Evolutionary Procedural 2D Map Generation using Novelty Search

Andreas Scheibenpflug[1a,2], Johannes Karder[1a], Susanne Schaller[1b], Stefan Wagner[1a], Michael Affenzeller[1a,2]

[1]University of Applied Sciences Upper Austria
[a]Heuristic and Evolutionary Algorithms Laboratory
[b]Bioinformatics Research Group
Softwarepark 11, 4232 Hagenberg, Austria

[2]Johannes Kepler University Linz
Institute for Formal Models and Verification
Altenberger Strasse 69, 4040 Linz, Austria

{ascheibe, jkarder, sschalle, swagner, maffenze}@heuristiclab.com

ABSTRACT

This paper presents an evolutionary approach to procedural content generation of 2D maps for computer games. To provide better adaptability to the map designer's vision, user preference is incorporated to guide the algorithm. A cooperative method utilizes novelty search as a source of diverse solutions, which are then further optimized by multiple, subsequent genetic algorithms. We compare the results to a second approach based on multi-objective optimization, which takes the two conflicting goals of optimizing towards user preference and finding novel solutions as objective functions to build a Pareto front of maps.

Keywords

Procedural Content Generation; Search-based Procedural Content Generation; Novelty Search; Genetic Algorithm; HeuristicLab

1. INTRODUCTION

Search-based procedural content generation (PCG) [3] is an approach for (semi-)automating the content generation process and therefore lowering game production costs by using algorithms and artificial intelligence techniques. It is desirable to allow designers to oversee and interfere with the content generation process if needed. We present two approaches for generating and evolving maps for the video game Hedgewars[1] using evolutionary algorithms. For control over the generated content, the map designer must provide a sketch of a map. The first proposed algorithm is a cooperative approach where novelty search [2] and a memory of previously found best solutions are used for repeated seeding of a standard genetic algorithm (GA) that optimizes towards the designer's sketch. The second approach is based on multi-objective optimization combining the novelty of a solution and its difference from the sketch into objective functions.

We compare the two proposed techniques and show the achieved improvements compared to pure novelty search and optimization. The achieved quality as well as coverage of the solution space is analyzed to show different strength of the approaches. The results show that the cooperative approach has proven to be a good choice for generating maps as it can be configured to either produce more novel or more similar results.

[1]http://www.hedgewars.org/

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: http://dx.doi.org/10.1145/2908961.2909047

2. EVOLUTIONARY PROCEDURAL CONTENT GENERATION

Novelty search is a technique for preventing premature convergence. The hypothesis of novelty search is that the trajectory to good solutions may lead through multiple, lower quality areas in the search space. It maximizes the behavioral distance of discovered solutions and is able to evolve a wide range of different solution candidates. A solution candidate S_n is a polygon represented as a set of points $P(x, y)$: $S_n = \{P_1, P_2, P_3, \ldots, P_n\}$ in the Cartesian coordinate system. The fitness function is defined as the Euclidean distance between a solution candidate $S(x, y)$ and the sketch $T(x, y)$:

$$Quality(S, T) = \sqrt{(S(x) - T(x))^2 + (S(y) - T(y))^2} \quad (1)$$

The novelty is measured as the average Euclidean distance between the behavioral characterization ($BCDist$) of one individual S to its k nearest neighbors S_i:

$$Novelty(S) = \frac{1}{k} \sum_{i=0}^{k} BCDist(S, S_i) \quad (2)$$

To assess the behavior of an individual, measures such as e.g., its diameter, area, perimeter or number of points are used. The main objective is to find shapes that are pleasing to the user but also incorporate a novel aspect. For the game, the resulting geometry is filled by a random color and exported. The game uses the polygon as ground and places players on it.

A new algorithm that combines novelty search with a standard genetic algorithm in a two step approach is proposed. Every n generations, a new genetic algorithm is started and its population is created from the solution archive, solutions

from the previous executions (if available) and filled up with randomly generated solutions. The second approach is based on multi-objective optimization, where an NSGA-II [1] tries to minimize the fitness function (Equation (1)) and maximize the behavioral distance (Equation (2)).

3. EXPERIMENTS

We compare the two different approaches and additionally conduct two experiments with pure novelty search and a standard genetic algorithm. The cooperative approach has two configurations: Configuration I (Conf. I) has a higher rate of optimization and configuration II (Conf. II) uses a higher degree of novelty search. Each configuration is repeated 20 times. Three different problem instances (i.e. desired level designs) are specified, containing 6 (level 1), 15 (level 2) and 25 (level 3) points. All experiments have been performed using the optimization environment HeuristicLab[2] [4].

4. RESULTS AND CONCLUSION

As a baseline, we first show the results of the standard genetic algorithm and novelty search. The GA focuses on those areas in the space where the points of the map are located, whereas the points found by novelty search are more equally distributed over the whole search space. The cooperative approach allows to control the solution space coverage to look more like the one of the genetic algorithm or novelty search, depending on the used configuration. This can be seen in Figure 1.

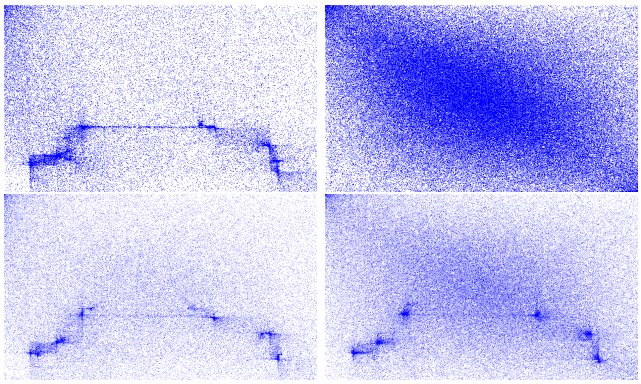

Figure 1: Sample solution spaces and their coverage for level 1. On the upper left the genetic algorithm, on the upper right novelty search, in the lower area the cooperative approach (Conf. I left, Conf. II right).

Figure 2 shows the best qualities achieved by the GA, novelty search and the cooperative approach for all three problem instances. It shows that, independent of configuration, the first executions of the genetic algorithm bring the most quality improvement. The subsequent runs then fine tune the results based on previously discovered solutions and bring smaller quality improvements. In Conf. I the GA runs for more generations and therefore generally generates better results in terms of quality compared to configuration II. The NSGA-II does not reach the qualities that the GA

[2]http://dev.heuristiclab.com

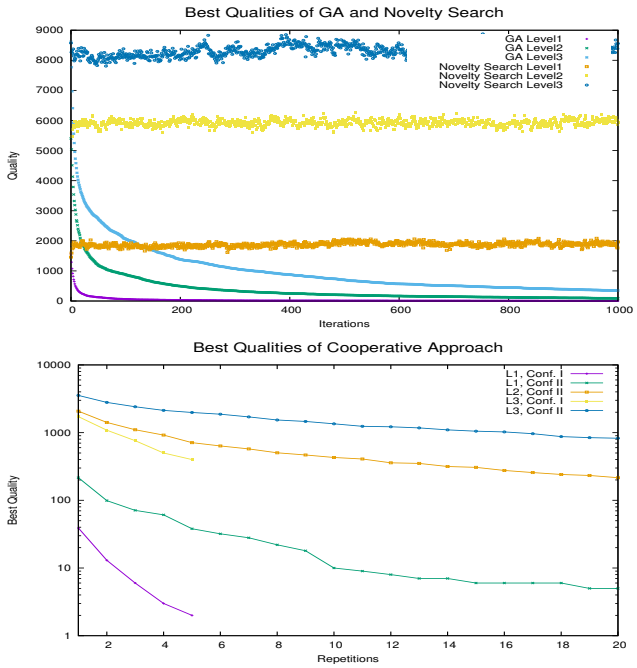

Figure 2: Average best found qualities for the GA, novelty search and the cooperative approach.

reaches. It is also outperformed by both configurations of the cooperative approach in terms of quality. Compared to all other algorithms, the NSGA-II produces a lot less unique individuals and has a higher number of average occurrences of points that are discovered multiple times. It seems therefore to be not so well suited for this application.

5. ACKNOWLEDGMENTS

The work described in this paper was done within the COMET Project Heuristic Optimization in Production and Logistics (HOPL), #843532 funded by the Austrian Research Promotion Agency (FFG).

6. REFERENCES

[1] K. Deb, A. Pratab, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[2] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[3] J. Togelius, G. Yannakakis, K. Stanley, and C. Browne. Search-based procedural content generation: A taxonomy and survey. *Computational Intelligence and AI in Games, IEEE Transactions on*, 3(3):172–186, 2011.

[4] S. Wagner, G. Kronberger, A. Beham, M. Kommenda, A. Scheibenpflug, E. Pitzer, S. Vonolfen, M. Kofler, S. Winkler, V. Dorfer, and M. Affenzeller. *Advanced Methods and Applications in Computational Intelligence*, volume 6 of *Topics in Intelligent Engineering and Informatics*, chapter Architecture and Design of the HeuristicLab Optimization Environment, pages 197–261. Springer, 2014.