Slot Machine Base Game Evolutionary RTP Optimization

Conference Paper *in* Lecture Notes in Computer Science · April 2017

DOI: 10.1007/978-3-319-57099-0_45

3 authors:

Delyan Keremedchiev
New Bulgarian University

Petar Tomov
Bulgarian Academy of Sciences

Maria Barova
Bulgarian Academy of Sciences

Slot Machine Base Game Evolutionary RTP Optimization

Delyan Keremedchiev(✉), Petar Tomov, and Maria Barova

Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences, acad. Georgi Bonchev Str,
Block 2, 1113 Sofia, Bulgaria
d_keremedchiev@bas.bg
http://www.iict.bas.bg/

Abstract. Slot machines are casino gambling machines with three or more reels which spin when a button is pushed. The machine pays are based on patterns of symbols visible on the front of the machine when it stops. Most of the modern slots consist of a base game, free games and bonus games. The base game is the core of the playing process. Player's money are usually taken as bet in the base game and no bet is taken during free games or bonus games. Each slot machine has a parameter called return to player (RTP). RTP is the average amount of money which a player will get back, in average, after each spin of the reels. The total RTP (measured in percents) can be in the range between 75% and 98%. Its components are: base game RTP, free games RTP, bonus games RTP. The base game also controls how often free games will be activated and how often bonus games will be played. In this paper an evolutionary optimization algorithm for optimization of slot machine base game RTP by rearrangement of the symbols in the reels, is proposed. The problem itself is a combinatorial problem and the fitness function used checks all possible slot machine winning screens.

Keywords: Slot machine · Gambling · Genetic algorithms · Return to player · Optimization

1 Introduction

Slot machines are electronic gambling devices which are popular all over the world. The most popular slot machines consist of five reels. The reels start spinning when the button is pushed. Nowadays slot machines are computerized with PRNG embedded in them. In 1984 Inge Telnaes received a patent for a device titled, "Electronic Gaming Device Utilizing a Random Number Generator for Selecting the Reel Stop Positions" (US Patent 4448419) [1]. In the beginning slot machines were mechanical. They had a lever on one (the left) side of the machine (because of this lever, machines were known as one-armed bandits), which were used for reels spinning activation. The machine pays according to

I. Dimov et al. (Eds.): NAA 2016, LNCS 10187, pp. 406–413, 2017.
DOI: 10.1007/978-3-319-57099-0_45

symbol patterns, visible on the screen, when the reels stop. Slot machines are the most popular gambling method in casinos and constitute about 70 percent of the average US casino income [2].

A gambler playing a slot machine has credit inserted - either cash or by printed ticket or loaded by the attendant. The machine is activated by means of a lever (or a button), or by pressing a touchscreen. The objective of the game is to win money from the machine, which usually involves matching symbols on reels (mechanical or virtual) that spin and stop to reveal one or several symbols. Most games have a variety of winning combinations of symbols. If a player matches a combination according to the given patterns, the slot machine rewards the player [6].

Each machine has a table that lists the number of credits the player will receive if the symbols listed on the pay table line up on the pay line of the machine. Some symbols are wild and can represent many (or all) of the other symbols to complete a winning line [3]. Symbols are statistically distributed on the reels. Some symbols show up more often than others. Some symbols pay more than others, according to the pay table. Slot machines are usually adjusted to pay out winnings of 75 to 98% of the money that the players bet. It is known as a theoretical payout percentage or RTP (return to player). The minimum RTP varies among jurisdictions and is subject of law regulations [6].

$$\mathrm{RTP} = \mathrm{SUM(bet)}/\mathrm{SUM(win)} * 100 \tag{1.1}$$

This research is motivated by a previous work based on GA (genetic algorithm) optimization of slot machine RTP [5]. In this work RTP is optimized for the base game of the slot machine. Less important, free games and bonus game winnings frequency are also optimized. The source code used in this research is available as an open-source project in Github global repository [4]. The rest of this paper is organized as follows: Sect. 2 presents the model proposed. Section 3 is about some experiments and results. The final Sect. 4 is a conclusion and presents some ideas for further research.

2 The Model Proposed

The model proposed is based on Genetic Algorithms (GAs). GA is applied in RTP, free games and bonus game winning frequency, as parallel computing evolutionary optimization. Unlike [5,6], the fitness function is calculated by an exact numerical method instead of Monte-Carlo simulation.

2.1 RTP Optimization

Most of the modern slot machines are computerized. They have virtual reels with symbols distributed on them. Stops of the virtual reels are selected by randomly generated numbers. RTP of the game is directly dependant on symbols distribution on the reels. In most of the cases the total RTP is the sum of base game RTP, free games RTP and bonus game RTP. Usually symbols (and their positions) are selected manually by mathematicians. From mathematical point of view, slot machine reels are discrete distribution of symbols. Such distribution can be generated by discrete optimization, according given constraints like desired RTP, free games and bonus game winning frequency. Multi-criteria cost function is converted into a single criteria by Euclidean distance calculation. [4] Slot machine reels are presented as chromosomes in the GA optimization. The cost function is an exact numeric calculation of RTP in order to make solution space exploration more efficient.

2.2 Genetic Algorithms

Genetic algorithms (GAs) are search heuristic algorithms inspired by the process of natural selection [8,9]. GAs are routinely used to select points (candidate solutions) from the solutions space. By application of the techniques for inheritance (crossover), mutation and selection, the generated solutions get closer to the optimum. GAs are classified also as population based algorithms because each point in the solution space represents an individual inside the GA population. Each individual has a set of properties which are subject to mutation and modification (usually a crossover). The traditional representation of the properties is binary, as a sequence of zeros and ones, but other encodings are also possible (a binary tree for example) [5].

The optimization usually starts with a randomly generated population of individuals, but this may vary in different implementations. The optimization process is iterative and the population in each iteration is called generation. A fitness value is calculated for each individual of the generation. The fitness value usually represents the objective function which is subject to optimization. The fittest individuals in the population are selected (according to a selection rule) and recombined (crossover and/or mutation) to form a new generation. This new generation is used at the next iteration of the algorithm. The algorithm termination is usually achieved either by reaching the maximum number of generations or by reaching the desired level of the fitness value [5].

In order to run GAs, the researcher should provide: 1. Genetic representation of the solution space (the solution domain); 2. An appropriate fitness function to evaluate the solution domain. Once these two conditions are met, GAs can proceed with the population initialization and the iterative population improvement by repetitive application of the selection, crossover, mutation and individuals evaluation [5].

2.3 Implementation

GA chromosomes are two-dimensional arrays of symbols. All symbols are marked as integer numbers. Each GA chromosome is a point in discrete finite solution space. Valid reel should consist of integer numbers listed in Table 1. The slot machine model in this research consists of symbols from 3 to 12 as regular winning combination symbols. Symbols1 is used as a special symbols called wild (it substitutes all other symbols). Symbol16 is used as a special symbol called scatter. Scatter pays everywhere on the screen and it is not needed to form special line for combination. Scatter is also responsible for the game to play extra reels spins, called free games. Symbol15 is used as a special symbol called bonus symbol. The bonus symbol is responsible the game to enter bonus mode, which is extra game inside the slot game play. The optimization process runs on parallel CPUs. There is a central node with global GA population and group of calculating nodes. Each calculating node has a subset of the global population as described in [7]. Initialization of the global population is done by randomly generated reels. The size of the global population vary from several chromosomes to hundreds or thousands of chromosomes. The central node operation is as follows:

Table 1. Genetic algorithm parameters.

Parameter	Value
Generation gap	0.97
Crossover rate	0.98
Mutation rate	0.02
Maximum generations	100
Number of individuals	37
Number of variables	315
Inserted rate	100 %

1. Generate random global population;
2. Distribution for each local node:
    2.1 Random selection of local population replacement;
    2.2 Distribute local populations;
    2.3 Collect local optimization results;
    2.4 Stop if predefined number of distributions reached;
    2.5 Repeat from 2.1;
3. Finish;

Local population size is fixed on 37 or 57 usually (it is subject of empirical estimation). Local GA's parameters are described in Table 1.

Local node genetic algorithm is as follows:

1. Load subset of the global population into the memory;
2. Optimization:
    2.1 Select parents;
    2.2 Crossover;
    2.3 Mutation;
    2.4 Calculate RTP and frequencies;

  2.5 Keep newly generated chromosome;
  2.6 Stop if predefined number of generations reached;
  2.7 Repeat from 2.1;
3. Report results;

The first step of local GA optimization is selecting parents and result vector. As a second step, binomial crossover is applied. In order mutation to be valid an element from randomly selected chromosome replaces an element into the result chromosome. In the third step, a mutation is done. The final fourth step is related to fitness value calculation. The main difference between this work and those in [5,6] is that in the fitness value calculation the Monte-Carlo simulation is not used. A full generation of the possible game screens is applied instead. In this approach only base game reels can be optimized which is a disadvantage. When there is a complex game play in free games mode and bonus game mode the only approach could be the Monte-Carlo simulation.

Multi-criteria problem is converted to a single-criteria problem by the formula of the Euclidean distance. Because there are only three criteria the problem is not so interesting from multi-criteria point of view. The dominant criteria is the RTP, when frequencies (winning, free spins mode and bonus mode) are not so important.

The maximum number of recombinations is used as an optimization termination. Manual observation/termination of the process is also possible. The final solution, found by GA, is an integer matrix. This matrix is directly applicable as slot machine reels strips. For example, if there is a slot game with 5 reels (visible on the screen as 5 columns and 3 rows), and each reel consists of 63 symbols, the final GA solution would be an integer matrix of 5×63 values (refer to [4] for more details).

3 Experiments and Results

All experiments have been done on an open-source slot machine simulator [4] (5×3 screen) with a particular pay Table (Table 1) and 50 winning lines (Fig. 1). All winning combinations are paid from left to right. The lowest winning is 2 - for a combination of 3 symbols SYM12 (Table 2). The highest winning is 500 -

Table 2. Slot machine pay table. Each column represents the winning of one particular symbol (10 possible symbols in this game). Each row shows the winning of the symbols when the combination is of 2, 3, 4 or 5 symbols.

	SYM03	SYM04	SYM05	SYM06	SYM07	SYM08	SYM09	SYM10	SYM11	SYM12
2 of	30	20	0	0	0	0	0	0	0	0
3 of	150	75	50	40	5	4	4	3	3	2
4 of	250	100	100	75	50	40	20	6	5	4
5 of	500	250	175	150	100	50	30	25	15	10

Fig. 1. Slot machine winning lines. There are 50 possible lines. On each of these lines (from left to right) win patterns can appear. Win patterns are formed by 2, 3, 4 or 5 symbols.

for a combination of 5 symbols SYM03 (Tab. 1). There are 10 regular symbols and 1 scatter symbol, which form the winning patterns on the screen.

All experiments are done by elitism rule keeping the best solution found. Global population expands, while local populations are fixed to 37 chromosomes. The maximum number of local recombinations is 100. All initial reels are randomly generated.

The target RTP was selected to 90, which is the lowest legal value for the Bulgarian gambling market. Thirty independent runs of the algorithm were done and it is visible that fast convergence was achieved in the beginning (Fig. 3). GA convergence is faster and smoother than DDE (Fig. 2), presented in [6].

The results were compared with those presented in [5,6], which is related to slot machine RTP optimization by GA and DDE, but with Monte-Carlo simulation as fitness value. The convergence is a little bit faster than that achieved in [5,6]. There are common differences in the models, but the general optimization idea is similar. In the case of this research the main disadvantage is the cost function. It is very time consuming and It is more time consuming and much slower than the cost functions used in both previous implementations.

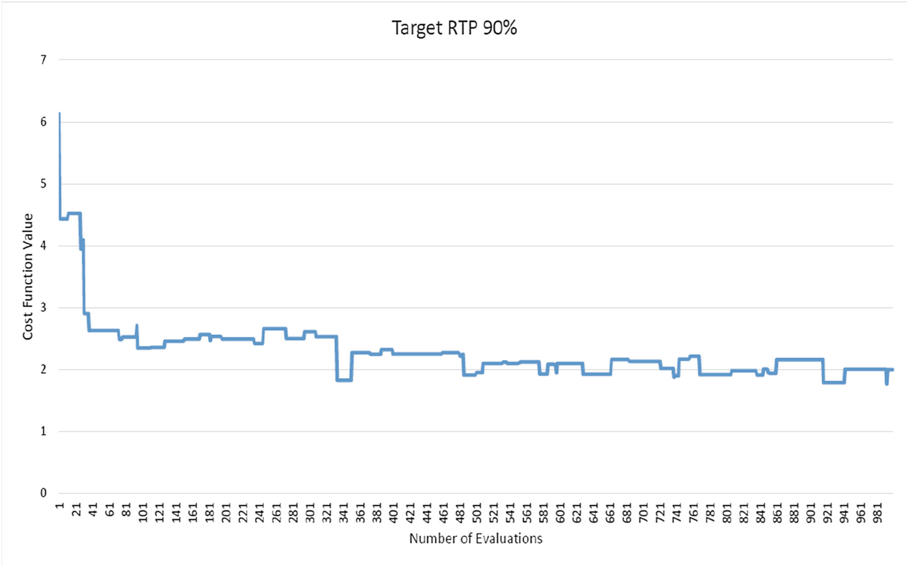

Fig. 2. Discrete differential evolution RTP optimization.

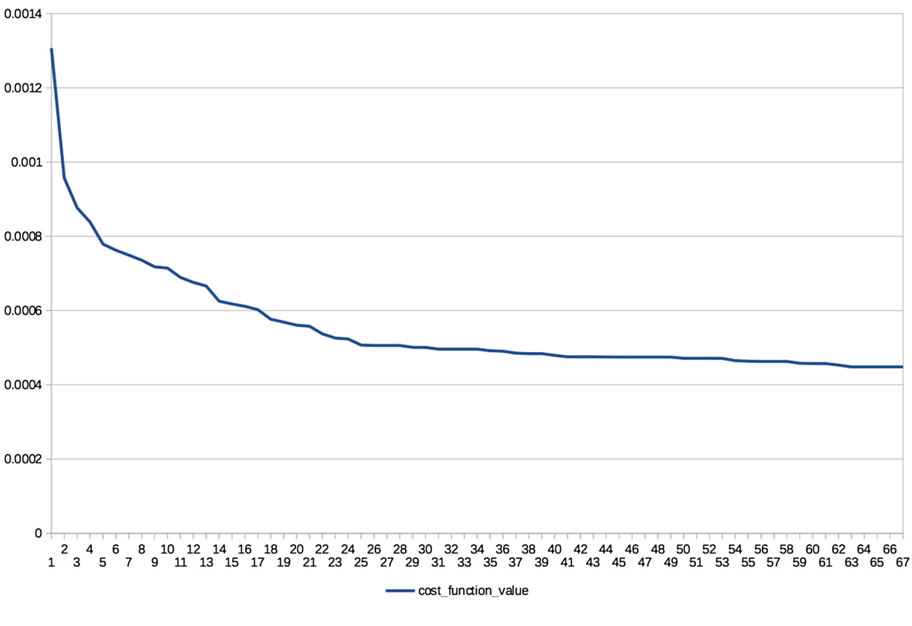

Fig. 3. Average convergence of thirty runs with mean value of 0.0005560259 and standard deviation of 0.0001492231.

4 Conclusions

The experiments show that the use of GA with exact numerical calculation of RTP as fitness function is very efficient and improves the slot game parameters by better adjustment of RTP, free games and bonus game frequencies. The optimization convergence depends on the probabilistic nature of GA. The biggest disadvantage is the exact RTP calculation which is time consuming and slows down the optimization process. The implementation of Discrete Diferential Evolution (DDE) instead of GA could be used for further research.

Acknowledgements. This work was supported by private funding of Velbazhd Software LLC.

References

1. Inge, S.: Electronic gaming device utilizing a random number generator for selecting the reel stop positions. US 4448419 A, Published 1984–05-15 (1984)
2. Cooper, M.: How slot machines give gamblers the business. The Atlantic Monthly Group (2005). Accessed 21 Apr 2008
3. Observer, C.: How to Play Slots. CasinoObserver.com (2013). http://casinoobserver.com/how-to-play-slots.htm. Accessed 06 Mar 2013
4. Balabanov, T.: Slot machine base game evolutionary RTP optimization as parallel implementation with MPI (2016). http://github.com/TodorBalabanov/SlotMachineBaseGameEvolutionaryOptimization/
5. Balabanov, T., Zankinski, I., Shumanov, B.: Slot machines RTP optimization with genetic algorithms. In: Dimov, I., Fidanova, S., Lirkov, I. (eds.) NMA 2014. LNCS, vol. 8962, pp. 55–61. Springer, Cham (2015). doi:10.1007/978-3-319-15585-2_6
6. Balabanov, T., Zankinski, I., Shumanov, B.: Slot machine RTP optimization and symbols wins equalization with discrete differential evolution. In: Lirkov, I., Margenov, S.D., Waśniewski, J. (eds.) LSSC 2015. LNCS, vol. 9374, pp. 210–217. Springer, Cham (2015). doi:10.1007/978-3-319-26520-9_22
7. Balabanov, T., Zankinski, I., Barova, M.: Distributed Evolutionary computing migration strategy by incident node participation. In: Lirkov, I., Margenov, S.D., Waśniewski, J. (eds.) LSSC 2015. LNCS, vol. 9374, pp. 203–209. Springer, Cham (2015). doi:10.1007/978-3-319-26520-9_21
8. Eiben, A.E., Raué, P.-E., Ruttkay, Z.: Genetic algorithms with multi-parent recombination. In: Davidor, Y., Schwefel, H.-P., Männer, R. (eds.) PPSN 1994. LNCS, vol. 866, pp. 78–87. Springer, Heidelberg (1994). doi:10.1007/3-540-58484-6_252
9. Ting, C.-K.: On the mean convergence time of multi-parent genetic algorithms without selection. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 403–412. Springer, Heidelberg (2005). doi:10.1007/11553090_41