Using Ant Colony Optimisation for map generation and improving game balance in the Terra Mystica and Settlers of Catan board games

Rommel Dias Saraiva
University of Fortaleza
Fortaleza, Brazil
rommelds@unifor.br

Alexandr Grichshenko
Innopolis University
Innopolis, Russia
a.grishchenko@innopolis.university

Luiz Jonatã Pires de Araújo
Innopolis University
Innopolis, Russia
l.araujo@innopolis.university

Bonfim Amaro Junior
Federal University of Ceará
Russas, Brazil
bonfimamaro@ufc.br

Guilherme Nepomuceno de Carvalho
Federal University of Ceará
Russas, Brazil
guilhermenepomuceno@alu.ufc.br

ABSTRACT

Game balancing is one of the most challenging features to be implemented in a typical game design process. Approaches for evaluating and achieving game balancing include extensive playtesting - which typically requires several iterations of games with subtle adjustments in the components and adopted strategies that resemble brute force - and algorithmic solutions that use qualitative and measurable design goals when developing game components. The literature contains examples of methods that employ artificial intelligence to generate maps in computer games that offer balanced and fairness of starting conditions for the players. The use of such methods for tabletop games, however, has been scarce in the academic literature, for the best of the authors' knowledge. This paper investigates the application of the ant colony optimisation metaheuristic to generate content and improving game balance for two well-known tabletop games, namely, Terra Mystica and Settlers of Catan. The resultant configurations satisfy complex game-dependent requirements while optimising a model for game balancing. Moreover, the results showed to be promising when compared with existing game maps and setup.

CCS CONCEPTS

• **Design and analysis of algorithms** → Evolutionary algorithms; • **Mathematical optimization** → Bio-inspired optimization; • **Algorithmic game theory and mechanism design** → Representations of games and their complexity; • **Mathematics of computing** → Combinatorial optimisation.

KEYWORDS

Ant colony optimisation, Terra Mystica, Settlers of Catan, procedural content generation

ACM Reference Format:
Rommel Dias Saraiva, Alexandr Grichshenko, Luiz Jonatã Pires de Araújo, Bonfim Amaro Junior, and Guilherme Nepomuceno de Carvalho. 2020. Using Ant Colony Optimisation for map generation and improving game balance in the Terra Mystica and Settlers of Catan board games. In *FDG'20: Foundations of Digital Games, September 15–18, 2020, Malta.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3402942.3409778

1 INTRODUCTION

Ensuring game balancing is one of the most crucial and non-trivial tasks faced by game developers [17]. In the past decades, many computational techniques have been proposed to optimise map generation, ranging from employing cellular automaton [35] to various machine learning algorithms [29] and playtesting resulting content using artificial agents [36]. The results of applying methods of combinatorial optimisation have generally been encouraging. However, while studies have been able to produce playable maps, they acknowledged that issues with balance were not satisfactorily addressed [23] [21]. One of the reasons is the complexity of corresponding objective functions and non-linearity of constraints used in map generations [9].

The recent literature shows that evolutionary techniques have been the prevalent algorithmic solution for procedural content generation (PCG) for games[33]. The main approaches for PCG include collecting feedback from human players, direct evaluation based on preexisting knowledge about the content and playtesting with artificial agents [33]. Most of the research in this field focuses on playtesting by humans, with significantly less research on the use of AI for playtesting. Moreover, digital games have received significantly more attention in the literature than board games regarding the automatic design of components [32].

The presence of complex features for implementing game balancing often require the use of artificial intelligence (AI) methods. The aim of the paper is to create balanced maps for tabletop games using

FDG'20, September 15–18, 2020, Malta

ACM ISBN 978-1-4503-8807-8/20/09…$15.00
https://doi.org/10.1145/3402942.3409778

an Ant Colony Optimisation (ACO) algorithm. To the best of authors' knowledge, the use of this particular evolutionary algorithm has not been reported in the literature. In this study, we consider two popular modern board games: Settlers of Catan (or simply Catan) and Terra Mystica (TM). Catan is a multiplayer Euro-style game in which players assume the role of settlers aspiring to build their holdings by trading valuable resources. The map in Catan consists of 19 hexagons arranged in a grid as shown in Fig. 1. Each hexagon represents one of five available resources: wood, brick, sheep, wheat or stone. The number in each hexagon corresponds to the required result of two D6 dice roll. TM also belongs to the class of Eurogames, which emphasises strategic development and careful resource management. Players control a faction striving to accumulate resources and spread their influence across the board. TM map is comprised of 113 tiles (hexagons) as presented in Fig. 2. One of the following terrain types can be assigned to each tile: desert, plains, swamp, lake, forest, mountain, wasteland and river.

Figure 1: An example of a map configuration for Settlers of Catan.

Figure 2: Original TM map (Score = 9.0)

The remaining part of this paper is organised as follows. Section 2 outlines map generation and playtesting techniques reported in the literature, and overviews the ACO metaheuristics and some of its industrial applications. Section 3 describes the implementation of ACO and the design of objective functions for TM and Catan maps. Next, Section 4 presents the results of applying ACO to both problems. Then, Section 5 discusses the results and the implications for game development. Lastly, Section 6 summarises the contributions of this paper and reflects on possible future research directions.

2 RELATED WORK

This section presents a brief review of studies on AI techniques applied to the designing of digital and board games. We consider two important areas of game development: playtesting and PCG. This section also reviews the ACO metaheuristic and some examples of its use in industrial applications.

2.1 Artificial intelligence in game development

The required time and the cost associated with human playtesting have motivated developers and researchers to consider methods for automating this process. One of the most frequently used algorithms in this field is the Monte-Carlo tree search (MCTS) [2]. MCTS is a greedy algorithm for discovering the best decision based on outcomes it had learned [4]. It starts by making random decisions, computing their results and then constructing a search tree based on the results. Several studies aiming to develop an AI player have employed variations of MCTS. Examples of board games addressed by such an approach include Carcassonne [16] and Settlers of Catan [6, 30]. Although these studies succeeded in implementing an agent capable of making complex moves, they agree that a relatively experienced human player can beat AI agents. Borovikov et al. argued that one of the reasons for the underperformance of AI techniques is that it often based on mimicking human behaviour during playtesting [2].

One of the prevalent approaches in the recent literature on automatic PCG is the use of evolutionary algorithms (EA) [26]. It typically requires the design of an evaluation function for quality assessment and a metaheuristic to evolve the candidate solution for a certain problem [33]. Most of the research in this field focuses on map or levels generation for digital games. Examples of this approach include the use of genetic programming for implementing an efficient terrain generator[12], obtaining playable and balanced maps in Siphon using genetic algorithms [22], employing multi-objective heuristic to generate maps for real-time strategy game [31] and applying Tabu search heuristic for generating balanced maps for Terra Mystica [8, 14]. Despite the good results obtained by these approaches for several games in different media, these studies agree that there is still some degree of game imbalance left.

There has been some research on algorithmic approaches applied to Settlers of Catan in the domain of AI playtesting and generating strategies. For example, Pfeiffer employed reinforcement learning of strategies based on tree models for mapping state-action values [24]. Another example was shown in Szita et al.'s study on the use of Monte-Carlo Tree Search (MCTS), which showed to be a competitive method to play against computer-controlled and human players [30]. Smith and Mateas employed answer set programming (ASP) and explored the possibility of generating maps via a program comprised of logical facts about a desirable map [28]. One more promising research direction is the application of wave function collapse (WFC), particularly to problems involving 3D terrains [18]. While comparatively extensive work has been directed to Settlers, little research has been dedicated to Terra Mystica, probably related to the presence of a larger universe of actions and game states.

Brown and Marco Scirea surveyed algorithmic PCG methods under restrictions on computational resources for wargames and other board game types [3]. Interestingly, most of the literature focuses on PCG for computer games. For example, several methods have been introduced for generating material for computer role-playing games including maps [19, 25] and story [15, 34]. The increasing number of alternatives gives rises to the need for orchestrating or integrating different techniques and domains, as discussed by Liapis et al. [20]. This study, however, focuses on the demonstration of the usefulness of the ACO metaheuristic as an approach for implementing PCG in tabletop games and the formalisation of the objective function and constraints in the domain of Terra Mystica and Settlers of Catan.

2.2 Ant colony optimisation

ACO is a stochastic algorithm inspired by real ants who help each other find the best food source with the use of pheromone trails [7]. ACO has been one of the most often used metaheuristics in problems that can be reduced to searching an optimal path in graphs [10]. It has been demonstrated to be an effective technique for solving complex optimisation problems. Noteworthy examples include solving the well-known travelling salesman problem (TSP) [11], the vehicle routing [5] and job scheduling [13]. ACO has also been widely used in the field of bioinformatics for studying DNA sequencing [1] and protein folding [27].

3 METHODOLOGY

The following section outlines the design of the objective functions for TM and Catan based upon inferred requirements to achieve map balance. Furthermore, the detailed description of ACO implementation for this domain is provided.

Since the aim of the paper is to increase balance in maps, some interesting scenarios that could potentially improve the user experience will be omitted (e.g. two identical tiles in Catan next to each other can provide for exciting trading, but would harm balance if the player monopolises the resource). The requirements were elicited by conducting thorough discussions with members of a large online gaming community [1] and tabletop club in Innopolis University.

3.1 Constraints and objective for maps in Terra Mystica

A balanced map in TM has to satisfy requirements regarding the placement of tiles. For this study, we consider four requirements that have the most significant effect on the playability and the balance of the maps. Firstly, it has to ensure that no land hexagon has a neighbor of the same terrain type to prevent some starting positions from being disproportionally advantageous (REQ1). The second requirement states that river hexagons should have between one and three river neighbours (REQ2). This condition prevents the occurrence of lakes, which impede the development of some factions, e.g. Mermaids. Furthermore, it is necessary that all river hexagons form a single connected component (REQ3). Along with REQ2, this requirement promotes the resemblance between generated and original maps. The last requirement involves terraforming - one of the most crucial actions that a player can take to advance his/her faction. Terraforming a terrain type to a different one costs a certain number of spades, which forces a player to spend valuable resources. For example, Fig. 3 demonstrates how two spades are required to terraform a mountain (grey) terrain to a yellow (desert) terrain. To facilitate players' development in the early stages of the game, a land hexagon should have a neighbour that can be terraformed by spending exactly one spade (REQ4).

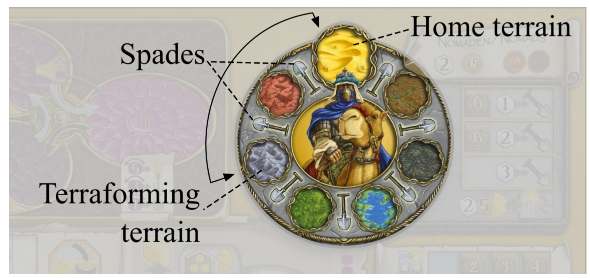

Figure 3: A player board indicating the terraforming circle.

The objective function for modelling balance in a TM map computes the number of violations of each requirement and returns a linear sum of all violations. The aim of the optimisation algorithm is to minimize the objective function to a global optimum of 0, i.e. no requirement is violated. The function is defined over 113 dimensions, where each dimension represents a single hexagon on the map and can assume one of eight possible discrete values (number of terrain types). Given that a TM consists of 77 land tiles (11 of each land terrain type) and 36 river tiles, the search space contains approximately $3.7 * 10^{89}$ potential solutions.

3.2 Achieving map balance in Settlers of Catan

A map in Catan consists of two main elements that contribute to the game balance: tiles of 5 different types and the probability assigned to each tile. The first step for achieving a balanced map is to ensure that two or more tiles of the same type are not located next to each other (REQ1). This requirement promotes fair distribution of tiles across the map and prevents a single player from monopolising a certain resource. The second requirement deals with the probabilities of obtaining a resource. In a perfectly balanced game, the probability of receiving a particular resource should be proportional to the number of tiles with this resource on the game board. Wheat, sheep and wood form one group of resources with an equal number of tiles; stone and brick - another. Therefore, among elements in each group, assigned probabilities of two D6 rolls should be approximately the same (REQ2). For a better understanding of REQ2, see table 1 with the example of a tile distribution that satisfies it. The last requirement ensures that tiles with D6 rolls of 8 and 6 are not situated next to each other, as these outcomes have the highest

[1] https://boardgamegeek.com/thread/2208002/designing-modular-board-tm-using-algorithms

probability of occurring and would result in a highly unbalanced region of the map (REQ3).

Table 1: REQ2 arrangements

Tile No.	Brick		Stone	
	D6 rolls	Probability	D6 rolls	Probability
1	11	0.056	3	0.056
2	9	0.111	10	0.083
3	11	0.056	4	0.083
total	10.3	**0.223**	5.67	**0.222**

The design of the evaluation function for Catan follows a similar pattern as for TM. The function determines how many times each requirement was violated and returns the total number of violations as the map score. The goal of the algorithm is to minimise the function to the global optimum of 0.

3.3 An ACO implementation for Terra Mystica

Each ant in the ACO is an abstract computational agent that iteratively searches for promising solutions. For TM, each candidate solution is represented by a graph of 9×13 vertices (the hexagons). Each vertex represents a board position for which a terrain type must be chosen and each edge represents a connection between neighbouring vertices. As a consequence, the target problem is addressed by simulating several artificial ants moving on the graph in order to figure out the best arrangement of terrain types for 113 hexagons. Vertices at the positions $\langle 2, 13\rangle$, $\langle 4, 13\rangle$, $\langle 6, 13\rangle$ and $\langle 8, 13\rangle$ correspond to blank terrain types due to the shape of the map.

At each iteration of the ACO, ants select the next terrain type to be assigned to a vertex following a stochastic mechanism that is guided by the pheromone trail. When an ant is located in vertex i, the corresponding terrain type is selected among the available options. Assuming that terrain type j has the availability to be chosen, it can be selected with a probability that is proportional to the value stored in position (i, j) of the pheromone matrix. At the beginning of each iteration, each ant selects the terrain type for all positions. In general, the k-th ant, when in vertex i, selects terrain type j with probability p^k_{ij} given by (1).

$$p^k_{ij} = \frac{\tau^\alpha_{ij}}{\sum_{l \in N_i} \tau^\alpha_{il}} \tag{1}$$

where τ_{ij} is the amount of pheromone deposited in position (i, j), $\alpha \geq 0$ is an input parameter that controls the influence of τ_{ij} and N_i is the set of available terrain types. It is noteworthy that our ACO algorithm does not take into account any heuristic information. The pheromone trail is then updated after all ants generated the solutions, which can increase or decrease the level of trails. We use the following rule to update pheromone concentration:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_k \Delta\tau^k_{ij} \tag{2}$$

where ρ is the pheromone evaporation coefficient and $\Delta\tau^k_{ij}$ is the amount of pheromone deposited by the k-th ant, which is given by:

$$\Delta\tau^k_{ij} = \begin{cases} \frac{Q}{L_k}, & \text{if ant } k \text{ assigns terrain type } j \text{ to vertex } i. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where L_k is the score of the k-th ant and Q is a constant calibrated beforehand. After all iterations, the output indicates the best arrangement of terrain types found by ACO.

3.4 An ACO implementation for Settlers of Catan

For Settlers of Catan, each candidate solution is a map (e.g. Fig. 1) which can also be represented by a graph of 19 vertices and 42 edges, as shown in Fig. 4. Each vertex is a hexagon for which a tuple $\langle t, n\rangle$ must be selected, with type $t \in \{wood, wheat, sheep, brick, stone\}$ and number $n \in \{2, \ldots, 6, 8, \ldots, 12\}$. The vertices that represent two neighbours hexagons on a map are connected by an edge.

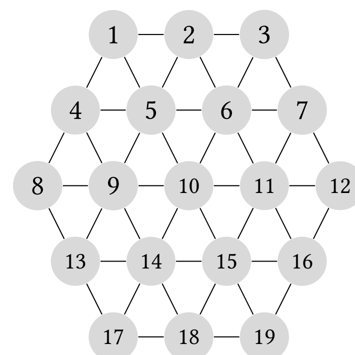

Figure 4: Graph representation for Settlers of Catan solutions.

Firstly, the algorithm generates all possible combinations of types and numbers. For example, for wood we have $\langle wood, 2\rangle$, $\langle wood, 3\rangle$, $\ldots$, $\langle wood, 6\rangle$, $\langle wood, 8\rangle$, $\ldots$, $\langle wood, 12\rangle$. Enumerating all possible combinations of types and numbers gives rise to a total of 51 tuples. This happens because we also consider a particular configuration, namely, $\langle desert, 7\rangle$. The algorithm aims to assign 19 tuples to the hexagons in such a way that it maximises balance.

Similarly to ACO for Terra Mystica, ants select the next tuple to be assigned to a vertex. A tuple j at the board position i can be selected with a probability proportional to the value stored in position (i, j) of the pheromone matrix. More specifically, each ant k, when in vertex $i = 1, \ldots, 19$, selects the tuple j according to (1). For Settlers of Catan, N_i is the set of available tuples, whereas α and τ_{ij} have the same interpretation as in Terra Mystica. The pheromone trails are then updated using (2) and (3).

4 RESULTS

The ACO algorithm for Terra Mystica was executed 30 times and the stopping criterion was satisfied when the running time of the program exceeds 5 minutes. The timeout constraint was selected based on the ACO convergence time in preliminary tests. The mean score obtained by the ACO across all runs was 147.1667 with the standard deviation of 92.3. On average, ACO achieved convergence after 48.6 seconds. The correlation between convergence time and the resulting score is shown in Fig. 5. Surprisingly, enabling the ACO to run for a longer period of time does not improve the overall results.

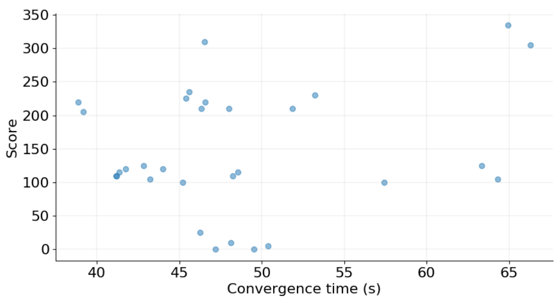

Figure 5: Convergence time vs. Score

Next, Fig. 6 shows how the score of the progression of the solution improves across the iterations for the best map in TM. Improvements after the generation 127 tend to be less significant. As a result, extending the running time beyond this time threshold would not be productive.

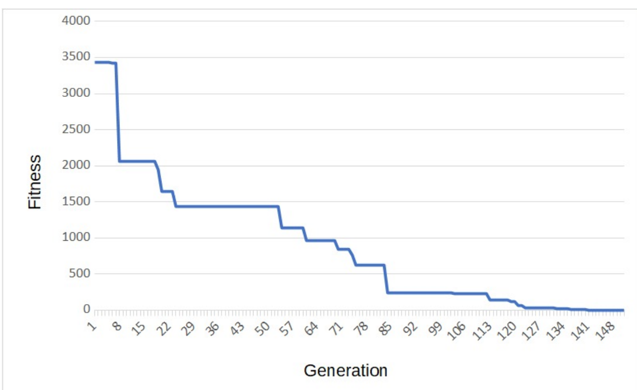

Figure 6: Score vs. generation for TM maps

ACO was able to obtain the optimal score of 0 for two instances. The best found TM map is shown in Fig. 7. The comparison between TM maps generated by ACO and existing maps considering requirement violations is shown in Table 2.

Table 2: Number of times each requirement for TM map is violated.

Map	Score (F_{tot})	REQ1	REQ2	REQ3	REQ4
Best ACO Map	0	0	0	0	0
Original Map	9	0	3	0	6
Fire and Ice	12	0	3	0	9
Fjords	9	0	0	0	9

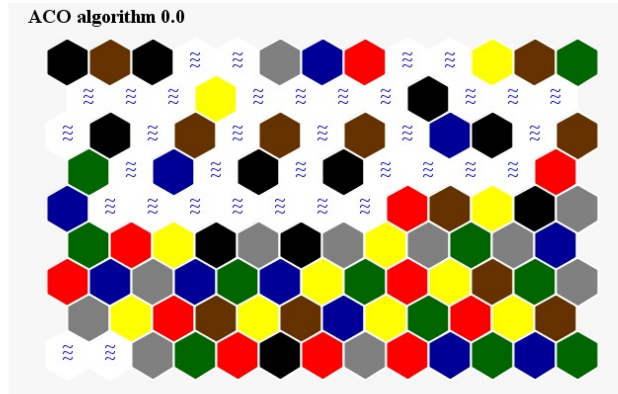

Figure 7: Best TM map (Score = 0.0)

Figure 8: Fire and Ice TM map (Score = 12.0)

Figure 9: Fjords TM map (Score = 9.0)

In order to be consistent and produce reliable analysis, the stopping criterion and number of runs for generating Catan maps were the same as for TM. Given a significantly smaller search space and less complex requirements, ACO performed much better when applied to Catan, achieving the optimal score of 0 for every instance. Average convergence time for Catan was also considerable smaller - 2.23 seconds. The best map for Catan is shown in Fig. 10.

Next, Fig. 11 shows how the score of the solution improves for the best map in Settlers of Catan. Similarly to TM, improvements

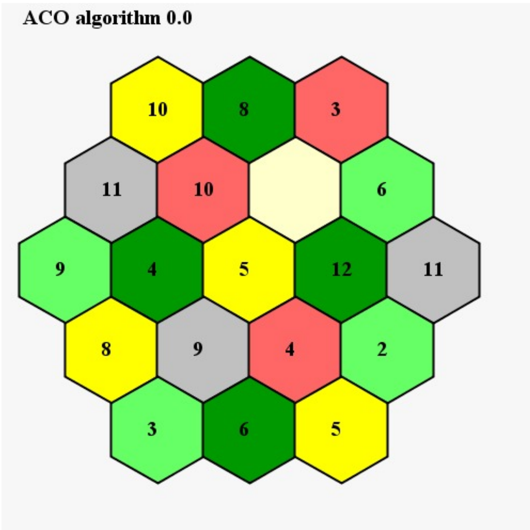

Figure 10: Best Catan map (Score = 0.0)

after relatively few generations tend to have less effect on the score of the map. Once more, extending the probing time for ACO beyond the generation 37 could lead to a waste of computational resources.

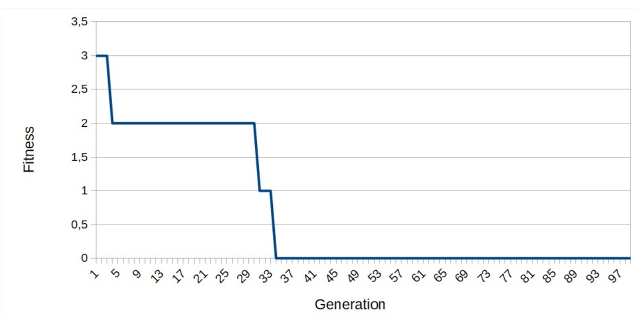

Figure 11: Convergence time vs. Score

5 DISCUSSION

Based on results described in Section 4, it is possible to observe that evolutionary heuristics such as ACO are capable of producing satisfactory PCG solutions in a reasonable time. As can be inferred from table 2, TM maps produced by ACO are superior (i.e. violate fewer requirements) to officially released maps. Such results can have major commercial implications, leading to potential new releases of more balanced maps with minimal cost of development and playtesting. Moreover, this approach could be used by regular players who want to experience new playing scenarios, thus constantly maintaining interest in the game. However, the analysis of the maps with optimal score exposed certain limitations in the used objective functions. For example, in Fig. 7 there are exactly two connected river components, which is acceptable according to the REQ3, but, there is a noticeable concentration of river hexagons on the top half of the map. Hence, many small islands (less than 4 land hexagons) are introduced, which favours only a limited number of factions. The issue implies that more comprehensive requirements (e.g. size of an island should be at least 5) are necessary to achieve fully playable and balanced maps. Also, a comprehensive playtesting of the generated maps by equally skilled players would be necessary to fully assess the impact of the requirements we introduce. Unlike in TM, the fitness function for Catan successfully captured requirements needed for balance, as no discrepancies were found in maps with optimal scores.

Besides the quality of the generated content, it is also important to consider the time required to produce acceptable solutions. Contrary to digital games, generation of board games' components are usually offline, thus minimisation of the runtime is not a priority. It is reasonable to expect that increasing the runtime for a search algorithm will result in a better final score. However, the results suggest that such reasoning cannot be applied here (see Fig. 5). From the figure we can see that almost no datapoints are located in the lower-right corner of the graph, corresponding to a supposed trade-off between time and score. Moreover, the worst obtained TM solution required nearly the longest time for ACO to converge (60 seconds), while satisfactory solutions converged much faster. The results imply that ACO is a suitable approach for improving map balancing. For TM and Catan, this could result in new content for online platforms like Tabletop Simulator on Steam.

This study ignores the effects that parameter tuning can have on the overall results since the employed hyperparameters successfully achieved optimal solutions. However, the authors recognise that by implementing more complex requirements into the evaluation functions of both TM and Settlers of Catan is likely to result in sub-optimal solutions. Therefore, the latter scenario would require tuning the parameters for the ACO, namely the number of ants, number of iterations and initial pheromone.

6 CONCLUSION

This study describes how the ACO can be employed for map generation in Catan and Terra Mystica board games. These are popular Eurogames with global competitions and an active community. The success of ACO in map generation addresses the demand of the gaming community for more engaging and balanced maps by demonstrating acceptable solutions with better characteristics than commercially available maps (see Table 2. This paper also presented the formalisation of requirements ensuring map balance, which lays the groundwork for further research on the use of metaheuristics for PCG.

One promising future research direction is the development of a comprehensive AI agent based on reinforcement learning (RL) for various PCG tasks. Studies of RL applications for generating maps in digital and board games could enrich the knowledge among the PCG community and enhance the designers' toolbox in creating innovative and engaging titles. Given the limitations of objective functions mentioned in the previous section, further research should focus on eliciting more comprehensive requirements and experimenting with a wider range of optimisation techniques. Lastly, board games offer an exciting sandbox for the use of AI and other advanced computer techniques (e.g. machine learning) for academics and practitioners.

REFERENCES

[1] Christian Blum, Mateu Yábar Vallès, and Maria J Blesa. 2008. An ant colony optimization algorithm for DNA sequencing by hybridization. *Computers & Operations Research* 35, 11 (2008), 3620–3635.

[2] Igor Borovikov, Yunqi Zhao, Ahmad Beirami, Jesse Harder, John Kolen, James Pestrak, Jervis Pinto, Reza Pourabolghasem, Harold Chaput, Mohsen Sardari, et al. 2019. Winning isn't everything: Training agents to playtest modern games. In *AAAI Workshop on Reinforcement Learning in Games*.

[3] Joseph Alexander Brown and Marco Scirea. 2018. Procedural Generation for Tabletop Games: User Driven Approaches with Restrictions on Computational Resources. In *International Conference in Software Engineering for Defence Applications*. Springer, 44–54.

[4] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* 4, 1 (2012), 1–43.

[5] Bernd Bullnheimer, Richard F Hartl, and Christine Strauss. 1999. An improved ant System algorithm for thevehicle Routing Problem. *Annals of operations research* 89 (1999), 319–328.

[6] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. 2008. Monte-Carlo Tree Search: A New Framework for Game AI.. In *AIIDE*.

[7] Alberto Colorni, Marco Dorigo, Vittorio Maniezzo, et al. 1992. An Investigation of some Properties of an" Ant Algorithm".. In *Ppsn*, Vol. 92.

[8] Luiz Jonatã Pires de Araújo, Alexandr Grichshenko, Rodrigo Lankaites Pinheiro, Rommel D Saraiva, and Susanna Gimaeva. 2020. Map Generation and Balance in the Terra Mystica Board Game Using Particle Swarm and Local Search. In *International Conference on Swarm Intelligence*. Springer, 163–175.

[9] Fernando de Mesentier Silva, Scott Lee, Julian Togelius, and Andy Nealen. 2017. AI-based playtesting of contemporary board games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*. ACM, 13.

[10] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. 2006. Ant colony optimization. *IEEE computational intelligence magazine* 1, 4 (2006), 28–39.

[11] Marco Dorigo and Luca Maria Gambardella. 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation* 1, 1 (1997), 53–66.

[12] Miguel Frade, Francisco Fernandéz de Vega, and Carlos Cotta. 2009. Breeding terrains with genetic terrain programming: the evolution of terrain generators. *International Journal of Computer Games Technology* 2009 (2009).

[13] Marc Gravel, Wilson L Price, and Caroline Gagné. 2002. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research* 143, 1 (2002), 218–229.

[14] Alexandr Grichshenko, Luiz Jonatã Pires de Araújo, Susanna Gimaeva, and Joseph Alexander Brown. 2020. Using Tabu Search Algorithm for Map Generation in the Terra Mystica Tabletop Game. *International Journal of Machine Learning and Computing (IJMLC, ISSN: 2010-3700)*.

[15] Ken Hartsook, Alexander Zook, Sauvik Das, and Mark O Riedl. 2011. Toward supporting stories with procedurally generated game worlds. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*. IEEE, 297–304.

[16] Cathleen Heyden. 2009. Implementing a computer player for Carcassonne. *Master's thesis, Department of Knowledge Engineering, Maastricht University* (2009).

[17] Ahmed Khalifa and Magda Fayek. 2015. Literature review of procedural content generation in puzzle games.

[18] Hwanhee Kim, Seongtaek Lee, Hyundong Lee, Teasung Hahn, and Shinjin Kang. 2019. Automatic Generation of Game Content using a Graph-based Wave Function Collapse Algorithm. In *2019 IEEE Conference on Games (CoG)*. IEEE, 1–4.

[19] Aliona Kozlova, Joseph Alexander Brown, and Elizabeth Reading. 2015. Examination of representational expression in maze generation algorithms. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 532–533.

[20] Antonios Liapis, Georgios N Yannakakis, Mark J Nelson, Mike Preuss, and Rafael Bidarra. 2018. Orchestrating game generation. *IEEE Transactions on Games* 11, 1 (2018), 48–68.

[21] Tobias Mahlmann, Julian Togelius, and Georgios N Yannakakis. 2012. Spicing up map generation. In *European Conference on the Applications of Evolutionary Computation*. Springer, 224–233.

[22] Jonas Juhl Nielsen and Marco Scirea. 2018. Balanced Map Generation Using Genetic Algorithms in the Siphon Board-Game. In *International Conference in Software Engineering for Defence Applications*. Springer, 221–231.

[23] Gonçalo Pereira, Pedro A Santos, and Rui Prada. 2009. Self-adapting dynamically generated maps for turn-based strategic multiplayer browser games. In *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*. ACM, 353–356.

[24] Michael Pfeiffer. 2004. Reinforcement learning of strategies for Settlers of Catan. In *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*.

[25] William L Raffe, Fabio Zambetta, Xiaodong Li, and Kenneth O Stanley. 2014. Integrated approach to personalized procedural map generation using evolutionary algorithms. *IEEE Transactions on Computational Intelligence and AI in Games* 7, 2 (2014), 139–155.

[26] Noor Shaker. 2016. Intrinsically motivated reinforcement learning: A promising framework for procedural content generation. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–8.

[27] Alena Shmygelska and Holger H Hoos. 2005. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC bioinformatics* 6, 1 (2005), 30.

[28] Adam M Smith and Michael Mateas. 2011. Answer set programming for procedural content generation: A design space approach. *IEEE Transactions on Computational Intelligence and AI in Games* 3, 3 (2011), 187–200.

[29] Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. 2018. Procedural content generation via machine learning (PCGML). *IEEE Transactions on Games* 10, 3 (2018), 257–270.

[30] István Szita, Guillaume Chaslot, and Pieter Spronck. 2009. Monte-carlo tree search in settlers of catan. In *Advances in Computer Games*. Springer, 21–32.

[31] Julian Togelius, Mike Preuss, and Georgios N Yannakakis. 2010. Towards multiobjective procedural map generation. In *Proceedings of the 2010 workshop on procedural content generation in games*. 1–8.

[32] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. 2010. Search-based procedural content generation. In *European Conference on the Applications of Evolutionary Computation*. Springer, 141–150.

[33] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3, 3 (2011), 172–186.

[34] Josep Valls-Vargas, Santiago Ontanón, and Jichen Zhu. 2013. Towards story-based content generation: From plot-points to maps. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*. IEEE, 1–8.

[35] Roland Van Der Linden, Ricardo Lopes, and Rafael Bidarra. 2013. Procedural generation of dungeons. *IEEE Transactions on Computational Intelligence and AI in Games* 6, 1 (2013), 78–89.

[36] Georgios N Yannakakis and Julian Togelius. 2011. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing* 2, 3 (2011), 147–161.