Exploring the flexibility of a design tool through different artificial agents

Gerard Serra, David Miralles, Maria Casals and **Míriam López**
Grup de recerca en Tecnologies Media
La Salle-Universitat Ramon Llull
Quatre Camins, 30 08022
Barcelona, Catalonia, Spain
gerard.serra@salle.url.edu

Abstract

Many machine learning approaches are focused on defining artificial agents able to find solutions to a certain problem given fixed design tools or parameters to optimize. In order to do that, creators must have certain knowledge of the solution space to define design parameters that ensure enough exploration allowing agent to find its best configuration. However, this approach may limit artificial agents since they are restricted by their initial conditions of a certain design problem. In addition, specific initial conditions also limits them to scale across multiple challenges.

In this paper, we explore how the definition of more general design tools can allow artificial agents to better explore the solution space and generalize through multiple design problems. To do that, we compare design artifacts produced by an artificial agent that learns to construct 2D shapes with a fixed number of pieces to another artificial agent that also learns to add or remove pieces from its design proposal. We demonstrate how by allowing more freedom in design, an artificial system is able to produce more novel artifacts with higher performances in multiple scenarios.

Introduction

Design can be described as a process of co-evolution of both problem and solution spaces (Maher and Poon 1996; Dorst and Cross 2001; Howard, Culley, and Dekoninck 2008). In this iterative process that involves generating and evaluating solutions, knowledge is acquired, augmenting designers' capabilities to generate creative designs based on previous experience (Gero and Kannengiesser 2004; Boden 2004). To support their creative role (Dorst and Cross 2001), many computational design tools have been defined (Philippa and Michael) ranging from early design exploration to more advanced design phases such as refinement and optimization. In this work, we focus on the creation of computational design tools that can allow an artificial agent to explore the design space for a given problem. We inherit from Wiggins (Wiggins 2006) formalization about Boden's creativity concepts (Boden 2004) considering the exploration of possible design proposals, named artifacts, mainly as a search in a conceptual space. This way, a the set of rules and actions to generate artifacts must be defined, playing a crucial role in the solution space exploration. However, this definition is often strictly related to an specific problem space limiting system's possibilities to generalize to multiple problems. In our approach, problem space will be defined in (Serra and Miralles 2019) environment that allows us to test 2D shapes as design proposals on multiple physically based design problems. We decided to use this environment since we can define different types of problems such as collecting elements, moving through or protecting areas which has already been resolved by humans. We are very interested on the possible emergence of human solutions for this problems but specially new solutions that may inspire future proposals. In addition, physically based environments allow us to simulate how different proposals may behave in a future real scenario. This is specially relevant to understand why some proposals perform better and which constrains may appear during the exploration of the solution space. To generate these proposals, we explore how the definition of different design tools directly affects on system's generative and learning capabilities. We are specially interested in how the same computational tool can be applied in multiple problems. In order to define these tools, we consider that flexibility is one of the key aspects to allow computational systems to explore problem space and re-adapting from possible non-favorable initial conditions while generalizing better across different scenarios. As shown in (Ha 2019), by allowing an agent to learn optimal physical configurations for a given task it improves its performance and it facilitates its policy learning. To do that, in our experiments, we compare different population-based search algorithms with two different constructive methods. The first one, based on learning to optimize a shape with an already defined number of pieces. This approach can benefit the algorithm to find solutions, but it requires that creators know the problem space, since a possible number of pieces must be proposed for the solution. In contrast, the second method consists in allowing the agent to freely modify its shape by adding or removing pieces. We evaluated each method capabilities to generate creative designs by comparing their artifacts produced considering both their performance and novelty (Ritchie 2007; Maher and Fisher 2012). Although the first constructive method is more efficient in finding possible solutions, the second method can even provide more novel valid proposals in multiple scenarios. By the combination of modular blocks within multiple environment and without any previ-

Proceedings of the 11th International Conference on Computational Creativity (ICCC'20)
ISBN: 978-989-54160-2-8

ous knowledge, our system has been able to generate design proposals from scratch that resembles to human proposals. These results show the importance of defining tools that can perform more actions to explore the solution space rather than focusing solely on the complexity of the algorithm.

Related work

Our work is based on previous research on evolutionary computing (Eiben, Smith, and others 2003) that has demonstrated its capabilities to solve complex challenges in multiple environments (Lehman et al. 2018). These techniques have been widely used in robotics ranging from optimizing already defined morphologies (Joachimczak, Suzuki, and Arita 2015; Nygaard et al. 2018) to even generating them from scratch (Sims 1994; Lipson and Pollack 2000). We highlight the work on Evolutionary Design from (Bentley 1999), showing how evolutionary strategies can also be applied to producing novel and functional designs. To do that, a formalism must be defined in order to explore actions on certain solution space (Wiggins 2006). We draw inspiration from Shape Grammars firstly introduced by (Stiny 1980). A Shape Grammar (SG) consists of a computational formalism that allows automatic shape generation by providing a finite set of shapes and rules that will be applied to these shapes. By continually applying this rules, original shapes are transformed and complex structures can emerge from this process presenting similarities to creative design process theories (Maher and Poon 1996; Dorst and Cross 2001; Gero and Kannengiesser 2004). SG have been often combined with Evolutionary Algorithms (Duarte 2005; O'Neill et al. 2010; Lee and Tang 2009). This combination has also been referred as a Grammatical Evolution (GE) (O'Neill and Ryan 2001; Dempsey, O'Neill, and Brabazon 2009). The main advantage of combining a Shape Grammar with an Evolutionary Algorithm is its exploratory capabilities of the solution space. Many of this projects focus on the generative power of shape grammars rather than their possibilities as a design language (Knight 2000) that can be shared between humans and artificial agents. In addition to that, previous research lines are focused on solving an specific problem rather than aiming a more general knowledge acquisition (O'Neill et al. 2010; Lee and Tang 2009). Other studies (Ha 2019) have also explored how by allowing agents to also optimize their initial design conditions they can perform better given a certain problem. Specially the work of (Pathak et al. 2019), demonstrates how a better generalization can be achieved by the usage of modular elements to construct. In our approach we want to define a computational tool that can be used for both humans and artificial agents to solve multiple problems. In contrast to other previous work, we are interested both in the performance and the novelty of artifacts produced with our tool and its capabilities to be used in multiple problems. In addition, we want to show how, by being less restrictive and allowing more actions, our tool has a direct impact on the emergence of design by providing a wide range of creative solutions without losing performance.

Methodology

We have performed our experiments in Coevo environment (Serra and Miralles 2019) that allow us to test a collection of physically based scenarios with specific design problems. This environment has been inspired by (Brockman et al. 2016) with the focus on 2D design creation allowing both humans and artificial agents to generate proposals. Each scenario has a fixed simulation time and conditions to evaluate design proposals. A total of five scenarios have been proposed as a benchmark for our comparative study (Figure 1).

- **E0 - Collect balls.** Each proposal is evaluated by the number of falling balls collected. We have two variants based on design proposal position: left side (E0.1) or at the middle of the scenario (E0.2).
- **E1 - Move along an inclined plane.** Each proposal is evaluated based on the total distance moved within the simulation steps that the experiments lasts.
- **E2 - Move through a different medium**. Each proposal is evaluated on total distance moved from an initial free fall position and experimenting a drag force $\vec{F_s}$ when entering the different medium (Equation 1).

$$\vec{F_s} = -sv^2 A\,\hat{v} \tag{1}$$

  where s refers to the medium properties (density and drag coefficient), v refers to the speed of the proposed shape, A the frontal surface of the shape that is pushing through the new medium and $\hat{v}$ the velocity unit vector. The area is computed by extracting the cross section of the proposal's shape. Then, the greater the area, the more difficult to reach the bottom.

- **E3 - Protect area.** Each proposal is evaluated by counting the number of randomly generated balls that hit the highlighted orange area (Figure 1).

These scenarios provide diversity within the solutions that may emerge on the solution space exploration. While E0 and E3, require larger number of blocks to be solved, E1 and E2 perform better with less number of blocks. Based on this knowledge, we can define constructive rules and initial configurations that allow the system optimally solving the scenario. However, this definition may not be optimal for other scenarios or even limit system capabilities to solve other specific scenario. So, we want to explore how to define a system that can perform better globally while allowing a proper exploration of the problem space. This aspect will be discussed in the following sections.

Design tool

As we mentioned earlier, to generate design proposals we use a language inspired by shape grammar formalism (Stiny 1980). This grammar is defined by the following elements:

- **Initial shape**: single block with fixed dimensions.
- **Shape**: finite set of proposals defined by rules
- **Rule**: transformations applied to shapes (Figure 2).

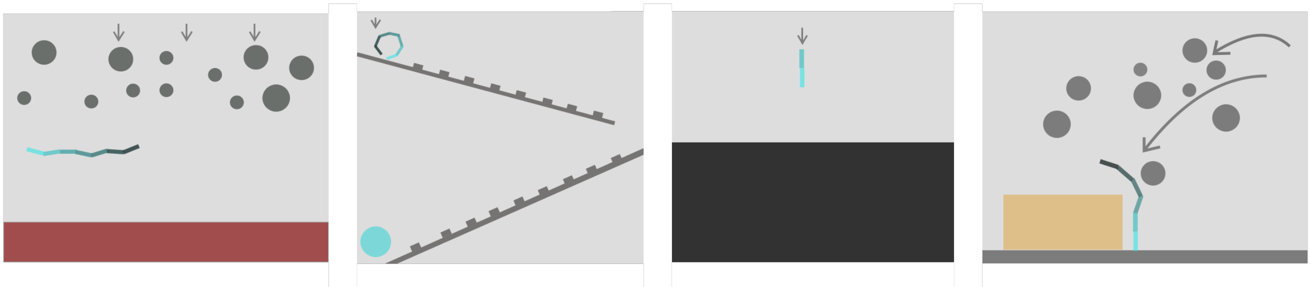

Figure 1: Scenarios. From left to right: collect falling balls, move along inclined plane, move thought different medium and protect target

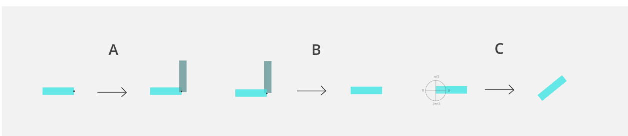

Figure 2: This shape grammar consist on three basic rules: add block (A); remove block (B); block rotation (C)

These constructive rules can allow emergence of complex shapes by concatenating simple blocks. In order to allow more freedom in design, block overlapping has been permitted.

Once the language has been defined, an internal representation of these operations and rules has been created in order to allow an artificial agent to learn how to construct a proposal(Figure 3). In our approach each shape is described by one integer which corresponds the length of the unit block, followed an open-ended stream of angles in radians which describe all the shape, as is showed in the equation (2);

$$design\,proposal = [\,length,\, \phi_1,\, \phi_2,\, \dots\,, \phi_N\,] \qquad (2)$$

where N is the number of blocks that make up a shape.

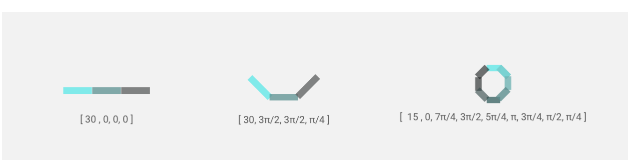

Figure 3: Internal representation of three different shapes generated using this shape grammar.

In our experiments, we maintained block length to 30 units to remain consistent across all the scenarios and artificial agents.

Artificial Agents Definition

For this study, we created three different artificial agents that learn to generate design proposals using previously described tools. All these AI agents are based on evolutionary techniques that learn to optimize its shape to fit the problem of each scenario. Then, we summarize each agent used:

- **Fixed Genetic algorithm**: this agent is based on a simple genetic algorithm (Mitchell 1998) that selects best candidates using roulette-wheel selection via stochastic acceptance (Lipowski and Lipowska 2012). Crossover is performed by combining selected candidates representation and we also add a mutation value that randomly changes angles $(0, 2\pi)$ to add noise when defining a new population
- **Fixed CMA-ES** : this agent is based on Covariance-Matrix Adaptation Evolution Strategy (Hansen, Müller, and Koumoutsakos 2003) adapted to optimize a shape with a certain number of blocks. Initial population is randomly generated. Then, each new population is generated within time from multiple distribution of mean and co variances (one for each block) based on previous generation performance. Note that the number of distributions depends on the initial number of blocks defined for that certain experiment.
- **Variable Genetic algorithm**: similar to the first agent, this approach is also based on a genetic algorithm. Its main difference is that a mutation value for adding and removing pieces has been also added. This allows the agent to optimize also the number of pieces required and explore possible valid morphologies for each scenario.

We have chosen to define our agents based on these evolutionary techniques as being ones of the most simple and popular amongst researchers in the field. (Salimans et al. 2017; Prabhu et al. 2018). In addition, population-based search techniques make possible to explore many areas in these spaces at once (Miikkulainen 2019) so we have considered ideal for our experiments. All experiments are initialized with a fixed number of blocks and only the third one is able to add and remove blocks. This decision allows us to evaluate how an agent with more design capabilities performs in comparison with the other ones.

Experiments

Here we enumerate all the experiments performed with each artificial agent and scenarios described in previous section

As seen in Figure 4, each scenario and algorithm has been initialized with three different number of blocks (6,12 and 24, respectively). Each combination has been simulated for 200 generations with a population of 100 members each one initialized randomly at the beginning of the experiment. We decided to define this initial conditions to compare how the different agents behave in possible optimal or bad initial configurations. Since we are specially interested on the global performance and novelty of each agent we consider that initializing the agent with three different number of blocks gives us a general idea on how the agent is able to adapt and provide different solutions within a limited number of iterations (200 generations). Finally, we repeated each experiment 10 times to have enough data to extract design patterns. This makes a total of 450 experiments to be analyzed

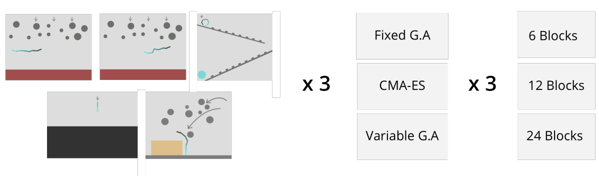

Figure 4: A total of 45 combinations can be performed considering given variables: scenario, agents and number of blocks

All design proposals are placed in the same corresponding initial position and evaluated individually using each scenario specific fitness function.

Results

In this section, we present the results based on the design proposals generated by each agent. Our goal is to evaluate agents capabilities to produce creative designs. We have considered (Ritchie 2007) approach for evaluating individual creativity by its produced artifacts rather than from the process used. To do that, we evaluate each individual artifacts based on (Maher and Fisher 2012) proposal that considers three parts for evaluation:

- **Value**: performance measure of the design.
- **Novelty**: similarity from the rest of the proposals.
- **Surprise**: how an artifact can exceed the value and novelty expectations of the already defined patterns found in the solution space.

Value

Since we have captured and analyzed all the designs produced we measure the value by computing the fitness obtained by the best member of each generation from each experiment. This parameter also gives us an idea of each agent's performance within generations. As we can see in Figure 5, there is a common behavior between configurations with fixed number of initial blocks reaching high fitness in most scenarios when their number of blocks is optimal for that scenario. In contrast, performs worse when this initial number is not optimal. For example, In Scenarios 0.1 and 0.2, only the configurations that start with 24 blocks are able to reach higher fitness. This behavior is also seen in Scenario 3, in which configurations with higher amount of blocks perform better. Opposite to that, in Scenario 2, configuration with lesser number of blocks perform better reaching maximum fitness faster. In this Scenario 2, the agent based on GA-24-fixed is the only place where this agent does not find a solution. We also observe that both fixed agents are also able to reach higher fitness within generations with the exception of the fixed ones that started with only 6 blocks.

	Fitness performance comparison					
	Fixed G.A		**CMA-ES**		**Variable G.A**	
E0.1	0.24	1	0.27	1	0.95	1
E0.2	0.3	0.71	0.32	0.81	0.54	0.66
E1	0.43	0.97	0.05	0.99	0.4	0.98
E2	0.65	1	1	1	1	1
E3	0.12	0.88	0.13	1	0.94	1

Table 1: Comparison between worst fitness and best fitness obtained by each agent configuration. As shown, agent with variable number of blocks performances are more similar.

In contrast to that, as also shown in Table 1, the agent with a variable number of blocks is able to perform better, no matter the number of blocks it is initialized. Similar to previous agents, proposals generated by this third agent are able to reach higher fitness in all scenarios expect from Scenario 0.2, also the worst scenario for the other agents. Using its constructive capabilities is able to optimize the number of blocks needed to solve the scenario. One exception to this behavior is in Scenario 1 with the with the agent starting form the lowest number of blocks (6), that the third agent has not been able to reach higher fitness.

Novelty and surprise

In terms of agent novelty, we have decided to evaluate each group of generated design proposals based on how similar are from each other. This approach is based on (Maher and Fisher 2012) which proposes evaluating similarity using distance of potentially creative designs and later on clustering them based on that. Since each agent proposes a large number of designs, we have only considered the ones a threshold performance higher than 0.9. In addition, to further reduce the number of proposals, we randomly pick only 15 proposals of each roll-out for agent. This ensures having enough representatives of each agent agent while maintaining a small data set for our similarity comparison. Then we have multiple data sets of valuable design proposals generated by each artificial agent. These two decisions ensure that selected design proposals are valuable to the given problem, while we can also evaluate how different they are across agents.

Then, we must define an efficient comparison method for determining which data set contains more novel designs. To do that, we decided to generate an image containing each proposal. To ensure enough resolution, we centrally place each proposal in a 300x300 image and then we reduce their dimensionality into 2 components using Principal Compo-

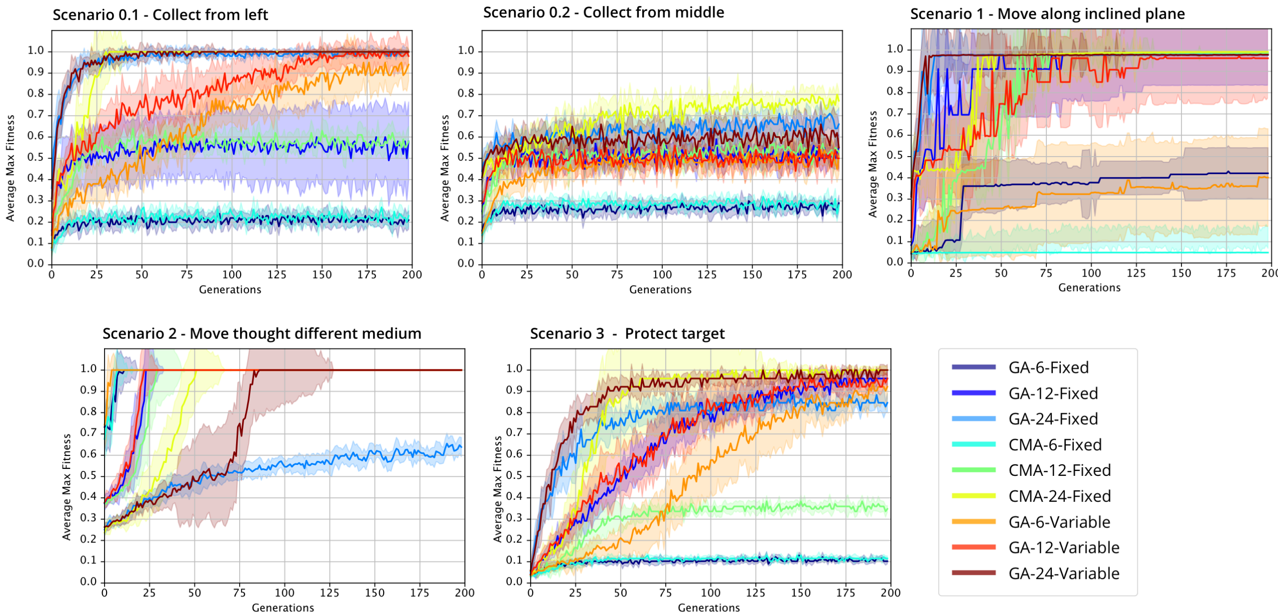

Figure 5: Learning process from each scenario and agent configuration considering 10 random rollouts

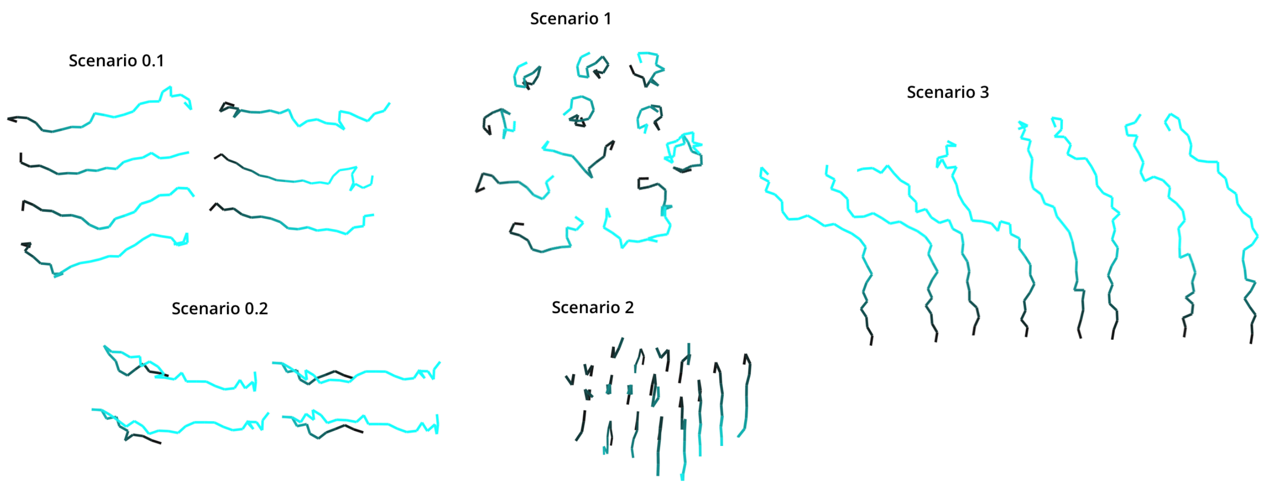

Figure 6: Random artifacts selected from third agent (G.A Variable) proposals. As shown different proposals can emerge from simple parts in each scenario.

nent Analysis (Wold, Esbensen, and Geladi 1987). This reduction helps us to visualize similar proposals closer in a 2D space allowing us to navigate between them and understand better similarity relations. We decided to use this approach to standardize all the design proposals within a single measurement since each agent may provide solutions with different number of pieces. We computed this value using (Pedregosa et al. 2011) tools. Then, for each scenario, we have placed each proposal on 2D dimensional space based on these two PCA components and clustered them.

For clustering, we propose Mean Shift algorithm (MS) (Comaniciu and Meer 2002). We decided to use the MS algorithm because it does not predetermine the number of clusters. We are interested in the emerge of clusters from our current distribution of proposals. Then from each cluster we randomly selected multiple representative proposals to compare them visually. As an example of this selection Figure 6 provides a visual overview on the divergence of solutions present in each scenario.

Regarding novelty between agents, in general, there is not a significant differentiation between the novelty produced by agents with fixed initial number of blocks than the others. All three agents are able to produce similar design proposals considering their number of blocks. However, as observed in Figure 7, the number of blocks strongly conditions the shape of generated proposals. As an example, proposals

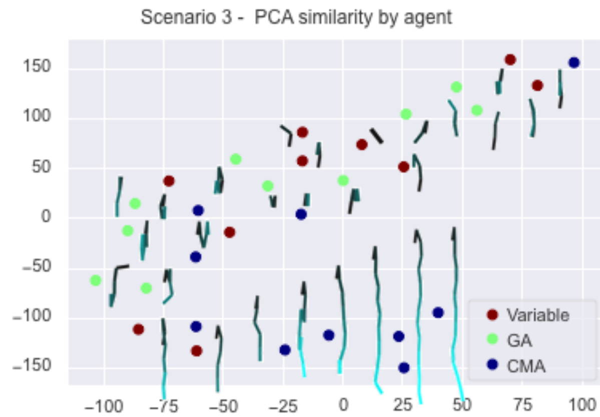

Figure 7: A total of 30 selected proposals (10 from each artificial agent) from Scenario 3 distributed on a 2D space. We can observe how a rich number of proposals are being generated by each artificial agent

from agent that CMA-ES with 24 blocks resemble a lot each other. In contrast, the agent with variable number of blocks is able to converge to a wide range of solutions with multiple number of blocks. This flexibility in design results in a greater dispersion of the generated artifacts.

Finally in terms of surprise, unexpected results specially emerged on Scenario 1. This scenario was originally designed expecting proposals with a circular shape similar to wheel. Each design agent has been able to produce not only circular shapes but also a large number of different shapes able to move along given inclined plane.

Discussion and future work

In this section, we discuss the results presented before and the findings based on the proposals generated by our artificial agents. We also include current limitations of our approach and plans for future work.

To perform our analysis, we have compared all the designs produced by our three different artificial agents in a total of 5 different scenarios. In general, all three agents have been able to produce valuable artifacts for each scenario. In terms of performance, Scenario 0.2 has been proved to be the most difficult one, directly lowering the performance obtained by the agents. In this particular scenario only the proposals generated by CMA-ES agent with 24 blocks has been able to surpass the value of 0.8 in fitness. In other scenarios, Variable G.A agent has been the unique one that has generated proposals with higher fitness no matter the initial number of blocks. This result evidences how an approach that allow more freedom in designing influences positively in exploration of the design space ending up in a richer number of high valuable generated artifacts. In contrast, both Fixed G.A and CMA-ES agents highly depend on initial parameters having less capabilities to adapt to each scenario. Then, only when initial parameters are beneficial, their performance is better reaching higher fitness faster than the others. One limitation of the current work is related to the initial conditions given to the system in terms of number of blocks and allowed iterations(200). As shown in our results, some of these configurations may limit agent's capabilities of finding optimal solutions. However, this has not happened in the flexible agent which, despite being affected in the iterations necessary to find optimal solutions, its exploratory capabilities allowed it to find solutions regardless of its initial conditions. This supports our approach that defining flexible constructive methods allows our computational tools to generalise better since we are not embedding scenario specific knowledge that may affect negatively in other situations.

Since the number of valuable proposals has been large in all the scenarios, the definition of metrics and tools to evaluate, compare and clustering them based on similarity has become crucial in our work. We have defined a comparison method inspired by (Maher and Fisher 2012) work on evaluating novelty as a distance between individual proposals. In our approach, we showed how by generating images from each shape and using PCA (Wold, Esbensen, and Geladi 1987) we can efficiently visualize and select proposals for novelty evaluation. Then, regarding novelty, our results evidence how by using these simple design tools a wide diversity of proposals emerge in all the scenarios and agents. This behavior is also stronger in Variable G.A agent since is not influenced by its initial conditions, its solution space exploration is higher. Our results also suggest how population-based algorithms combined with simple design tools inspired by shape grammars can be a powerful combination for iteratively exploring multiple solution spaces

As we seen in Figure 6, the same tools are able to generate a rich diversity of proposals for each scenario. Then, designers role in this creative environment can be focused on defining problem space and collaborating with artificial agents to propose solutions to propose proper solutions to that problem. Our current environment is limited to only five different scenarios. However, new evaluation techniques can be applied to each of them or even new scenarios can be created and tested using our artificial agents.

An interesting future work would be to explore how problem space definition by designers can influence the novelty of designs generated by artificial agents. It has been shown that the most complex scenario (Move along a plane) is the one that produced a greater emergence of novel design proposals. In natural evolution, the environment plays an important role in diversity, however, more research should be done to determine if this also happens on digital environments.

In future work, we will also investigate how these design tools can also be used by humans and how they can collaborate with different artificial agents to solve together a given challenge. In our work, many artifacts that resemble to human designs have emerged through each artificial agent learning process. However, also unique designs that we have not initially thought about have also emerged. It would be very interesting to explore how human creative capabilities can be augmented by collaborating with agents with no prior knowledge given a design problem.

Conclusions

In this paper we are especially interested in how by providing an artificial agent with more degrees of freedom in its creation tools, it can better adapt to multiple design challenges by offering proposals of greater value and novelty. To do that, we presented a new set of design tools to construct complex proposals by concatenating minimal blocks. The environment provided (Serra and Miralles 2019) together with the tools created, allowed us to define and evaluate problems such as collecting elements, moving or protecting areas that have been already been solved by humans in different ways. Our results suggest that the degrees of freedom given to the tool allowed the system to generate more novel designs with higher performance providing also solutions that are not influenced by initial design considerations based on the expected solution of a given problem.

To show that in our studies, we have defined three population-based different evolutionary agents that have generated design proposals for a total of five different scenarios. Each agent is initialized with a fixed number of blocks that can use to construct and only one has been allowed to change this number during its learning process. By defining the initial number of blocks we are providing some knowledge on the solution space since some environments can be solved optimally depending on this number. However, this knowledge is related to a certain set of solution that the creator may have in mind limiting the system to explore other solution spaces. In addition to that, it cannot be generalized in different scenarios, since this knowledge that can be beneficial in some scenarios is a limitation in others. As an example, E0 and E3 involve that the solution includes larger number of blocks than scenarios E1 and E2. Then, agents initialized with the optimal number of blocks learn faster than others that may not even reach higher fitness due to their initial definition 5. Then when defining these systems, creators must consider initial configuration as a key aspect in their design. This requires an initial human effort to understand the problem and also an initial limitation since the creators are already embedding their knowledge in the tool they are creating. However, our results suggest that flexible agent does not show this limitation in the given scenarios. In contrast to fixed ones, variable agent is able to reach optimal solution spaces despite the fact of being initialized in a less beneficial solution space or even with a configuration that has no possible solutions to the given problem. As a result of this, our artificial agent has been able to construct valid design proposals across multiple scenarios surpassing the other two agents in terms of performance and novelty. Is also specially relevant that this agent is also able to find novel solutions with high performance compared to fixed agents initialized on optimal spaces. Our results suggest that allowing more degrees of freedom influences the ability to innovate by reconfiguring its morphology, augmenting the space of possibilities and exploring new paths within this space in each scenario. Especially in E1, by continually adding pieces, different new shapes emerge to the wheels, such as spirals or S-shaped morphologies similar to sleds. This phenomenon may be related to the evolutionary path followed by the solutions provided by the variable agent since all the possibilities found by the fixed agent end up in the wheel as an optimal shape.

We hope that our results encourage computational creativity community to continue working on the definition of flexible design tools that allow artificial agents to better adapt to multiple environments.

Acknowledgments

This work has been partially funded with the support of the European Social Fund and the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement of the Catalan Government for the pre-doctoral FI grant No. 2019 FI_BI 00171.

References

Bentley, P. J. 1999. *Evolutionary design by computers*. Morgan Kaufmann.

Boden, M. A. 2004. *The creative mind: Myths and mechanisms*. Routledge.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Comaniciu, D., and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* 24(5):603–619.

Dempsey, I.; O'Neill, M.; and Brabazon, A. 2009. *Foundations in grammatical evolution for dynamic environments*, volume 194. Springer.

Dorst, K., and Cross, N. 2001. Creativity in the design process: co-evolution of problem–solution. *Design studies* 22(5):425–437.

Duarte, J. P. 2005. A discursive grammar for customizing mass housing: the case of siza's houses at malagueira. *Automation in construction* 14(2):265–275.

Eiben, A. E.; Smith, J. E.; et al. 2003. *Introduction to evolutionary computing*, volume 53. Springer.

Gero, J. S., and Kannengiesser, U. 2004. The situated function–behaviour–structure framework. *Design studies* 25(4):373–391.

Ha, D. 2019. Reinforcement learning for improving agent design. *Artificial life* 25(4):352–365.

Hansen, N.; Müller, S. D.; and Koumoutsakos, P. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation* 11(1):1–18.

Howard, T. J.; Culley, S. J.; and Dekoninck, E. 2008. Describing the creative design process by the integration of engineering design and cognitive psychology literature. *Design studies* 29(2):160–180.

Joachimczak, M.; Suzuki, R.; and Arita, T. 2015. Improving evolvability of morphologies and controllers of developmental soft-bodied robots with novelty search. *Frontiers in Robotics and AI* 2:33.

Knight, T. 2000. Shape grammars in education and practice: history and prospects. *International Journal of Design Computing* 2(67).

Lee, H. C., and Tang, M. X. 2009. Evolving product form designs using parametric shape grammars integrated with genetic programming. *AI EDAM* 23(2):131–158.

Lehman, J.; Clune, J.; Misevic, D.; Adami, C.; Altenberg, L.; Beaulieu, J.; Bentley, P. J.; Bernard, S.; Beslon, G.; Bryson, D. M.; et al. 2018. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *arXiv preprint arXiv:1803.03453*.

Lipowski, A., and Lipowska, D. 2012. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications* 391(6):2193–2196.

Lipson, H., and Pollack, J. B. 2000. Automatic design and manufacture of robotic lifeforms. *Nature* 406(6799):974.

Maher, M. L., and Fisher, D. H. 2012. Using ai to evaluate creative designs. In *DS 73-1 Proceedings of the 2nd International Conference on Design Creativity Volume 1*.

Maher, M. L., and Poon, J. 1996. Modeling design exploration as co-evolution. *Computer-Aided Civil and Infrastructure Engineering* 11(3):195–209.

Miikkulainen, R. 2019. Creative ai through evolutionary computation. *arXiv preprint arXiv:1901.03775*.

Mitchell, M. 1998. *An introduction to genetic algorithms*. MIT press.

Nygaard, T. F.; Martin, C. P.; Torresen, J.; and Glette, K. 2018. Exploring mechanically self-reconfiguring robots for autonomous design. *arXiv preprint arXiv:1805.02965*.

O'Neill, M., and Ryan, C. 2001. Grammatical evolution. *IEEE Transactions on Evolutionary Computation* 5(4):349–358.

O'Neill, M.; McDermott, J.; Swafford, J. M.; Byrne, J.; Hemberg, E.; Brabazon, A.; Shotton, E.; McNally, C.; and Hemberg, M. 2010. Evolutionary design using grammatical evolution and shape grammars: Designing a shelter. *International Journal of Design Engineering* 3(1):4–24.

Pathak, D.; Lu, C.; Darrell, T.; Isola, P.; and Efros, A. A. 2019. Learning to control self-assembling morphologies: a study of generalization via modularity. In *Advances in Neural Information Processing Systems*, 2292–2302.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research* 12(Oct):2825–2830.

Philippa, M., and Michael, B. V. An ontology of computational tools for design activities.

Prabhu, S. G. R.; Seals, R. C.; Kyberd, P. J.; and Wetherall, J. C. 2018. A survey on evolutionary-aided design in robotics. *Robotica* 36(12):1804–1821.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; and Sutskever, I. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.

Serra, G., and Miralles, D. 2019. Coevo: a collaborative design platform with artificial agents. *arXiv preprint arXiv:1904.13333*.

Sims, K. 1994. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 15–22. ACM.

Stiny, G. 1980. Introduction to shape and shape grammars. *Environment and planning B: planning and design* 7(3):343–351.

Wiggins, G. A. 2006. Searching for computational creativity. *New Generation Computing* 24(3):209–222.

Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2(1-3):37–52.