A bounds-driven analysis of “Skull and Roses” cards game

Gragera Aguaza, Alonso[1,a)] Baffier, Jean-françois[1,2,b)]
Suppakitpaisarn, Vorapong[3,4,c)]

Abstract: In this paper, we analyze the game with large number of short states named *Skull & Roses* from a computer science point of view. We describe the game in a formal way, and use that formulation to obtain boundaries and average value on the game length, state-space size, and the game tree size. As a result, we can imply that developing an AI strategy for *Skull & Roses* could be more difficult than Shogi or Go. Since the game learning curve is relatively short for human, the game could be another evidence to show differences between humans and computers in game playing strength.

1. Introduction

The *Skull & Roses* game was created by Hervé Marly, illustrated by Rose Kipik, and edited by *lui-même* in 2010. After that, the game gains its popularity very quickly. As the moment this paper was submitted, the rule of the game has been officially published in German, French and English. The game received the international price of as d'or, jeu de l'année, Cannes 2011.

Compared to Go and Shogi, this game learning curve is much shorter for humans. In our experiments, players who have completed 5-6 games can enjoy this game, develop his strategy, and have a chance to win the game over experienced players.

On the other hand, *Skull & Roses* tends to be one of the hardest game with partial information. The game has three game stages with totally differentiated gameplays, but decision on each one affect the progress on the others. This number of differentiated game stages is larger than most of known turn-based games. However, the number of turns on each stage is relatively small. In short, humans can quickly enjoy the game thanks to the large number of short stages.

In this work, we used this fragmented game structure to develop a method to find upper bounds of the following numbers:

→ the state-space size

→ the search tree size

Also, we run simulations to find the average of the following numbers:

→ the game length

→ the branching factor

All numbers are relatively large compared to other games analyzed in the literature. This fact indicates that it should be hard to develop a competitive AI strategy for *Skull & Roses* . Because of this, *Skull & Roses* could be another game to show the difference between humans and computers in game playing strength. This would give us more insight about games with large number of short stages.

Also, it took Hervé Marly 15 years to balance the game[3]. Investigations on the search space bounds can be interesting to compare the balance found by a computer to the hard balancing work of a famous board-game designer.

1.1 Notations

There are four main parameters of *Skull & Roses* that we will consider in this paper: the number of players $P \geq 2$, the initial number of skulls $S \geq 1$, the initial number of roses $R \geq 1$, and the number of wins (successful challenges) required $W \geq 1$.

The original game is the one with $3 \leqslant P \leqslant 6$, $R = 3$, $S = 1$, $W = 1$ and the expansion – *Skull & Roses RED* – brings P up to 12.

2. Game Rules

Below, we formally describe the rules of *Skull & Roses* . Please refer to Fig. 1 as the decomposition of the different stage of a round.

Game sequence:

(1) Play a round.

(2) If last challenger is still playing, he takes the initiative.

(3) If last challenger is not playing, next player, $(challenger + 1) \mod P$, takes the initiative.

(4) Repeat until winning condition is reached.

[1] Graduate School of IST, The University of Tokyo
[2] JFLI, CNRS; LRI, Université Paris-Sud
[3] Global Research Center for Big Data Mathematics, NII
[4] JST, ERATO, Kawarabayashi Large Graph Project
[a)] alonso@is.s.u-tokyo.ac.jp
[b)] jf.baffier@is.s.u-tokyo.ac.jp
[c)] vorapong@nii.ac.jp

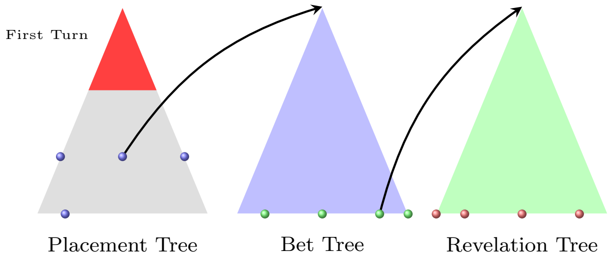

Fig. 1 Decomposition of a Round Tree
Each colored node represents a new associate tree; only leaves of a Bet Tree lead to a new Revelation Tree; only leaves of the Revelation Tree lead to a new round.
The composition of the trees is different depending if we consider state-space or search-tree, however the decomposition follows the same idea.

Round sequence:

(1) Do one *First turn.*

(2) Until a challenge is set, every player in order do either a *Card turn* or set a challenge.

(3) Every player in order do either a *Bet turn* or pass (definitively), until the challenger is decided.

(4) The challenger does a *Revelation turn,* until a skull is revealed or the number of roses revealed match the maximum bet.

(5) Let w_p be the number of games that Challenger p has won so far. If the challenge is successful, increase w_p by one. Otherwise, do a *Discard turn.*

Starting Condition:

(1) Each player has R roses and S skulls and 0 win.

(2) Player 1 takes the initiative.

Winning condition:

(1) Winning W challenges.

(2) Be last player standing.

First turn: Each player places simultaneously a card in his own pile.

Card turn: Place a card on the top your pile.

Bet turn: Increase the current bet.

Revelation turn:

> If your own pile is not empty, turn up the top card of your own pile. Otherwise, turn up the top card of any player's pile.

Discard turn:

> If the revealed skull is your card, choose a card to discard by yourself from your card. Otherwise, let the owner of the revealed skull choose one of your card to discard randomly. For both cases, the players other than you do not know which card is removed.

Successful challenge: The challenge is successful, if the challenger revealed a number of roses equal to the bet without revealing a skull.

Unsuccessful challenge: The challenge is unsuccessful, if the challenger revealed a skull in the process of the challenge.

3. Bounds

In this section, we will derive upper bounds for state-space size and game-tree size. As there are a large number of variables to be considered in this analysis, we group those variables into four groups as follows:

- *Round Variables*
  - g - the next player to make a decision
  - s_p, r_p - the number of skulls and roses remained at Player p on the beginning of this round
  - w_p - the number of games that Player p has won so far.
- *Placement Variables*
  - $d_p \in \{s, r\}^*$ - the current placement of Player p.
- *Betting Variables*
  - $b_p \in \{t, f\}$ - the current state of Player p for this round. $b_p = t$ if Player p has already decided to pass (not to bet anymore in this round). $b_p = f$ otherwise.
  - m - the current maximum bet
- *Revelation Variables*
  - v_p - the number of cards of Player p that have been revealed

3.1 State-Space Upper Bound

It is obvious that the variables presented previously in this section are enough to describe the state space of our game. Hence, we know that the state space size cannot be larger than the size of the domain sets of those variables. In this subsection, our derivation will be focused on the upper bound of that domain size for each variable group. The bound is also the upper bound for the size of state space. Our results in this subsection can be summarized as follows:

$$\begin{aligned}
&\text{Round Variables} &&: P\left[(S+1)(R+1)(W+1)\right]^P \\
&\text{Placement Variables} &&: (R+S)\left[2\binom{S+R}{S}\right]^P \\
&\text{Betting Variables} &&: P(R+S)2^P \\
&\text{Revelation Variables} &&: (R+2)+(R+1)^{P-1}
\end{aligned}$$

Combining them to form the global upper-bound, we get:

$$\begin{aligned}
\text{State Space Size} \leq\ & P^2(R+S)^2 \cdot \left[R+2+(R+1)^{P-1}\right] \\
& \times \left[4(S+1)^2(R+1)(W+1)\binom{S+R}{S}\right]^P .
\end{aligned}$$

Lemma 1. *The domain size of round variables cannot be larger than* $P[(S+1)(R+1)(W+1)]^P$.

Proof. This lemma is obvious since we know that $g \in \{1, \ldots, P\}$, $s_p \in \{0, 1, \ldots, S\}$, $r_p \in \{0, 1, \ldots, R\}$, and $w_p \in \{0, 1, \ldots, W\}$. □

Lemma 2. *The domain size of placement variables cannot be larger than* $(R+S)\left[2\binom{S+R}{S}\right]^P$.

Proof. Assume that there are s_p skulls and r_p roses remained at player p. Let $M(\ell)$ be the number of methods to place ℓ cards from those skulls and roses. We know that $M(\ell) \leq \binom{s_p+r_p}{r_p} \leq \binom{S+R}{R}$.

Let ℓ_p be the number of cards placed by Player p, and $\ell^* = \min_p \ell_p$. By the game rule, we know that the number of card placed by each player cannot be different by more than one, i.e. $\ell^* \leq \ell_p \leq \ell^* + 1$ for all p. Hence, the placement for each player is at most $M(\ell^*)+M(\ell^*+1) \leq 2\binom{S+R}{R}$. The number of placements possible for all player is at most $(M(\ell^*) + M(\ell^* + 1))^P$ assuming that the minimum number of cards among all players is ℓ^*. Thus, the number of placement is at most $\sum_{\ell^*=0}^{S+R-1} (M(\ell^*) + M(\ell^* + 1))^P \leq (S+R)\left[\binom{S+R}{R}\right]^P$. □

Lemma 3. *The domain size of betting variables cannot be larger than* $P(R+S)2^P$.

Proof. m cannot be larger than the number of cards held by all player at the beginning, $P(R+S)$, and $b_p \in \{t, f\}$ for all Player p. Thus, the domain size is at most $P(R+S)2^P$. □

Lemma 4. *The domain size of revelation variables cannot be larger than* $(R+2) + (R+1)^{P-1}$

Proof. Let Player p^* be the one who won the bet. By the game rule, he has to reveal all of his cards before begin revealing the others. Let ℓ_{p^*} be the number of cards placed by Player p^*. If $v_{p^*} \neq \ell_{p^*}$, v_p must be 0 for all $p \neq p'$. v_{p^*} can be a number between 0 and $R+1$. Hence, our domain size is $R+2$ in this case.

When $v_{p^*} = \ell_{p^*}$, the other v_p can be larger than 0. The revelation step stops when a skull is revealed, so the number of cards of Player $p \neq p^*$ revealed cannot be larger than R, i.e. $v_p \in \{0, 1, \ldots, R\}$ for $p \neq p^*$. For this case, our domain size is at most $(R+1)^{P-1}$.

By summing up the results from both cases, we know that the domain size cannot be larger than $(R+2) + (R+1)^{P-1}$. □

3.2 Game Tree Size Upper Bound

Our analysis for the game tree size has several steps, which can be stated as follows:

(1) We find the upper bound of the number of rounds required to finish the game. As shown in Lemma 5, we can show that the number cannot be larger than $\mathcal{R} = (R+S+W)P$.

(2) In one round, we find the upper bound for the number of nodes in our game tree that is in placement step. In Lemma 6, we found that the number is at most $\mathcal{P} = 2^{(R+S)+1}$.

(3) Every nodes in the placement step have two types of children. Player can continue placing their cards or start the betting step. We will consider the decision to bet as a single node, which is a child of all nodes in the placement step. We consider that node as a root of the subtree that has a different structure from the tree for the placement step. We call that subtree as *betting tree*, and we show that the number of leaves of the subtree is at most $\mathcal{B} = P^{(R+S)P+1}$ in Lemma 7.

(4) The descendant of every leaves of the betting tree are nodes for the revelation step. We call the subtree of revelation nodes for each leaves of betting tree as *revelation tree*. We show that the number of leaves of this subtree at most $\mathcal{V} = (P-1)^{R(P-1)+1}$ in Lemma 8.

(5) We know that the number of leaves for our game tree is at most $\mathcal{PBV}$ for one round. Those leaves will be a parent of the beginning of the next round. As we know that there will be at most $\mathcal{R}$ rounds played, the number of leaves of our game tree is at most $\mathcal{PBV}^{\mathcal{R}}$.

Lemma 5. *The maximum number of rounds before the game ends is at most* $(R+S+W)P$.

Proof. On the beginning of the game, $s_p = S$, $r_p = R$, $W - w_p = W$ for all Player p. In every round, $\sum_p (s_p + r_p + (W - w_p))$ decreases by 1. The game stops before the number drops to 0. Hence, there are at most $(R+S+W)P$ rounds played. □

Lemma 6. *For each round, the number of nodes in placement stage cannot be larger than* $2^{(R+S)+1}$.

Proof. In each step, current player can choose to place a skull or rose. Each placement node has two placement children. Hence, a tree of placement nodes in one round is a binary tree. Because the depth of that binary tree is at most $R+S$, the number of nodes is at most $2^{(R+S)+1}$. □

Lemma 7. *The number of leaves for each bet tree is at most* $P^{(R+S)P+1}$.

Proof. Consider a node that represents a state that q players has not been passed and t cards remained for betting, i.e. $q = ||\{p|b_p = f\}||$ and $t = \sum_p ||d_p|| - m$. Let $B(q,t)$ be the number of leaves of a subtree rooted at that node. We know that

$$B(q,t) = \sum_{s=0}^{s-1} B(s,t) + B(q,t-1)$$

for any $q \leq 2$ and $q \leq 1$, and $B(q,t) = 1$ otherwise. $\sum_{s=0}^{s-1} B(s,t)$ represents the case when the player decides to bluff with higher bet, and $B(q,t-1)$ represents the case when the player decides to pass. We note that the player who turns the game into the betting step cannot pass on his first turn. Therefore, the number of leaves we have in our betting tree is

$$B'(q,t) = \sum_{s=0}^{s-1} B(s,t),$$

when t of the number of players remained in this round, and

Table 1 The upper bound of state-space size and tree size we get from our analysis for each game parameter we can have in *Skull & Roses RED*

(P, S, R, W)	State-space size	Tree size
$(2, 1, 3, 2)$	10^{9}	10^{66}
$(3, 1, 3, 2)$	10^{13}	10^{226}
$(4, 1, 3, 2)$	10^{16}	10^{506}
$(5, 1, 3, 2)$	10^{20}	10^{919}
$(6, 1, 3, 2)$	10^{24}	10^{1475}
$(7, 1, 3, 2)$	10^{27}	10^{2181}
$(8, 1, 3, 2)$	10^{31}	10^{3043}
$(9, 1, 3, 2)$	10^{34}	10^{4068}
$(10, 1, 3, 2)$	10^{38}	10^{5262}
$(11, 1, 3, 2)$	10^{42}	10^{6628}
$(12, 1, 3, 2)$	10^{45}	10^{8169}

q is the number of cards placed during the placement step. By induction, we can show that

$$B'(q, t) \leq q^{t+1} \leq P^{(R+S)P+1}.$$

□

Lemma 8. *The number of leaves for each revelation tree is at most* $(P-1)^{R(P-1)+1}$.

Proof. The player who won the bet can choose to reveal a card from the other $P-1$ players. As there are at most R roses placed for each player, the player cannot pick more than $R(P-1)+1$ cards. Hence, the revelation tree is an $(R(P-1)+1)$-depth tree in which each node has at most $P-1$ children. The number of leaves is at most $(P-1)^{R(P-1)+1}$. □

In Table 1, we calculate the upper bound we get for each game parameter we can have in *Skull & Roses RED.*

4. Simulations Results

Besides the analysis in the previous section, we also perform a simulation to find an average branching factor and game length. For the game length, we run 10^7 games with the assumption that players make decision randomly. The possibility that players select each of their choices is set to be uniform. For the average branching factor, we also perform 10^7 uniformly random simulations. We collect the number of children of each node we visit in those simulations, and take an average value.

The simulation results are shown in Table 2. In the same table, we also show the comparison between Connect Four, Backgammon, Chess, Shogi, Go, and *Skull & Roses* . It can be obviously seen in the table that *Skull & Roses* is more complicated than the other games in every factors.

5. Conclusion and Future Works

In this paper, we perform both the analysis and simulation to show that *Skull & Roses* is one of the most difficult game for AI strategy developer. Beside the results that *Skull & Roses* is significantly more complicated for AI strategy, we know from this work that those analysis are significantly harder for the game with large number of short game states, compared to the other games analyzed before. The simulation is also very complicated, and developing monte-carlo-based strategy could be very challenging task.

Table 2 *Skull & Roses* results compared with Connect Four, Backgamon, Chess, Shogi, Go:
A *Skull & Roses* game, $S\&R(P, S, R, W)$, is defined by a quadruplet of the players, starting skulls, starting roses, and required challenges

Game	Game length	Branching factor	State-space	Game Tree
Connect Four [1], [7]	36	4	10^{13}	10^{21}
$S\&R(3, 1, 3, 2)$	63	2	10^{13}	10^{226}
Backgammon [5]	55	250	10^{20}	10^{144}
$S\&R(6, 1, 3, 2)$	434	16	10^{24}	10^{1475}
$S\&R(9, 1, 3, 2)$	1304	43	10^{34}	10^{4068}
$S\&R(12, 1, 3, 2)$	2858	149	10^{45}	10^{8169}
Chess [4]	80	35	10^{47}	10^{123}
Shogi [2], [8]	115	92	10^{71}	10^{226}
Go [1], [6]	150	250	10^{171}	10^{360}

Besides the case when P, S, R, W can be any natural number, we are also interested in more specific case when some of those variables are set to be a small constant. Specifically, we are improving our upper bound for the case when $S = 1$, $R = 3$, $W = 2$ as in *Skull & Roses RED.* We are also looking for other evidences to show that developing AI strategy for *Skull & Roses* is difficult.

Acknowledgements

The authors would like to thank Mr. Yuki Kawata, Mr. Chihiro Komaki, Mr. Frédéric Maillasson, Prof. Hiroshi Imai, and anonymous reviewers for giving us several useful comments during the course of this research.

References

[1] Allis, V. L.: Searching for solutions in games and artficial intelligence, PhD Thesis, University of Limburg, Maastricht (1994).

[2] Iida, H., Sakuta, M. and Rollason, J.: Computer shogi, *Artificial Intelligence*, Vol. 134, No. 1-2, pp. 121–144 (2002).

[3] Marly, H.: Rules of Skull & Roses, (online), available from ⟨http://www.skull-and-roses.com/pdf/Skull_rules_Us.pdf⟩ (accessed 2013-09-23).

[4] Shannon, C. E.: Programming a computer for playing chess, *Philos. Mag. (7)*, Vol. 41, No. 314, pp. 256–275 (1950).

[5] Tesauro, G.: Practical Issues in Temporal Difference Learning, *Machine Learning*, pp. 257–277 (1992).

[6] Tromp, J. and Farneback, G.: Combinatorics of Go, *Proceedings of 5th International Conference on Computer and Games*, Torino, Italy (2006).

[7] Tromp, J.: John's connect four playground, (online), available from ⟨http://homepages.cwi.nl/ tromp/c4/c4.html⟩ (accessed 2013-09-23).

[8] Yen, S.-J., Chen, J.-C., Yang, T.-N. and Hsu, S.-C.: Computer Chinese Chess, *ICGA Journal*, Vol. 27, No. 1, pp. 3–18 (2004).