On Monte Carlo Tree Search With Multiple Objectives

Michael Painter

Pembroke College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2024

TODO: list

1. Copy contributions from conformation and write out properly
2. THTS section, as it defines notation. Put macros in text/abreviations.tex
3. Copy DENTS and BTS into thesis and copy into common notation
4. Define the toy problems, D-Chain and the one with the entropy trap
5. Have all outline finsihed

Acknowledgements

TODO: acknowledgements here

Abstract

TODO: abstract here

Contents

List of Figures

List of Tables

List of Notation

Markov Decision Processes

a_t . . . . . TODO

r_t . . . . . TODO

s_t . . . . . TODO

t . . . . . TODO: free variable for current timestep?

Trial Based Heuristic Tree Search

n . . . . . Number of trials run

T . . . . . Computation time limit

List of Abbreviations

Trial Based Heuristic Tree Search

`CNODE` . . . . . The chance node class.

`cnode` . . . . . An instance of `CNODE`, i.e. a chance node instance.

`DNODE` . . . . . The decision node class.

`dnode` . . . . . An intance of `DNODE`, i.e. a decision node instance.

MCTS . . . . . Monte Carlo Tree Search

`mcts_mode` . . . TODO: definition of `mcts_mode`

MENTS . . . . Maximum ENtropy Tree Search

`THTS++` . . . . . TODO: thts++

UCT . . . . . Upper Confidence Bound applied to Trees

1 Introduction

Contents

TODO: chapter structure (i.e. in the introduction section I give some background in the field(s), cover the main contributions of this thesis, etc, etc).

1.1 Overview

TODO: list

- Give some context around MCTS (and talk about exploration and exploitation), and why we might use it
  - Larger scale than tabular methods
  - Can do probability and theory stuff (and some explainability, by looking at stats in the tree the agent used)
  - Can use tree search with neural networks to get some of the above (and use for neural network training as in alpha zero)

- Argument from DENTS paper for exploration > exploitation (in context of planning in a simulator)
- Give high level overview of Multi-Objective RL, and why it can be useful
- Give an idea of how my work fits into MCTS and MORL as a whole
- Discuss research questions/issues with current literature (i.e. introduce some of the ideas from contributions section below)

1.2 Contributions

TODO: I'm not too happy with this section, I think most of the questions could be phrased better, but nothing I'm writing sounds quite right to me at the moment. I also just tried to come up with as many questions as I could with the idea of I don't need to keep them all

TODO: Also inline acronyms used,

Throughout this thesis, we will consider the following questions related to Monte Carlo Tree Search and Multi-Objective Reinforcement Learning:

Q1 - Exploration: When planning in a simulator, how can an algorithm best exploit the time it is given to produce the best recommendation action?

Q2 - Entropy: Entropy is often used as an exploration objective in RL, but can it be used soundly in MCTS?

Q3 - Complexity: MCTS algorithms typically run in $O(nAH)$, but are there algorithms that can improve upon this?

Q4 - Scalability: How can the scalability (with respect to size of environments) of multi-objective MCTS methods be improved?

Q5 - Curse of Dimensionality: To what extent do Multi-Objective MCTS methods suffer from the curse of dimensionality? (basically Multi-Objective complexity)

Q6 - Tree Evaluation: How can we best evaluate a search tree produced by a Monte Carlo Tree Search algorithm? TODO: add a matching contrib

Q7 - Multi-Objective Evaluation: Can we apply methods from the MORL literature to theoretically and empirically evaluate Multi-Objective MCTS?

TODO: some words about how below is the contributions we're making in this thesis and expand these bullets a bit more

- Max Entropy can be misaligned with reward maximisation (**Q2 - Entropy**)
- Boltzmann Search Policies - BTS and DENTS (**Q1 - Exploration**, **Q2 - Entropy**, and eval **Q6 - Tree Evaluation**)
- Use the alias method to make faster algorithms (**Q3 - Complexity**)
- Simple regret (**Q1 - Exploration**)
- Contextual regret introduced in CHMCTS (**Q4 - Scalability**, **Q7 - Multi-Objective Evaluation**)
- Contextual Zooming and CHMCTS (runtimes cover **Q5 - Dimensionality**, results **Q6 - Tree Evaluation**)
- Simplex maps (**Q4 - Scalability**, **Q5 - Dimensionality**)
- Contextual Simple Regret (**Q6 - Tree Evaluation**, **Q7 - Multi-Objective Evaluation**)

TODO: Would like to do the comparing different types of eval, even if not listing it as a research question (compare giving it X seconds per decision and evaluating that policy (SLOW), and comparing policy extracted from the tree)

TODO: can make an argument that the best bound achieved by theory is given by letting temperature go to max. Which is consistent with the exploring bandits results

1.3 Structure of Thesis

TODO: a paragraph with a couple lines to a paragraph about each chapter. This is the high level overview/intro to the thesis paragraph. I.e. this section is "this is the story of my thesis in a page or two"

1.4 Publications

TODO: update final publication when submit

The work covered in this thesis also appears in the following publications:

- Painter, M; Lacerda, B; and Hawes, N. "Convex Hull Monte-Carlo Tree-Search." In *Proceedings of the international conference on automated planning and scheduling. Vol. 30. 2020*, ICAPS, 2020, (see Appendix B).
- Painter, M; Baioumy, M; Hawes, N; and Lacerda, B. "Monte Carlo Tree Search With Boltzmann Exploration." In *Advances in Neural Information Processing Systems, 36, 2023*, NeurIPS, 2023, (see Appendix C).
- Painter, M; Hawes, N; and Lacerda, B. "Simplex Maps for Multi-Objective Monte Carlo Tree Search." In *TODO*, **Under Review at conf_name**, (see Appendix D).

2 Background

Contents

TODO: Introduce that going to introduce notation and give the building blocks this thesis builds off

2.1 Multi-Armed Bandits

TODO: Introduce tree search using multi-armed bandits? UCB and exploring bandits

- Would like to think a bit about some of the bandits work that sample actions (from adversarial I think), because they were similar to boltzmann search but I hadn't seen details about those works when writing dents

2.2 Reinforcement Learning

TODO: list

- MDPs definition
- Basic results and definitions we use
- Talk about entropy and some of that work (probably a subsection)

Definition 2.2.1. *A* Markov Decision Process *is a tuple* $M = (S, A, R, p, H)$*, where* S *is a set of states,* A *is a set of actions,* $R(s, a)$ *is a reward function* $S \times A \to \mathbb{R}$*,* $p(\cdot|s, a)$ *is a next state probability distribution* $S \times A \to S$ *and* $H \in \mathbb{N}$ *is a finite-horizon time bound.*

2.3 Trial-Based Heuristic Tree Search and Monte-Carlo Tree Search

TODO: list

- Give high level overview of MCTS (why use it etc)
- Outline that I'll present this as here is THTS, and then here's the THTS routines for MCTS

In this section we introduce `THTS++` [3], which is an open-source, parallelised extension of the Trial-based Heuristic Tree Search schema [2]. This schema is a generalisation of Monte Carlo Tree Search (MCTS), as presented in Section **??**. In `THTS++` trees consist of *decision nodes* and *chance nodes*. Decision nodes output actions that can be taken by the agent, and chance nodes output *outcomes* that may be random and may depend on the action taken. As such, each decision node has an associated *state* and each chance node has an associated *state-action pair*. In this work, we are considering fully-observable environments, but `THTS++` can be generalised to consider *partially-observable* environments. We give `THTS++` implementations of the standard Upper Confidence Bound applied to Trees (UCT)

algorithm and Maximum ENtropy Tree Search (MENTS) in Sections 2.3.2 and 2.3.3 respectively.

In MCTS we run trials, either for some fixed number of trials, or some timelimit, where each trial is split into four stages: (1) selection, which samples states and actions for the trial, corresponding to a path down the tree; (2) expansion, which creates any new nodes in the tree; (3) initialisation, which initialises values in new nodes in the tree; (4) backup, which updates values at all nodes visited on the trial.

TODO: add MCTS figure here?

```
1 def foo(bar):
2         print("helloworld!")
```

2.3.1 Trial Based Heuristic Tree Search

TODO: list

- Copy DENTS MCTS section presentation, make a notation `node`(s_t) for the node at state s_t
- Present thts++
- Indicate what parts are new versus the original paper (context function, optionally running `mcts_mode` and mutli-threading)
- Small comment about multi-threading and two-phase locking used to avoid deadlock
- TODO: probably not necessary to say - but thought of nice/concise way of explaining it (a node can lock children, not parent, if need info from parent, then it has to put a thread safe copy in the context)
- Define terms precisely and consistently, for example `mcts_mode` (say that notation and terminology varies widely in literature, e.g. does uct run in mcts mode or not?)

In THTS++ we run trials for either some fixed number of trials n, or some time limit T. Each trial consists of three steps: (1) sample a context, which is used to store variables that are associated with a specific trial, and is passed to the following three functions; (2) selection, which samples states, actions and outcomes for the trial, corresponding to a path down the tree; (3) initialisation, which creates any new nodes in the tree and initialises their values; (4) backup, which updates values at all nodes visited on the trial.

Decision nodes follow the interface:

```
1 class DNODE:
2     # children : dictionary[A] -> DNODE
3     def initialise(state (s_t), depth (t), context)
4     def select_action(context)
5     def backup(trial_return (R_t), context)
```

And chance nodes:

```
1 class CNODE:
2     # children : dictionary[S] -> DNODE
3     def initialise(state (s_t), action (a_t), depth (t), context)
4     def sample_outcome(context)
5     def backup(trial_return (R_t), context)
```

The `run_trial`function can be written as:

```
1 def run_trial:
2     # root_node : DNODE
3     # mcts_mode : bool
4     t = 0
5     state = root_node.state
6     while (not selection_phase_ended(t,mcts_mode)):
7
8 def selection_phase_ended(t,mcts_mode):
9     if
```

urgh BRAIN POOP

TODO - copy the descriptions from DENTS, and adapt and add the psuedocode

2.3.2 Monte-Carlo Tree Search

TODO: list

- Give overview of MCTS
- Give UCT in terms of THTS schema
- Define terms precisely and consistently in terms of THTS functions, maybe `mcts_mode` should go here
- Define the value initialisation of THTS using a rollout policy for MCTS
- Talk about the things that are ambiguous from literature (e.g. people will just say UCT, which originally presented doesn't run in `mcts_mode`, but often assumed it does)
- Should talk about multi-armed bandits here?

2.3.3 Maximum Entropy Tree Search

TODO: list

- Define MENTS here

2.4 Multi-Objective Reinforcement Learning

TODO: list

- MOMDP definition
- (Expected) utility
- Define an interface for pareto front and convex hull objects
- Define CHVI
- Should talk about multi-objective and/or contextual multi-armed bandits here?

- I'm planning on aligning this section with the recent MORL survey [1]
- Mention some deep MORL stuff, say that this work (given AlphaZero) is adjacent work

2.5 Multi-Objective Monte Carlo Tree Search

TODO: I think this whole section can just go in litrev

TODO: list

- Define the old methods (using the CH object methods, so clear that not doing direct arithmetic)
- Mention that old method could be written using the arithmetic of CHMCTS (but they don't)
- TODO: write about & make sure its implemented - its because just updating for 1 is more efficient in deterministic, and say that the additions can be implemented as updating for 1 value when determinstic
- Different flavours copy UCT action selection, but with different variants
- Link back to contributions and front load our results showing that all of the old methods don't explore correctly

2.6 Sampling Random Variables

TODO: list

- Talk about the alias method here
- Reference to chapter 4 section where talk about using this with THTS

3 Literature Review

Contents

TODO: currently this is a copy and paste of what I originally wrote for background chapter 2. Deleted parts which are irrelevant for litreview here (and vice versa for the background section).

TODO: I'm also going to use this as a space to paste papers I should write about as they come up while writing later chapters

3.1 Multi-Armed Bandits

TODO: Just UCT/dont lit review?

3.2 Reinforcement Learning

TODO: list

- Talk about entropy and some of that work (probably a subsection)

3.3 Trial-Based Heuristic Tree Search and Monte-Carlo Tree Search

3.3.1 Trial Based Heuristic Tree Search

TODO: THTS paper

3.3.2 Monte-Carlo Tree Search

TODO: list

- Talk about the things that are ambiguous from literature (e.g. people will just say UCT, which originally presented doesn't run in `mcts_mode`, but often assumed it does)
- Should talk about multi-armed bandits here?

3.3.3 Maximum Entropy Tree Search

TODO: MENTS

3.4 Multi-Objective Reinforcement Learning

TODO: list

- Should talk about multi-objective and/or contextual multi-armed bandits here?
- Bunch of the work covered in recent MORL survey [1]
- Mention some deep MORL stuff, say that this work (given AlphaZero) is adjacent work

3.5 Multi-Objective Monte Carlo Tree Search

TODO: I think this whole section can just go in litrev

TODO: list

- Define the old methods (using the CH object methods, so clear that not doing direct arithmetic)
- Mention that old method could be written using the arithmetic of CHMCTS (but they don't)
- TODO: write about & make sure its implemented - its because just updating for 1 is more efficient in deterministic, and say that the additions can be implemented as updating for 1 value when determinstic
- Different flavours copy UCT action selection, but with different variants
- Link back to contributions and front load our results showing that all of the old methods don't explore correctly

3.6 Sampling Random Variables

TODO: Just citing the alias method papers?

4

Monte Carlo Tree Search With Boltzmann Exploration

Contents

4.1 Introduction

TODO: list

- high level overview of DENTS work
- discuss how DENTS answers the research questions from introduction chapter
- state clearly that we're in single objective land here
- Comment about work exploring multi-armed bandits motivating this work

4.2 Boltzmann Search

TODO: list

- Recall MENTS
- Define BTS using THTS functions
- Define DENTS using THTS functions
- Discuss alias method variant (and complexity analysis) in a subsection?

4.3 Toy Environments

TODO: list

- Define D-chain stuff from the paper
- Define the D-chain with entropy trap
- Front load some results still

4.4 Theoretical Results

TODO: list

- add theoretical results

4.5 Empirical Results

TODO: list

- DChain
- GridWorlds
- Go

4.6 Full Results

TODO: there's a lot of figures for the D-chain environment, work out how to best fit them in? Or put them in this seperate section?

5 Convex Hull Monte Carlo Tree Search

Contents

5.1 Introduction

TODO: list

- high level overview of CHMCTS work
- discuss how CHMCTS answers the research questions from introduction chapter
- moving into multi-objective land now
- Comment about CHVI and prior MOMCTS work motivating this

5.2 Contextual Tree Search

TODO: list

- Discuss need for context when doing multi-objective tree Search
  - Use an example env where left gives (1,0) and right gives (0,1), optimal policy picks just left or just right, but hypervolume based methods wont
  - Use previous work on these examples and show they dont do well bad
- Discuss how UCT = running a non-stationary UCB at each node, so given above discussion, there is work in contextual MAB
- Introduce contextual regret here

5.3 Contextual Zooming for Trees

TODO: list

- Give contextual zooming for trees algorithm
- Discussion on the contextual MAB to non-stationary contextual MAB stuff (CZT is to CZ what UCT is to UCB) (and what theory carry over)

5.4 Convex Hull Monte Carlo Tree Search

TODO: list

- Give convex hull monte carlo tree search
- Contextual zooming with the convex hull backups

5.5 Results

TODO: list

- Results from CHMCTS paper
- Get same plots from C++ code, but compare expected utility, rather than the confusing hypervolume ratio stuff

6

Simplex Maps for Multi-Objective Monte Carlo Tree Search

Contents

6.1 Introduction

TODO: list

- high level overview of simplex maps work
- discuss how simplex maps answer the research questions from introduction chapter
- staying in multi-objective land now
- Motivated by CHMCTS being slow

6.2 Simplex Maps

TODO: list

- Define simplex map interface
- Give details on how to efficiently implement the interface with tree structures
- (Good diagram is everything here I think)

6.3 Simplex Maps in Tree Search

TODO: list

- Come up with better title for section
- Use simplex maps interface to create algorithms from the dents work
- Give a high level idea of what δ parameter is (used in theory section)

6.4 Theoretical Results

TODO: list

- Convergence can build ontop of DENTS results
- Runtime bounds (better than $O(2^D)$ which is what using convex hulls has)
- Simplex map has a diameter δ (i.e. the furthest away a new context could be from a point in the map)
- Bounds can then come from that diameter (which is a parameter of the simplex map/algorithm) and DENTS results

6.5 Empirical Results

TODO: list

- Results from MO-Gymnasium
- Compare algorithms using expected utility

7 Conclusion

Contents

TODO: Something about we'll conclude by looking back at contributions and possible future work.

7.1 Summary of Contributions

TODO: go through each of the research questions and contributions, and write about how the work answers the research questions

7.2 Future Work

TODO: outline some avenues of potential future work

Appendices

A

List Of Appendices To Consider

- Multi Armed Bandits, maybe
- MAybe from of the things in background are more appropriate as appendices?

B

Convex Hull Monte Carlo Tree Search - ICAPS2020

TODO: something about paper appearing in ICAPS 2020 & reference to the chapter that covers this content.

TODO: c&p paper

C

Monte Carlo Tree Search with Boltzmann Exploration

TODO: something about paper appearing in NeurIPS 2023 & reference to the chapter that covers this content.

TODO: c&p paper

D

Simplex Maps for Multi-Objecitve Monte Carlo Tree Search

TODO: something about paper submitted to XXX & reference to the chapter that covers this content.

TODO: c&p paper

Bibliography

[1] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

[2] Thomas Keller and Malte Helmert. Trial-based heuristic tree search for finite horizon mdps. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 23, pages 135–143, 2013.

[3] Michael Painter. THTS++,https://github.com/MWPainter/thts-plus-plus.