Improved Heuristics for Sushi Go and Applicability of Probabilistic Scoring Functions.

Cuervo, Ricardo (210920932, r.cuervo@se21.qmul.ac.uk), Deuskar, Rohan (220217275, ec22054@qmul.ac.uk), Thanga Jawahar, Reshi Krish (220532499, ec22185@qmul.ac.uk)

Abstract. Sushi Go is a multi-player game of diminishing incomplete information, with simultaneous playing. As such, it has certain characteristics that make it stand apart from mainstream AI research. However, not much research has been conducted on AI agents, game search methods and tailored heuristics to tackle game complexities. In this paper we propose alternative approaches to improved heuristics. However, since the implementation is constrained by the use of Sushi Go on TAG and by computing constraints, we limit the scope of experimentation to finetuning heuristics within aggressive compute budget limits. Specifically, we analyse algebraic formulations for 3 heuristic models. Furthermore, we lay theoretical ground for further assessment of a heuristic based on Dynamic Probabilistic Scoring, which although computationally more complex, it has the potential to address the shortcomings of the standard heuristic on TAG and of other simpler algebraic scoring methods.

1. Introduction

Sushi Go provides a framework to study intelligent agents, specifically in cases of simultaneous action selection under evolving conditions of information availability. The game is of partial information during the first turns of each round, when a player does not see other players' cards. The level of stochasticity diminishes as game progresses, becoming deterministic. We define a **Game State** based on two components: the cards on the table at any point in time and the cumulative points from previous rounds. For example, a game state in round 2/3, turn 2/8 is illustrated in table 1.

	cards on table	cumulative points
player 1	Tempura, Sashimi	10
player 2	Tempura, Dumpling	8
player 3	Wasabi, Squid Nigiri	11
player 4	2Maki, 3Maki	7

Table 1. Game State Example

A **Game Action** is defined as each player selecting one card and placing it on the table. Players turn their respective cards visible at the same time – which means that an agent's choice of card is made independently from other players' cards in the same turn.

The player with highest score wins. However, the scoring methodology is relatively complex and highly conditional on probabilities of getting combos within each round and in the case of Pudding cards, across rounds. Different game strategies may focus on maximizing specific combo points (e.g., a player may want to maximize Maki icons over securing a Sashimi triplet). Thus, an action by an intelligent agent must simultaneously consider other cards on the table, past rounds (and cumulative points), probabilities of getting the expected combo cards and strategies to maximize scores.

We estimate that the game search is very wide with a depth of 21 (8 turns in each one of 3 rounds, but 8th card is determined by previous action). For the first round in a 4-player game, there are 32 cards played among set of 108 (2.6699E+27 combinations). In the second round, 32 cards are played among remaining 76 (2.6956E+21 combinations). In last round, 32 cards are played among remaining 44 (2.1091E+10 combinations). Thus, the entire game space has 1.518E+59 options. Because of the size of the search tree, we consider an MCTS agent and propose different heuristics to optimize search.

This paper presents an introductory discussion on the game and previous research on agents and search methods for multi-player games with sequential playing, followed by a review of proposed heuristics that build on current MCTS heuristic in TAG. We discuss a heuristic with significant performance improvement potential based on dynamic calculation of probabilities but argue that its complexity and runtime exceed project budget constraints. We experiment on 3 computationally simpler heuristics but fail to outperform current MCTS TAG implementation.

2. TAG

The Sushi Go implementation on TAG was first proposed by Embring et. al. [1] One of the shortcomings of this implementation is related to the heuristic used by agents, which "simply calculates the current score accumulated by the player with no consideration to possible combos or future rewards". Thus, a good agent using this heuristic would have to explore a large amount of tree branches before making a decision, to ensure that the probabilities associated with combos in future turns are adequately captured. Under conditions of limited budget, such extended search is not possible.

The search can also be optimized by pruning the tree. Saffidine et. al. [2] proposed Simultaneous Move Alpha-Beta (SMAB) pruning for games where players act simultaneously, rather than sequentially, and tested it on a Goofspiel game implementation. According to authors, results showed a "considerable drop in node expansions, even though not nearly as much as with

Alpha-Beta in the sequential setting, but certainly enough to be very promising".

Perez and Oommen proposed Multi-Minimax [3], a new algorithm for multi-player games without a fixed sequential play ordering and tested it on Snake Game. Multi-Minimax performed better compared with other similar AI strategies including Max-n algorithm, Paranoid algorithm, and Best-Reply Search (BRS). However, for the purpose of this research, we are constrained by the Sushi-Go implementation on TAG.

3. Background

Valuation methods are used in games such as chess to provide a rough idea of the state of the game and are used in a heuristic function to value the impact of an action. The best-known system assigns 1 point to a pawn, 3 to a knight or bishop, 5 to a rook and 9 to a queen. However, they fail to capture considerations regarding specific game states where the value of a piece can differ considerably from the standard valuation. A well posted knight can be more valuable than a passive rook.

Likewise, we considered point valuation systems for Sushi Go that can support more accurate heuristics to determine best actions (cards to be played) at any given point in time, avoiding full-depth MCTS rollouts. This is intended to overcome the limitation of the standard Sushi Go TAG implementation which uses current score accumulated by the agent with no consideration to possible combos or future rewards. The standard heuristic in the TAG implementation is so basic that OSLA agents beat MCTS agents (39% vs. 21% of wins), probably due to "lack in advanced heuristics". Furthermore, authors state need for "well-tuned heuristic, that could take into consideration different possible combos, long-term rewards such as puddings, or the other decks and other player's cards"

For our implementation, we opted for an MCTS agent based on the standard MCTS agent on TAG, and explored alternative heuristics based on improved scoring methodologies.

4. Method

3 types of scoring methods are considered in designing the heuristic function, ranked in increasing order of computational complexity: Empirical Scoring, Algebraic Scoring and Dynamic Probabilistic Scoring.

Empirical Scoring: Contrary to chess, Sushi-Go is a poorly studied game. Some anecdotal information can be found over the internet from bloggers and game fans, providing card scoring methodologies. [include reference]. Agreeordie.com proposes a static point value system [4] where a squid nigiri card gets dual values of 3 / 4.5, depending on a wasabi card. Similarly, other cards are valued based on max. potential score and number of cards needed. The problem is that such methodology does not consider specific game states. A Wasabi card on hand is worth nothing if all nigiri cards have been previously played.

The main advantage of Empirical Scoring is its simplicity to implement, and low computational load for games where computing budget is scarce. However, it does not address shortcomings from current TAG heuristic implementation.

Algebraic Scoring. Scoring formulas that consider the game state. Specifically, we propose the following heuristics:

Basic heuristic: *playerScore/maxScore*.
This heuristic compares the player score with the maximum score present in the game to see if agent is winning in each game state.

Zoomed heuristic :
(playerScore-minScore)/(maxScore-minScore)
This heuristic takes the basic heuristic and focuses more on the range of values which have impact on the game. I.e. the relevant score range where every player lies. Thereby giving us a good reference of how well the agent is performing in comparison with the other agents.

Potential heuristic: This heuristic adds a potential value to the score before passing it on to the agent, by taking reference of how many of the potential based cards are in the players field and in the other players hands. It is intended to capture potential higher scores from combos, as follows:
Tempura : If there is an ungrouped tempura in the player field and there are tempuras available in player hands, a potential value of 2 is added.
Sashimi: If there are ungrouped sashimi in the player field and there are tempuras available in player hands, a potential value of 2 (if 1 ungrouped sashimi) or 4 (if 2 ungrouped sashimi) is added.
Maki and Pudding: The end result addition of score is verified at every game state.

Like Empirical Scoring, Algebraic Scoring is simple to calculate and implement. However, performance is compromised in certain game states where the potential value of an action deviates more profoundly from standard valuations.

Dynamic Probabilistic Scoring (DPS): We propose a scoring methodology that assesses the value of an action (score of a card that can be played) based on count of cards already played and probabilities that a particular card may be played, which are conditional on other cards played before. This approach captures the dynamic nature of the game.

To further illustrate this point, consider the game presented in table 2, with a scenario in which no Squid Nigiri cards are played in round 1. This makes a future

Wasabi card more valuable as the probability of a Squid Nigiri card in round 2 is higher. The Wasabi card would yield additional 6 points. The probabilistic scoring values it at 5.64. Similarly, fractions of sashimi and tempura cards played in round 1 drive variations of card values going into round 2.

	# cards Round 1 Shuffle	Points per Card	Cards played Round 1	# cards Round 2 Shuffle	Points per Card
Tempura	14	1.390	6	8	1.760
Sashimi	14	2.780	4	10	2.713
Dumpling	14	1.7	0	14	2.8
Wasabi	6	3.998	2	4	5.647
Squid Nig.	5	3	0	5	3
Salmon Ni	10	2	10	0	2
Egg Nigiri	5	1	5	0	1
2Maki	12	0.949	0	12	0.612
3Maki	8	1.424	0	8	0.918
1Maki	6	0.475	0	6	0.306
Pudding	10	0	5	5	0
Chopstick	4	1.648	0	4	1.866
Total	108		32	76	

Table 2. Value Points per Card at Shuffle

The main drawback of the proposed DPS is that it is very complex to implement and computationally heavy because:

- **PDF – Probability Distribution Functions**: It involves dynamically calculating the probability distribution of each card at any point in time (i.e. at each game state) to determine higher-probability scenarios, which is a dynamic variable driven by cards previously played. For example, the value of a
- Tempura card is conditional on the probability of getting 2 tempura cards in same round, which is a function of the number of Tempura cards available in the shuffle (108 cards in round 1, 76 cards in round 2 of a 4-player game). Likewise, the value of a Sashimi card is conditional on the probability of getting 3 cards in same round.
- **Conditional Probabilities**: Final scoring is conditional on other cards. For example, if no wasabi card is played in round 1, the value of a wasabi card is higher in second round. But if all nigiri cards are played in first round, the value of the wasabi card in second round is zero.

We speculate that DPS can have two significant benefits: (1) more effective pruning – by dynamically assessing probabilities of cards and values in the stochastic turns of each round, it can prune actions (cards) of lower value. (2) more accurately calculating future rewards during a Monte Carlo rollout, limiting need for a deep search while providing accurate value estimates. For example, a Monte Carlo search can be limited to depths equal to remaining turns in current round and use DPS scores to probabilistically estimate potential reward from future rounds. However, the dynamic calculation of conditional probabilities based on remaining cards on shuffle makes the DPS coding very complex. Furthermore, we also believe that the allocated time for playing in the proposed tournament would not be sufficient to guarantee that it DPS can be adequately explored in determining best action (card to play). Thus, we propose analysis of DPS to be considered in future work and rather focus current project on Algebraic Scoring and Heuristics listed above.

5. Experimental Study

We conducted experimentation on two levels: 'Knowing the Game' and "Assessing Agent Performance'.

'Knowing the Game' Experimentation

We conducted two 5000 game tournaments, one with 4 Random Agents and the other with 4 standard MCTS agents. The objectives were two-fold: (1) assess early strategies that have a higher likelihood to win, and (2) learn range of max-scores for different games. The rationale for the former is anchored on games such as tic tac toe or chess, where not all initial actions are equally likely to yield a win. The rationale for the latter is based on fact that Sushi Go is not a zero-sum game. Certain cards have a score-multiplying effect and thus, how such cards are played will impact total scores.

Table 3 lists the # of games where winner played a card first on any given round using standard MCTS and random agents, as per standard TAG implementation).

	Random Test		MCTS Test		Total Cards	
	A	% of games	A	% of games	# of cards	% of cards
Tempura	1779	11.9%	2096	14.0%	14	13.0%
Sashimi	1814	12.1%	137	0.9%	14	13.0%
Dumpling	1742	11.6%	2157	14.4%	14	13.0%
Wasabi	871	5.8%	1641	10.9%	6	5.6%
SquidNigiri	832	5.5%	3669	24.5%	5	4.6%
SalmonNigiri	1413	9.4%	4753	31.7%	10	9.3%
EggNigiri	756	5.0%	415	2.8%	5	4.6%
Maki_2	1152	7.7%	15	0.1%	12	11.1%
Maki_3	1698	11.3%	27	0.2%	8	7.4%
Maki_1	882	5.9%	24	0.2%	6	5.6%
Pudding	1412	9.4%	33	0.2%	10	9.3%
Chopsticks	649	4.3%	33	0.2%	4	3.7%

Table 3. Winner 1st card Frequency.
A: # of games where winner played card first on any round

Standard MCTS players do not play Maki cards first. While Maki cards represent 24.1% of the deck of cards, MCTS winner agents only played them first in 0.5% of the cases. We argue that this is significantly driven by suboptimal current heuristic which only considers accumulated points. Maki points are only awarded at the end of each round. Thus, they would be valued zero points during the round. Conversely, Wasabi and Nigiri cards are played first more often, with 10.9 % of winning games playing wasabi first, and 24.5% / 31.7% for Nigiri (Squid / Salmon) played first.

It is important to notice that in 945 of the 5000 MCTS games, an agent played a wasabi card without a nigiri card following, meaning that the extra points were not captured by the agent playing the wasabi card, resulting in a game loss for such agent.

'Assessing Agent Performance' Experimentation

We run 3 1200-game tournaments to test each one of the proposed Algebraic Scoring Heuristics against basic MCTS agents using standard heuristic on TAG, in 4-player games, Results are listed in table 4.

	4-Player Games - % wins on 1200 Game Tournament			
Player 1	Player 1	Basic MCTS	Basic MCTS	Basic MCTS
Basic Heuristic MCTS	20.0%	28.3%	26.1%	25.6%
Zoom Heuristic MCTS	21.2%	26.1%	24.1%	28.7%
Potential Heuristic MCTS	20.1%	24.6%	27.6%	27.7%

Table 4. Proposed MCTS Agent vs. Standard MCTS Agents – Game Performance.

The proposed heuristics performed worse than the default heuristic. One possible reason is that while the proposed heuristics provide a good reference of the agent's position in reference to opponent positions, in the case that player 1 is the player with max points there is no guidance on which action will be good as the heuristic will return 1.

6. Discussion

The proposed heuristics fail to capture the potential value of future cards and are outperformed by the standard MCTS configuration in TAG proposed by [1]. We analysed the impact that an MCTS agent can have in the game. Table 5 presents quartile analyses of winner points to measure points impact from intelligent agents. Specifically, it compares winning score distribution on 5000 4-player games. While the max. score seems to hover around 60 points, MCTS agents MCTS agents consistently get higher scores (additional 5 to 6 points). So, we conclude that the value of earlier cards (under incomplete information) is a significant factor on total score and thus, likelihood to win. This lends more weight to the argument in favour of a forward-looking heuristic that captures dynamicity and conditional probabilistic nature of card values.

	Random Players	MCTS players
min	22	26
25%	32	37
50%	35	40
75%	39	43
max	61	60

Table 5. Win Score Distribution.

We also looked at the relative importance of winning first round as indicator of game winner. In the test with 5000 games (4 MCTS players), winner of first round was also game winner in 47% of the games. In 23% of the games, the score difference between round winner and game winner was 3 points or less. This also supports case for a strong heuristic in first round, especially in early turns when game is more stochastic.

7. Conclusions and Future Work

Based on analyses previously described, we conclude that a strong heuristic based on Dynamic Probabilistic Scoring has the potential to deliver higher performance. However, this is not a simple task as the card probabilities change. Figure 1 illustrates the probability function of drawing X Sashimi cards (up to 14) in a game as a function of cards previously dealt (0 cards corresponds to first round, 5 cards represent a scenario for round 1, 10 cards represent a scenario for round 2).

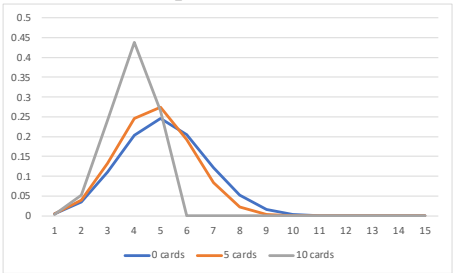

Figure 1. Probability Function, Sashimi Cards.

Future work should tackle DPS implementation issues and applicability of it on:

- Improved heuristics
- Best action ranking
- Opponent Modelling
- Tree Pruning

However, proposed DPS also has its limitations beyond computational considerations. Its value is highest during the first 3 stochastic turns of each round. After turn 4, the game round becomes deterministic and thus, DPS should be explored in conjunction with multi-player minimax methods to model game outcomes based on remaining round cards and opponents' games.

We argue that this implementation would provide a more levelled playing field against other methodologies such as Reinforcement Learning, as proposed by [5].

8. References

[1] Carl-Magnus Embring Klang , Victor Enhörning, Alberto Alvarez, Jose Font, 2021, Assessing Simultaneous Action Selection and Complete Information in TAG with Sushi Go!

[2] Abdallah Saffidine, Hilmar Finnsson, Michael Buro, 2012, Alpha-Beta Pruning for Games with Simultaneous Moves

[3] Nicolas Perez and B. John Oommen, 2019, Multi-Minimax: A New AI Paradigm for Simultaneously-Played Multi-player Games.

[4] www.agreeordie.com

[5]Alexander Soen, 2019, Making Tasty Sushi using Reinforcement Learning and Genetic Algorithms