

A Demonstration of Pathfinding-Based Puzzle Generation with Adaptive Difficulty

Matthew McConnell, Richard Zhao

Department of Computer Science, University of Calgary
Calgary, Alberta, Canada, T2N 1N4
matthew.mcconnell1@ucalgary.ca, richard.zhao1@ucalgary.ca

Abstract

In this demonstration paper, we showcase an adaptive puzzle-generation game designed to dynamically adjust puzzle difficulty in real-time for individual users. The game utilizes a genetic algorithm to procedurally generate pathfinding-based puzzles tailored specifically to each player’s skill level and interaction patterns. A player-modeling mechanism continuously monitors user behaviors and interactions, enabling the game to match puzzle complexity to each player’s abilities. By adaptively calibrating challenge levels, this system seeks to enhance player engagement, reduce frustration, and maintain an optimal difficulty balance.

Introduction

Maintaining player engagement in puzzle games is challenging due to varied skills and interaction patterns. These puzzles provide only a fixed set of challenges with a predetermined progression of difficulty (Scirea 2020). Static difficulty can cause frustration or boredom, affecting retention (Zohaib 2018). Dynamically adjusting difficulty is critical, significantly impacting user experiences.

This demonstration features a puzzle game inspired by *Cosmic Express*¹ (Hazelden, Davis, and Tyu 2017), in which the player draws a path to guide a container from start to finish, picking up and delivering cargo boxes to designated locations. To successfully complete a puzzle, the path must allow the container to travel along the path from the starting point to the ending point without going backward, while ensuring each cargo box is delivered to a drop-off location. If the container is unable to reach the ending point, or if any cargo box is not delivered correctly when the container reaches the end, the puzzle fails and the player must redraw the path. This type of game was selected due to its accessibility for beginners, while simultaneously fostering the development of planning, critical thinking, and problem-solving skills. We refer to our game as the *Adaptive Problem-Solving Game (APSG)*², detailed in McConnell and Zhao (2025). Our game was developed in Unity.

AI Approach and Implementation Details

In our APSG, we generate the puzzles using a genetic algorithm, which produces a set of paths and special points representing the puzzle solution. These set of points are stored as an $n \times n$ grid, based on the size of the puzzle. A player model provides the genetic algorithm a difficulty level to target. We set the target range as an integer between one and ten. Figure 1 shows some examples of generated puzzles of various difficulty levels. The genetic algorithm targets the specified difficulty using a fitness score calculated from the following factors:

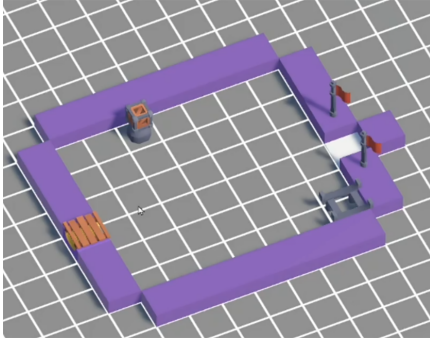
- The length of the solution path in grid cells.
- Number of corners in the solution path.
- Number of empty spaces in the puzzle.
- Number of cargo boxes for pick-up.
- Number of cargo boxes in orthogonal positions.

For all factors except the number of empty spaces, a larger number implies a higher difficulty. For the number of empty spaces, it is the inverse - a larger number implies a lower difficulty, as more spaces potentially allow for more ways to solve the puzzle. Having multiple possible solutions reduces the difficulty.

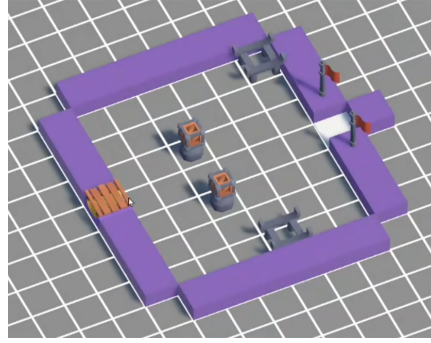
The genetic algorithm is based off of the NSFI-2POP algorithm (Scirea 2020), which has many benefits in constrained multi-objective optimization problems (Scirea 2020).

For the player model, we can specify hard constraints (that must be satisfied for a difficulty level change to occur) and soft constraints (that become factors in determining whether the difficulty should be changed). During testing, we determined that we would use one hard constraint - the *number of attempts before a solution*, as we set a hard limit on the number of attempts the player can have before they are unable to get a puzzle with a higher difficulty in the next game.

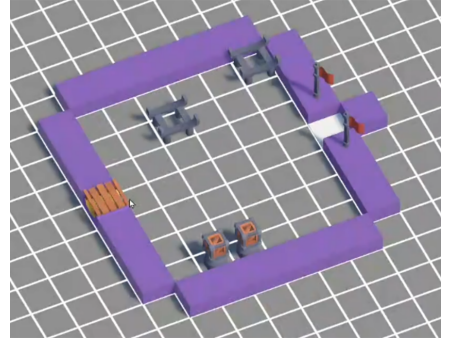
The soft constraints we used are *time taken to reach solution*, *number of backtracks (removing a portion of the puzzle to try again)*, *number of times the puzzle state was reset*, and *number of times the puzzle was almost solved (missing less than 25% of special points)*. The player model takes in the soft constraints score in tandem with the validity of passing the hard constraint, and suggests a new difficulty.



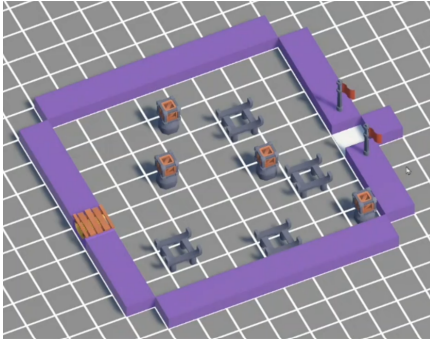
(a) Difficulty Level 1.



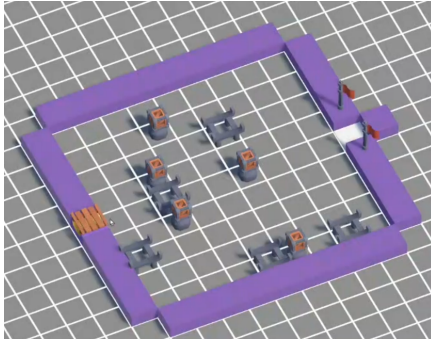
(b) Difficulty Level 2.



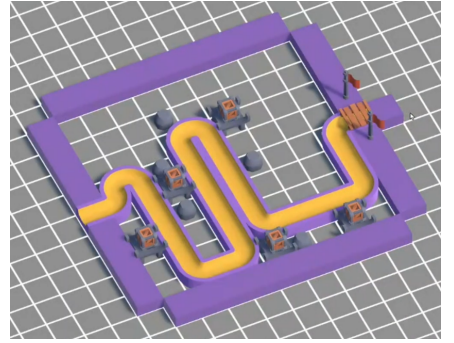
(c) Difficulty Level 3.



(d) Difficulty Level 4.



(e) Difficulty Level 5.



(f) Difficulty Level 5 with a solved path.

Figure 1: Examples of dynamically generated puzzles of various difficulties. The cargo boxes are shown in brown colors and the drop-off locations are shown in gray. (f) shows a solution path of the puzzle in (e).

For the genetic algorithm, the crossover function mixes the path and grid configurations of two parent puzzles to create new children. Each child inherits part of the path and grid structure from one parent, and the rest from the other - aiming to blend traits and explore new puzzle variations. The fitness function is used to evaluate the current analytical difficulty of a given puzzle based on the factors listed in the previous section. Details of the genetic algorithm implementation can be found in McConnell and Zhao (2025).

Discussions

The generation of playable puzzles within a practical time frame for real-time gameplay presents a significant challenge, requiring a balance between computational efficiency and puzzle variability. In our implementation, a typical puzzle can be generated in approximately seven seconds of runtime on a machine equipped with an Intel Core i5-9300H @ 2.40 GHz CPU, with more complex and difficult puzzles requiring additional computation time compared to simpler ones. This performance enables the feasibility of real-time puzzle generation. However, as parameters such as population size, generation limits, and maximum grid dimensions are increased, the system is capable of producing substantially larger and more complex puzzles, albeit at the cost of considerably longer runtimes.

The systematic approach to adaptive puzzle generation

developed in this work has the potential to be extended to more complex and non-linear puzzle types. For example, puzzles involve mathematics, logic, or computational thinking, represent promising opportunities for broadening the applicability of the system. A comparable framework could be employed to dynamically generate practice problems tailored to grade-school curricula, particularly within mathematics and programming, thereby supporting personalized and adaptive learning experiences to enhance engagement in learners and reduce frustration.

Related Works

Dynamic difficulty adjustments in games have been explored in the past. Zohaib (2018) provides a review of past techniques, including probabilistic methods, dynamic scripting, search methods such as Upper Confidence Bound for Trees (UCT), neural networks, and reinforcement learning, etc. More recently, Mortazavi, Moradi, and Vahabie (2024) further expand the categories to include control theory and regression models, and divide them into rule-based and data-driven approaches. Data-driven player modeling has specifically been used by Zook and Riedl (2012), whereas rule-based approaches can take different forms, such as by specifying how in-game skills build on each other (Sarkar and Cooper 2020; Stoneman, Miller, and Cooper 2022). In serious games, dynamic adaption can be done to adjust to dif-

ferent play styles and learning styles (Lindberg and Laine 2018), among other usages. Our work adds to the research by exploring the different constraints that could factor into such dynamic systems.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant, the NSERC Canada Graduate Scholarships (CGS-M), and the Alberta Graduate Excellence Scholarship (AGES). We thank members of the Serious Games Research Group and the reviewers for their feedback.

References

- Hazelden, A.; Davis, B.; and Tyu. 2017. Cosmic Express.
- Lindberg, R. S.; and Laine, T. H. 2018. Formative evaluation of an adaptive game for engaging learners of programming concepts in K-12. *International Journal of Serious Games*, 5(2): 3–24.
- McConnell, M.; and Zhao, R. 2025. From frustration to fun: an adaptive problem-solving puzzle game powered by genetic algorithm. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 21.
- Mortazavi, F.; Moradi, H.; and Vahabie, A.-H. 2024. Dynamic difficulty adjustment approaches in video games: a systematic literature review. *Multimedia Tools and Applications*, 83(35): 83227–83274.
- Sarkar, A.; and Cooper, S. 2020. Evaluating and comparing skill chains and rating systems for dynamic difficulty adjustment. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, 273–279.
- Scirea, M. 2020. Adaptive puzzle generation for computational thinking. In *International Conference on Human-Computer Interaction*, 471–485. Springer.
- Stoneman, A. D.; Miller, J. A.; and Cooper, S. 2022. Effects of player-level matchmaking methods in a live citizen science game. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 18, 199–206.
- Zohaib, M. 2018. Dynamic difficulty adjustment (DDA) in computer games: A review. *Advances in Human-Computer Interaction*, 2018(1): 5681652.
- Zook, A.; and Riedl, M. 2012. A temporal data-driven player model for dynamic difficulty adjustment. In *Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment*, volume 8, 93–98.