Antagonistic Procedural Content Generation Of Sparse Reward Game

Shaoyou, Xie*
School of Animation and Digital Arts, Communication University of China, China
yuusukei@cuc.edu.cn

Wei, Zhou
School of Animation and Digital Arts, Communication University of China, China
void_hyacinth@cuc.edu.cn

Honglei, Han
School of Animation and Digital Arts, Communication University of China, China,
hanhonglei@cuc.edu.cn

ABSTRACT

With the development of the game industry, procedural content generation technology has been widely used in game automatic content generation. And with the help of this technology, designers can quickly generate unlimited levels, implement real-time level generation, preset difficulty level generation, and so on. However, due to the difficulty of providing real-time and continuous feedback from game elements, sparse reward game is difficult to meet the expected results in content generation. This paper uses search-based procedural content generation combined with auxiliary task reinforcement learning to implement automatic content generation of sparse bonus games by means of confrontation generation. In this method, hierarchical reinforcement learning can smoothly evaluate the fitness of the generated candidate individuals, and screen individual populations according to the obtained fitness**.** This generation method is based on agent simulation, and generates runnable levels through free exploration of agents, which is more novel than traditional rule-based generation. In the end, we successfully implemented the procedural content generation of a typical sparse reward game, proving that our method is feasible. Different sparse reward games have different task complexity. By modifying the levels and tasks of hierarchical reinforcement learning, we can extend this method to other sparse reward games.

CCS CONCEPTS

• **Computing methodologies**; • **Artificial intelligence**; • **Search methodologies**; • **Discrete space search**; • **Applied computing**; • **Computer games**;

KEYWORDS

Computer games, PCG, Sparse reward game, Reinforcement learning

ACM Reference Format:
Shaoyou, Xie*, Wei, Zhou, and Honglei, Han. 2021. Antagonistic Procedural Content Generation Of Sparse Reward Game. In *The 16th International Conference on the Foundations of Digital Games (FDG) 2021 (FDG'21), August 03–06, 2021, Montreal, QC, Canada.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3472538.3472580

FDG'21, August 03–06, 2021, Montreal, QC, Canada

ACM ISBN 978-1-4503-8422-3/21/08…$15.00
https://doi.org/10.1145/3472538.3472580

1 INTRODUCTION

With the improvement of computer computing ability, procedural content generation technology appears, which can help game designers quickly generate game content that meets their needs, increase game playability, implement unlimited levels, adaptive difficulty adjust and other functions. At present, there are many algorithms of procedural content generation with different ideas applied to the automatic generation of all kinds of games. Such as generation and testing based on random algorithm, heuristic method based on search, PCGML (procedural content generation via machine learning) based on machine learning, etc. [1]. Using these methods can help game designers solve various types of game procedural content generation problems.

Sparse reward refers to the problem that it is difficult for the agent to obtain positive reward in the process of exploration, which leads to slow learning or even unable to learn. For example, in the Go chess, it is difficult for agents to get reward feedback during the game, and the global reward obtained at the end of the game is too sparse and the state space is huge, which leads to the slow learning or even impossible to learn for traditional reinforcement learning agent.

At present, the application of common procedural content generation technology in sparse reward game is difficult to obtain novel and runnable automatic generated content. Generated content with novelty means that the generated content is not constrained by fixed generation rules, and has innovative content similar to that created by human designers. These contents even have the level details that human designers have not found. For example, generation and testing are difficult to simulate the interaction details in the real game process, and the generated content lacks novelty. Another example is the method of generating procedural content based on machine learning, which needs many data sets support. In view of this, this paper proposes a new method of combining search-based procedural content generation with layered intensive learning to help evaluate agent's reward feedback and successfully complete game levels by layering environmental goals in a way that is counter-generated. Perform fitness analysis based on the data obtained in the agent simulation process, and the fitness analysis results will be used in evolutionary calculations to achieve procedural content generation for sparse reward games.

In this paper, we implement the procedural content generation of Sokoban game, and get the game content that can run. Through the analysis of the generated results, we find that the generated content has some characteristics similar to the artificially designed level, and these generated contents are different. which shows that our proposed content generation algorithm of sparse reward game can solve the problem of procedural content generation of sparse reward game and has strong practicability. Based on this method, the sparse reward tasks are layered by the characteristics of different sparse reward games. We can apply this method to different sparse reward games, and then implement the procedural content generation of generalized sparse reward games.

The contributions of this paper are as follows: 1. A novel sparse reward game generation model is proposed, which successfully generates the procedural content of sparse reward game. 2. Hierarchical reinforcement learning is introduced into the field of procedural content generation to solve the problem of quality evaluation of sparse reward game candidate content generation. 3. By analyzing the generated results, it is proved that our method can generate novel content.

2 RELATED WORK

There are many mature cases of procedural content generation technology in the field of game and computer technology, such as Rogue and Elite. These two games use automatic algorithm generation to reduce the cost of game development and save the space occupied by the game. In recent years, the technology of procedural content generation has gradually formed several different branches. According to Togelius J, Yannakakis G N, Stanley K O, et al. [2], procedural content generation including generation and testing methods, search based procedural content generation methods, and machine learning based procedural content generation methods.

Bento D S et al. [3], present a system that uses custom hardness metrics and novelty to generate the initial states of Sokoban game. C. Browne [4], using generation and testing methods, writes a game description language to express game content, and implements a general game system to play, measure and explore game content, and then synthesizes new games through the evolution of existing rule sets. However, constrained by the fixed rule generation, this method requires the designer to implement the given detailed generation constraint rules, which leads to the lack of novelty and time-consuming of the generated content.

A. Summerville et al. [5], a machine learning model trained on existing content to generate game content. Simon Liu, et al. [6], using PCGML method based on machine learning to implement the content generation of tower defense game. But using this method needs a lot of training data support, which is suitable for the content expansion of existing games, but not suitable for the first design of the game for procedural content generation.

A. Baldwin et al. [7] designed a hybrid startup tool, which takes the search-based procedural content generation as the core and the interactive interface is the auxiliary. The designer can adjust the desired generation results according to the needs, so that the designer can participate in the content generation. W. L. Raffe et al. [8] proposed to integrate multiple evolutionary processes to achieve personalized procedural content generation, which can be used to design a personalized game that can adapt to different player preferences and skills. But in the sparse reward game, it is difficult for the evaluation agent to evaluate the candidate content well, which leads to the poor effect of search based procedural content generation.

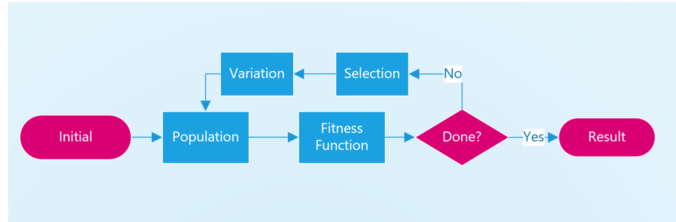

Figure 1: Process of search based procedural content generation method

At present, in the aspect of sparse reward task, the research to solve sparse reward task have gradually developed and matured with the efforts of many researchers, such as Dietterich T G. [9]. In this paper, a new hierarchical reinforcement learning method is proposed, which decomposes the objective Markov decision process into a smaller Markov decision process, and decomposes the value function of the objective Markov decision process into an additive combination of the value functions of the smaller Markov decision process. Nachum o, Gu s, Lee h, et al. [10], developed a general hierarchical reinforcement learning algorithm to solve the general sparse reward problem, so that the agent is not limited to specific tasks or specific decisions.

3 METHODOLOGY

In this paper, we use a search-based procedural content generation method for sparse reward game procedural content generation. In the iterative generation process, we use the evaluation function and call it to assign a number or vector to the content, which is called fitness or the value of the content. When generating new candidate content, it depends on the adaptability value assigned to the previous evaluation content instance: the purpose is to generate new content with higher value.

As shown in Figure 1, the whole process can be summarized as: initialization, generation, and evaluation of fitness. When the generation goal has been achieved, the generation will be ended, and the candidate individuals will be screened, regenerated and mutated to generate new candidate content. This process is repeated until the fitness of the candidate content meets the generation requirements, then the generation is finished, and the candidate content that meets the requirements is the result of this generation.

When we evaluate the sparse reward game, we need to find the most suitable evaluation method to ensure that the generation process is fast and accurate. The common methods proposed by G. N. Yannakakis and J. Togelius [11] include direct evaluation, simulation-based evaluation, and interactive evaluation.

For sparse reward games, the evaluation speed of direct evaluation is very fast. However, because of its evaluation based on some characteristics of the candidate content, it is difficult to find the problems that many designers may ignore in the real game situation. For example, in the Sokoban game, the indicator to decide whether the level can run is not the position of the obstacle, but whether the player can push the box to the end through the combined action.

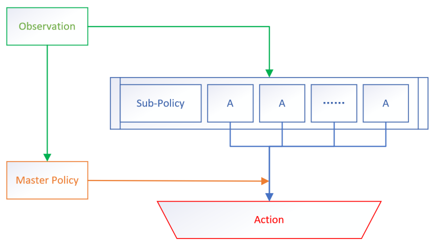

Figure 2: Flow chart of hierarchical reinforcement learning

Due to its cost and reliability, interactive evaluation is difficult to be applied to many procedural content generation in engineering, which is time-consuming and laborious, contrary to the purpose of this paper. In contrast, simulation-based evaluation does not need to set too many restriction rules like direct evaluation, it evaluates the candidate content through the information collected in the process of agent simulation. Because of its characteristics of simulating the actual game, the evaluation method based on simulation can better find the problems that players will encounter in the actual game, so it is more convenient and reliable. Therefore, we choose simulation-based evaluation as the evaluation method of sparse reward game procedural content generation.

However, because of the scattered and rare characteristics of sparse reward games, traditional intensive learning intelligence is difficult to get enough training from the environment. So the simulation evaluation of sparse reward games often takes a lot of time. In the simulation of sparse reward task with large space complexity, the agent cannot even complete the task. For example, in the case of the Sokoban game used in this paper, the agent can get the feedback of victory only when all boxes are pushed into the target, while the agent cannot get any feedback when exploring in the environment most of the time.

To effectively evaluate the generated candidate content, we choose hierarchical reinforcement learning method as the evaluation method. Hierarchical reinforcement learning is produced to deal with the sparse reward problem in practical problems, as shown in Figure 2. It splits the complex problem and generates several easier subtasks. The observation value is assigned to the subtask strategy and the main task strategy. Through the combination of the two, the corresponding action can be generated. After obtaining the action, the agent can interact with the environment, and then update the strategy to implement the simulation. For each layer of reinforcement learning, we use Q-Learning algorithm, and its pseudo code is shown in Algorithm1. Through the exploration of environment by low-level agents, we can get the reward feedback needed by high-level agents for training, and then achieve the simulation goal.

In this paper, we take Sokoban game as an example. The top-level problem is how to push all boxes to the target point, the first level sub goal is how to make the agent push a box to the target point, and the second level sub goal is how to make the agent push the box. By applying two-level reinforcement learning, we decompose complex problems into multi-level simple problems to solve. By solving the simulation problem, we can use the search-based procedural content generation method to generate the sparse reward game content automatically.

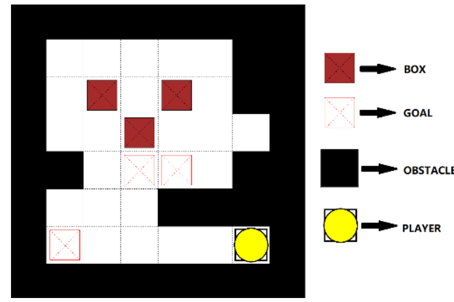

Figure 3: Map of box pushing game

Algorithm 1 Q-Learning

Initialize Q(s, a) arbitrarily;
Repeat for each episode:
  Initialize s;
  Repeat for each step of episode:
    Choose a from s using policy derived from Q;
    Take action a, observer, s';
    $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_{a'} Q(s',a') - Q(s,a)],\ s \leftarrow s'$;
  Until s is terminal;
  If Terminal-Condition then:
    Return succeed
  End
End
Return Fail

4 EXPERIMENT

In this paper, we take the game of Sokoban as an example. Sokoban is a typical sparse reward game. It can get positive feedback of game victory only when all boxes are pushed to the right target, while a single box is pushed to a target position and cannot get positive feedback. In addition, there are few elements to generate the game. Because it is a typical sparse reward game with simple structure,, we think it is suitable to use the Sokoban to verify our method.

In this experiment, as shown in Figure 3, we use 6 * 6 size of the Sokoban game map and include three groups of boxes and target points. The purpose of this experiment is to study the feasibility of search based procedural content generation method in the field of sparse reward games. Therefore, in order to obtain the experimental results quickly, we use smaller maps and fewer target points and target boxes for the experiment.

In the genetic algorithm part, in order to ensure that there is enough different genetic information in the population, we found that when the population capacity is set to 30, we can ensure that we can obtain the individuals with the required fitness in a small number of breeding times. In addition, considering that the population tends to be similar after about 10 times of reproduction, we set up two populations to ensure the simulation time.

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma max_{a'} Q(s',a') - Q(s,a)\right] \quad (1)$$

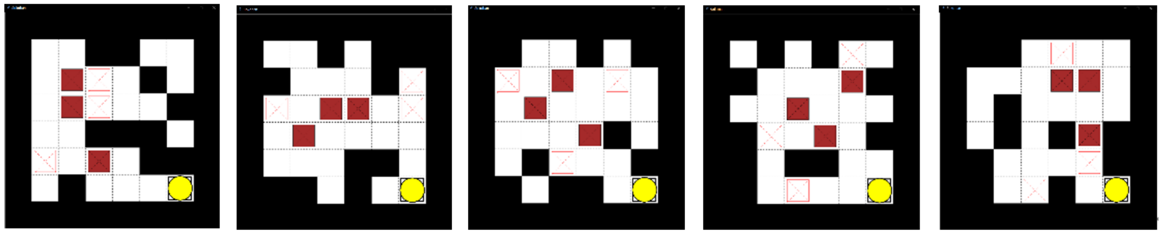

Figure 4: Levels obtained after five simulations

Finally, in the simulation-based evaluation part, we use hierarchical reinforcement learning as the evaluation method, and divide the target task into two layers. The low-level reinforcement learning is used to increase the reward of high-level reinforcement learning in the environment. The reward formula of the underlying reinforcement learning state is shown in equation 1. Among them, Q is the value, α is the learning rate of each agent learning, s is the current state, a is the selected action, and γ is the reward decay value. To ensure that the agent can make more exploration attempts and ensure the simulation efficiency under the condition of as high greedy as possible, we set the parameters as $\alpha = 0.01$, $\gamma = 0.9$. Like low-level reinforcement learning, high-level reinforcement learning obtains reward feedback according to the results of low-level reinforcement learning to implement the complete simulation process. To make agents pay more attention to long-term reward, we set the future reward attenuation value γ to 0.95, while the learning rate α remains unchanged.

After five simulations, we get five levels, as shown in Figure 4. After manual testing, we verify that all generated levels are runnable. Each level contains 3 targets and 3 boxes, and the number of obstacles is 10, which meets the generated target.

By observing the five generated results, we find that the details are different, which is different from the fixed design ideas of human designers. These generated contents have a certain degree of complexity under the condition of ensuring that the level can run, and the generated puzzles are different.

5 CONCLUSION

In this paper, a method based on search and hierarchical reinforcement learning is proposed to solve the problem that it is difficult for agent to obtain positive reward in the process of exploration. Different from the traditional method of automatic generation of procedural content, to generate more rich and novel game content, we adopt the method of procedural content generation based on search, and further extend the method. We introduce hierarchical reinforcement learning as a simulation agent to realize the real evaluation of candidate content. In this paper, Sokoban game is used to verify the method. After analyzing all the generated results, we find that it not only can run, but also has the advantage of novelty that traditional generation methods do not have, which shows that the method is feasible. In the future work, we can extend this method to other sparse reward game procedural content generation.

REFERENCES

[1] G. N. Yannakakis and J. Togelius. 2018. Artificial Intelligence and Games. Springer,New York.
[2] Togelius J, Yannakakis G N, Stanley K O, *et al.* Search-based procedural content generation: A taxonomy and survey[J]. IEEE Transactions on Computational Intelligence and AI in Games, 2011, 3(3): 172-186.
[3] Bento D S, Pereira A G, Lelis L H S. Procedural generation of initial states of sokoban[J]. arXiv preprint arXiv:1907.02548, 2019.
[4] C. Browne, "Automatic generation and evaluation of recombination games," Ph.D. dissertation, Queensland Univ. Technol., Queensland, Australia, 2008.
[5] A. Summerville *et al.*, "Procedural Content Generation via Machine Learning (PCGML)," in IEEE Transactions on Games, vol. 10, no. 3, pp. 257-270, Sept. 2018, doi: 10.1109/TG.2018.2846639.
[6] Liu S, Chaoran L, Yue L, *et al.* Automatic generation of tower defense levels using PCG[C]//Proceedings of the 14th International Conference on the Foundations of Digital Games. 2019: 1-9.
[7] Baldwin A, Dahlskog S, Font J M, *et al.* Mixed-initiative procedural generation of dungeons using game design patterns[C]//2017 IEEE conference on computational intelligence and games (CIG). IEEE, 2017: 25-32.
[8] Raffe W L, Zambetta F, Li X, *et al.* Integrated approach to personalized procedural map generation using evolutionary algorithms[J]. IEEE Transactions on Computational Intelligence and AI in Games, 2014, 7(2): 139-155.
[9] Dietterich T G. The MAXQ Method for Hierarchical Reinforcement Learning[C]//ICML. 1998, 98: 118-126.
[10] Nachum O, Gu S, Lee H, *et al.* Data-efficient hierarchical reinforcement learning[J]. arXiv preprint arXiv:1805.08296, 2018.
[11] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," IEEE Trans. Affective Comput., 2011