Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 8784065, 8 pages
https://doi.org/10.1155/2021/8784065

Research Article

Slot Machine RTP Optimization Using Variable Neighborhood Search

Pantelis-Arsenios Kamanas, Angelo Sifaleras, and Nikolaos Samaras

Department of Applied Informatics, School of Information Sciences, University of Macedonia, 156 Egnatia Str., Thessaloniki 54636, Greece

Correspondence should be addressed to Angelo Sifaleras; sifalera@uom.gr

Received 11 August 2020; Accepted 27 April 2021; Published 5 May 2021

Academic Editor: Ramon Sancibrian

This work presents a Variable Neighborhood Search (VNS) approach for solving the Return-To-Player (RTP) optimization problem. A large number of software companies in the gaming industry seek to solve the RTP optimization problem in order to develop modern virtual casino gambling machines. These slot machines have a number of reels (e.g., three or more) that spin once a button is pushed. Each slot machine is required to have an RTP in a particular range according to the legislation of each country. By using a VNS framework that guides two local search operators, we show how to control the distribution of the symbols in the reels in order to achieve the desired RTP. In this study, optimization refers only to base game, the core of slot machine games, and not in bonus games, since a bonus game is triggered once two, three, or more specific symbols occur in the gaming monitor. Although other researchers have tried to solve the RTP problem in the past, this is the first time that a VNS methodology is proposed for this problem in the literature with good computational results.

1. Introduction

Slot machines, also known as fruit machines, are the most popular gambling games in casinos. They are either electronic or electromechanical devices that consist of a number of reels (n) that spin independently and a screen with a number of rows (m). Each reel contains several symbols in specific arrangements and quantities. Also, it has a number of positions at which they stop, and this number is called stops [1]. The stops are being shown on the display of a slot machine, while the reels are spinning in front of the player. Slot machines as mechanical devices had cylinders as spinning reels holding the symbols that were printed on a strip. Nowadays, slot machines are mostly electronic, and they can be found either in traditional casinos or in online casinos as software. However, in virtual slots, once the reels start spinning, a Pseudo-Random Number Generator (PRNG) is used to get the stops.

The display of a virtual slot machine shows the $(m \times n)$ window of the game, game rules, coin values, line bet, payout schedule, and more options. The player inserts cash or a paper ticket with a barcode in “ticket-in, ticket-out” machine. Once the player presses a button, in case the slot machine has the minimum amount of bet, the reels of the base game start spinning. After that, the reels produce the outcome of the game, which consists of all symbols that remained visible through the window on the display and the payment of the combination in pay table of these symbols that are greater than zero. If the visible window contains symbols that trigger a bonus game (e.g., free spins in base game or to another feature game), then the player has a big chance to win much more money than he/she bets.

The gambling industry is one of the most profitable industries globally. With more than US$ 450 billion dollars estimated value in 2020 and with a projected value of more than US$ 640 billion by 2027, the gambling market is definitely one of the largest and most profitable in the world [2]. It is also worth mentioning that these figures are only related to the regulated markets, as the grey and black markets contribute with hundreds of billions of dollars to the total market value. There are hundreds of companies of different types of services that compose the gambling industry: from casinos and software developers to casino game developers and payment solutions. The growth of remote gaming, with online and mobile gambling market size, is expected to grow by more than 10% by 2027, ensuring that the gambling industry will become even more profitable in the years to come. Hundreds of thousands of dollars are spent every year from several companies in order to participate in exhibitions around the world, with the most famous and attractive being held in Las Vegas, Paris, London, South Africa, and Singapore. A list of the most well-known conferences and trade shows with a Gaming, Sports Betting, and Affiliate theme may be found online [3], for example, Paris Games Week, iGB Live!, and ICE Totally Gaming.

Return-To-Player (RTP) is one of the two most important parameters for a player to choose if he/she is going to play the game, but it is also the main gambling game parameter which is under government regulation [4]. It is the win as percentage of the bet and is the phrase that casinos and casino game makers use to describe the long-term theoretical expected payback percentage from all wagers on a slot machine, video poker machine, or Video Lottery Terminal (VLT). If a slot machine player is going to play a hundred rounds at 1$ on a slot game with a theoretical slot RTP of 93%, with all other things being equal, he/she would expect to get a return of 93$ in wins from his staked 100$. A player wins back most of his wager. The remaining 7% of his bet, in this game, is the house edge, the money that casino earns off each bet. The formula for the computation of the RTP and also some necessary notations can be found with the formulae below:

$$\text{RTP} = \frac{\text{Coins Out}}{\text{Coins In}} \times 100\%, \tag{1}$$

or

$$\text{RTP} = \frac{\text{Total Prize}}{\text{Cycle} \times \text{Paylines}} \times 100\%. \tag{2}$$

Total Prize: the amount of money that a window instance wins

Cycle: how many times the reels spun in a Monte Carlo simulation

Pay Lines: the number of lines that the game has enabled

Coins In: total amount of coins bet

Coins Out: total amount of coins won

In this paper, Monte Carlo simulation with 100,000 and 1,000,000 separate runs will be used for the RTP calculations. In simulations, formula (2) is used, where the cycle represents the number of runs. In addition, there is another type of simulation called "Full Cycle." This latter type of simulation is used in the design phase (i.e., mathematical) of a slot machine game. It is an exhaustive cycle through all possible stop positions. The Full Cycle simulation allows an exact match to the calculations in the spreadsheet but is not capable of calculating coinciding statistics across a series of feature games.

The structure of this work is as follows: In Section 2, we discuss similar approaches for the RTP optimization problem, focusing on recent research work and the methods they utilize. Next, we introduce our methodology in Section 3. Section 4 presents the results of our methods on several games. Finally, we draw our conclusions and provide some research directions for future work in Section 5.

2. Related Work

Although the RTP optimization problem is one of the most important parts of the slot machine game design, there are not many scientific articles that present a heuristic or metaheuristic approach to solve this problem. Only three papers have been published which present a metaheuristic approach to solve the RTP optimization problem and are briefly presented in Table 1.

Balabanov et al. [4] presented a solution for the slot machine RTP optimization problem, where the symbols distribution of the base game is being controlled and, also, some other features like free spins frequency or bonus game frequency. The fitness function in this work was the absolute difference between the desired RTP and the obtained RTP. For Monte Carlo simulations, 1,000,000 or 100,000 separate runs were used in order to calculate the obtained RTP. They used Genetic Algorithms (GAs), with a randomized initial population and appropriate crossover and mutation operators. The target RTP of the example that the authors presented was 99%, starting from 91%. So, the algorithm increases the RTP in a small range in case the game already exists (a quality starting RTP). However, in most cases, the mathematicians of slot games usually try to develop a new slot machine game (from scratch) that has initial reels with randomized symbol distribution and the RTP might be equal, for example, to 630% (big random value). So, it is not a trivial task for such algorithms to handle the case of large reductions and the huge difference between the target RTP and the initial RTP. The decrease of the RTP in slot machine game is the hardest problem for the mathematicians, since they have to solve the RTP optimization problem with hand adjustments and this usually requires a long time. Furthermore, the algorithm's behavior in the case where special symbols appear in reels (e.g., wild, scatter) is also unknown.

The same authors in [5] also proposed a Differential Evolution (DE) algorithm, since the problem of optimal symbols distribution on the reels is discrete and its convergence was found to be faster than GAs. Additionally, they also used Monte Carlo simulations with 1,000,000 separate runs over ten trials for better accuracy in base game. Their model of slot machine game contained only simple symbols and not scatter, wild, or bonus games (scatter represents symbols that pay everywhere on the screen and substitute all other symbols). The authors proposed a multicriteria optimization approach as future work, for the symbol frequencies, the free spins frequency, the game volatility, and the bonus game frequency.

Table 1: Key literature contributions on RTP optimization.

Reference	Approach	Simulation	Special symbols	Runs	Games tested
This work	VNS	Monte Carlo	Yes	100,000 & 1,000,000	3
					3
[4]	GA(s)	Monte Carlo	No	100,000 or 1,000,000	1
[5]	DE	Monte Carlo	No	1,000,000	1
[6]	DDE and GA(s)	Full Cycle	Yes	All possible	1

Keremedchiev et al. [6] proposed a Genetic Algorithm methodology for the RTP optimization problem, in bonus and free games frequency as parallel computing evolutionary optimization. The fitness function of base game that was used was computed with Full Cycle method instead of Monte Carlo simulations. In current algorithm, the model of slot machine game does contain as features bonus game, simple symbols that pay on each pay line, scatter symbols, and wild and bonus symbols that trigger the bonus game. This experiment, as it refers in it, showed that the use of Genetic Algorithm with exact numerical calculation of RTP as fitness function was very efficient and improved the slot game parameters by better adjustment of RTP, bonus game, and free games frequency. Finally, the biggest drawback was that the exact RTP computation was time-consuming and slowed down the optimization process. The authors also proposed a Discrete Differential Evolution (DDE) implementation as a further research.

3. Research Methodology

Variable Neighborhood Search (VNS) constitutes a metaheuristic solution method which is systematically changing a neighborhood structure. VNS was initially proposed by Mladenović et al. [7, 8]. Nowadays, a large number of successful VNS applications have been reported in a wide range of areas [9–13]. In this work, a Variable Neighborhood Descent (VND) heuristic schema was applied in base game's RTP and it takes all symbols into account. The VND metaheuristic features a strong local search phase (intensification) utilizing a number of different neighborhood structures but lacks a diversification phase (shaking method) as compared to other VNS variants. The VND variant has been applied in several research works in the literature [14, 15].

The proposed solution method is analytically described in the following subsections and an overview is depicted in Figure 1.

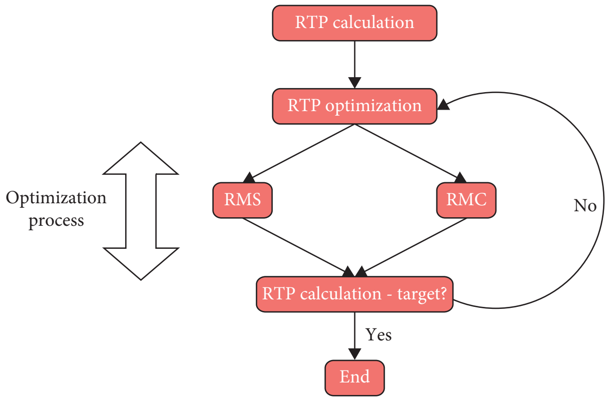

Figure 1: Solution process.

3.1. Solution Representation. A slot machine consists of n reels where the symbols are represented by a string and are equally distributed. The solution representation is a two-dimensional array with the distribution of the symbols on each reel. For example, in case of a five-reel (number of columns) slot machine game with 48 symbols per reel (number of rows), the solution array (denoted as "Reels") would have the dimension of 48×5.

Three inputs of the proposed algorithm are (i) the maximum number of iterations as a termination criterion (also a maximum CPU solution time can be used), (ii) the target (desired) RTP of the user ($\mathrm{RTP}_{\mathrm{des}}$), and (iii) the allowed deviation denoted (R) from the target RTP. Usually, the computation of the precise RTP is quite difficult and, due to this reason, game developers are satisfied if the RTP belongs in a specific range. Hence, if $\mathrm{RTP}_{\mathrm{curr}} \in [\mathrm{RTP}_{\mathrm{des}} - R, \mathrm{RTP}_{\mathrm{des}} + R]$, then the algorithm stops. Assume, for example that if we are given $R = 0.03$, $\mathrm{RTP}_{\mathrm{des}} = 0.83$, and $\mathrm{RTP}_{\mathrm{curr}} = 0.80$, then the algorithm stops. An extra feature of the algorithm is that it allows the user to declare a minimum number of each symbol per reel. Thus, a fourth input of the proposed algorithm is an additional two-dimensional array containing the minimum number of symbols per reel and is denoted as "minNSR." The number of columns of this array is equal to the number of reels, while its number of rows is equal to the number of distinct symbols. Thus, $\min \mathrm{NSR}(i, j)$ denotes the minimum required number of occurrences of the i symbol in the j-reel.

3.2. Neighborhood Structures. The proposed VND methodology utilizes two neighborhood structures denoted as "RMS" and "RMC." Both these local search operators aim to reduce the RTP as follows:

(i) **RMS:** initially, the symbols are sorted in ascending order based on their corresponding payments. The Replace Maximum Symbol (RMS) local search operator replaces symbol_i with the biggest payment by symbol_{i-1} that has the next biggest payment in pay table. The players of a slot machine game may win a monetary award in the case where a combination of at least two identical symbols occurs in the pay lines. The symbol that has the biggest payment in these combinations is considered the most expensive symbol. However, the replacement by the RMS operator is permitted only after checking the input array min NSR (minimum number of symbols per reel requested by the game developer); otherwise, the next most expensive symbol$_{i-1}$ is considered for potential replacement.

(ii) **RMC:** the RTP increases when the placement of the symbols in the reels leads to minimum winning combinations of symbols (i.e., combinations of two identical symbols only) with a high frequency. The Replace Minimum Combination (RMC) local search operator replaces symbol$_i$ which participates in a minimum winning combination with symbol$_j$ that does not participate in such winning combinations (e.g., it participates in winning combinations of three or more identical symbols). Such symbols may participate in combinations with a little larger payment, but the frequency of their combinations is much lower.

3.3. *VND Metaheuristic Algorithm.*

The VND solution approach is depicted in Algorithm 1. Also, Monte Carlo simulation (RunSimulation) is used for the calculation of the current RTP$_{\text{curr}}$ with 100,000 or 1,000,000 separate runs. The two neighborhood structures RMS and RMC are also denoted as N_1 and N_2, respectively. Moreover, the number of different neighborhood structures is denoted as $l_{\max}$ and in this work it is equal to two.

The proposed VND is initialized with a randomized initial symbol distribution per reel (the list of distinct symbols is given by the game developer) and terminates once a maximum number of iterations is performed rather than not finding an improvement. This is due to the fact that, in each iteration, a Monte Carlo simulation might return an approximate value of the RTP and not the exact RTP that would result after a (time-consuming) Full Cycle RTP computation. Thus, in some cases, although the two local search operators do not make any improvement, a better RTP value can be calculated by the Monte Carlo simulations. Moreover, each time the neighborhood is changed, the search is performed in a different (next) reel. Thus, the VND starts the optimization using the first reel (Reel$_{\text{curr}}$ = 1) and afterward sequentially applies the local search operators in each one of the n reels.

3.4. *Illustrative Examples.*

Without loss of generality, let us assume a hypothetical slot machine game with n reels, with each one of them containing 12 equally distributed symbols. The symbols are represented as string variables in an array (R) and the pay table is stored as integer values in another array (P).

$$P = \begin{bmatrix} 0 & 2 & 4 & 16 & 64 \\ 0 & 0 & 8 & 32 & 128 \\ 0 & 0 & 8 & 32 & 128 \\ 0 & 0 & 16 & 64 & 512 \end{bmatrix}. \tag{3}$$

Next, an example of the minNSR array is presented, where for simplicity we show only one of the columns corresponding only to the first reel. Each index represents how many of $sym0i$ symbols should remain in current reel and not get replaced by other, for example, minNSR[2] = 3 meaning that $sym02$ cannot occur less than three times in the first reel. This way, it will not be possible to replace $sym02$ with $sym03$.

$$\min \text{NSR}^T = [2 \;\; 3 \;\; 1 \;\; 1]. \tag{4}$$

The RMC operator changes the symbols of the first reel as follows:

$$\begin{bmatrix} sym01 \\ sym02 \\ sym02 \\ sym02 \\ sym01 \\ sym01 \\ sym03 \\ sym02 \\ sym01 \\ sym03 \\ sym01 \\ sym04 \\ sym04 \end{bmatrix} \Rightarrow \begin{bmatrix} sym02 \\ sym02 \\ sym02 \\ sym02 \\ sym01 \\ sym01 \\ sym03 \\ sym02 \\ sym01 \\ sym03 \\ sym01 \\ sym04 \\ sym04 \end{bmatrix}. \tag{5}$$

The i-row of the pay table contains the payments of each symbol $sym0i$ (arrays start from one) for combinations of one, two, three, four, and five, respectively, for example, $sym03$'s payment takes place in the third row of pay table $P[3]$. The RMC local search operator replaces $sym01$ with $sym02$.

If the RMS operator was applied, then $sym04$, that is, the most expensive symbol according to pay table (combination of five pays 512), will be replaced by $sym03$ which is the next most expensive but cheaper than this symbol. So, in the first reel $sym04$ will be replaced by $sym03$ as follows:

$$\begin{bmatrix} sym01 \\ sym02 \\ sym02 \\ sym02 \\ sym01 \\ sym01 \\ sym03 \\ sym02 \\ sym01 \\ sym03 \\ sym01 \\ sym04 \\ sym04 \end{bmatrix} \Rightarrow \begin{bmatrix} sym01 \\ sym02 \\ sym02 \\ sym02 \\ sym01 \\ sym01 \\ sym03 \\ sym02 \\ sym01 \\ sym03 \\ sym01 \\ sym04 \\ sym03 \end{bmatrix}. \tag{6}$$

```
Data: $\text{RTP}_{\text{des}}$, $\text{iteration}_{\text{max}}$, $R$, minNSR, Reels, $l_{\text{max}}$
Result: Reels
$\text{Reel}_{\text{curr}} \longleftarrow 1$;
$\text{RTP}_{\text{curr}} \longleftarrow$ RunSimulation;
Difference $\longleftarrow \text{RTP}_{\text{curr}} - \text{RTP}_{\text{des}}$;
repeat
  $l \longleftarrow 1$;
  repeat
    Select $\text{Reels}' \in \mathcal{N}_l(x)$ such that $\text{RTP}(\text{Reels}') < \text{RTP}(\text{Reels})$;
    $\text{RTP}_{\text{curr}} \longleftarrow$ RunSimulation;
    NeighborhoodChange $(\text{Reels}, \text{Reels}', l)$;
    $\text{Reel}_{\text{curr}} \longleftarrow (\text{Reel}_{\text{curr}} + 1) \bmod (n)$;
  until $l = l_{\text{max}}$;
until stopping criterion;
```

Algorithm 1: VND.

4. Experimental Results

All the computational experiments have been done using a Dell Inspiron 5558 laptop with an Intel Core i5-5200U CPU at 2.20 GHz, 8 GB DDR3 RAM at 1600 MHz, and running 64-bit Microsoft Windows 10 Professional. The experiments have been done using three commercial slot machine games developed by the Zeusplay (https://zeusplay.com) casino game developer company.

Le Mystere Du Prince [16], a five-reel slot machine game with wild and scatter as special symbols

Amun's Book [17], a five-reel slot machine game that holds one special symbol called book and acts as a wild and scatter

Wild Charger [18], a five-reel slot machine game that holds one wild and acts as expanding and sticky at the same time (symbol developed by Zeusplay)

All the above games have ten pay lines (winning lines) and they are all paid from left to right. The smaller winning is two monetary units for a combination of two symbols for each game. The details (e.g., pay table, symbols) of the first slot machine game are presented below:

$$P = \begin{bmatrix} 0 & 2 & 5 & 25 & 100 \\ 0 & 0 & 5 & 25 & 100 \\ 0 & 0 & 5 & 25 & 100 \\ 0 & 0 & 5 & 25 & 100 \\ 0 & 0 & 10 & 50 & 125 \\ 0 & 0 & 15 & 75 & 250 \\ 0 & 0 & 15 & 75 & 250 \\ 0 & 0 & 20 & 100 & 400 \\ 0 & 2 & 25 & 125 & 750 \\ 0 & 2 & 25 & 125 & 750 \\ 0 & 20 & 50 & 200 & 5000 \\ 0 & 10 & 250 & 2500 & 9000 \end{bmatrix}. \tag{7}$$

In this game, special symbols wild and scatter are included. The distinct symbols of the game are

$$S = \begin{bmatrix} sym01 \\ sym02 \\ sym03 \\ sym04 \\ sym05 \\ sym06 \\ sym07 \\ sym08 \\ sym09 \\ sym10 \\ sym11 \\ scatter \\ wild \end{bmatrix}. \tag{8}$$

Each $S[i]$ refers to payments of $P[i]$ element for one, two, three, four, and five combinations of this symbol. The only differences with the other two games, in base game, are the pay table and the number of distinct symbols. Amun's Book and Wild Charger consist of ten and eight symbols, respectively. The minNSR array for these two games has been set the same as Le Mystere Du Prince's.

$$\text{minNSR} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{9}$$

According to the above minNSR table, at least one symbol on each reel must exist. In the tables below, all the necessary information about the optimization is presented, such as the number of separate runs with Monte Carlo simulation, the target RTP, the initial RTP, and the RTP that was finally obtained. The allowed deviation for the RTP

(R variable) was set equal to 3% for the first game, meaning that if 77% $\leq \text{RTP}_{\text{curr}} \leq$ 83%, then the algorithm terminates since a good solution was found. The output is a set of reels with the modified symbols and the best RTP that has been found. For the second game and the third game, the R value was set equal to 2% and 1%, respectively. These three games are presented in order to show that the algorithm can indeed optimize games with either large or small difference between the initial RTP and the target RTP. The RTP of the first game was decreased from 715% to 80%, while the RTP of the second game was decreased from 57% to 40%, and the RTP of the third game was decreased from 107% to 53%, approximately. The results per game are presented in Tables 2–4, respectively.

The required solution time and number of iterations are presented in Table 5.

The RTP reduction through the whole process of optimization for each iteration of each game is presented in Figures 2–7. It can be noticed that Monte Carlo simulation with 1,000,000 separate runs is more precise and needs less iterations to find one acceptable solution rather than the version with 100,000 separate runs. That happens since the larger the sample (runs) is, the more exact the results will be. A larger sample always leads to more reliable results.

Also, it can be noticed that Amun's Book RTP needs 110 iterations to finish for 100,000 runs and 149 iterations for 1,000,000. The difference with the other two games which makes these big changes in RTP is due to the fact that the wild symbol is also a scatter, so once the wild symbol is modified to another simple symbol, the RTP rapidly decreases.

Creating an exciting as well as attractive slot game, consists of a variety of elements, depending on the target group. There are several types of players, some of which prefer to risk on a game that might not return much when in base game, but they have a bonus feature that is more likely to return big wins. In order to create such games, a combination of acceptable pay tables, hit rates, and bonus feature triggering frequency must be taken in consideration. The most common logic is to have a risky base game, with a low RTP but with recurring small wins, so that the player does not immediately lose his/her money, which triggers a bonus feature more frequently and gives the player the confidence that he/she is able to win big prizes.

Table 2: Le Mystere Du Prince.

Runs	RTP initial (%)	RTP target (%)	RTP obtained (%)
100,000	718.374	77–83	82.025
1,000,000	712.661	77–83	78,996

Table 3: Wild Charger.

Runs	RTP initial (%)	RTP target (%)	RTP obtained (%)
100,000	56.105	38–42	41.277
1,000,000	57.226	38–42	38.758

Table 4: Amun's Book.

Runs	RTP initial (%)	RTP target (%)	RTP obtained (%)
100,000	111.005	52–54	52.015
1,000,000	104.852	52–54	52.803

Table 5: Results for the three games.

Game	Simulation runs	Time (secs)	Iterations
Le Mystere Du Prince	100,000	353	2456
Le Mystere Du Prince	1,000,000	786	501
Wild Charger	100,000	2	2062
Wild Charger	1,000,000	15	350
Amun's Book	100,000	35	110
Amun's Book	1,000,000	223	149

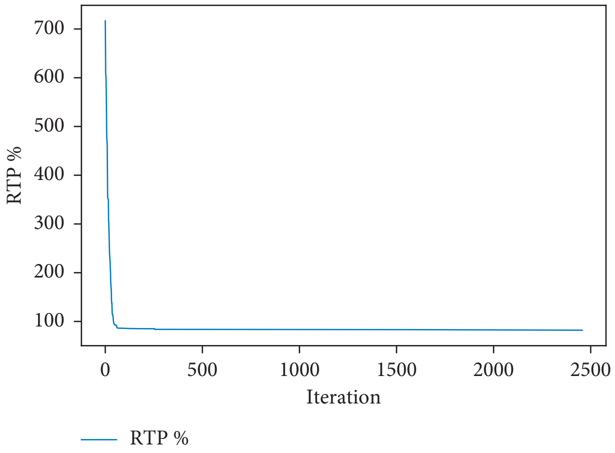

Figure 2: Le Mystere Du Prince: 100,000 runs.

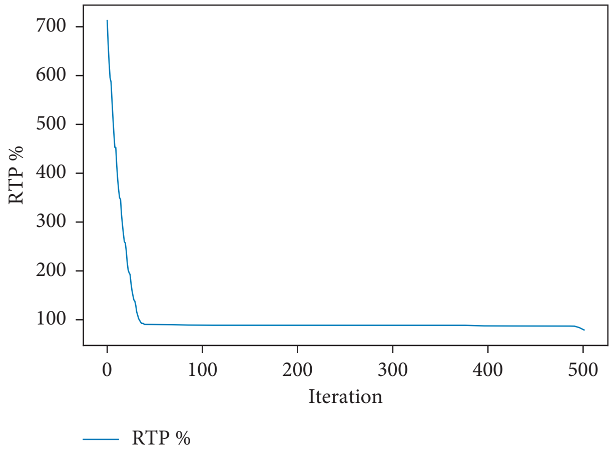

Figure 3: Le Mystere Du Prince: 1,000,000 runs.

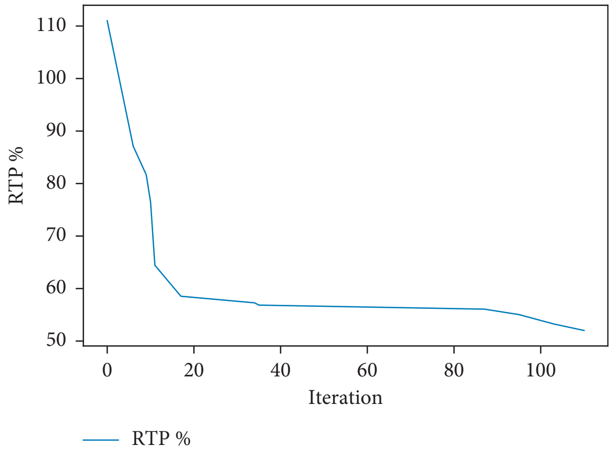

Figure 4: Amun's Book: 100,000 runs.

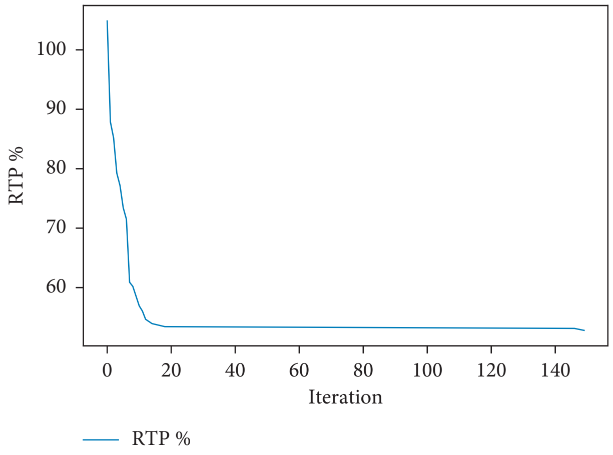

Figure 5: Amun's Book: 1,000,000 runs.

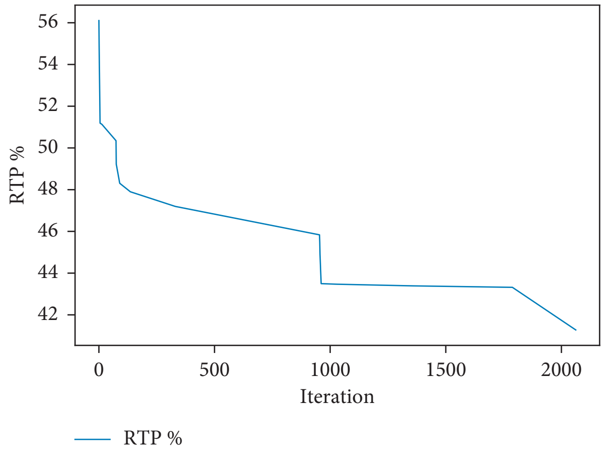

Figure 6: Wild Charger: 100,000 runs.

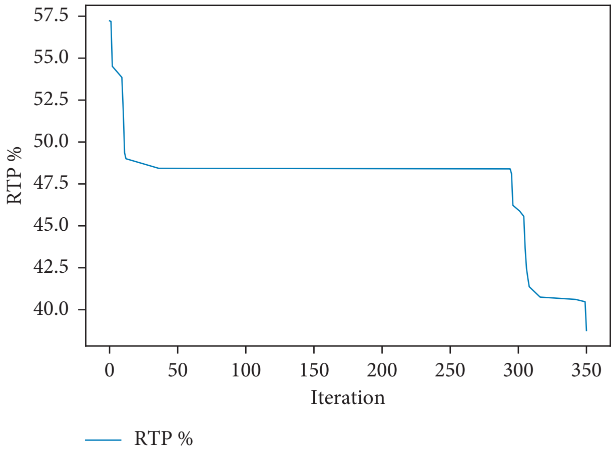

Figure 7: Wild Charger: 1,000,000 runs.

5. Conclusions and Future Work

The experiments showed that the use of VNS was very efficient in developing slot games from scratch, where the symbol distribution is not uniform. As shown in the figures of the previous section, the required number of iterations of the proposed method depends on the initial difference between the initial RTP and target RTP. Also, in cases where a symbol contains two types of special symbols (e.g., such as in the second game where the "scatter" symbol is also a "wild" symbol), a large RTP decrease can be achieved by converting this symbol to another.

As the complexity of games increases, the required solution time increases too, and this fact shows the need for speeding up the whole process. As future research, an implementation of much more complex games RTP can be proposed using parallel CPU and GPU computing techniques to decrease the computational time for finding an acceptable (RTP) solution. Furthermore, the proposed algorithm is a single-criterion approach for the RTP optimization. Another research idea is to develop a multicriteria method for optimizing RTP, volatility, and hit rate (i.e., how frequently a player is expected to stop on a winning combination).

Data Availability

The proposed methodology is applied to three commercial casino games, which can be found online at the company's website (public URLs are included in the references).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work has been funded by the company Zeusplay (Research Committee of the University of Macedonia (Grant no. 81359)).

References

[1] C. Barboianu, *The Mathematics of Slots: Configurations, Combinations, Probabilities*, INFAROM Publishing, 2013.

[2] Research and Markets: Global Gambling Market Analysis & Forecasts 2012-2019 & 2020-2027, 2020. https://www.researchandmarkets.com.

[3] Tradefest: The Best Gaming, Sports Betting and Affiliate Trade Shows and Conferences in 2020 & 2021, 2020. https://tradefest.io/en/tag/gambling-and-casinos.

[4] T. Balabanov, I. Zankinski, and B. Shumanov, "Slot machines RTP optimization with genetic algorithms," in *Numerical Methods and Applications (NMA 2014)*, I. Dimov, S. Fidanova, and I. Lirkov, Eds., pp. 55–61, Springer, Berlin, Germany, 2015.

[5] T. Balabanov, I. Zankinski, and B. Shumanov, "Slot machine RTP optimization and symbols wins equalization with discrete differential evolution," in *Large-Scale Scientific Computing (LSSC 2015)*, I. Lirkov, S. D. Margenov, and J. Waśniewski, Eds., Springer, Berlin, Germany, pp. 210–217, 2015.

[6] D. Keremedchiev, P. Tomov, and M. Barova, "Slot machine base game evolutionary RTP optimization," in *Numerical Analysis and its Applications (NAA 2016)*, I. Dimov, I. Faragó, and L. Vulkov, Eds., pp. 406–413, Springer, Berlin, Germany, 2017.

[7] P. Hansen, N. Mladenović, J. Brimberg, and J. Moreno-Perez, "Variable neighborhood search," in *Handbook of Metaheuristics. International Series in Operations Research & Management Science*, M. Gendreau and J. Y. Potvin, Eds., Springer, Berlin, Germany, 2019.

[8] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.

[9] I. Alharkan, K. Bamatraf, M. A. Noman, H. Kaid, E. S. Abouel Nasr, and A. M. El-Tamimi, "An order effect of neighborhood structures in variable neighborhood search algorithm for minimizing the makespan in an identical parallel machine scheduling," *Mathematical Problems in Engineering*, vol. 2018, Article ID 3586731, 2018.

[10] C. Alves, P. Bras, J. M. Valerio de Carvalho, and T. Pinto, "A variable neighborhood search algorithm for the leather nesting problem," *Mathematical Problems in Engineering*, vol. 2012, Article ID 254346, 2012.

[11] R. Benmansour, A. Sifaleras, and N. Mladenović, "Variable neighborhood search," in *Proceedings of 7th International Conference, ICVNS 2019*, vol. 12010, Springer, Cham, Switzerland, Rabat, Morocco, October 2019.

[12] A. f. Ling, "A VNS metaheuristic with stochastic steps for max 3-cut and max 3-section," *Mathematical Problems in Engineering*, vol. 2012, Article ID 475018, 16 pages, 2012.

[13] A. Sifaleras, S. Salhi, and J. Brimberg, "Variable neighborhood search," in *Proceedings of 6th International Conference, ICVNS 2018*, vol. 11328, Springer, Cham, Switzerland, Sithonia, Greece, October 2018.

[14] A. Duarte, J. Sánchez-Oro, N. Mladenović, and R. Todosijević, "Variable neighborhood descent," in *Handbook of Heuristics*, R. Martí, P. M. Pardalos, and M. G. C. Resende, Eds., pp. 341–367, Springer International Publishing, Berlin, Germany, 2018.

[15] A. Sifaleras and I. Konstantaras, "Variable neighborhood descent heuristic for solving reverse logistics multi-item dynamic lot-sizing problems," *Computers & Operations Research*, vol. 78, pp. 385–392, 2017.

[16] Zeusplay: Le mystere du prince, 2021, https://zeusplay.com/portfolio_page/le-mystere-du-prince.

[17] Zeusplay: Amun's Book, 2021, https://zeusplay.com/portfolio_page/amuns-book.

[18] Zeusplay: Wild charger, 2021, https://zeusplay.com/portfolio_page/wild-charger.