University of Basel

Implementing and Evaluating Entropy and Simulation based Algorithms for Wordle using Deterministic POMDP Modeling

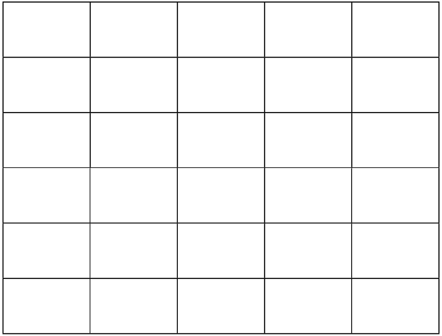
Floyd Peiszan <floyd.peiszan@unibas.ch>

Natural Science Faculty, University of Basel

Artificial Intelligence Research Group

27. January 2025

Wordle Game Example



Example Wordle game: Green indicates correct letters in the correct positions; yellow indicates correct letters in the wrong positions; gray indicates incorrect letters.

Wordle Game Example

M	A	G	I	C

Example Wordle game: Green indicates correct letters in the correct positions; yellow indicates correct letters in the wrong positions; gray indicates incorrect letters.

Wordle Game Example

M	A	G	I	C

Example Wordle game: Green indicates correct letters in the correct positions; yellow indicates correct letters in the wrong positions; gray indicates incorrect letters.

Wordle Game Example

M	A	G	I	C
S	M	O	K	Y

Example Wordle game: Green indicates correct letters in the correct positions; yellow indicates correct letters in the wrong positions; gray indicates incorrect letters.

Wordle Game Example

M	A	G	I	C
S	M	O	K	Y

Example Wordle game: Green indicates correct letters in the correct positions; yellow indicates correct letters in the wrong positions; gray indicates incorrect letters.

Wordle Game Example

M	A	G	I	C
S	M	O	K	Y
B	A	S	I	C

Example Wordle game: Green indicates correct letters in the correct positions; yellow indicates correct letters in the wrong positions; gray indicates incorrect letters.

Wordle Game Example

M	A	G	I	C
S	M	O	K	Y
B	A	S	I	C

Example Wordle game: Green indicates correct letters in the correct positions; yellow indicates correct letters in the wrong positions; gray indicates incorrect letters.

Objectives

- **Implement an algorithm to generate a policy.**
  - A policy refers to the chain of guesses to find the solution.
- **Derive a theoretical model.**
  - Allows to apply algorithms from the literature.
- **Evaluate algorithmic performance.**
  - Compute the average scores.

State Transition System

- **Deterministic Models Are Insufficient.**
  - The hidden word is unknown. $\rightarrow$ The feedback is uncertain.

Wordle as DET-POMDP:

- **State:** All subsets of possible solution words.
- **Action:** Set of valid guess words.
- **Observation Function:** Maps actions to feedbacks, handling the uncertainty.
- **Transition Function:** Filters word list of a state according to action and observation.

State Transition System

- **Deterministic Models Are Insufficient.**
  - The hidden word is unknown. $\rightarrow$ The feedback is uncertain.

Wordle as DET-POMDP:

- **State:** All subsets of possible solution words.
- **Action:** Set of valid guess words.
- **Observation Function:** Maps actions to feedbacks, handling the uncertainty.
- **Transition Function:** Filters word list of a state according to action and observation.

Entropy in Information Theory

Entropy:
In information theory, **entropy** quantifies the amount of uncertainty in a system. It is typically measured in bits using the base-2 logarithm.

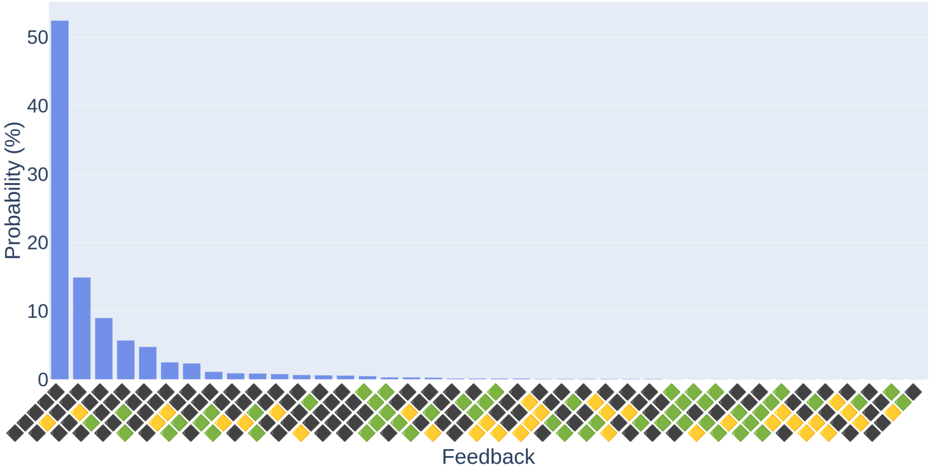
- **Example:**
  - 3 bits of entropy correspond to $2^3 = 8$ equally likely outcomes.
  - Reducing entropy by 1 bit: $2^2 = 4$ outcomes.

Entropy in Information Theory

Entropy:
In information theory, **entropy** quantifies the amount of uncertainty in a system. It is typically measured in bits using the base-2 logarithm.

- **Example:**
  - 3 bits of entropy correspond to $2^3 = 8$ equally likely outcomes.
  - Reducing entropy by 1 bit: $2^2 = 4$ outcomes.

Feedback distribution for ‘FLUFF’



Maximum Information Gain Heuristic

- **Guess Assessment:**
  - A guess can be assessed by its **average entropy reduction**.
- **Maximize** entropy reduction to narrow down the solution space.
- **Entropy Formula:** (Shannon 1948)

$$H(a) = \sum_{o \in O} P(o) \log_2 \frac{1}{P(o)}$$

  - a: The guess word.
  - O: Set of all observations (distinct feedback patterns).

Maximum Information Gain Heuristic

- **Guess Assessment:**
  - A guess can be assessed by its **average entropy reduction**.
- **Maximize** entropy reduction to narrow down the solution space.
- **Entropy Formula:** (Shannon 1948)

$$H(a) = \sum_{o \in O} P(o) \log_2 \frac{1}{P(o)}$$

  - a: The guess word.
  - O: Set of all observations (distinct feedback patterns).

Tie Breaking

Scenario: The **average entropy reduction** does not lead to a decision.

Version A

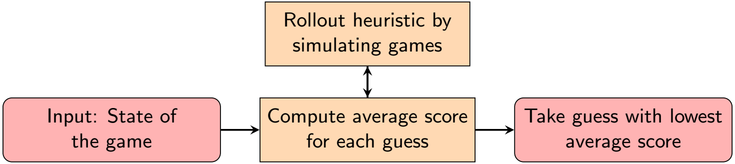
Use **alphabetical** word order.

Version B

Prefer **possible solution** words, before reverting to **alphabetical** word order.

Rollout Algorithm (Bhambri et al. 2023)

- **Idea:** Improve heuristic decision-making by simulating multiple scenarios to optimize the next guess.



- **Experiments:** Only consider the n most promising guesses based on heuristic evaluation.

Upper Confidence Bound for Trees (UCT) Algorithm

- **Key Motivation:**
  - Previous approaches have no guarantee about the quality of the solution.
- **Search Tree:**
  - Iteratively build a tree by simulating the game (rollouts).
  - Tree allows to use knowledge from previous rollouts.
- **Optimality:**
  - **UCT formula** balances exploration and exploitation.
  - Converges to optimal policy with infinite rollouts.

UCT Formula

- Evaluate for each successor node.
- Maximum UCT value decides the next successor node.

$$UCT = B\sqrt{\frac{\log \text{VisitsCurrentNode}}{\text{VisitsSuccessorNode}}} - \text{CostSuccessorNode (Eyerich et al. 2010)}$$

- **Exploration term:**
  - $\sqrt{\frac{\log(\text{VisitsCurrentNode})}{\text{VisitsSuccessorNode}}}$
    Root: Encourages less frequently visited nodes.
    Log: Reduces exploration as visits increase.

- **Exploitation:**
  - CostSuccessorNode
    Costs: Favors lower costs.

UCT Formula

- Evaluate for each successor node.
- Maximum UCT value decides the next successor node.

$$UCT = B\sqrt{\frac{\log \text{VisitsCurrentNode}}{\text{VisitsSuccessorNode}}} - \text{CostSuccessorNode (Eyerich et al. 2010)}$$

- **Exploration term:**
  - $\sqrt{\frac{\log(\text{VisitsCurrentNode})}{\text{VisitsSuccessorNode}}}$
    Root: Encourages less frequently visited nodes.
    Log: Reduces exploration as visits increase.
- **Exploitation:**
  - CostSuccessorNode
    Costs: Favors lower costs.

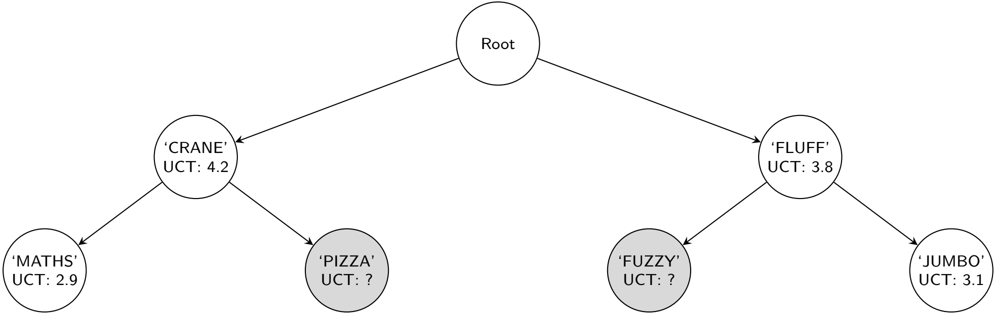
UCT Formula

- Evaluate for each successor node.
- Maximum UCT value decides the next successor node.

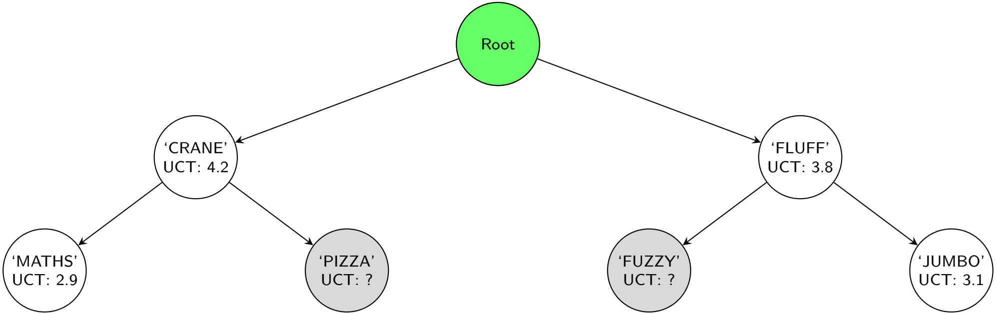
$$UCT = B\sqrt{\frac{\log \text{VisitsCurrentNode}}{\text{VisitsSuccessorNode}}} - \text{CostSuccessorNode (Eyerich et al. 2010)}$$

- **Exploration term:**
  - $\sqrt{\frac{\log(\text{VisitsCurrentNode})}{\text{VisitsSuccessorNode}}}$
    Root: Encourages less frequently visited nodes.
    Log: Reduces exploration as visits increase.
- **Exploitation:**
  - CostSuccessorNode
    Costs: Favors lower costs.

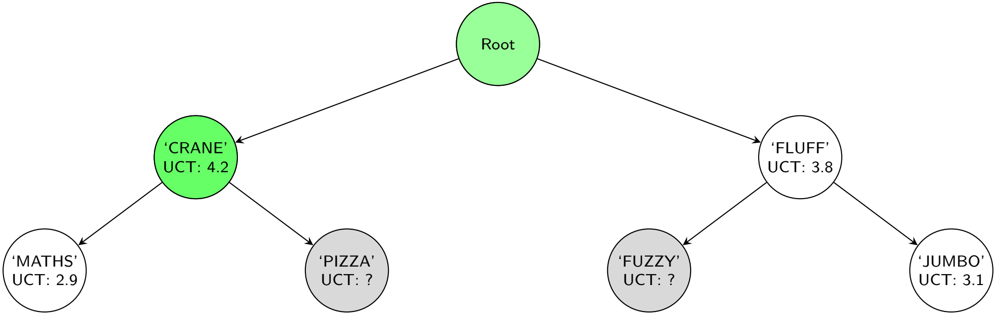
UCT Tree Search Example - Rollout: Tree Traversal



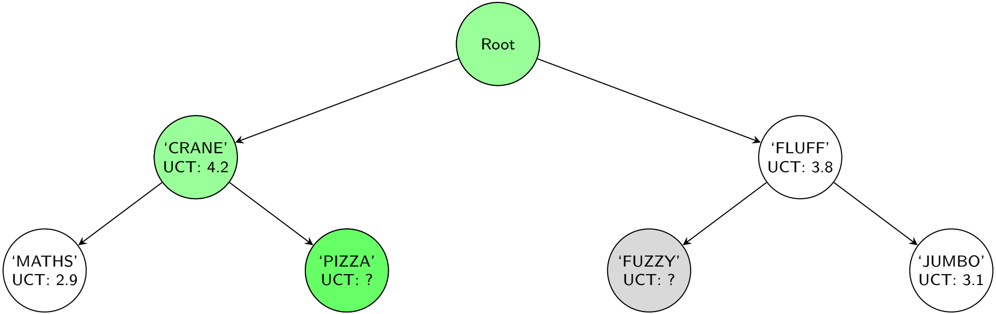
UCT Tree Search Example - Rollout: Tree Traversal



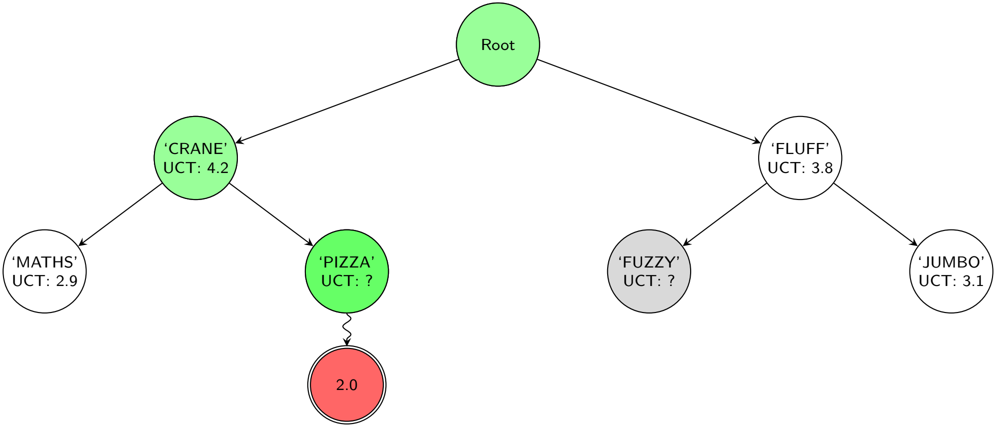
UCT Tree Search Example - Rollout: Tree Traversal



UCT Tree Search Example - Rollout: Tree Traversal



UCT Tree Search Example - Rollout: Tree Traversal

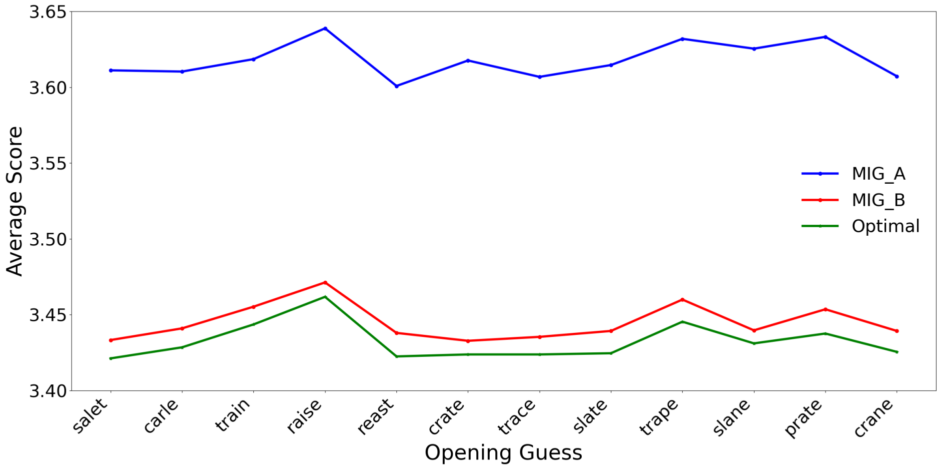


Experiment Overview

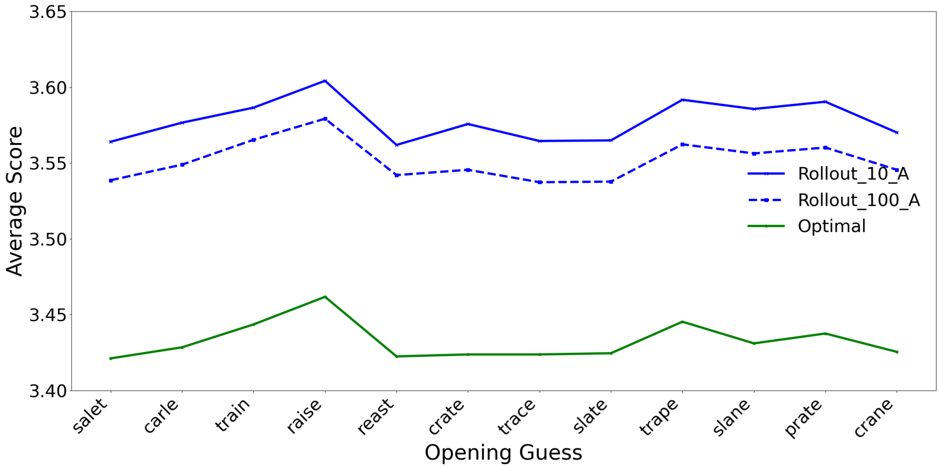
- Measure **accuracy** of the algorithms:
  - Average score across all possible solutions with fixed opening guess word.
  - Use multiple opening words that performed well in the literature.
- Controlled **Hardware**:
  - All experiments were conducted at the sciCORE[1] scientific computing center at the University of Basel.

[1] Available at: http://scicore.unibas.ch/

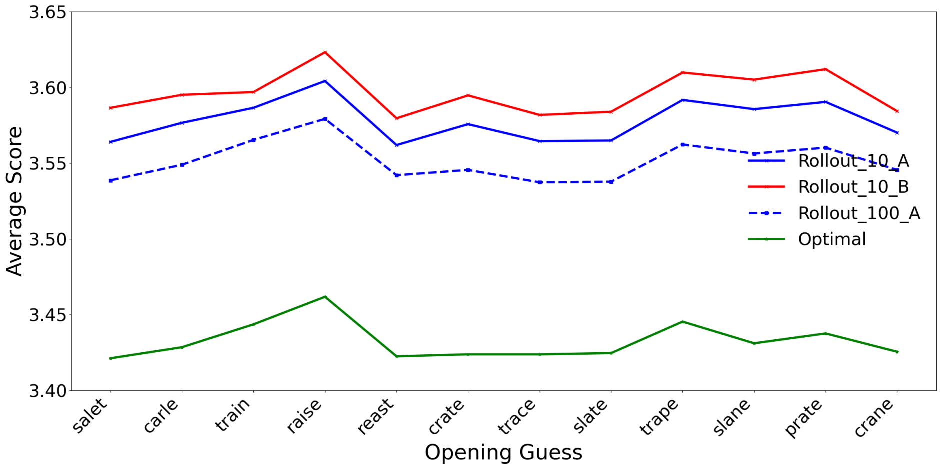
Maximum Information Gain Heuristic Scores



Rollout Algorithm Scores



Rollout Algorithm Scores



UCT Algorithm Scores

Average scores for the UCT algorithm with ‘SALET’ as opening guess.

Number of rollouts	UCT_A	UCT_B
50,000	3.6605	3.6566
100,000	3.6605	3.6566

Optimal Score:
The **optimal average score** achievable is **3.421** (Bertsimas and Paskov 2024).

Conclusion

- **Heuristic Approach:**
  - Maximum Information Gain heuristic performs exceptionally well.
  - Scores with 'SALET': $MIG\_A : 3.61$, $MIG\_B : 3.43$
- **Rollout Approach:**
  - Use a base heuristic to achieve consistent results.
  - Scores with 'SALET': Between 3.54 and 3.56
- **UCT Approach:**
  - Theoretically guarantees optimal solutions through balanced exploration and exploitation.
  - Scores with 'SALET': Around 3.66

Bibliography (1)

- Bertsimas, Dimitris and Alex Paskov (2024). "An Exact Solution to Wordle". In: **Operations Research** 0.0. DOI: 10.1287/opre.2022.0434. URL: https://doi.org/10.1287/opre.2022.0434.
- Bhambri, Siddhant et al. (2023). "Playing Wordle Using an Online Rollout Algorithm for Deterministic POMDPs". In: **IEEE Conference on Games, CoG 2023, Boston, MA, USA, August 21-24, 2023**. IEEE, pp. 1–4. DOI: 10.1109/COG57401.2023.10333228. URL: https://doi.org/10.1109/CoG57401.2023.10333228.
- Bonet, Blai (2009). "Deterministic POMDPs Revisited". In: **Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)**. Arlington, Virginia, USA: AUAI Press, pp. 59–66.
- Eyerich, Patrick et al. (2010). "High-Quality Policies for the Canadian Traveler's Problem". In: **Proceedings of the AAAI Conference on Artificial Intelligence**. Vol. 24. 1, pp. 51–58.
- Kocsis, Levente and Csaba Szepesvári (2006). "Bandit Based Monte-Carlo Planning". In: **European conference on machine learning**. Springer, pp. 282–293.
- Littman, Michael Lederman (1996). **Algorithms for sequential decision-making**. Brown University.

Bibliography (2)

- Sanderson, Grant (2022). **Solving Wordle using information theory**. URL: https://www.youtube.com/watch?v=v68zYyaEmEA [Accessed: 24.11.2024]. Youtube.
- Shannon, Claude E. (July 1948). "A Mathematical Theory of Communication". In: **Bell System Technical Journal** 27.3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x. URL: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Scores Table MIG and Rollout

Opening Guess	MIG_A	MIG_B	Rollout_10_A	Rollout_10_B	Rollout_100_A	Optimal
salet	3.6112	**3.4333**	3.5641	3.5866	*3.5387*	*3.4212*
carle	3.6104	**3.4410**	3.5767	3.5952	3.5490	3.4285
train	3.6186	**3.4553**	3.5866	3.5970	3.5654	3.4436
raise	3.6389	**3.4713**	3.6043	3.6233	3.5793	3.4618
clout	3.6976	**3.5235**	3.6613	3.6773	3.6354	-
soare	3.6380	**3.4648**	3.6013	3.6242	3.5784	-
salut	3.6644	**3.4898**	3.6324	3.6475	3.5983	-
lants	3.6803	**3.5343**	3.6803	3.6976	3.6549	-
reast	*3.6009*	**3.4380**	*3.5620*	*3.5797*	3.5421	3.4225
crate	3.6177	**3.4328**	3.5758	3.5948	3.5456	3.4238
trace	3.6069	**3.4354**	3.5646	3.5819	3.5374	3.4238
slate	3.6147	**3.4393**	3.5650	3.5840	3.5378	3.4246
trape	3.6320	**3.4600**	3.5918	3.6099	3.5624	3.4454
slane	3.6255	**3.4397**	3.5857	3.6052	3.5564	3.4311
prate	3.6333	**3.4536**	3.5905	3.6121	3.5603	3.4376
crane	3.6073	**3.4393**	3.5702	3.5844	3.5456	3.4255

Runtime Table MIG and Rollout

Opening Guess	MIG_A (ms)	MIG_B (ms)	Rollout_10_A (ms)	Rollout_10_B (ms)	Rollout_100_A (ms)
salet	5.84	5.74	1507.53	1183.66	14486.84
carle	6.32	7.20	1244.55	1529.84	16290.10
train	6.61	6.27	1838.25	1485.88	15690.97
raise	5.85	6.73	990.71	1219.78	13128.42
clout	8.44	7.60	3763.18	2989.56	30749.71
soare	6.37	6.59	1223.43	1275.22	13870.86
salut	7.66	6.47	2676.67	2073.36	22281.57
lants	6.92	8.35	3118.62	3243.33	34920.17
reast	6.53	6.56	1468.83	1496.09	16096.90
crate	6.92	6.18	1548.09	1238.82	13077.92
trace	7.25	6.96	1575.95	1574.52	16676.68
slate	5.93	5.66	1461.38	1182.59	12624.15
trape	7.02	7.26	1408.99	1776.83	18915.19
slane	6.69	5.90	1533.87	1250.32	13196.11
prate	7.00	7.19	1385.20	1742.94	18664.90
crane	7.10	6.22	1659.33	1326.16	14139.39

UCT Scores and Runtimes

Average Scores with the opening guess 'SALET'

Number of rollouts	UCT_A	UCT_B	UCT_A_O	UCT_B_O
50.000	3.6605	3.6566	3.4937	3.4907
100.000	3.6605	3.6566	-	-

Average Runtimes with the opening guess 'SALET'

Number of rollouts	UCT_A (ms)	UCT_B (ms)	UCT_A_O (ms)	UCT_B_O (ms)
50.000	19819.81	20545.45	22094.34	22055.80
100.000	38858.05	43131.73	-	-

Formal Definition of Wordle

Definition

The Wordle game can be described as:

- **State:** A state $s \in S$ is the set of solution words consistent with feedback from previous guesses. At s_i, $g \in s_i$ is a possible solution.
- **Action:** An action $a \in A$ is a valid guess word. Applying a_{s_i} in s_i transitions the game to s_{i+1}.
- **Observation:** Feedback $o_i \in O$ for action a_{s_i} is:

$$O = \{(f_1, f_2, f_3, f_4, f_5) \mid f_x \in \{g, y, b\}\},$$

  where g: green, y: yellow, b: grey.
- **Initial State:** $b_0 = s_0$, the set of all possible solution words.
- **Goal State:** A state $s_i \in S$ with one word, the last action a_{i-1}.
- **Transition Function:** $f(s_i, a_{s_i})$ removes words inconsistent with the feedback from a_{s_i}.
- **Observation Function:** $o(s_i, a_{s_i})$ generates feedback for a_{s_i} based on the true solution word.
- **Costs:** The cost is $c(s_i, a_{s_i}) = 1$ for each action $a_{s_i} \in A$.