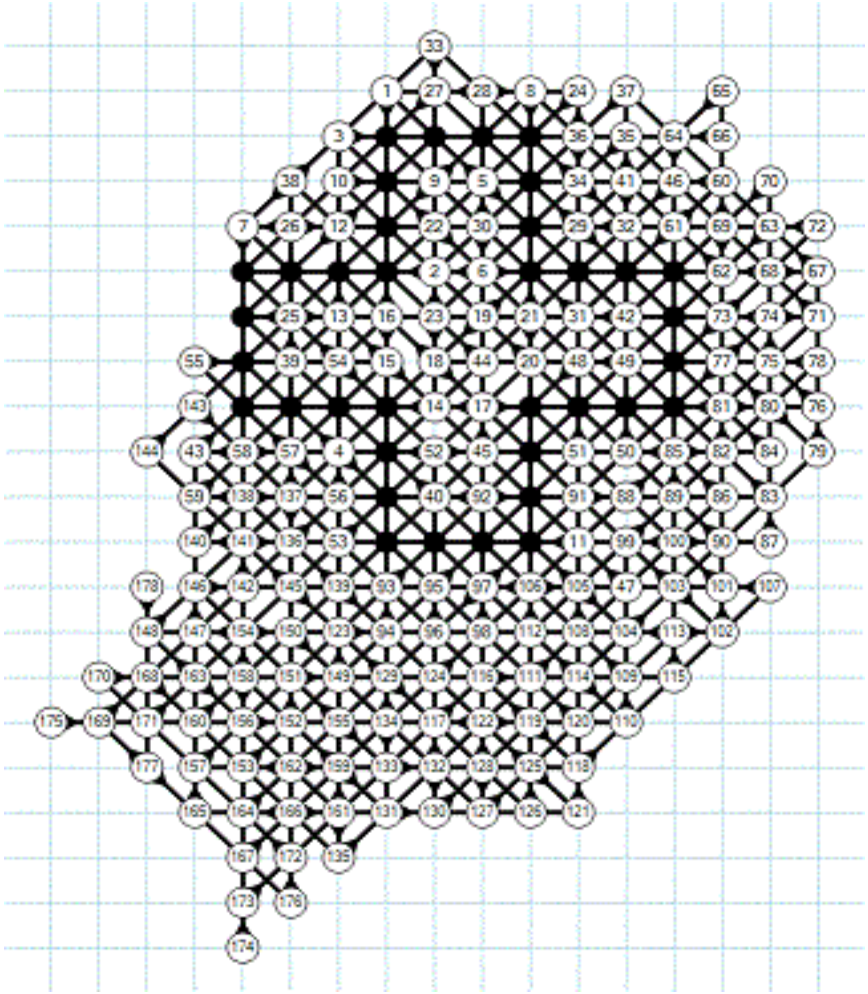
- For each move of the best sequence:

  $w_i = w_i + 1$

- For each possible move of each state of the best sequence:

  $w_j = w_j - \exp(w_j) / \Sigma_k \exp(w_k)$

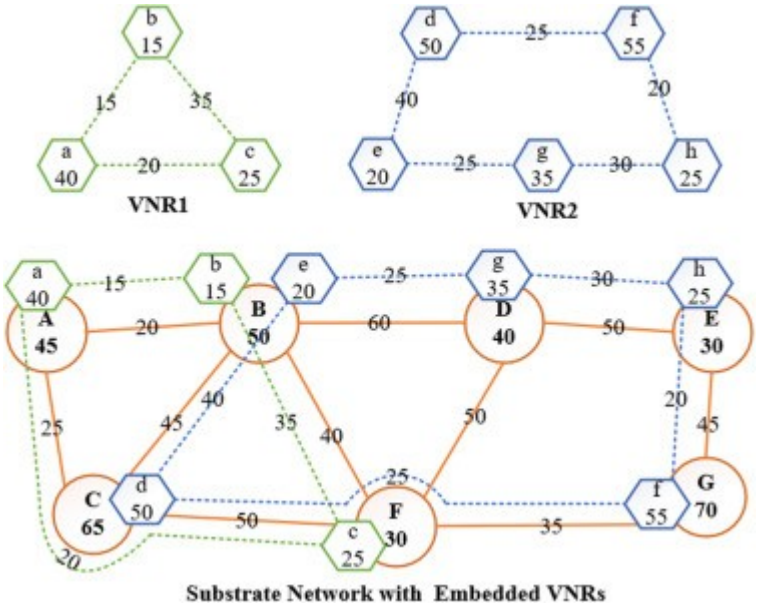
Morpion Solitaire



World record [Rosin 2011]

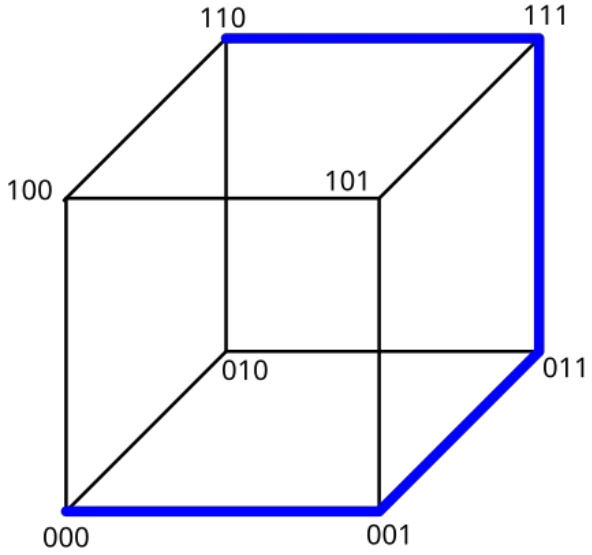
Virtual Network Embedding

- MCTS for 5G network slicing [Elkael 2023]



Snake in the Box

- Find a long path in an hypercube :



Dimension	Meta-NRPA-fe	Best Known Score
7	50	50
8	97	98
9	188	190
10	**373**	370
11	**721**	712
12	**1383**	1373
13	**2709**	2687

Table 5: Comparison of Meta-NRPA with known lower bounds on the Snake-in-the-Box

- Improved lower bounds [Dang & al. 2023]

Nested Rollout Policy Adaptation

- Morpion Solitaire [Rosin 2011]
- CrossWords [Rosin 2011]
- Traveling Salesman Problem with Time Windows [Cazenave et al. 2012]
- 3D Packing with Object Orientation [Edelkamp et al. 2014]
- Multiple Sequence Alignment [Edelkamp et al. 2015]
- SameGame [Cazenave et al. 2016]
- Vehicle Routing Problems [Edelkamp et al. 2016, Cazenave et al. 2020]
- Graph Coloring [Cazenave et al. 2020]
- RNA Design [Cazenave & Fournier 2020]
- Network Traffic Engineering [Dang & al. 2021]
- Refutation of Spectral Graph Theory Conjectures [Roucairol & Cazenave 2022]
- Slicing 5G [Elkael et al. 2023]
- Snake in the Box [Dang et al. 2023]
- Latin Square Completion and Kakuro [Cazenave 2024]
- Flexible Job Shop Scheduling [Kobrosly et al. 2025]
- …

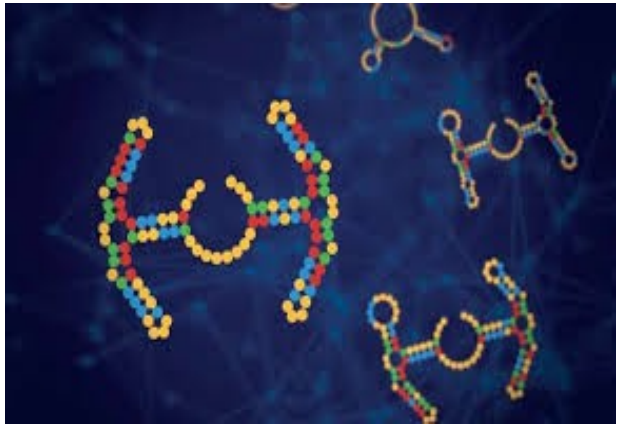
RNA Design

RNA Design

- Molecule Design as a Search Problem
- Find the sequence of nucleotides that gives a predefined structure
- Useful for synthetic biology, medicine, and nanotechnology
- GREED-RNA: Greedy Local Search

RNA Design

- Find a sequence that has a given folding



eterna
Solve Puzzles. Invent Medicine.

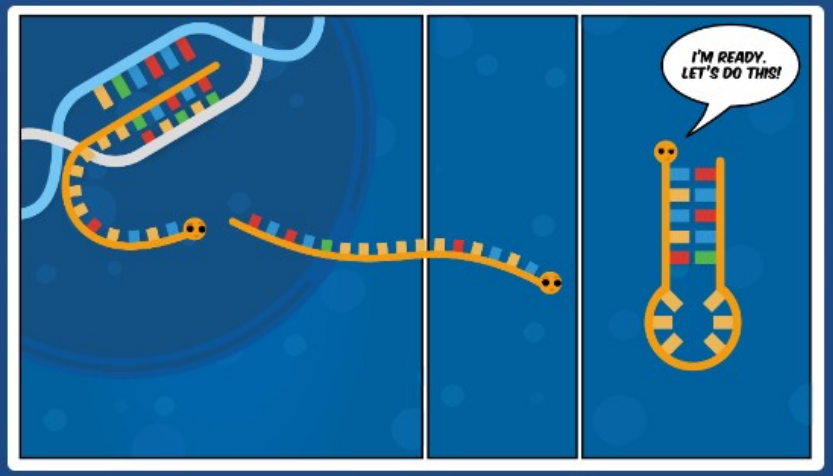
Log in Register

HOME PLAY LEARN COMMUNITY NEWS ABOUT

What is RNA?

RNAs are tiny molecules in the cells of every living thing. They copy information from DNA and use it to make things happen in the cell.

Like DNA, RNA is made up of four bases. Each RNA folds into a shape that determines its function, and the shape is defined by the pattern of the bases.



Eterna 100

- Human experts have managed to solve the 100 problems of the benchmark
- No program has so far achieved such a score.
- The best score in 2018 was 95/100 by NEMO: NEsted MOnte Carlo RNA Puzzle Solver

NEMO

- NEMO uses two sets of heuristics
- General ones that give probabilities to pairs of bases.
- More specific ones that are tailored to puzzle solving.

GNRPA

Let w_{ib} be the weight associated to move b at index i in the sequence. In NRPA the probability of choosing move b at index i is:

$$p_{ib} = \frac{e^{w_{ib}}}{\sum_k e^{w_{ik}}}$$

We propose to try Generalized NRPA (GNRPA) [9] for Inverse Folding and to replace it with:

$$p_{ib} = \frac{e^{w_{ib}+\beta_{ib}}}{\sum_k e^{w_{ik}+\beta_{ik}}}$$

where we use for β_{ib} the logarithm of the probabilities used in NEMO.

Learning a Prior for Monte Carlo Search by Replaying Solutions to Combinatorial Problems

Tristan Cazenave

LAMSADE, Université Paris Dauphine - PSL, CNRS, Paris, France

Abstract. Monte Carlo Search gives excellent results in multiple difficult combinatorial problems. Using a prior to perform non uniform playouts during the search improves a lot the results compared to uniform playouts. Handmade heuristics tailored to the combinatorial problem are often used as priors. We propose a method to automatically compute a prior. It uses statistics on solved problems. It is a simple and general method that incurs no computational cost at playout time and that brings large performance gains. The method is applied to three difficult combinatorial problems: Latin Square Completion, Kakuro, and Inverse RNA Folding.

1 Introduction

Monte Carlo Tree Search (MCTS) has been successfully applied to many games and problems [4]. It has superhuman performances in two player complete information games such as Go and Chess [32].

Nested Monte Carlo Search (NMCS) [6] is an algorithm that works well for puzzles and combinatorial problems. It biases its playouts using lower level playouts. At level zero NMCS adopts a uniform random playout policy. Learning of playout strategies combined with NMCS has given good results on combinatorial problems [28]. Other applications of NMCS include Single Player General Game Playing [24], Cooperative Pathfinding [2], Software testing [26], heuristic Model-Checking [27], the Pancake problem [3], Games [10], the Inverse RNA Folding problem [25] and retrosynthesis [30].

Online learning of a playout policy in the context of nested searches has been further developed for puzzles and combinatorial problems with Nested Rollout Policy Adaptation (NRPA) [29]. NRPA has found new world records in Morpion Solitaire and crosswords puzzles. NRPA has been applied to multiple problems: the Traveling Salesman Problem with Time Windows (TSPTW) [11,13], 3D Packing with Object Orientation [15], the physical traveling salesman problem [16], the Multiple Sequence Alignment problem [17] or Logistics [14]. The principle of NRPA is to adapt the playout policy so as to reinforce the best sequence of moves found so far at each level.

The use of Gibbs sampling in Monte Carlo Tree Search dates back to the general game player Cadia Player and its MAST playout policy [19].

Monte Carlo Search for combinatorial problems can be much improved using a prior. A prior is a heuristic that is used in playouts to sample in a non uniform way. It favors some moves in the playout according to the heuristic. The use of a bias or the

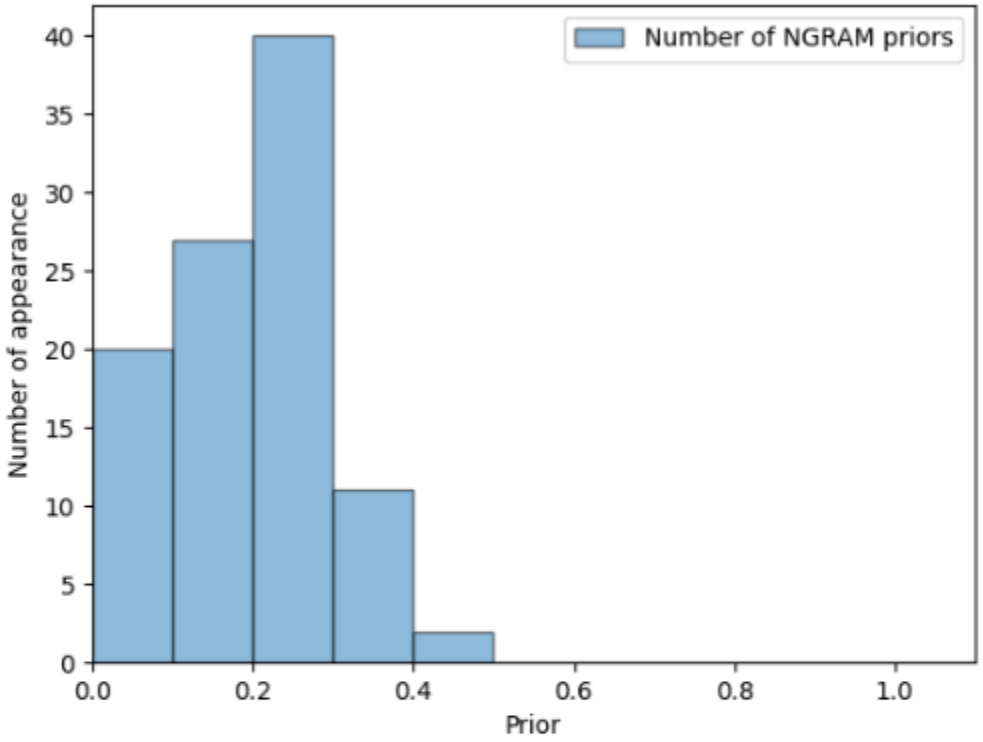


Fig. 4: The distribution of the priors for Inverse RNA Folding. The y-axis gives the number of priors in each range of values. There are 6 possible moves for a '(' and 4 possible moves for a '.' in the target structure. This makes 10 possibilities for the previous move in the NGRAM and again 10 possibilities for the current move. Therefore there are 100 different priors. On the contrary of LSC and Kakuro the distribution of the priors is mainly on small values. The smallest prior is equal to 0.010083 and the greatest prior is equal to 0.437825.

Table 3: Number of Eterna100 problems solved by different algorithms and various search time limits in seconds. GNRPA is much better than NRPA. The NGRAM prior is better than the NEMO prior. The temperature for the NGRAM prior is $\tau = 6$. Sampling with the NGRAM prior is better than sampling with the NEMO prior. Sampling with a prior is much better than uniform sampling.

Algorithm	32s	64s	128s	256s	512s	1,024s	2,048s	4,096s
Sampling	11	11	11	12	14	16	16	17
Sampling NEMO prior	51	55	57	60	61	61	62	64
Sampling NGRAM prior	57	65	68	69	69	69	69	69
NRPA	28	33	41	48	57	59	61	65
GNRPA NEMO prior	68	69	74	77	78	79	81	81
GNRPA NGRAM prior	70	75	78	79	80	81	82	85

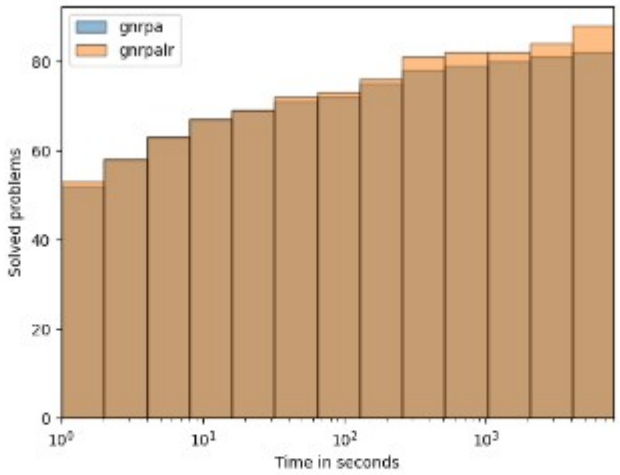


Figure 2: Comparison of GNRPA and GNRPALR for Inverse RNA Folding. The number of repetitions is set to 0 for GNRPALR. GNRPALR is eight times faster than GNRPA. It solves 88 problems in 4,096 seconds when GNRPA solves 82. The relative performance of GNRPALR improves with more search time. The tests are made using the 100 problems of Eterna100.

RNA Design

- Montparnasse
- Multi Objective Generalized Nested Rollout Policy Adaptation with Limited Repetitions
- Base Pair Distance (BPD), Hamming Distance, …
- Stop search at a level if the same best sequence is found a second time.
- Prior on CG, GC and A.

RNA Design

(a) Gladius: problem 90

(b) Shooting Star: problem 99

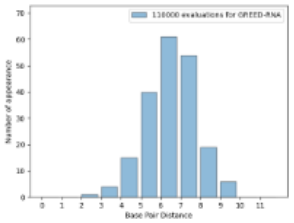
(c) Teslagon: problem 100

RNA Design

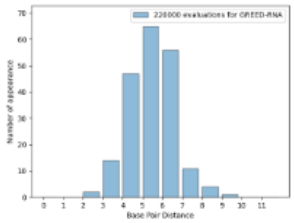
Table 1: Distributions of the BPD of the various algorithms after 270 000 evaluations for problem 99.

BPD	0	1	2	3	4	5	6	7	8
GREED-RNA	6	22	49	66	38	17	2	0	0
MOGRLS	19	46	63	39	22	7	2	2	0
PN	28	72	64	28	8	0	0	0	0
MOGNRPALR	120	78	2	0	0	0	0	0	0

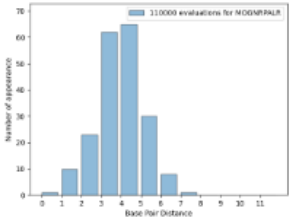
RNA Design



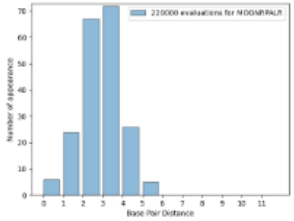
(a) Distribution of the BPD for problem 90 after 110 000 evaluations by GREED-RNA.



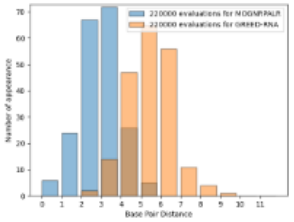
(b) Distribution of the BPD for problem 90 after 220 000 evaluations by GREED-RNA.



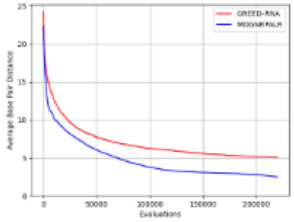
(c) Distribution of the BPD for problem 90 after 110 000 evaluations by MOGNR-PALR.



(d) Distribution of the BPD for problem 90 after 220 000 evaluations by MOGNR-PALR.



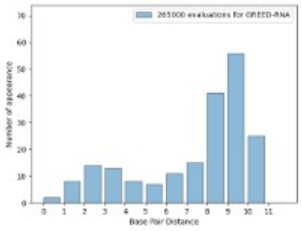
(e) Comparison of the distributions after 220 000 evaluations between GREED-RNA and MOGNRPALR.



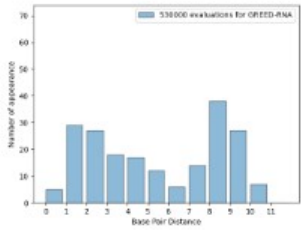
(f) Evolution of the average BPD of GREED-RNA and MOGNRPALR for problem 90.

Fig. 5: Comparison of the BPD on problem 90 for GREED-RNA and MOGNRPALR for increasing numbers of evaluations. 220 000 evaluations by one process takes one day. GREED-RNA is stuck and does not solve the problem while MOGNRPALR progresses and solves the problem 6 times out of 200 runs.

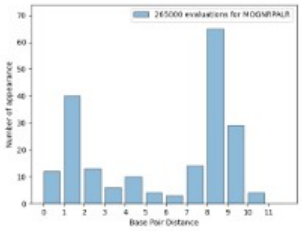
RNA Design



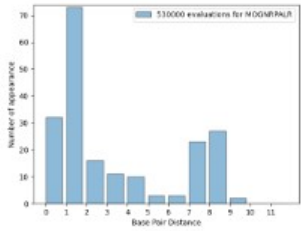
(a) Distribution of the BPD for problem 100 after 265 000 evaluations by GREED-RNA.



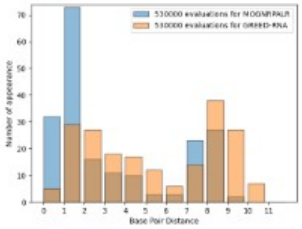
(b) Distribution of the BPD for problem 100 after 530 000 evaluations by GREED-RNA.



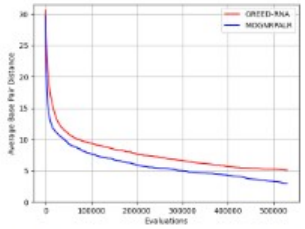
(c) Distribution of the BPD for problem 100 after 265 000 evaluations by MOGN-RPALR.



(d) Distribution of the BPD for problem 100 after 530 000 evaluations by MOGN-RPALR.



(e) Comparison of the distributions after 530 000 evaluations between GREED-RNA and MOGNRPALR.



(f) Evolution of the average BPD of GREED-RNA and MOGNRPALR for problem 100.

Fig. 6: Comparison of the BPD on problem 100 for GREED-RNA and MOGNRPALR for increasing numbers of evaluations. 530 000 evaluations by one process takes one day. GREED-RNA solves problem 100 less frequently than MOGNRPALR.

RNA Design

- The most difficult problems from Eterna are solved within one day.
- Eterna consists of puzzles for the secondary structure.
- Next step : 3D design.

3D RNA Design

Why tertiary RNA design for drug discovery?

- Therapeutic RNAs (aptamers, ribozymes, miRNA scaffolds) depend on 3D shape
- Most inverse folding tools focus on secondary structure only
- Need a practical method for short/medium RNAs (< 100 nt)

Goal: design sequences whose **predicted 3D structure** matches a target 3D structure.

BeeRNA formulates RNA inverse folding as a structure-driven optimization problem.

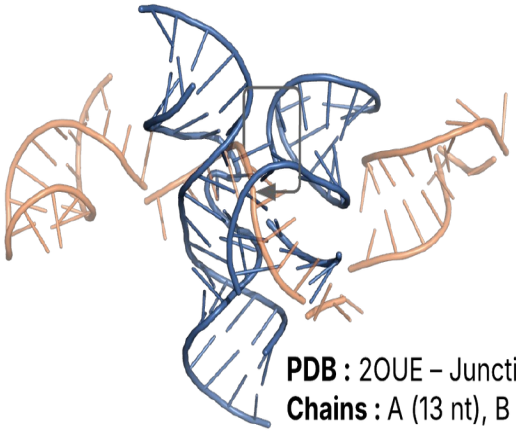
Input

- Target tertiary structure T_{3D} (PDB)
- Corresponding secondary structure Tsec

Objective

$$S^{*} = \arg\min_{S} \mathrm{RMSD}\left(F(S),\ T_{3D}\right)$$

where $F(S)$ is the predicted 3D structure obtained using RhoFold [1].



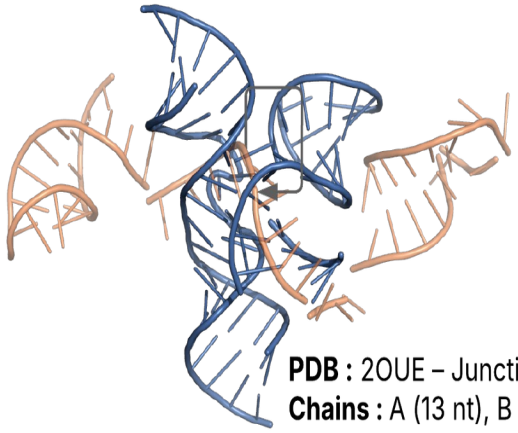
PDB : 2OUE – Junctionless Hairpin Ribozyme
Chains : A (13 nt), B (12 nt), C (17 nt), D (19 nt)
Length : 61

Recovery : 98.4%
RMSD : 19.34Å
GDT_TS : 11.59%

Most prior inverse folding methods [2, 3, 4, 5] evaluate performance using native sequence recovery:

- The percentage of nucleotides that are identical to the native sequence

Why structure-based evaluation matters



PDB : 2OUE – Junctionless Hairpin Ribozyme
Chains : A (13 nt), B (12 nt), C (17 nt), D (19 nt)
Length : 61

Recovery : 98.4%
RMSD : 19.34Å
GDT_TS : 11.59%

Predicted structure (orange) vs. sequence structure (blue) for 2OUE RNA. Despite 98.4% native recovery, RMSD and GDT-TS show a poor structural match.

Limitation:

- High sequence identity (e.g., native sequence recovery) does **not** guarantee accurate tertiary folding.
- Tertiary contacts depend on long-range interactions

BeeRNA instead evaluates candidates by comparing their predicted 3D structure to the target.

BeeRNA uses an Artificial Bee Colony (ABC) metaheuristic [6] to explore the sequence space.

Population

- Candidate RNA sequences (food sources)
- Initialized under base-pairing and GC-content constraints

Search phases

- **Employed bees:** local sequence mutations
- **Onlooker bees:** probabilistic selection of good candidates
- **Scout bees:** replacement of stagnant solutions

Evaluating tertiary structures is computationally expensive.
BeeRNA reduces cost using a two-stage fitness.

Stage 1: secondary structure filter

- Compute Base Pair Distance (BPD) to target T_{sec}
- Reject sequences with $\mathrm{BPD} > 0$

Stage 2: tertiary structure evaluation

- Fold remaining sequences using RhoFold
- Compute RMSD to the target structure

This focuses expensive evaluations on structurally plausible candidates.

Datasets

- RNASolo (short RNAs) [7]
- RFAM-derived 3D set (broad ncRNA lengths) [8, 9]
- 14-structure benchmark (diverse PDBs) [10]

Metrics

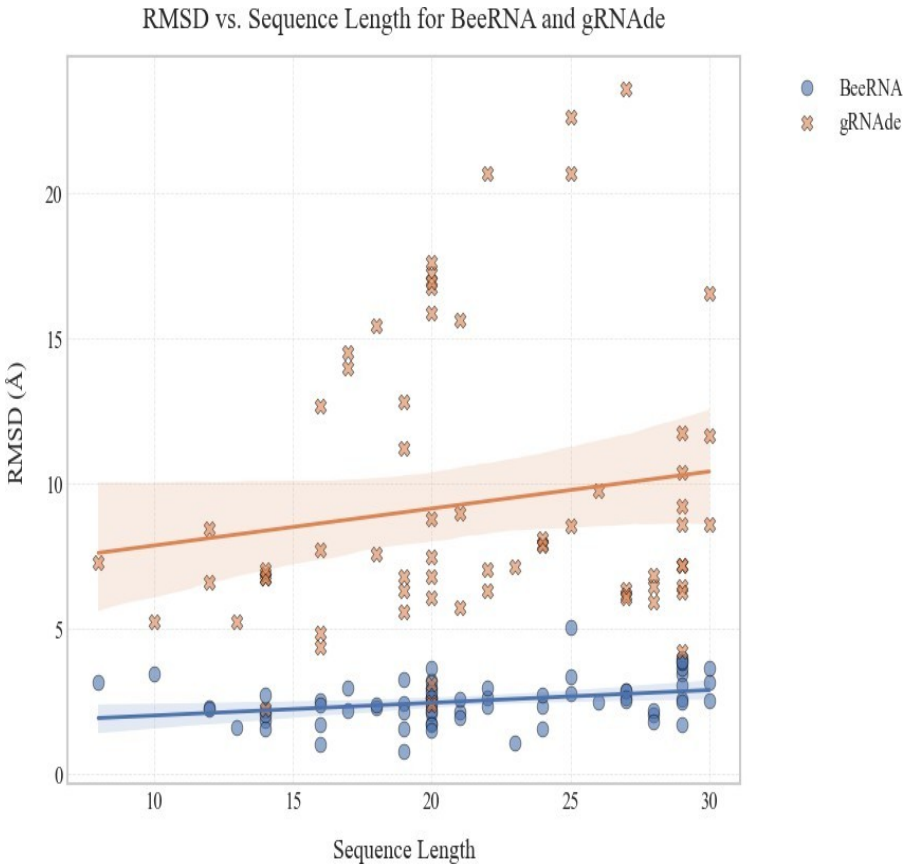
- RMSD (primary)
- GDT-TS (secondary)

RNASolo: BeeRNA vs gRNAde

Average performance

Metric	BeeRNA	gRNAde
RMSD (˚A)	**2.50**	9.33
GDT-TS (%)	**26.91**	18.97

BeeRNA consistently achieves low RMSD on short RNAs.



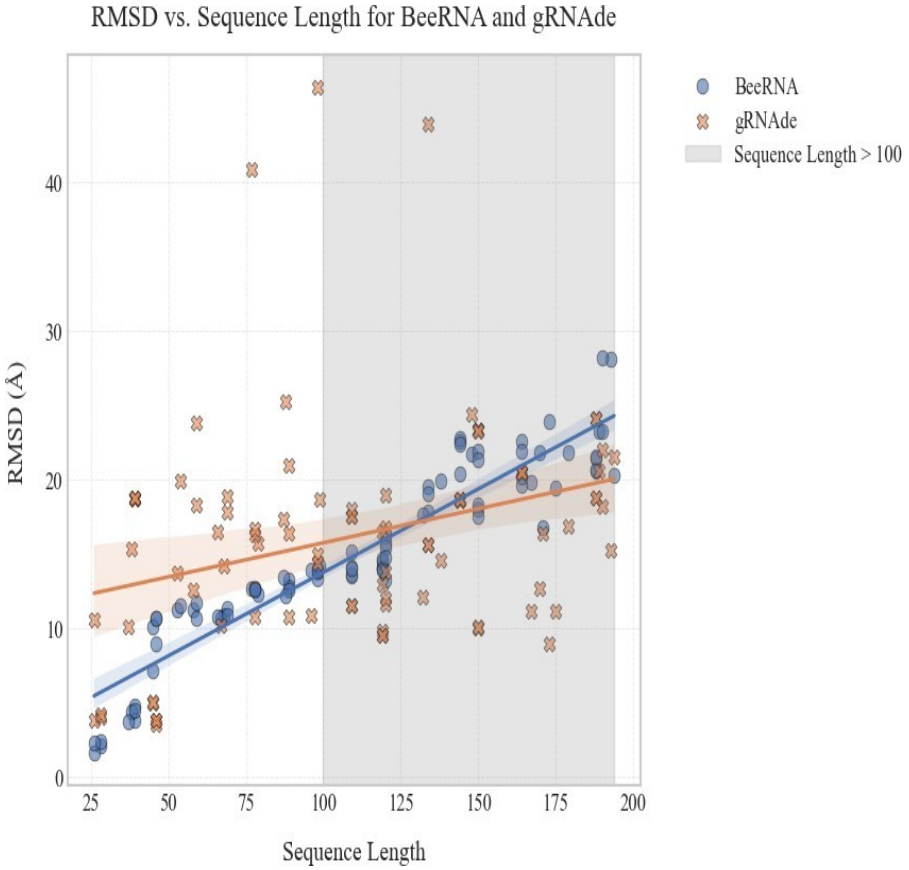
RMSD vs sequence length on RNASolo. BeeRNA remains stable as length increases, while gRNAde degrades.

RFAM: BeeRNA vs gRNAde

Average performance

Metric	BeeRNA	gRNAde
RMSD (˚A)	**14.98**	16.24
GDT-TS (%)	**11.56**	9.77

BeeRNA performs best on short and medium-length RNAs.



RMSD vs sequence length on RFAM. Shaded region highlights sequences longer than 100 nt, where performance degrades for both methods.

14-structure benchmark: BeeRNA vs gRNAde vs RiboDiffusion

PDB	Len	BeeRNA	gRNAde	RiboDiff.
1F27	19	**2.21**	14.94	3.47
1LNT	22	**3.69**	17.51	8.08
354D	23	**10.68**	17.38	12.89
1L2X	27	**3.00**	3.93	3.47
1Q9A	27	**2.65**	6.57	4.35
1CSL	28	**2.93**	3.36	12.25
1ET4	35	**11.34**	13.51	12.18
1XPE	46	20.92	14.26	**10.85**
1X9C	60	**21.00**	25.19	26.79
2OUE	61	**21.00**	22.14	24.71
4FE5	67	11.80	**8.42**	9.78
2GDI	78	**8.00**	15.73	8.49
2GCS	122	24.00	25.71	**1.66**
2R8S	159	26.00	20.20	**5.38**

- BeeRNA achieves the lowest RMSD on 10/14 structures.
- Strong performance on short and medium RNAs (≤100 nt).
- Higher RMSD on long targets is partly due to no initialization reaching BPD = 0, requiring a fixed penalty.

- **BeeRNA** brings training-free metaheuristic search to tertiary RNA inverse folding
- Two-stage evaluation (BPD → RMSD) makes 3D design practical
- Strong results for short/medium RNAs (<100 nt) relevant to therapeutics

Limitations / next

- Scaling to long RNAs (search space + expensive folding calls)
- Future: multi-objective optimization, improved initialization

Conclusion

- Monte Carlo Search has many applications to Chemistry and Biology:
  - Modeling Gene Regulatory Networks
  - Retrosynthesis
  - Drug Discovery
  - RNA Design