Exploiting the PuyoVS2 RNG
A statistical study

BLue & BrouH

***Abstract*—This article presents a probabilistic and statistical analysis of the random number generator in Puyo VS 2, showing its biases could be exploited to one's advantage. A case study of a few game situations provides an insight on how a player can knowingly use the bias to make better short-term decisions. The study also shows that exploiting this bias will benefit the player on the long run, for instance, in the course of a « first to 100 » battle.**

***Index Terms*—Puyo Puyo - Probabilities & Statistics - Random Number Generator - Monte Carlo Method**

I. Introduction

To keep this paper short, we will not remind the basic game rules of Puyo Puyo 2, but focus on those relevant to the proposed study.

We also focus on classic rules mainly used in competitive gameplay.

However, mathematical concepts will be explained to be understandable to the average player with a basic math background and the reader will be warned before math-heavy sections.

A. Puyo Puyo 2: basic rules and definitions

The two adversaries are each receiving the same pseudo-random sequence of colored puyo pairs (*tsumos*), with the first two pairs being picked from a 3-color only subset.

While the number of colors can be set from 3 to 5, we only focus on the widely used 4-color gameplay. The next tsumos will thus be picked from a 4-color subset. Consequently, only 16 different tsumos exist, some of which are equivalent from a player's perspective, but may not be to the game's algorithms: for instance, the red-blue (`RB`) and blue-red (`BR`) tsumos.

The pseudo-random tsumo sequence is generated by a custom algorithm that differs from one game version to the other. These differences seem to take the form of variance-reducing countermeasures, and might have been implemented in order to offset the apparent long streaks of unwanted tsumos introduced by a *possibly* fully random algorithm, such as Puyo Puyo 2's.

Therefore, given currently available information, this study only applies to PuyoVS2.

B. PuyoVS2 RNG and pair picking algorithm

One of the inherent issues of applying random number generators (RNG) to such a small subset of tsumos is that you might end up picking (or not picking) a specific pair over and over for a relatively long streak.

As situations in Puyo Puyo battles often leave you a very small time frame to react, such long streaks make the player mostly rely on pure luck and are ultimately undesirable.

However, implementing a fair filter algorithm to offset those deficiencies can prove difficult: one must keep long-term tsumo frequencies even, but reduce short-term variance.

According to Hernan (creator of the online fan game), PuyoVS2 bases its tsumo-picking algorithm on the Tetris Grand Master randomizer. The tsumo-picking algorithm works as follows:

1) an history buffer of the last N_p tsumos picked is kept;
2) an iteration counter N_c is initialised at a predefined value: this value is the maximum number of attempts at getting a pair that has not recently been dealt to the player;
3) a RNG picks a candidate tsumo from the set of 16, according to a uniform probability distribution (each tsumo has an equal chance of getting picked);
4) the candidate is checked against the history:
   a) if $N_c = 0$, the candidate is selected whether or not it already belongs to the history,
   b) else, the candidate is either selected if it doesn't match any tsumo in the history buffer, or N_c gets

decremented and a new attempt is made at the check (go to step *3*)

5) once a candidate has been selected, it gets pushed to the history buffer, replacing the oldest entry.

The pseudo-random nature of RNG, and the deterministic nature of the filter algorithm makes the outputs repeatable, as a RNG using the same *seed* value will ultimately generate the very same sequence of outputs. This explains the fact that the two players can ultimately obtain the very same sequence of tsumos.

3 parameters of the picking algorithm are currently unknown:

- the size of the history buffer N_p;
- the maximum number of attempts at picking a different pair N_c;
- whether some pairs are considered equivalent or not, when checking a candidate against the history buffer (*i.e.* is `RB` equivalent to `BR`).

While waiting for better data, we will consider arbitrary values in the following study (see subsection I-D). An updated version of this paper will be published once the accurate parameters are known.

C. Proposed study

Ignoring the game algorithms, a player would base his decisions on a uniform probability distribution and hope that a specific tsumo has the same chance of being given to him in the near future than any other does.

The previously introduced filter could alter the uniform distribution in two major ways:

- short-term effects, meaning the probability of getting a specific color in the next few tsumos depends on whether or not it already appears in the few previously given pairs;
- long-term effects, meaning some tsumos are going to be less likely than others (specifically regarding unicolor tsumos).

As those long-term effects are likely to be insignificant from a gameplay perspective, we will focus on short-term effects. So we will study a set of cases in which the history buffer could impact short-term expectations of getting a puyo of a specific (expected) color, for up to the next 3 unknown tsumos.

This best describes the issue of the filtering algorithm from a player standpoint. From a mathematical standpoint, this boils down to studying convergence, expected value, variance, and confidence interval of the overall tsumo-picking algorithm. This analysis is performed for every non-equivalent history buffer state.

These non-equivalent states correspond to history buffers filled with a varying number of different tsumos containing at least one puyo of the expected color. So, if we say our trigger color is green, we will have buffers with one, two, three and four different tsumos, each with at least one green puyo. Since it doesn't matter how many of the same pair is in the buffer, a history buffer containing two of the same tsumo will have the same effect as a buffer containing only one and will be considered as such. The blank spaces in the buffer states will be filled with arbitrary pairs that don't have any green puyo. We will consider every possible combination of green tsumos positions in the buffer: Indeed, as we consider the next three draws, some positions are not equivalent as they will be shifted out of the buffer earlier than others.

The results will be applicable to situations where a player is aware of a specific tsumo history and is expecting to get a specific single color in the few next puyo pairs. For instance, the results could benefit players undecided as to whether they should trigger their current chain or try to quickly expand upon it (say, by one or two): if the desired trigger is too unlikely to appear soon enough, the player might as well continue chaining and wait for a better opportunity. On the contrary, expanding the chain with the currently known upcoming tsumos could alter the history buffer in a way that will make it less likely to hit the right trigger soon enough.

D. Assumptions and unknown parameters

Here's a summary of the assumptions made to complete the statistical study:

- the pseudo-RNG obeys a uniform probability distribution;
- the settings include 4-color puyos (normal difficulty), and we're far enough into the current battle so that the history buffer is clear of the initial 3-color subset;
- history buffer length N_p is set to 4;
- maximum number of checks N_c is set to 3;
- the filter algorithm distinguishes between all tsumo permutations, *i.e.* `RB` and `BR` are different tsumos and will not affect one another[1];
- any tsumo formed with the expected trigger color is equally worth to the player, so the specific situations where a player expects a very precise pair are not accounted for;
- only the probabilities for the upcoming 3 pairs are evaluated, as this is considered enough to expand a chain by 1, keeps the required computational power acceptable and and more complex situations

[1]From a player standpoint, those pairs are equivalent.

can boil down to these cases once the buffer is completely renewed.

II. Statistical analysis

A. Method

The following section explains how we model the problem using Bernoulli trials and study their Monte Carlo convergence. If not interested, you may skip to the result analysis in section II-D.

Deducing the actual values of the new probabilities induced by the filtering algorithm is non-trivial to perform analytically, so we propose doing so through numerical simulations.

We have re-implemented the tsumo-picking algorithm, complete with RNG and filter, and made it generate the upcoming pairs.

For each pre-defined starting buffer state, the algorithm generated $N_s = 300\,000$ sequences of 3 upcoming tsumos. This attempts to offset variability of any quantity of interest (expected value, variance), which should ensure proper convergence to their true value according to the law of large numbers. This convergence will be verified first.

We will evaluate the cumulative probabilities of getting a trigger color in either one, two or three rolls. Thus, the probability of hitting a specific color on the second upcoming pair includes the probability of getting it right away with the first tsumo.

Studying the situation where a player hopes to get a specific trigger color boils down to what is called a Bernoulli trial, in which the outcome is either success (getting the expected trigger color withing the next x upcoming pairs) or failure (which is always an option).

As with all Bernoulli trials, success is given a value of 1, and failure 0. Succeeding at the first upcoming pair, implies succeeding at the second and third upcoming ones. Our study then comes down to computing estimations of the expected value and standard deviation of successes over the 300 000 generated sequences for each considered history buffer state (adjusted for all equivalent permutations).

Estimating the expected value then equates to calculating the average number of successes in each given situation. By noting that, statistically speaking, the probability of an event occurring is the ratio of favorable outcomes to total number of trials, the estimation of the expected value of successes simply translates to the probability of hitting a trigger color in this given situation. In mathematical language, this can be written: Let X be our Bernoulli random variable (taking the value 1 in case of success, 0 otherwise). Then the true expected value of that random variable is:

$$E[X] = \lim_{N_s \mapsto \infty} \frac{1}{N_s} \sum_{i=1}^{N_s} x_i = p_s, \tag{1}$$

where x_i is the outcome of the i-th Bernoulli trial and p_s is the exact probability of success. This corresponds to the ratio of successes obtained for an infinite number of trials, which indeed gives the exact probability of success. However, we will only be able to estimate an approximate probability $\tilde{p}_s$ for a finite, but very large value of N_s:

$$\tilde{p}_s = \frac{\text{Number of successes}}{\text{Number of trials}}. \tag{2}$$

Our estimator of the probability will thus be the estimator of the expected value :

$$\tilde{E}[X] = \frac{1}{N_s} \sum_{i=1}^{N_s} x_i = \tilde{p}_s, \tag{3}$$

We will however be certain that:

$$\lim_{N_s \mapsto \infty} \tilde{p}_s = p_s. \tag{4}$$

So if we want a better estimation of the probability, we simply have to increase the number of trials, which can be done at the cost of computational time.

Studying standard deviation of the previous Bernoulli trial will expose short-term fluctuations of success rate around the expected value. This will be particularly useful when considering whether the results are of any use to the *average* (or rather, *insane*) Puyo Puyo player.

In the same fashion as we did with the expectation, we can write our estimator for the standard deviation, which will be a converging estimate of the true value as N_s goes to infinity. Let us call $\tilde{\sigma}[X]$ the estimated standard deviation of X. By noting that it is related to the estimated variance $\tilde{V}[X]$ by the simple equations:

$$\tilde{\sigma}[X] = \sqrt{\tilde{V}[X]}, \tag{5}$$

$$\tilde{V}[X] = \tilde{\sigma}[X]^2, \tag{6}$$

we can access $\tilde{\sigma}[X]$ by estimating $\tilde{V}[X]$, which is easier to do. We get:

$$\tilde{\sigma}[X] = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} (x_i - \tilde{p}_s)^2} \tag{7}$$

Without getting into details, this estimator is *biased* regarding the variance and the standard deviation, and we will rely on an estimator which is unbiased in terms of variance but still very slightly biased in terms of standard deviation, written as follows:

$$\tilde{\sigma}[X] = \sqrt{\frac{1}{N_s - 1}\sum_{i=1}^{N_s}(x_i - \tilde{p}_s)^2} \tag{8}$$

This method of estimating probabilistic quantities by statistics over a large sample of data is called the Monte Carlo method, commonly used in numerical analysis. Several advantages include having a known rate of convergence, as well as being able to check the convergence while the experiment is running whatever that experiment may be, because the method is non-intrusive and can be implemented on anything. The estimation error of the Monte Carlo method converges to zero in the order of $\frac{1}{\sqrt{N_s}}$, so if one wants to decrease the error tenfold, he must increase the number of trials hundredfold.

B. Analytical probabilities for a uniform distribution

Remember, we only focus on probabilities of hitting a specific trigger color, whichever the tsumo it is in.

If one considers that upcoming tsumos are selected using a uniform probability distribution (each tsumo is always as likely as another to come up), then the relevant probabilities can be deduced this way:

as there are 16 independant tsumos with a four color rule, and 7 of those containing any chosen color (which comes down to 4 different tsumos to the player's eyes), we instantly get the probability p_{t1} of hitting the trigger in the next upcoming pair from equation (19):

$$p_{t1} = \frac{7}{16} \approx 43.75\%. \tag{9}$$

Hitting the desired trigger color on a specific draw will always be p_{t1} and we can also say that missing the trigger on that draw will have a probability of $1 - p_{t1} = \frac{9}{16}$.

The probability of hitting the trigger color in any of the two upcoming tsumos amounts to hitting it on the first draw or missing and hitting hit on the second one, which translates to the following equation:

$$p_{t2} = \frac{7}{16} + \frac{9}{16} \times \frac{7}{16} \approx 68.36\%. \tag{10}$$

The same reasoning applies to getting the trigger on any of the next 3 draws:

$$p_{t3} = \frac{7}{16} + \frac{9}{16} \times \frac{7}{16} + \frac{9}{16} \times \frac{9}{16} \times \frac{7}{16} \approx 82.20\%. \tag{11}$$

Those are the probabilities a player could think of naively. The next subsections will focus on studying how actual probabilities differ from that.

C. Effect of history on long-term tsumo frequencies

If one wants the game to be fair, the filter algorithm should not impact long-term probabilities of a particular color or tsumo over the others, with respect to analytical values discussed in the previous subsection.

Using our reimplementation, we generated a million different tsumos in accordance to the filtering algorithm, and computed their actual appearance frequency.

Analytically, unicolor tsumos should appear $1/16^{th}(= 0.0625)$ of the time, while the player expects equivalent bi-color tsumos to appear $2/16^{th}(= 0.125)$ of the time.

Tsumo	*Probability*
RR	0.062190
GG	0.062652
YY	0.062678
BB	0.062717
BR	0.124441
BG	0.125103
BY	0.125182
RG	0.125316
RY	0.124764
GY	0.124957

TABLE I: Actual (perceived) appearance frequencies of tsumos as impacted by the filter algorithm

Table I sums up our observed frequencies. Given our relatively small sample size, the fluctuations are well within our margin of error.

As such, results confirm the filtering algorithm has no long-term impact on fairness between tsumos.

D. Monte Carlo Convergence

As well as proving the Monte Carlo method has been correctly used, studying convergence of the probabilities of getting a particular color in the next 1, 2 or 3 pairs will show how actual probabilities stand in comparison with naive analytical values. It also tells us about short-term fluctuations of those probabilities by showing how fast their values converge. Those fluctuations are what usually matters to a player, but convergence alone cannot quantify their effect ; they will be discussed in section II-F.

The figures of this section are to be read as follows:

- red, green and blue solid lines are respective probabilities of hitting the desired color for the upcoming 1, 2 and 3 pairs. Don't confuse them with probabilities of getting a red, green or blue puyo;
- horizontal axis bears the number of Bernoulli trials;
- vertical axis bears the estimated probabilities after a given amount of Bernoulli trials (horizontal axis), which accounts for the long-term convergence;

- dashed lines are the respective analytical values (see section II-B).

For any given buffer state, many permutations exist with regards to the position of incriminating tsumos. Those permutations induce small differences in actual converged probabilities. However, keeping track of those differences is way too difficult from a player's perspective.

To keep those aspects simple enough, we will aggregate all permutations of buffer states equivalent to the player's eyes when summing up probabilities of relevance later on.

However, each simulation actually accounted for these permutations and ran over a sample of 25 000 trials each time. Thus, each buffer state ended up going through an equal amount of 300 000 trials, by repeating equivalent permutations when needed.

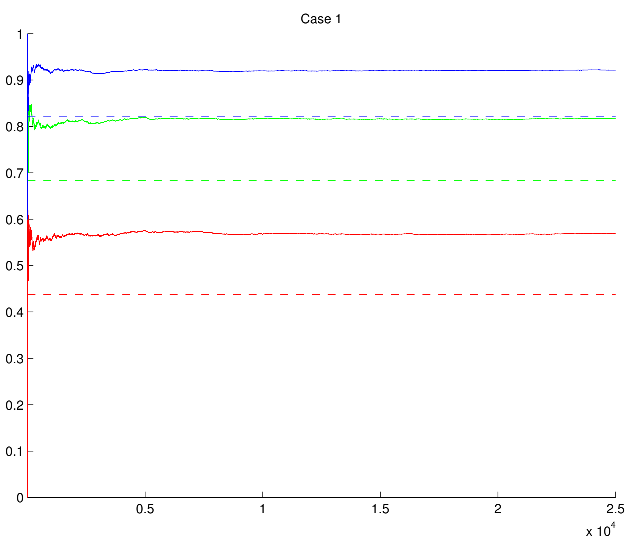

Fig. 1: Actual convergence for a buffer with green puyos in 0 different tsumos

Figure 1 illustrates convergence for situations where the history buffer has no puyos of the expected color. A notable event is the fact that the odds of getting the desired color are way higher than expected. When considering the second upcoming pair, they actually match those one would think he has of getting his color in not two, but three draws.

This is quite a noticeable shift over the naive expectations: odds of getting the desired color are higher by up to 14 points (57.9% instead of 43.75% expected on the first draw; 82.04% instead of 68.36% expected on the second draw).

On the contrary, figure 2 illustrates the opposite situation with a buffer full of different tsumos, each one

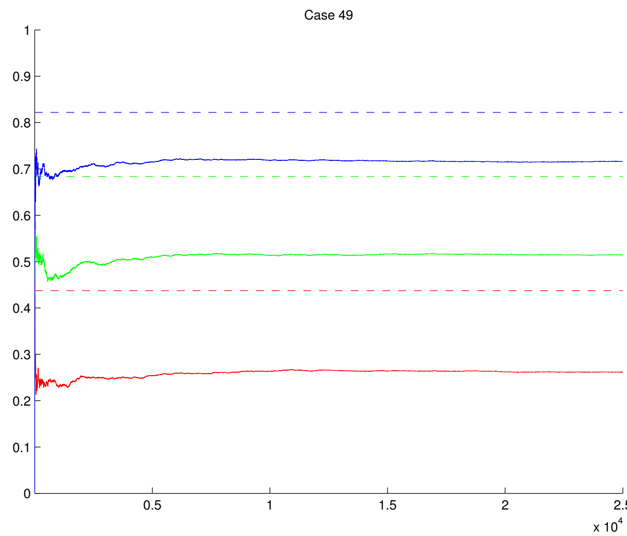

Fig. 2: Actual convergence for a buffer with green puyos in 4 different tsumos

formed with at least one puyo of the expected color. Actual probabilities are indeed much lower than expected (for instance, 26.14% instead of 43.75% expected on the first draw). The extent of the difference seems to be determined by the number of attempts at getting a pair absent of the history: the more attempts are made, the lower the odds. The lower bound seems to be linked to the size of the history buffer.

As explained before, convergence not only depends on the number of different tsumos containing the expected color in the history buffer, but also on the position of these tsumos in the buffer.

Indeed, position is an important factor when considering probabilities of hitting a color from the second upcoming tsumo onwards, as oldest tsumos get shifted out of the buffer first.

Figures 3 and 4 both illustrate the convergence for a buffer with two incriminating tsumos. But figure 4 demonstrates convergence when incriminating tsumos are erased from the buffer later than the buffer state allows in figure 3.

Based on our observations, the following conclusions are of a relative importance to the player:

- position permutations within a buffer state only impact probabilities on the second and third draw;
- there are significant differences (up to 17 points) between actual and analytical probabilities;
- short-term variability seems high enough to warrant a study of its own, as fluctuations last for more occurrences than one could expect in a typical match.

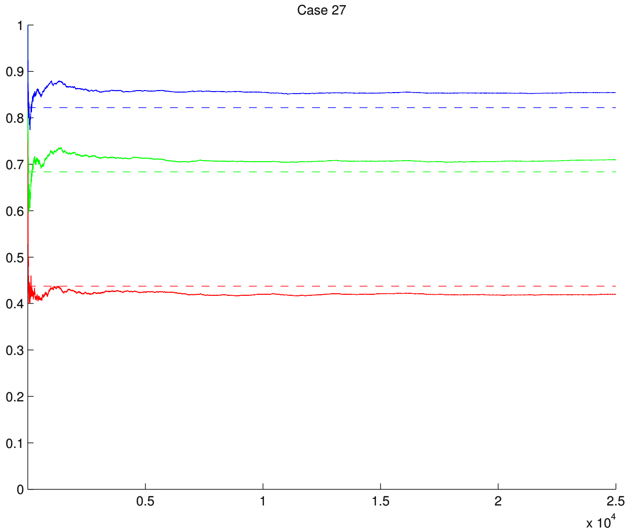

Fig. 3: Actual convergence for a buffer with green puyos in 2 different tsumos, erased earlier on

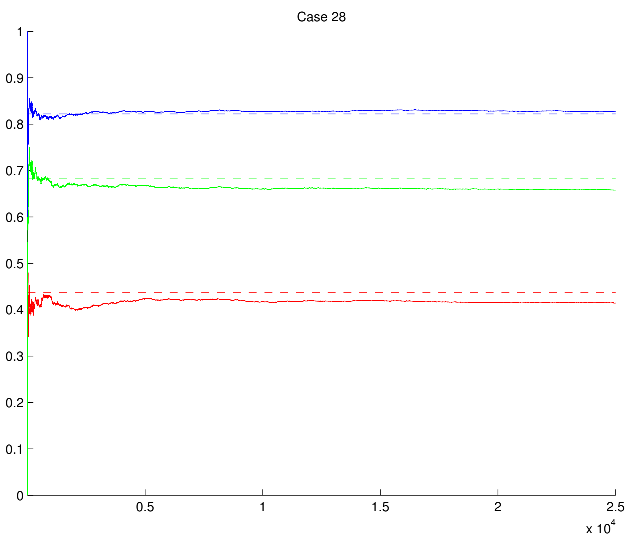

Fig. 4: Actual convergence for a buffer with green puyos in 2 different tsumos, erased later on

The next subsection will provide a table summary of the actual probabilities for each buffer state of importance to the player (permutation-induced fluctuations being averaged).

E. Expected value

Estimated expected value of our Bernoulli random variable, given a sufficiently large sample size, is the computed probability of hitting our trigger color, accounting for the filtering algorithm.

This boils down to computing the final converged value in each of our previous simulations, while grouping the results in buffer states that include all position permutations.

Resulting data can be used by a player as long as he memorizes how many different tsumos with his target trigger color are in the recent history buffer.

Tables II through VI provide actual probabilities of hitting the target color in the coming 1, 2, or 3 draws, as well as the analytical values for comparison (given a number of incriminating tsumos in the buffer).

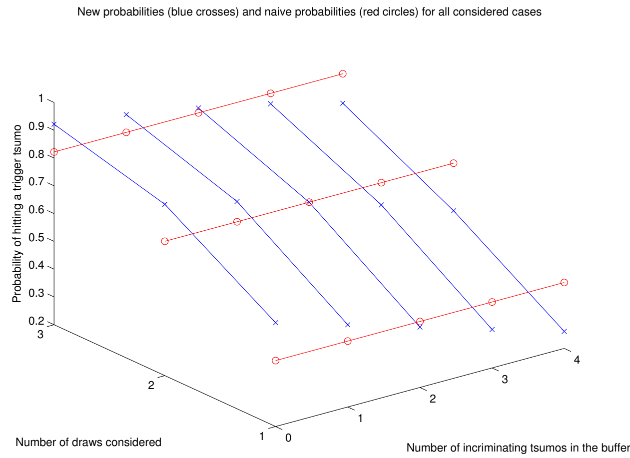

Fig. 5: Expected values

Figure 5 sums up all these tables in a single 3 dimensional plot.

Draw #	*Analytical probability*	*Actual probability*
1	0.4375	0.574230
2	0.6836	0.816927
3	0.8220	0.922777

TABLE II: Actual probabilities obtained for a buffer with 0 incriminating tsumos

Draw #	*Analytical probability*	*Actual probability*
1	0.4375	0.495650
2	0.6836	0.756367
3	0.8220	0.885420

TABLE III: Actual probabilities obtained for a buffer with 1 incriminating tsumos

Draw #	*Analytical probability*	*Actual probability*
1	0.4375	0.418903
2	0.6836	0.682913
3	0.8220	0.839670

TABLE IV: Actual probabilities obtained for a buffer with 2 incriminating tsumos

Draw #	*Analytical probability*	*Actual probability*
1	0.4375	0.340430
2	0.6836	0.604410
3	0.8220	0.784526

TABLE V: Actual probabilities obtained for a buffer with 3 incriminating tsumos

Draw #	*Analytical probability*	*Actual probability*
1	0.4375	0.261347
2	0.6836	0.512667
3	0.8220	0.716390

TABLE VI: Actual probabilities obtained for a buffer with 4 incriminating tsumos

Notable facts include:

- only buffers with 2 incriminating tsumos give probabilities close enough to the analytical values so that it doesn't make a true difference to the player;
- extreme situations will actually have a great impact on about 1 out of 6 occurrences, which may become apparent in long matches such as « first to 100 » events;
- sometimes, actual probabilities will match probability levels naively expected for one less or one more draw, which can amount for a significant time frame difference in the context of a heated battle.

F. Standard deviation

If we showed in section II-D the true impact of the history buffer on the actual probabilities of getting a coveted color, the figures also illustrate quite an important short-term variance in the results, which may impair applicability of those theoretical results to real-world battles. A high variance (or standard deviation) means short-term applications are quite impractical.

We sum up the estimated standard deviation (see Eq. (8)) computed for our previous experiments in table VII. The analytical standard deviation of our Bernoulli trial without the filtering process has been listed as well for comparison. In this case, for the i-th draw with probability of success p_{ti}, standard deviation of the Bernoulli trial is given by:

$$\sigma_a[X] = p_{ti} \times (1 - p_{ti}) \tag{12}$$

Incriminating tsumos in buffer	*Draw #1*	*Draw #2*	*Draw #3*
0	0.493746	0.383910	0.265771
1	0.260501	0.202901	0.138724
2	0.254151	0.206490	0.136626
3	0.304275	0.267063	0.188027
4	0.551421	0.493911	0.368489
No filtering	0.496078	0.465071	0.382513

TABLE VII: Estimated standard deviation of the Bernoulli trials for different buffer states and reference (naive) case

G. 95 % confidence interval

In order to account for the short-term variability of the results, we have powerful statistical and probabilistic tools at our disposal. However, it is very important that we present results regarding this matter in a way that the player can use to assert the advantage he would almost certainly gain (with a given confidence level) from using the updated probabilities we propose, for a given number of situations where he has to hope for a specific trigger color. What we will present is, for all considered states of the buffer and considering 1, 2 or 3 draws, a graph in which we plot the number of expected successes against the number of trials ranging from 1 to 200, thus giving a good idea of what one can expect during a fairly long match. In addition to plotting the average (long-term) number of expected successes deduced from both the naive and our new probabilities, we will show a 95 % confidence interval of the results, meaning a "range" of number of successes in which 95 % of short-term scenarios will fall taking short-term variability (standard deviation) into account. The interval used here is necessarily centered around the average. The confidence level is arbitrary and we simply chose 95 % for convenience and because it is a fairly high confidence level. The reader should also know that the accuracy of this interval grows as more trials are done but the method used is considered accurate when the number of successes and failures both are over 5. More complex ways to compute the confidence interval can be used to account more precisely for those extremely short-term cases, but we have not tried to implement them as of now. If the community of players thinks it would be profitable to have such information, an update of the results will be provided.

So for example, for a given buffer state, looking for a trigger tsumo in the next X draws, if we look at a match where this situation arises once per battle, with success leading to winning the game for simplicity's sake and 20 battles are played, and let's say the naive average gives you 10 successes. If the new estimated average gives you 15 and the confidence interval has an upper bound of 18 and a lower bound of 12 successes, then it means that in the long run, using the new probability will give you five extra wins per 20-battle matches, but in any 20-battle match you play, you can be 95 % sure that you will get between 2 and 8 extra wins, so adopting this new strategy will be highly profitable even in this short a match. If the upper and lower bounds are 5 and 25, 95 % of the 20-battle match you play might go wrong with 5 extra losses so it may not be worth the trouble.

The rest of the section provides a fairly detailed mathematical development we performed in order to construct the confidence interval. The reader might want to skip directly to the results in section II-H.

In order to obtain this confidence interval, we have to consider the *Central Limit Theorem* which states that for a sequence of independant random variables defined on the same probability space (meaning they follow the same probability distribution), all with the same mean and same standard deviation, the difference between the arithmetic average of those random variables and their own mean follows a normal distribution. Before writing this in mathematical terms, the reader should note that, in our case, each of those random variables is a Bernoulli random variable X_i representing a trial with an (estimated) mean $\tilde{E}[X]$ and standard deviation $\tilde{\sigma}[X]$, explicitly given previously in this paper for all relevant cases of our study. The length of the sequence is the number of occurrences we consider and will be noted N_g.

The arithmetic average of the sequence of Bernoulli random variables is:

$$\bar{S}_{N_g} = \frac{1}{N_g}\sum_{i=1}^{N_g} X_i. \tag{13}$$

By noting that $V[a \times \sum_{i=1}^{N} X_i] = Na^2\tilde{\sigma}^2$, it follows that the variance of $\bar{S}_{N_g}$ is:

$$V[\bar{S}_{N_g}] = \frac{\tilde{\sigma}^2}{N_g} \tag{14}$$

and its standard deviation is :

$$\sigma_{\bar{S}_{N_g}} = \frac{\tilde{\sigma}}{\sqrt{N_g}}. \tag{15}$$

Following the *Central Limit Theorem*, the normal distribution $\mathcal{N}(0, V[\bar{S}_{N_g}])$ (with zero mean and variance $V[\bar{S}_{N_g}]$) is a good approximation of the distribution followed by $\bar{S}_{N_g} - \tilde{E}[X]$, given that N_g is sufficiently large:

$$(\bar{S}_{N_g} - \tilde{E}[X]) \sim \mathcal{N}(0, V[\bar{S}_{N_g}]). \tag{16}$$

For convenience, we will divide this quantity by its standard deviation so that the new quantity, which we will note Z, follows a standard normal distribution (zero mean and unit variance) and we can thus use the well-known table of values of its cumulative distribution function later:

$$Z = \frac{\bar{S}_{N_g} - \tilde{E}[X]}{\sigma_{\bar{S}_{N_g}}} \sim \mathcal{N}(0, 1). \tag{17}$$

If we can determine the interval in which this quantity has a 95 % probability of falling, then we can deduce the same interval for $N_g \times \bar{S}_{N_g} = \sum_{i=1}^{N_g} X_i$ which is the number of successes in N_g trials since the value of X_i is 1 in case of success, 0 in case of failure.

Before developing the explicit expression of the interval, let us introduce the cumulative distribution function $z \mapsto \phi(z)$ of the random variable Z following the standard normal distribution $\mathcal{N}(0, 1)$, defined by:

$$\phi(z) = P(Z \leq z). \tag{18}$$

This cumulative distribution function has an inverse function $y \mapsto \phi^{-1}(y)$ and the values of both of those can easily be found in tables or computed by relevant software.

If we write the probability of Z being bounded by a zero-centered interval of arbitrary length $2z$, we then need to find the value of z so that this probability is 95 %. If we can write this probability in terms of the cumulative distribution function by using Eq. (18), then we can use tables to determine z.

It is clear that

$$P(-z \leq Z \leq z) = P(Z \leq z) - P(Z \leq -z) \tag{19}$$

and, using the symmetry of the probability distribution function with respect to 0, that

$$P(Z \leq -z) = P(Z \geq z). \tag{20}$$

To be convinced of that, one can plot the probability density function of this distribution, shown in figure 6.

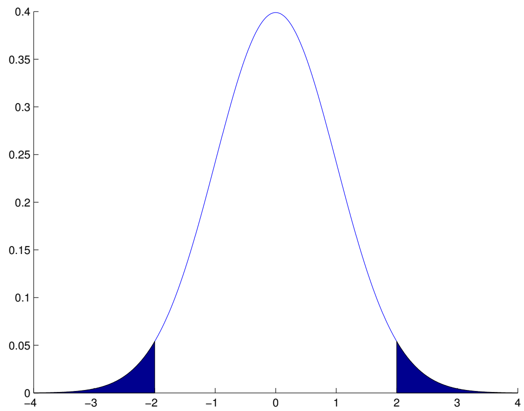

Fig. 6: Probability density function of $\mathcal{N}(0,1)$

The area under the curve for $z \leq -2$ and $z \geq 2$ are filled. We can see that the probability of being between 2 and -2 is equal to the probability of being inferior to 2 (the whole curve without the rightmost filled area) substracted by the probability of being inferior to -2 (leftmost filled area), that is to say Eq. (19). By the symmetry of the function with respect to 0, we can also see that the probability of being superior to 2 (rightmost filled area) is the same as the probability of being inferior to -2 (because the areas are the same), that is to say Eq. (20). Introducing Eq. (20) in Eq. (19) we get

$$P(-z \leq Z \leq z) = P(Z \leq z) - P(Z \geq z) \quad (21)$$

and since by definition $P(Z \geq z) = 1 - P(Z \leq z)$,

$$P(-z \leq Z \leq z) = P(Z \leq z) - (1 - P(Z \leq z)),$$
$$P(-z \leq Z \leq z) = 2P(Z \leq z) - 1,$$
$$P(-z \leq Z \leq z) = 2\phi(z) - 1. \quad (22)$$

Since we want $P(-z \leq Z \leq z) = 0.95$, we have $\phi(z) = \frac{1+0.95}{2}$. Lastly, using the inverse of the cumulative distribution function we get:

$$z = \phi^{-1}(\phi(z)) = \phi^{-1}(\frac{1+0.95}{2}) \approx 1.96. \quad (23)$$

Our 95 % confidence interval for the random variable Z is thus given by $-1.96 \leq Z \leq 1.96$, that is to say $-1.96 \leq \frac{\bar{S}_{N_g} - \tilde{E}[X]}{\sigma_{\bar{S}_{N_g}}} \leq 1.96$ following Eq. (17). The 95 % confidence interval for the number of successes $N_g \times \bar{S}_{N_g}$ is then given by:

$$N_g \times (\tilde{E}[X] - 1.96 \times \sigma_{\bar{S}_{N_g}}) \leq N_g \times \bar{S}_{N_g}$$
$$N_g \times \bar{S}_{N_g} \leq N_g \times (\tilde{E}[X] + 1.96 \times \sigma_{\bar{S}_{N_g}}) \quad (24)$$

$$N_g \times (\tilde{E}[X] - 1.96\frac{\tilde{\sigma}}{\sqrt{N_g}}) \leq N_g \times \bar{S}_{N_g}$$
$$N_g \times \bar{S}_{N_g} \leq N_g \times (\tilde{E}[X] + 1.96\frac{\tilde{\sigma}}{\sqrt{N_g}}). \quad (25)$$

If we call this interval $I_{95\%}$, we can write its explicit mathematical expression:

$$I_{95\%} = [N_g(\tilde{E}[X] - 1.96\frac{\tilde{\sigma}[X]}{\sqrt{N_g}}); N_g(\tilde{E}[X] + 1.96\frac{\tilde{\sigma}[X]}{\sqrt{N_g}})], \quad (26)$$

where the only variable is the number of occurrences N_g, since the expected value $\tilde{E}[X]$ and the standard deviation $\tilde{\sigma}[X]$ have been previously estimated using the Monte Carlo method.

H. Results

In figures 7 through 11, we plot the number of expected successes against the number of trials, for buffer configurations ranging from 0 to 4 incriminating tsumos, considering the upcoming one to three pairs. The red line describes the naive, analytical probabilities while the thin blue line describes the new estimated probability considering the buffer. The yellow band around it shows the 95 % confidence interval, bound by plain black lines.

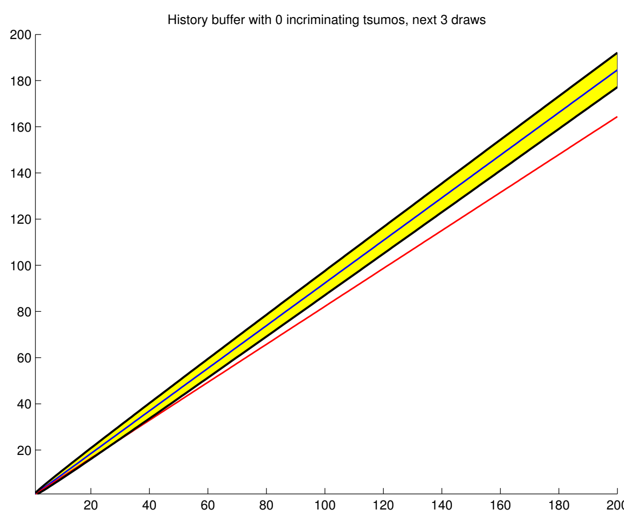

Fig. 7: 95% confidence interval for the upcoming 3 pairs with a buffer with 0 incriminating tsumos

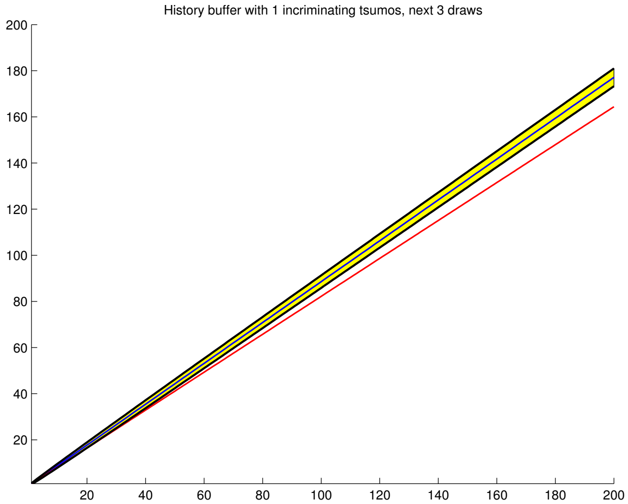

Fig. 8: 95% confidence interval for the upcoming 3 pairs with a buffer with 1 incriminating tsumos

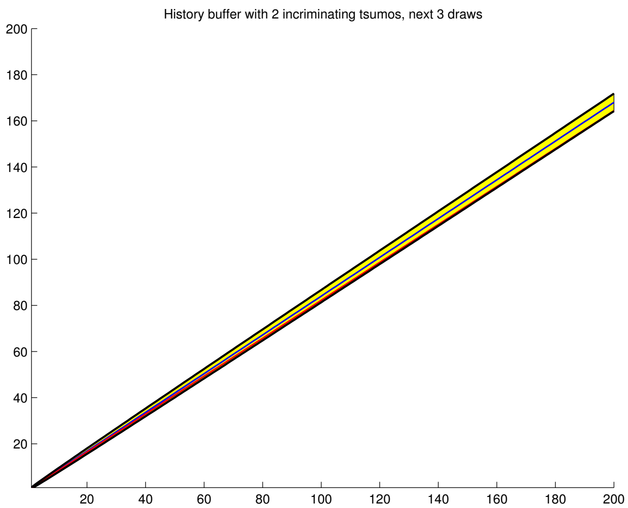

Fig. 9: 95% confidence interval for the upcoming 3 pairs with a buffer with 2 incriminating tsumos

Table VIII provides a summary of, depending on the buffer state and number of draws, how many battles it takes for the analytical naive expectation to fall out of the 95 % confidence interval. This number, N_g^{min}, can be deduced analytically from the confidence interval formula and is given by the following formula when the buffer state and number of draws considered yield estimated actual probability $\tilde{E}[X]$ and naive probability p_t:

$$N_g^{min} = \left\lceil \left(\frac{1.96\tilde{\sigma}}{\tilde{E}[X] - p_t} \right)^2 \right\rceil \tag{27}$$

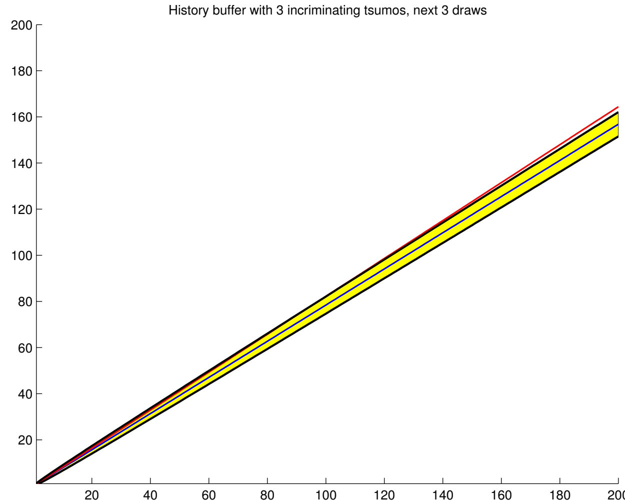

Fig. 10: 95% confidence interval for the upcoming 3 pairs with a buffer with 3 incriminating tsumos

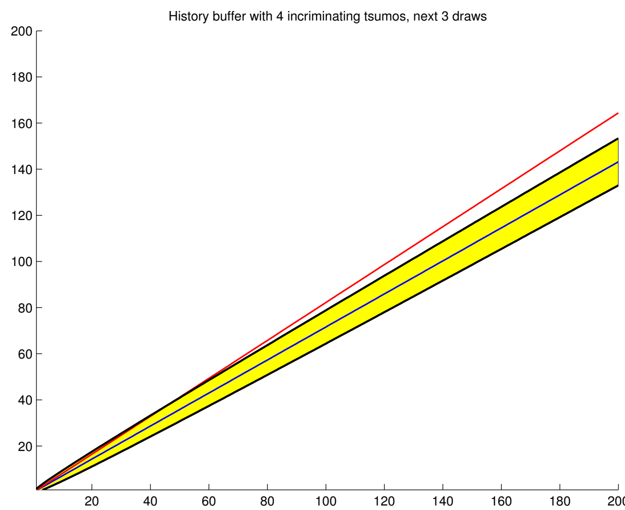

Fig. 11: 95% confidence interval for the upcoming 3 pairs with a buffer with 4 incriminating tsumos

where $x \mapsto \lceil x \rceil$ is the ceiling function, which takes the closest greater integer to its argument. The formula gives precise results when compared to figures 7 through 11.

Incriminating tsumos in buffer	*Draw #1*	*Draw #2*	*Draw #3*
0	51	32	28
1	74	30	19
2	609	3651071	221
3	38	43	97
4	38	33	47

TABLE VIII: Number of occurrences required for the naive expectation to fall out of the 95 % confidence interval, for every case considered

It shows from these results that short-term variability of the process we're simulating can be overcome without too much trouble. The number of games needed in order to be 95 % sure of having better information when making a decision is quite low compared to how long typical Puyo Puyo matches usually last, in most cases. In some other cases, the maximum number of occurrences is too low to see a shift between the analytical probabilities and the confidence interval, namely buffer states with 2 incriminating tsumos. Three things could be done in order to remedy that:

1) plotting the graphs for a larger number of occurrences. It then needs to be determined if the new maximum number is still acceptable as a number of occurrences in a «first-to-100» event;
2) Using a lower confidence level since this makes the interval thinner. Players have to decide how low they are willing to go, but a 90 % confidence level shortens N_g^{min} by quite a lot already;
3) Refining the value of the standard deviation by estimating it from a larger sample (requiring more computational time), hoping a more accurate value yields more satisfying results.

One more important thing to note, from a player's perspective, is that even if the confidence interval shifts below the analytical value, one can actually make better decisions based on that. Indeed, knowing with such confidence that a particular situation gives less chances of success than expected will benefit the player in the long run, as he can act accordingly and go for something more likely to happen. The next section will discuss possible in-game applications of these situations.

III. Case study

From the results compiled in table VIII, it appears to be clearly profitable to consider the results from our study in the context of a « first to 100 » event and even shorter matches. We're not saying exploiting our results will end up in a resounding victory, but will actually help a player make better short-term decisions.

A typical match will last for 100 to 199 battles, roughly averaging around 175 battles. During each of those battles, situations where critical decisions are taken hoping the upcoming pairs will fit a player's short-term strategy can occur more than once, so the number of occurrences in a match can grow to be quite large fairly quickly. Figuring out how often these decisions actually arise in a match depending on the skill level of both competitors is left for discussion but if anything, we might need to plot those graphs for a longer range of occurrences, not necessarily explore the very short-term situations.

In order to exploit these results, a player needs to figure out a way to effectively memorise the buffer state, either at the right time or during the whole battle. He also needs to know when it benefits to act aggressively with such knowledge and when he is in a solid defensive positions because of decreased or increased probabilities. We will assess these three points in the final subsections of this article, keeping in mind that it is up to the community to figure out precise strategies and discuss practical applicability.

A. *How to efficiently infer the history buffer*

Practical use of our results is only possible if players can easily infer the state of the history buffer to then match the situation with actual probabilities.

Considering a 4-slot buffer, one can watch the currently falling pair and the next two announced pairs to deduce 3 out of 4 needed tsumos. The fourth one can either be remembered from the last tsumo dropped, or by waiting for the next one to shift the preview by one.

While remembering the last dropped tsumo is relatively easy for oneself, it is a bit more tricky when observing the opponent in the heat of the battle, but the key point is that three out of four of the needed pairs are already in the mind of every player with a little experience since one needs to watch the upcoming two pairs and the current pair all the time in order to plan chain expansion ahead. In order not to fill his mind with too much extra information, the player should consider keeping a running count of the number of independent tsumos in the buffer containing the trigger color when he is building his chain while keeping the same trigger ready. This happens for example while you are building a tail or while you are adding missing links to your chain in order to make it fully ready.

Another good time to count the independent pairs of a specific color would be when one is not only expanding his main chain but also keeping a short combo on the side for either harass or defense, or to fuse it to the main chain. High probabilities of hitting a trigger is reassuring if the short combo is to be used to counter a harass, while low probabilities might indicate a good time to start the fusion. Deciding whether or not to harass with it however, might need more advanced counting like keeping a count for the opponent's trigger. When this probability is lower than expected, the probability of the harass being successful (essentially winning the game) goes up accordingly. In any case, while the short chain goes off, the player can take the time to figure out the count for his main chain's trigger and figure out in advance what is more likely to come up after the already shown two upcoming pairs and decide whether to expand the main chain to increase his maximum damage output or go for another short harass chain.

B. From an attacker point-of-view

When an attacker has both a long main chain and a short harass ready, he could ask himself how likely he is to be able to hit his main chain's trigger right after having harassed. Let's see how the filtering mechanism affects this likelihood. Consider the following board for the attacker:

Fig. 12: 3-1 stairs chain with 2-chain harrass

The main chain's trigger is the leftmost group of 3 yellow puyos. Assume the last tsumo that was placed on the board is *yellow-blue* which completed that trigger. The three tsumos at the top are the already known upcoming pieces, the currently falling pair being the *red-blue* tsumo on the left. Now, at this instant, the buffer is filled with precisely one tsumo containing the trigger color, which is the worst-case scenorio for this situation because having any other incriminating tsumo in the buffer means that the attacker would obtain his trigger anyway. If the last tsumo placed on the board did not contain any incriminating tsumo, it would be even more likely that the attacker gets his trigger. On the right side of the chain, the groups of red, green and blue puyos are setup to harass the opponent with a 2- or possibly 3-chain. Let's analyze how effective it would be to harass the opponent. The currently falling *red-blue* tsumo is a good candidate for harassing because it allows the attacker to trigger a power 2-chain with both blue and green groups popping on the second hit. In case the harass goes wrong and the opponent quickly counters with a more powerful short chain, the attacker wants to be able to trigger the main chain quickly. Let us consider the following situation. By observing the opponent's board, the attacker sees that the other player can possibly counter the harass with a 4-chain. While the power 2-chain is firing, the opponent will have time to place 2 tsumos on the board. In the same fashion, if the opponent successfully fires his 4-chain, the attacker will then have time to place 2 tsumos to try to trigger the main chain.

Using the naive probabilities, we assume our opponent as a 68.36 % chance of triggering his 4-chain. This means that out of all possible outcomes, the harass is immediately successful 31.64 % of the time (which is a then losing move without a backup plan). When the opponent counters, the attacker has a 68.36 % chance of triggering his main chain as well, which accounts for $0.6836 \times 0.6836 = 0.4673$ (46.73%) of all possible outcomes. The naive probability of success of the harassing strategy is thus $31.64 + 46.73 = 78.37\%$, which is already quite high. Now, considering the present buffer state, we know that the attacker has in fact a 75.6367 % chance of triggering his main chain would the opponent counter the harass. Since we do not know the opponent's buffer state, the probability of him triggering his 4-chain doesn't change. Without information, it is best to rely on that, thus his probability stays at 68.36 %. So, while the harass will still be immediately successful 31.64 % of the time, the attacker will now trigger his main chain $0.6836 \times 0.756367 = 0.5171$ (51.71%) out of all possible outcomes, bringing the global success rate of the harass strategy up to $31.64 + 51.71 = 83.35\%$, which is roughly a 5 points increase even for the worse case scenario.

While the increase in the success rate is not very large here, it shows that this kind of move will always have a success rate higher than naively expected when a filtering algorithm is in place. Also, higher levels of success could be reached by asserting the true probabilities of the opponent hitting his 4-chain trigger by inferring his buffer state and harassing when his chances are lower than he expects. His probability of triggering might be as low as 26.13 %, making the actual success rate

$0.7387 + 0.2613 \times 0.756367 = 0.9363$ (93.63%), this time a roughly a 15 points increase over the naive expectation, the strategy working nearly every single time in this case. This is a good example of how you can assess the effectiveness of a strategy (in advance, not during the game) by quantifying the delays allowed to react and using the probabilities.

An attacker might also want to look only at his opponent's buffer in order to wait for the moment where his opponent is the most vulnerable because of lowered probabilities of defense he isn't aware of. Consider the following board for the opponent:

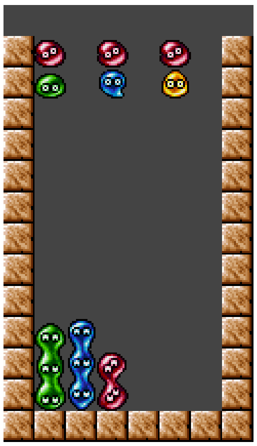

Fig. 13: Power 2-chain in the making

Assume the defender has just placed the *red-red* tsumo and is going for an early power 2-chain here, trying to trigger the chain with the red puyos and popping the blue and green groups at the same time on the second hit. The upcoming pairs for the opponent are presented like in the previous example, the currently falling pair being the *red-green* tsumo on the top left corner. Here the opponent needs to turn that tsumo around and place it on column 1, turn the next one around and place it on column 2, and then stack 2 red puyos on column 3 in order to trigger the power 2-chain. In this case, he needs to hit one more red puyo in the upcoming pairs to make that work. Let's assume the attacker is contemplating triggering a 2-chain of his own, not as powerful but just enough to cover column 3 with nuisance puyos that would prevent the defender from harassing, making him vulnerable to losing to the attacker's main chain which he will then easily have time to trigger. Let us assume as well that while the attacker tries to obtain a trigger for his 2-chain, the defender will have enough time to place 2 extra tsumos in addition to those already announced. If we do not have information about the opponent's buffer, his probability of sending a crippling amount of nuisance puyos to the attacker by triggering his power 2-chain is, like before, 68.36 % which would make that a very profitable move on his part (it clearly seems like a perfect opportunity to harass early). Now, it can easily be spotted by the attacker that the opponent's buffer is filled with 4 incriminating tsumos, that is to say all possible tsumos including a red puyo have just been given to him. We then deduce that his effective probability of hitting a red trigger in the next two pairs is 51.2667 %, which is an astonishing 17 points drop of the success rate. Still, the success rate being over 50 %, one could think that this stays a (although marginally) profitable move on the defender's part. But remember that even though it will be a dire situation for the attacker, the battle probably isn't over after getting hit with such a power 2-chain, and the attacker only has to win roughly 3 % of the time the opponent's harass was not blocked in order to bring the opponent's success rate under 50 %. When a move's success rate is high, one could say that since the probability of a come back from the opponent is very low, it can be assumed to be 0 % and it would be a good approximation. But when the success rate is close to making the strategy a losing move long-term, the element of luck in Puyo Puyo games almost guarantees that a decent opponent will make a come back enough of the time.

This example is more important than it seems, because it could easily be assumed that this kind of power 2-chain early in the game will very often require a long string of puyos of the trigger color, making it, as we have seen, a lot less effective than it appears to be and thus bringing down the overall efficiency of early harassing in PuyoVS2 compared to another Puyo game without a filtering algorithm.

C. *On the defense side*

Assessing the probability of hitting a trigger at a given time by inferring the current buffer is straightforward and is done by reading the correct probability in tables II through VI and evaluating if this is as favorable a situation as it gets or if there is still time to hope for a better spot. The results on the actual expected value of the probability show that basically, a buffer filled with 2 incriminating tsumos gives you probabilities roughly the same as the naive ones. Less incriminating tsumos in the buffer makes it more likely to hit the trigger and more make it less likely. One interesting fact one can deduce from that, is that a trigger group of 3 puyos in a chain is less likely to bet set off right after it has been finished if it has been built by a consecutive string of tsumos containing the trigger color (because the buffer is then filled with incriminating tsumos) than if it has been built progressively using a color that wasn't over-abundant in the recent drawn pairs. It follows that it is actually less vulnerable to build the current trigger of the chain using slower options and not what seems immediately profitable. Essentially, for a given string of upcoming tsumos, if you deplete your luck quickly by hitting the first three puyos of the trigger, the fourth one is unlikely to show up right afterwards, making you more vulnerable to harass than you think while the buffer gets emptied of its incriminating tsumos. This could prove useful when the main chain starts getting very long (over 13 hits when space on the board gets very scarce). 13+ chains always have a very high probability of being more powerful than the opponent's chain (well... against *most* players) because the probability of two players building such a long chain in a given match is very low. It then follows that adding any extra hit to the chain after reaching 13 must be done with a very good reason because it doesn't really increase the chances of winning by that much compared to the risks and vulnerability it creates. Players should consider not going for an extra hit in the already long chain if 3 of the 4 required puyos are given quickly in order not to increase that vulnerability.

It should be noted however, that this is not very relevant if the chain is smaller and you intend to expand your chain from this trigger right away, because then you don't need to place the fourth puyo very soon. Also, strings of exactly the same tsumo are not dangerous to build a trigger that you want to possibly set off right away because it gives a buffer filled with only 1 incriminating tsumo, which is still profitable. This particular case is not very frequent but still happens and is thus shown to be very efficient since two colors can then be used at the same time to expand while keeping higher-than-normal probabilities of hitting either one. A situation where this is of interest is when building the transition:

Fig. 14: Sandwich chain without transition and 2-colored string of upcoming tsumos

The upcoming tsumos on these boards are put above the chain and must be read from top to bottom, left to right. Here a long string of the same tsumo is given to the player. A typical transition to build with it would be:

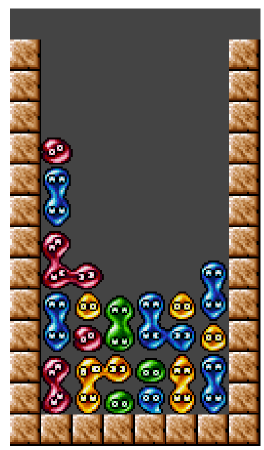

Fig. 15: 2-colored safe transition

Using two colors only to build the transition, one of them being the same as the first hit of the base is very safe because it is hard for the probability of those two colors to go down because of the history buffer. In fact, at the end of that string, the probability of hitting a blue puyo is the same as the probability of hitting a red puyo (trigger): 49.57 % in one draw, 75.64 % in two draws and 88.54 % in three draws, which is roughly 5.8 %, 7.3 % and 6.3 % more than the naive expectation, respectively, so safer overall. It also gives very good probabilities to continue expanding on the red trigger with blue puyos. It should be noted however that such strings are more likely to happen with permutations of *red-blue* and *blue-*

red tsumos, thus which rather gives a buffer filled with 2 incriminating puyos which is a rather neutral situation considering all cases.

On the contrary, it can also be shown that when confronted with a 3- or 4-colored tsumo string, some unsafe situations arise:

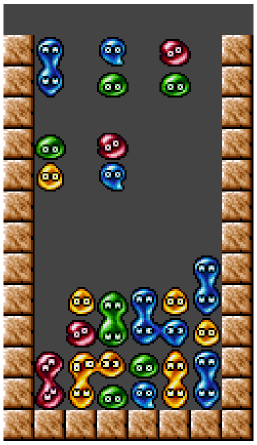

Fig. 16: Sandwich chain without transition and 4-colored string of upcoming tsumos

We propose the following transition involving a green trigger:

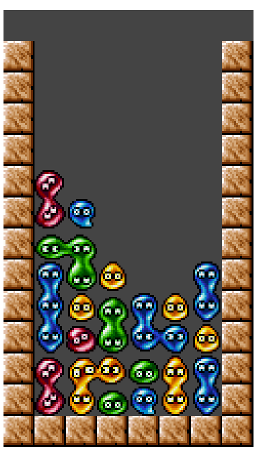

Fig. 17: Unsafe 3-colored transition

At the end of that string, the buffer is left with 3 independant tsumos including a green puyo, giving probabilities of hitting a green trigger in the next 1, 2 or 3 draws of 34.04 %, 60.44 % and 78.45 % respectively, which is roughly 10 %, 8 % and 4 % less than the naive expectation, so less safe overall.

We propose that it might be preferable to only take a 2-colored part of the string to build a smaller, but safer transition:

Fig. 18: Safe chaining with small transition

which, here, leaves the buffer with only 2 independent blue tsumos (neutral in terms of probabilities), and plan to complexify the transition later on (which is easy to do when only two colors have been involved so far):

Fig. 19: Expansion over safe chaining

However, this shouldn't restrict the player from attempting an unsafe transition. Vulnerable spots will inevitably show up in a battle as some simply come from the buffer state, as shown, over which the player has no control at all. It just boils down to whether or not one is ready to make a risky situation even riskier knowing the opponent's harassing tendency. In the same fashion as we did in subsection III-B, we could calculate how much a given transition building strategy is vulnerable to harass and see how close it is to being a losing move (if it is not already) when the opponent tries to harass during the process. If the strategy is a long-term losing one, then the player pretty much counts on the fact that his opponent is not going to harass him while the transition is being built, so a case could be made that the player might as well go for as complicated a transition as he can.

As a last point, we want to address a common defense situation in Puyo Puyo games which is when the attacker has already started his long, main chain and the defender has to catch up in terms of main chain power, while keeping an eye on how close the opponent's chain is to finish popping. It is very tempting in that case to "waste" trigger tsumos to build extra hits in advance. What one has to take into account in the light of our study is that these "wasted" trigger puyos will soon become the new trigger of the chain or not, but the trigger will come back to being of the same color. As the opponent's chain is almost gone, the player should avoid having wasted too much trigger puyos on extra hits because then the probability of hitting the final trigger puyo will be too low and make it unlikely that he will be able to counter. So we suggest that near the end of the opponent's chain (with a certain number of hits left which is to be determined, depending on the player's own speed and confidence), one should only continue expanding the chain on a relatively empty buffer with regards to the final trigger color, and rather start the chain a little early if the buffer is filled or about to be.

IV. Conclusion

In this article we presented a statistical and probabilistic analysis of the RNG mechanics of the Puyo Puyo fan game PuyoVS2, showing how its specificities, while transparent to a player, can be exploited to one's advantage with some extra memorisation effort. Even though, as stated at the beginning, the exact parameters of the randomizer used in the software are not known, we showed that the possible improvements in terms of strategy and decision making could be very significant, while at the same time giving the reader a well-needed course in math and showing off our pdf-making skills. We wrote this paper in a close way to what an actual scientific publication would look like because this is what we do and we felt like huge nerds looking into stuff like that. The requirements for an actual in-game use of the results of this study are being Japanese *AND* knowing the 42^d decimal of π but as shown in the last section, the efficiency of relatively general situations can also be evaluated on paper before a match in order to determine if a line of thinking is effective or not. This is done in chess, backgammon or even versus fighting games, but never in Puyo Puyo as far as we know, even though the player pool in our case seems to be almost exclusively constituted of smart people. We just propose a way this could be done with more accurate information on the randomness of the game but it is up to the community to take this kind of reasoning and use it to make Puyo Puyo a more strategically interesting game than it is already. And in the end, never forget that:

There is no sense crying over every mistake, you just keep on trying 'till you run out of cake !