Continuous Nested Monte Carlo Search and Nested Rollout Policy Adaptation for Chemical Processes Planning

Lotfi Kobrosly and Tristan Cazenave

LAMSADE, Université Paris Dauphine-PSL, Place du Maréchal de Lattre de Tassigny, Paris, France

1 Introduction

Industrial chemical processes aim to attain a final state of a system that satisfies a set of conditions (concentration of a component, its absorption by the system, its extraction rate, etc.). They require a well-planned sequence of external actions to such objective[5]. Many problems are bound by differential equations[7], rendering the choice of actions and their impact on the states rather complicated. They thus need a simulation procedure for accurate planning. The field of planning in Computer Science has been soaring recently with the advent of learning-based approaches (Reinforcement Learning[4], Monte Carlo methods[3], Transformers[6], etc.) in addition to Constrained Programming[13] or MINLP[8]. In our work, we present two novel algorithms, based on Monte Carlo Search, to take on the planning task in a simulated environment.

2 cNMCS and cNRPA

- Continuous Nested Monte Carlo Tree Search (**cNMCS**): starting from the Nested Monte Carlo Tree Search algorithm [2], we sample uniformly a number of actions and set these as the possible ones, and either play the rollout from there to get the final score or get the instant reward of each action. The rest of the algorithm behaves similarly to the original one.
- Continuous Nested Rollout Policy Adaptation (**cNRPA**): based on Nested Rollout Policy Adaptation [11], the policy takes the current state into consideration and we use two approaches to handle the continuity of the action space. 1- **A gaussian kernel** to choose the action when encountering a new state, according to actions of other visited states. Also, to adapt the policy values of states neighboring the ones visited through the last best sequence. 2- **Subdividing the search space into regions**: when encountering a state, the policy returns the action of the region to which the state belongs. After a specified number of visits to a region, it is subdivided for finer evaluation of the policy. To allow further exploration, when the policy returns an action, we sample around this action using a normal distribution. The standard deviation of this distribution decreases as we go through the rollouts.

3 Experimentation

We test our algorithms on a set of problems available with the `pc-gym`[1] [1] package in `python`: Continuously Stirred Tank Reactor (CSTR)[14], Nonsmooth Control[9], Crystallization of Potassium Sulfate[10], and Multistage Extraction Column (MEC)[7]. We compare them to a *Proximal Policy Optimization* (PPO)[12].

Problem	CSTR	Nonsmooth Control	Crystallization	MEC
Rollout cNMCS level 1	0.0475	3.8012	133.0	3.9241
Reward cNMCS level 1	**0.0026**	**0.0886**	129.4	**0.8599**
Gaussian cNRPA level 1	0.0212	3.7381	128.8	5.3561
By Region cNRPA level 1	0.0409	0.5476	134.5	6.9461
Rollout cNMCS level 2	0.0152	0.6923	130.6	3.3912
Reward cNMCS level 2	0.0055	1.4174	130.3	2.6613
Gaussian cNRPA level 2	0.0089	0.8317	**127.91**	4.0616
By Region cNRPA level 2	0.0907	0.5176	131.8	6.2739
PPO	0.0126	0.2011	135.8	2.7551

Table 1: Cumulative reward obtained for each algorithm.
Level 1: bandwidth=25, $n_{policies} = 300$; Level 2: bandwidth=5, $n_{policies} = 20$

4 Conclusion

The algorithms presented are indeed able to handle the continuity of the actions' and the observations' spaces and provide promising results. The random rollouts algorithms still do not outperform a PPO on all problems due to their evaluation approach which only evaluates the whole trajectory's performance. In contrast, a reinforcement learning approach will also evaluate each action's impact with the obtained reward, making it more suitable to these cases where we have setpoints for each timestamp during the process. This is why the reward-based cNMCTS outperforms the rest (and significantly faster) and confirms this interpretation.

[1] https://maximilianb2.github.io/pc-gym/

References

1. Bloor, M., Neto, J., Sandoval, I., Mowbray, M., Ahmed, A., Mercangoz, M., Tsay, C., Rio-Chanona, A.D.: pc-gym: Reinforcement learning envionments for process control (2024), https://github.com/MaximilianB2/pc-gym
2. Cazenave, T.: Nested monte-carlo search. In: Boutilier, C. (ed.) IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009. pp. 456–461 (2009)
3. Cheimarios, N., To, D., Kokkoris, G., Memos, G., Boudouvis, A.G.: Monte carlo and kinetic monte carlo models for deposition processes: a review of recent works. Frontiers in Physics **9**, 631918 (2021)
4. Devarakonda, V.S., Sun, W., Tang, X., Tian, Y.: Recent advances in reinforcement learning for chemical process control. Processes **13**(6), 1791 (2025)
5. Hahn, G.J., Brandenburg, M.: A sustainable aggregate production planning model for the chemical process industry. Computers & operations research **94**, 154–168 (2018)
6. Huang, X., Liu, W., Chen, X., Wang, X., Wang, H., Lian, D., Wang, Y., Tang, R., Chen, E.: Understanding the planning of llm agents: A survey. arXiv preprint arXiv:2402.02716 (2024)
7. Ingham, J., Dunn, I.J., Heinzle, E., Přenosil, J.E., Snape, J.B.: Modelling of Stagewise Processes, chap. 3, pp. 93–172. John Wiley Sons, Ltd (2007). https://doi.org/https://doi.org/10.1002/9783527614219.ch3, https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527614219.ch3
8. Liberti, L., Sager, S., Wiegele, A.: Series b preface (2021)
9. Lim, H.C.: Classical approach to bang-bang control of linear processes. Industrial & Engineering Chemistry Process Design and Development **8**(3), 334–342 (1969)
10. de Moraes, M.G.F., Lima, F.A.R.D., Lage, P.L.d.C., de Souza, M.B.J., Barreto, A.G.J., Secchi, A.R.: Modeling and predictive control of cooling crystallization of potassium sulfate by dynamic image analysis: Exploring phenomenological and machine learning approaches. Industrial & Engineering Chemistry Research **62**(24), 9515–9532 (2023). https://doi.org/10.1021/acs.iecr.3c00739, https://doi.org/10.1021/acs.iecr.3c00739
11. Rosin, C.D.: Nested rollout policy adaptation for monte carlo tree search. In: Ijcai. vol. 2011, pp. 649–654 (2011)
12. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
13. Timpe, C.: Solving planning and scheduling problems with combined integer and constraint programming. OR spectrum **24**(4), 431–448 (2002)
14. Uppal, A., Ray, W.H., Poore, A.B.: On the dynamic behavior of continuous stirred tank reactors. Chemical Engineering Science **29**(4), 967–985 (1974)