Optimizing RTP in Slot Machines While Preserving Reel Characteristics

Ashley Del Sesto
Department of Computer Science & Engineering
University of Nevada, Reno
Reno, Nevada
adelsesto@nevada.unr.edu

***Abstract*—This research presents a Genetic Algorithm (GA) approach for finding solutions to the multicriteria problem of optimizing Return-To-Player (RTP), symbol diversity, and bonus game hit frequency. Many slot machines incorporate reel characteristics (e.g., stacked wild symbols) as features in the base game to enhance player experience. By utilizing a stack-centric framework for our GA, this research demonstrates how to create several slot machine base game math models that yield favorable values for the listed criteria. Furthermore, this work uses a methodology not previously used in this field of GA study to promptly yield the exact RTP for a base game.**

***Index Terms*—genetic algorithm, evolutionary computing, slot machine, RTP, reels**

I. Introduction

Slot machines, or electronic gaming machines, are an iconic and lucrative product sector in the United States' gaming industry. These devices use reels, arrays of symbols that "spin inside the slot machine window" [1]. Each entry in the reel array corresponds to a symbol from the game's symbol list and a reel stopping position, or "reel stop" [1]. The symbol list S is a list of all symbols present on at least one reel. A slot machine contains at least one reel, usually containing three or five adjacent reels. The reel window displays a portion of each reel to the player. This window is hereby denoted as $m \times n$, where m is the reel window height, the number of symbols displayed vertically in the reel window; n is the number of reels. The reel window is the basis for most awards from a slot machine. Symbol patterns, otherwise known as pay combos, award credits or other prizes if the reel window contains said symbol pattern on viable positions specified by the slot machine. The symbol patterns for positions specified by line patterns are listed in P_l. The symbol patterns for positions specified by scatter patterns are listed in P_s. For each reel, a random number generator (RNG) is used to determine which reel stop is selected to display in the reel window [1].

The U.S. commercial gaming industry saw significant growth in 2023 with Q1 and Q3 being the two all-time highest grossing quarters for the industry [2]. This increase in demand for both land-based and iGaming products coincides with an increase in game complexity. Though having "developers innovating all the time" [3] is indicative of a healthy industry, slot game mathematicians may exercise caution in where they choose to innovate on their game design ideas. A slot game mathematician developing a game may adhere to preliminary benchmarks that the slot machine model must meet such as Return-to-Player (RTP) and bonus game hit frequency. RTP is the average percentage of a wager that a slot machine will award to a player.

$$\text{RTP} = \frac{\text{Credits Awarded}}{\text{Credits Wagered}} \tag{1}$$

Many slot machines include a bonus game where a player can win additional credits or prizes. The bonus game is a triggered event and may result from specific symbols being visible in the reel window when the reel stop; three or more of these bonus symbols is a common triggering event. The bonus game hit frequency is the average number of wagers a player will make before a bonus game is awarded.

Furthermore, a game may be prescribed reel characteristics such as stacked wilds or displaying at most one bonus symbol per reel on any given reel stop. Symbol stacking is terminology used to describe the consecutive ordering of a given symbol. Thus the phrase stacked wilds describes the consecutive ordering of wilds, symbols that substitute for many if not all symbols in the symbol list.

This research is motivated by previous work incorporating Genetic Algorithms (GAs) along with firsthand experience in developing slot machine game math models. If a slot machine's base game can be created by an algorithm, then a slot game mathematician can spend more time innovating on other game elements such as a bonus game. This work seeks to expand the use of GAs in slot machine math development by reducing the duration necessary to evaluate RTP and incorporate desired reel characteristics in an individual's chromosomal representation. RTP, symbol diversity, and bonus game hit frequency are optimized for the base game of a slot machine. The code used for this research can be found in the cited GitHub Repository [4].

The structure of this paper is as follows: Section II summarizes work related to slot machine RTP optimization and evolutionary computing. Section III presents the proposed framework for the GA. The particular slot machine game math model used is presented in Section IV. Section V details the results from this experiment. This work concludes in Section VI with summarized thoughts and prompts for continued research.

II. Prior Work

A. Solution Representation

In 2014, Balabanov, Zankinski, and Shumanov published the first paper related to evolutionary computing and slot machine RTP optimization [5]. These authors developed a 5-reel slot machine with 63 reel stops per reel and used a 3×5 reel window. They used a two-dimensional (2D) chromosome to represent this slot machine; each one-dimensional (1D) array in the 2D chromosome was mapped to a slot machine reel and each reel position mapped to a symbol in the symbol list. Thus the chromosome contained 315 values.

Each other paper mentioned in this section uses this representation for their solution space.

B. Genetic Algorithms

Reference [5] implemented a GA to optimize RTP for a slot machine model. The authors initialized each individual in this math model with a chromosome that yielded 90.88% RTP. Their solution targeted 99% RTP. They ran $100,000$ or $1,000,000$ Monte Carlo simulations to approximate RTP for each individual in each generation; they concluded that this method was successful yet time-consuming [5]. In 2021, Kamanas, Sifaleras, and Samaras identified an issue with this approach [6]. Slot game mathematicians do not always start with a pre-existing math model from which to adjust RTP. Thus the assumption Balabanov, Zankinski, and Shumanov made is not applicable to certain use cases.

In 2017, Keremedchiev, Tomov, and Barova revisited the GA approach to optimizing RTP but implemented an exact approach for calculating RTP called a full cycle calculation [7]. This full cycle calculation, though it yields the exact RTP for a slot machine base game, only calculates values for the base game, omitting RTP for any features or bonus games that may be triggered from the base game. The slot machine model in their research again used 5 reels with 63 reel stops per reel and used a 3×5 reel window. For this research Keremedchiev, Tomov, and Barova targeted 90% RTP for the base game RTP.

Though the authors stated that 90% RTP is the "lowest legal value for the Bulgarian gambling market" [7], this research may have produced more applicable results had they targeted a lower RTP. The slot machine model in this research included a bonus game trigger. RTP must be allocated for credits or prizes awarded in the bonus game, so targeting a lower value for base game RTP would free up RTP that slot game mathematician can instead allocate to the bonus game. While this approach resulted in faster RTP convergence than methods using Monte Carlo simulations, the full cycle calculation took more time to evaluate the objective function for each individual. Additionally, these authors optimized for RTP with slight consideration for bonus game hit frequencies for their objective function.

C. Evolutionary Computing

In 2015, Balabanov, Zankinski, and Shumanov applied a linear transformation to three criteria to create a single-objective Discrete Differential Evolution method [8]. The three criteria were RTP, prizes equalization, and symbol diversity. Prizes equalization is the concept of having each symbol equally contribute to the RTP; credit yield discrepancies are offset by symbol pattern hit frequency. Symbol diversity, as defined in [8], is "a parameter related to how many symbols of the same kind are next to each other in a single reel". Like the previous paper, this paper utilizes a 5-reel slot machine with 63 reel stops per reel and a 3×5 window. Their work implemented Monte Carlo simulations for approximating RTP and prizes equalization to be used in the objective function for each individual. Each individual ran ten sets of $1,000,000$ Monte Carlo simulations for a total of $10,000,000$ simulations to achieve better accuracy [8]. The authors again concluded that this method using Monte Carlo Simulations to approximate RTP was time-consuming.

In 2021, Kamanas, Sifaleras, and Samaras implemented Variable Neighborhood Search, or VNS, to attempt to optimize RTP [6]. These authors examined three popular iGaming slot games and explored constructing reel sets for these games via VNS. Each slot game model uses 5 reels and has a 3×5 window. The authors used $100,000$ or $1,000,000$ Monte Carlo simulations to approximate RTP for each individual. While Kamanas, Sifaleras, and Samaras successfully reduced RTP for each of these three games to yield reels that can be realistically implemented in a slot machine game math model, they concluded that there exists a need to "speed up the whole process" of approximating or calculating RTP[6].

III. Method

A. Base Game

This work includes an alternative method to calculating RTP for a slot machine base game. Thus game attributes such as full game RTP and full game hit frequency are not calculated nor approximated to via this method.

B. Chromosomal Representation

Previous research in this field has focused on generating slot machine reels that lack defining characteristics, opting for an entirely randomized ordering of symbols. While Balabanov, Zankinski, and Shumanov included symbol diversity as one of their criteria in [8], their methods did not consider the possibility for a game to require identical symbols located adjacently on a given reel. Symbol stacking is an incredibly common reel characteristic or feature in slot machines. This feature manifests in multiple ways: only one symbol such as a wild symbol or the highest-paying symbol can appear as a stack, or multiple symbols can appear as a stack. These stacks may additionally be restricted to specific reels.

This research primarily focuses on including, restricting, and monitoring symbol stacks in reels for a slot machine base game. To adhere to any stacking requirements, this work deviates from previous research in this field by changing the chromosomal representation. Reels are re-contextualized as concatenated segments, or stacks, of symbols; a segment may contain between 0 and X symbols where X is a non-zero integer determined by the slot game mathematician. Each

(1, 3)

TABLE I: A tuple depicting a stack of symbol 1 with length 3

1
1
1

TABLE II: A stack of symbol 1 with length 3

segment with two or more symbols must contain identical symbols.

This chromosome still uses a 2D array with a 1D array per reel, however each position in each 1D array is a tuple that represents a segment; the first element is the symbol and the second element is the segment length. Table I contains an example of a tuple entry and Table II shows the representation of said tuple as part of a reel. This methodology results in a set of reels that may vary in length. Though previous research used the same number of symbols per reel, it is not a requirement for several slot machine products.

Each tuple in the chromosome is initialized with a random symbol from symbol list S and a randomized, non-negative stack length with bounds determined by the slot game mathematician.

C. Chromosome Adjustments

Initializing and altering a chromosome may result in reels that contradict the reel characteristics prescribed for a game. Certain deterministic adjustments are performed upon a chromosome to yield valid reels. For instance, if a tuple depicts a stack of two or more bonus symbols, then the stack length, or the second element of said tuple, is overwritten with the a value of one. Other adjustments may include:

- Reducing the number of symbols in a stack to the symbol's prescribed maximum.
- Increasing the number of symbols in a stack to the symbol's prescribed minimum.
- Interpreting adjacent tuples with identical symbols as one stack.
- Setting the number of bonus symbols in a stack to zero if the stack appears in the same reel window as another bonus symbol.

These adjustments may force a tuple to have its stack length, or second element, set to zero. Setting each reel to have a large number of segments remedies possible drawbacks resulting from tuples with a stack length of zero. The adjustment strategy must be completely deterministic; when presented with two identical chromosomes, the chromosomes post-adjustment must also be identical.

D. Chromosome to Reels

After processing and applying any adjustments, the chromosome is temporarily interpreted as a 2D array as input for evaluating the objective functions. Each 1D array is interpreted as a reel populated by the tuples in ascending order relative to the array indices. Each tuple (X, Y) places symbol X into the array Y times; if Y is zero, then the tuple is ignored.

E. Objective Functions

This GA utilizes three objective functions for the following criteria: RTP, symbol diversity, and bonus frequency. Note that the equations referenced in these objective functions use symbol list S, symbol patterns for lines list P_l, symbol patterns for scatters list P_s, window reel height m, and number of reels n. Let ANY be a symbol that represent every $s \in S$. Let $B \subset S$ such that B contains all bonus symbols.

Let $p_{(s,r)}$ represent a winning pattern in the pay table where r reel stops on valid pattern positions contain symbol s. Depending on the context, $p_{(s,r)} \in P_l$ or $p_{(s,r)} \in P_s$.

Let $\text{Pay}_{(s,r)}$ represent the credit award or prize associated with pattern $p_{(s,r)}$; this value defaults to zero if the pay table does not contain prescribe credit award or prize for it. Note that Let the function $\text{Freq}(s, r)$ calculate the frequency of symbol s on reel r; this value defaults to zero s is zero.

Let the function $\text{Unique}(l, r, m)$ count the number of unique symbols on reel r starting at reel stop l, ascending relative to indices to observe m symbol positions. If $\text{Freq}(ANY, r) - (l + 1) < m$, then the remaining positions needed to count m symbols start at position 0.

Let C denote the cycle, the number of combinations for landing all possible reel stops on each reel; C is calculated as

$$C = \prod_{r=1}^{n} \text{Freq}(ANY, r). \tag{2}$$

The objective function for RTP is problem-dependent, based on the pay combos and the presence of any substitution symbols such as wilds. Instead of using a Monte Carlo simulation to approximate RTP or a full cycle calculation found in [7], this work utilizes a different approach to calculate exact RTP.

The two lists S_I and S_E are created to better represent the symbol patterns. For each $s \in S$, list S_I contains an inclusion symbol IN_s, a symbol representing a set of symbols that are accepted in place of symbol s. That is, if symbol 1 represents a wild symbol that substitutes for symbol 3, then symbol IN_3 indicates that either symbols 1 or 3 can be accepted in the pattern at the specified position. The frequency for IN_s is calculated as

$$\text{Freq}(IN_s, r) = \sum_{i \in IN_s} \text{Freq}(i, r) \ , \tag{3}$$

a summation of the frequencies for each symbol represented by IN_s.

For each $s \in S$, list S_E contains an exclusion symbol EX_s, a symbol representing a set of symbols that are not accepted in place of symbol s. It follows that $IN_s \bigcup EX_s = S$. The frequency for EX_s, similar to IN_s, is a summation of frequencies for each symbol represented by EX_s.

Each $p \in P_l$ is rewritten in terms of IN_s, EX_s, and ANY. For instance, the symbol pattern 3 of-a-kind (OAK) of symbol 4, $p_{4,3}$, is interpreted as

$$p_{4,3} = IN_4 \ IN_4 \ IN_4 \ EX_4 \ ANY \tag{4}$$

since reel 4 cannot have symbols 4 nor 1 in the line position; otherwise the pattern would evaluate to 4 OAK of symbol 4.

Depending on the pay amount for each symbol pattern, additional considerations may be required to calculate the exact RTP. The pay table developed for this research illustrated in Table V is set up such that the per line RTP is calculated as

$$\text{RTP} = \frac{\sum_{s\in S}\sum_{k=1}^{K}(\prod_{r=1}^{n}\text{Freq}(p_{(s,k)}[r-1],r))\times\text{Pay}_{(s,k)}}{C} \tag{5}$$

where K is the number of IN_s symbols in the pattern and $p_{(s,k)}[r-1]$ indexes the symbol used on reel r in pattern $p_{(s,k)}$. Since each reel stop on a reel has equal probability of landing, the number of pay lines is negligible and irrelevant for calculating RTP. For example, the RTP for 1 line is multiplied by 50, divided by the wagered amount, and multiplied by the original wager to expand the number of lines to 50. That said, for this research all symbol patterns on lines are paid from left to right, each symbol appearing on adjacent reels starting from the leftmost reel.

This work interprets symbol diversity as the percentage of identical symbols that appear in the same reel window in a single reel, deviating from the description in [8]. Since this work focuses on symbol stacks, cases such as *X Y X* appearing on the reels warrant observation. The objective function for symbol diversity, abbreviated as SymDiv, is calculated as

$$\text{SymDiv} = \frac{\sum_{r=1}^{n}\frac{\sum_{l=1}^{\text{Freq}(ANY,r)}\text{Unique}(l,r,m)}{\text{Freq}(ANY,r)\times m}}{n} \tag{6}$$

The objective function for bonus game hit frequency is problem-dependent, based on the number of bonus symbols and how many symbols are required to trigger bonuses. The pay table developed for this research contains one bonus symbol that uses scatter patterns to evaluate upon. It is a trivial matter of combinatorics to obtain the combinations of reels that stop with a bonus symbol in the reel window for three, four, or five bonus symbols. Each of these combinations, $p \in P_s$, is rewritten in terms of $Scatter\_IN_b$ and $Scatter\_EX_b$ where $b \in B$ is the bonus symbol. This work restrains each reel such that each reel can only contain up to one bonus symbol in the reel window. Thus the frequencies for $Scatter\_IN_b$ and $Scatter\_EX_b$ are easily calculated as

$$\text{Freq}(Scatter\_IN_b, r) = \text{Freq}(IN_b, r)\times m \tag{7}$$

$$\begin{aligned}\text{Freq}(Scatter\_EX_b, r) = {} & \text{Freq}(ANY, r) \\ & - \text{Freq}(Scatter\_IN_b, r)\end{aligned} \tag{8}$$

since each appearance of b is independent from one another. Due to our aforementioned restrictions, the following calculation shall suffice for bonus game hit frequency, abbreviated to BFreq.

$$\text{BFreq} = \frac{\sum_{b\in B}\sum_{k=1}^{K}\sum_{p\in P_s}\prod_{r=1}^{n}\text{Freq}(p_{(b,k)}[r-1],r)}{C} \tag{9}$$

F. Fitness Function

The Fitness function used in this research is

$$\begin{aligned}\text{Fitness} = 100 & - \min(10\times|\frac{\text{tRTP}-\text{RTP}}{\text{tRTP}}|, 6)^2 \\ & - 28\times\min(2\times|\frac{\text{tSymDiv}-\text{SymDiv}}{\text{tSymDiv}}|, 1) \\ & - \min(2\times|\frac{\text{tBFreq}-\text{BFreq}}{\text{tBFreq}}|, 6)^2,\end{aligned} \tag{10}$$

where tRTP is the targeted RTP, tSymDiv is the targeted symbol diversity, and tBFreq is the targeted bonus game hit frequency. It follows that this function yields zero if each these conditions are met:

- RTP $\geq 1.6\times$ tRTP.
- SymDiv $\geq 1.5\times$ tSymDiv.
- BFreq $\geq 4\times$ tBFreq.

These constraints disincentivize large differences in values that are difficult for the GA to learn from. In practice, slot game mathematicians avoid resolving similarly large differences, if possible. For example, it is easier to reduce 90% RTP to 60% than it is to reduce 200% RTP to 80%.

G. GA Operators

This research utilizes tournament selection, uniform crossover, and three mutation operators. The tournament selection process first selects one parent from a binary tournament of individuals selected via roulette wheel selection [9]. The selected parent is removed from the larger pool of individuals and a second parent is selected in the same manner. Two child individuals are created. If crossover is selected to occur for these two individuals, for each tuple in each array in the chromosome, they "compete" to determine which of them inherits said tuple from the first parent. The "loser" inherits the tuple in the same position from the second parent.

Mutation may occur in two separate manners independent of each other. A chromosome may first mutate via swapping one tuple with another tuple located within the chromosome. The probability for this mutation is labeled "Swap mutation rate" in Table III.

A second level of mutation is determined on a tuple-by-tuple basis for the individual's chromosome. A tuple can mutate in one of two ways with equal likelihood:

- The symbol assigned to the tuple (first element) shifts to the symbol directly ahead of it or behind it in symbol list S with modulo $|S|$ applied to prevent invalid indices. The two directions for shifting have equal probability.
- The stack length assigned to the tuple (second element) increments or decrements by one with modulo applied to prevent negative or exceedingly-long stack lengths. This modulo value is based on the assigned symbol's length parameters. Incrementing and decrementing the stack length have equal probability.

The probability for this mutation is labeled "Tuple mutation rate" in Table III. The four individuals, two parents and two children, are evaluated and the two individuals with the highest fitness are set aside in a pool of individuals. This process continues until no individuals are left in the initial pool, and the generation ends. The next generation starts by selecting two parents from the newer large pool of individuals, and the process repeats until the maximum number of generations has been met.

TABLE III: GA Parameters

Parameter	Value
Generation gap	1.0
Crossover rate	0.9
Swap mutation rate	0.1
Tuple mutation rate	0.01
Maximum generations	100
Number of individuals	50
Number of variables	2× number of segments[a]

[a]2 variables per tuple.

IV. Experimental Design

All experiments were performed using an Acer Nitro AN515-53 laptop with an Intel Core i5-8300U CPU at 2.30GHz, 8GB DDR4 RAM at 2304 MHz, and running 64-bit Microsoft Windows 10 Home with Windows Subsystem for Linux. The experiments used an original slot machine math model created solely for research purposes.

TABLE IV: Pay table features

Name	Feature
Target RTP (tRTP)	0.70
Target symbol diversity (tSymDiv)	0.45
Target bonus game hit frequency (tBFreq)	0.008
Number of reels (n)	5
Reel window height (m)	3
Reel window dimensions $(m \times n)$	3×5
Number of segments per reel	100
Symbol list S	$1 - 11$
Wild symbols	1
Symbols that are wild symbols	1
Symbols that wild symbols can substitute for	$2 - 10$
Bonus symbols	11
Minimum number of Bonus symbols to trigger bonus	3
Maximum number of Bonus symbols to trigger bonus	5
Minimum stack length for symbol 1	3
Maximum stack length for symbol 1	5
Minimum stack length for symbols 2-10	1
Maximum stack length for symbols 2-10	2
Minimum stack length for symbol 11	1
Maximum stack length for symbols 11	1

Table IV contains the features specific to the slot machine game math model used in these experiments. The symbol pay table for the line patterns is presented in Table V and symbol pay table for the scatter patterns is presented in Table VI. This base game math model targets 70% RTP as to create a surplus in RTP that the bonus game can leverage; it is assumed that the encompassing math model targets a legal RTP value. The target symbol diversity is 45% to counteract the large influence the chromosomal representation has on stacking symbols. The target bonus game hit frequency is 0.8%, or 1-in-125 games. This frequency is typically seen in some capacity in video slot machine products, hence it is appropriate for this experiment.

TABLE V: Line Pattern Symbol Pay Table

Symbol Name	Symbol Count				
	1	2	3	4	5
1	0	0[a]	0[a]	0[a]	0[a]
2	0	5	20	50	200
3	0	0	15	45	100
4	0	0	15	45	100
5	0	0	10	30	75
6	0	0	10	30	75
7	0	0	5	20	50
8	0	0	5	20	50
9	0	0	5	20	50
10	0	0	5	20	50

[a]Pays the amount for symbol 2.

TABLE VI: Scatter Pattern Symbol Pay Table

Symbol Name	Symbol Count				
	1	2	3	4	5
11	0	0	Bonus Trigger		

Each reel contains 100 tuples resulting in 500 total tuples and $1,000$ modifiable values. Each chromosome is initialized with randomly-selected values for each tuple. The assigned symbol is selected from the symbol list S with replacement, and the assigned segment length is selected via roulette wheel selection with replacement from Table VII. Once these selections are made for each tuple in the chromosome, the chromosome is adjusted using methods from III-C and the GA begins once the number of individuals prescribed by the GA's parameters in Table III have been created.

TABLE VII: Segment Distribution Table

Segment Length	Probability
1	0.8
2	0.2

V. Results and Analysis

This experiment performed 30 randomly-seeded runs of the GA. The GA calculated the exact RTP in a sufficient amount of time since the generations progressed in a timely fashion. The efficiency of this calculation is evident in the video recording of the GA performing one run [10].

In addition, this pay table was also ran on a hill climber (HC) variant of the GA program where the crossover probability was set to zero. Figures 1 - 8 illustrate the differences in performance between the GA and the HC; these reported statistics averaged the mean and maximum values at each generation between the 30 runs. The HC seemed to disregard targeting 70% RTP altogether as that value trended towards 480%. The GA meaningfully optimized all three criteria with each run's best individual ending within 1% of the optimum fitness.

One observation of note is that each criterion converged with a different number of generations completed. Though the averages for the Average and Maximum values for each criterion tend to close in at approximately 80 generations, the symbol diversity and bonus frequency values are relatively distant given their small target values.

The best set of reels from each GA show interesting results. From the 30 GA runs performed, only 16 of them included at least 1 wild stack and at least 1 bonus symbol on every reel. Since this GA did not penalize an individual for omitting a given symbol from a reel, those results are hardly surprising. The interesting details are instead found in the reels that are sometimes omitting particular symbols.

TABLE VIII: Best GA Individual Reel Omissions Table

Reel	Wild Stack	Bonus Symbol
1	2	0
2	2	1
3	3	1
4	1	1
5	2	2

Though 30 runs is an awfully small sample size when looking at the best individual from each run, Table VIII presents enlightening statistics. Judging from these numbers, the GA is more likely to omit wild stacks from reels $1-3$ instead of reels 4 and 5. This behavior leans toward a sound methodology for reducing RTP in a slot machine base game or free games bonus. There exist several slot machines that feature wild stacks only on reels $2-5$, omitting reel 1 altogether. Though those games still include wild stacks on reels 2 and 3, their presence is muted, creating a "pinch point" in the game where big wins are largely teased and seldom awarded.

Furthermore, the GA is more likely to leave off bonus symbols from reels 4 and 5 than reels $1-3$ for a similar reason. Though a bonus game triggers with 3 or more bonus symbols, nothing happens when only 2 bonus symbols are present in the reel window when the reels stop. Thus bonus symbols are considered to be "blockers" when they appear on the leftmost reels, prohibiting a majority line pattern pay wins from forming. The GA, like slot game mathematicians, is capable of capitalizing on the "blocking" nature of bonus symbols, a nuanced decision that a portion of slot machine players are oblivious to.

The source files for these results may be found in [4] for further analysis.

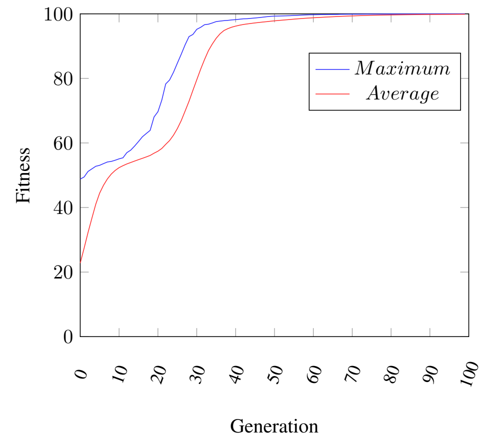

Fig. 1: Fitness Graph for GA

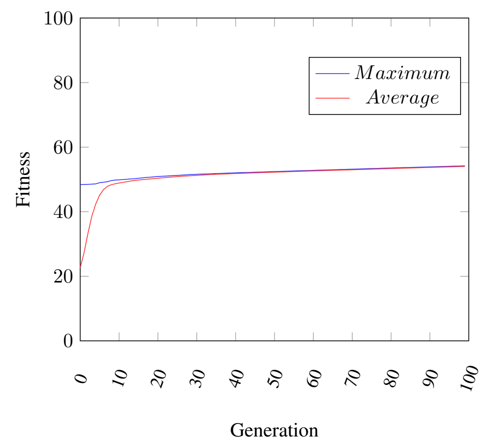

Fig. 2: Fitness Graph for HC

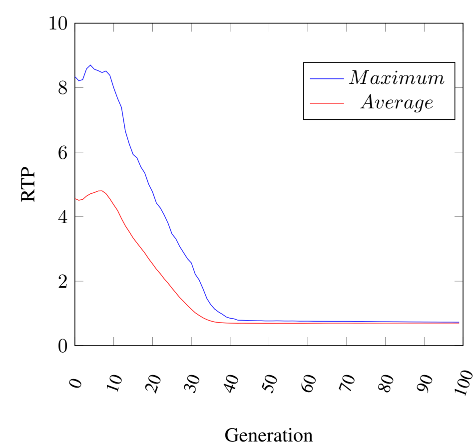

Fig. 3: RTP Graph for GA

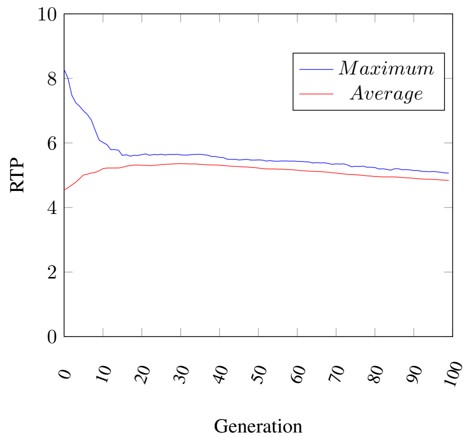

Fig. 4: RTP Graph for HC

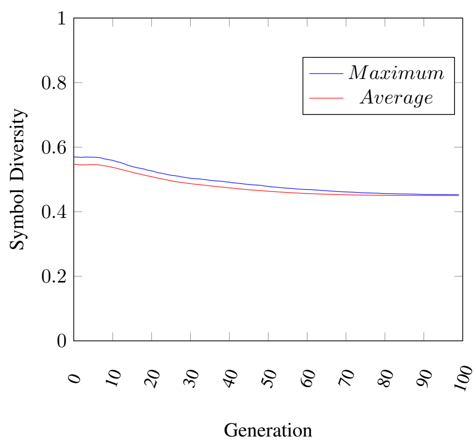

Fig. 5: Symbol Diversity Graph for GA

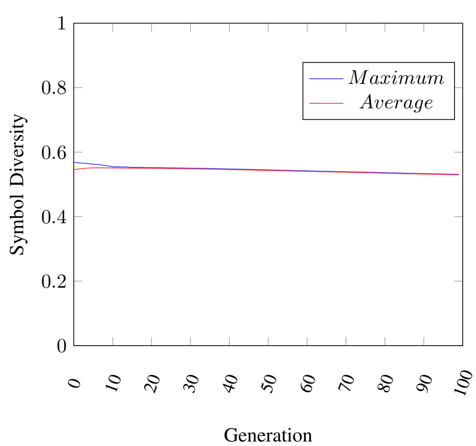

Fig. 6: Symbol Diversity Graph for HC

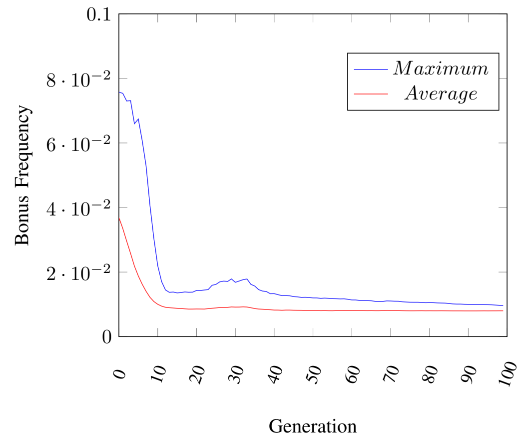

Fig. 7: Bonus Frequency Graph for GA

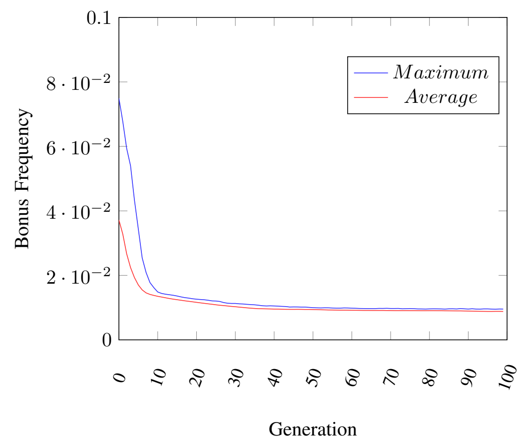

Fig. 8: Bonus Frequency Graph for HC

VI. Conclusions and Future Work

These experiments performed on both the Hill Climber and the Genetic Algorithm demonstrate the latter's aptitude in optimizing RTP, symbol diversity, and bonus game hit frequency. The method used for calculating the exact RTP of the base game seems vastly more efficient than the full cycle calculation used in [7] and $1,000,000$ Monte Carlo simulations per individual per generation per GA run. Through focusing on reel characteristics via the chromosomal representation, the GA created realistic slot machine game math models and even identified mathematical traits relevant to slot game mathematicians.

Developing a method specific to this GA to ensure that each symbol appears on reels determined by the user would serve as valuable future work. Additionally, it is worth exploring the implementation of more complex base game features for this GA to search for the limits of what it can do. This research will also benefit from installing a way for a slot game mathematician to input their pay table and exact RTP calculation function. This work involves several hyperparameters so the code in [4] must be hand-edited for custom use.

REFERENCES

[1] *Slots dictionary: Definitions of slot machine related terminology & lingo*, (accessed Dec. 16, 2023), Jul. 2018. [Online]. Available: https://www.worldcasinodirectory.com/articles/slots-dictionary.

[2] *Q3 2023 commercial gaming revenue reaches $16.17b, marking industry's highest-grossing third quarter*, (accessed Dec. 16, 2023), Nov. 2023. [Online]. Available: https://www.americangaming.org/new/q32023cgrt/.

[3] N. Baldo, *The increasing complexity of slots*, (accessed Dec. 16, 2023), Dec. 2022. [Online]. Available: https://texasholdemquestions.com/complexity-of-slots/.

[4] A. Del Sesto, *Slotmachinegareelcharacteristics*, (accessed Dec. 17, 2023), 2023. [Online]. Available: https://github.com/delsesto-ashley/SlotMachineGAReelCharacteristics.

[5] T. Balabanov, I. Zankinski, and B. Shumanov, "Slot machines rtp optimization with genetic algorithms," in *Numerical Methods and Applications: 8th International Conference, NMA 2014, Borovets, Bulgaria, August 20-24, 2014, Revised Selected Papers 8*, Springer, 2015, pp. 55–61.

[6] P.-A. Kamanas, A. Sifaleras, and N. Samaras, "Slot machine rtp optimization using variable neighborhood search," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–8, 2021.

[7] D. Keremedchiev, P. Tomov, and M. Barova, "Slot machine base game evolutionary rtp optimization," in *Numerical Analysis and Its Applications: 6th International Conference, NAA 2016, Lozenetz, Bulgaria, June 15-22, 2016, Revised Selected Papers 6*, Springer, 2017, pp. 406–413.

[8] T. Balabanov, I. Zankinski, and B. Shumanov, "Slot machine rtp optimization and symbols wins equalization with discrete differential evolution," in *Large-Scale Scientific Computing: 10th International Conference, LSSC 2015, Sozopol, Bulgaria, June 8-12, 2015. Revised Selected Papers 10*, Springer, 2015, pp. 210–217.

[9] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Syst.*, vol. 9, 1995. [Online]. Available: https://api.semanticscholar.org/CorpusID:6491320.

[10] A. Del Sesto, *Ga run recording*, (accessed Dec. 17, 2023), 2023. [Online]. Available: https://delsesto-ashley.github.io/SlotMachineGAReelCharacteristics/.