Controllable Exploration of a Design Space via Interactive Quality Diversity

Konstantinos Sfikas
konstantinos.sfikas@um.edu.mt
University of Malta
Msida, Malta

Antonios Liapis
antonios.liapis@um.edu.mt
University of Malta
Msida, Malta

Georgios N. Yannakakis
georgios.yannakakis@um.edu.mt
University of Malta
Msida, Malta

ABSTRACT

This paper introduces a user-driven evolutionary algorithm based on Quality Diversity (QD) search. During a design session, the user iteratively selects among presented alternatives and their selections affect the upcoming results. We implement a variation of the MAP-Elites algorithm where the presented alternatives are sampled from a small region (window) of the behavioral space. After a user selection, the window is centered on the selected individual's behavior characterization, evolution selects parents from within this window to produce offspring, and new alternatives are sampled. Essentially we define an adaptive system of local QD search, where the user's selections guide the search towards specific regions of the behavioral space. The system is tested on the generation of architectural layouts, a constrained optimization task, leveraging QD search through a two-archive approach.

CCS CONCEPTS

• **Computing methodologies → Search methodologies**; • **Human-centered computing →** *Interaction paradigms*; • **Applied computing →** *Architecture (buildings)*.

KEYWORDS

quality diversity, interactive evolution, human computer interaction, creativity support tools, floorplan generation

ACM Reference Format:

Konstantinos Sfikas, Antonios Liapis, and Georgios N. Yannakakis. 2023. Controllable Exploration of a Design Space via Interactive Quality Diversity. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3590616

1 INTRODUCTION

Interactive Evolutionary Computation (IEC) is a form of "systemic optimization that uses a real human's subjective evaluation in its optimization process" [9]. In its narrow definition, the user's subjective evaluation takes the role of fitness in an evolutionary optimization process [14]. IEC is advantageous in problems where the definition of a fitness function is hard or impossible, while their evaluation from real humans is feasible. IEC however can easily lead to user fatigue [14], and many solutions have been proposed to address this by e.g. showing a subset of the population, sharing the burden of IEC with multiple users through online interfaces [11], and fitness approximation via user models [6].

This paper introduces a novel IEC algorithm aiming to provide a high degree of user control without inducing user fatigue. We showcase that this is achievable by exploiting the illumination capabilities of Quality Diversity (QD) algorithms [10]. We envision a hybrid system of "User-Controlled QD Exploration", where the user's choices localize and control QD search within a part of the behavioral space. In this case, we modify MAP-Elites [8] and we constrain the algorithm's operation within a window that covers a small region of the feature map, where it locally expands the archive for a number of generations. Afterwards, design alternatives are sampled from within the window and presented to the designer. Finally, the user's selection determines where the window will move towards next. These steps summarize the functionality of the specific algorithmic implementation introduced here, which we refer to as *User Controlled MAP-Elites* (UC-ME).

We test UC-ME on the constrained design problem of generating architectural layouts, using a generative methodology that was introduced in [12]. This problem has multiple constraints, an ad-hoc representation and ad-hoc genetic operators. To apply UC-ME on a constrained problem, we draw inspiration from FI-MAP-Elites [12] and adapt UC-ME to work on the dual archives of feasible and infeasible elites. To test how UC-ME caters to different potential user goals, we utilize artificial users with different selection criteria.

2 RELATED WORK

Quality Diversity (QD) search [10] simultaneously optimizes and diversifies the population of generated solutions. As an archetypal QD method, MAP-Elites [8] operates by subdividing a feature space into cells. Each cell contains the fittest individual in that niche, as defined by multiple Behavioral Characterizations (BCs).

When a problem includes hard constraints, solutions are characterized as feasible or infeasible if they satisfy a set of criteria or not. Several methods [3, 7] combine QD with constraint solving, inspired by the FI-2-Pop GA [4] which evolves two populations (one with feasible and one with infeasible individuals) in parallel. In constrained QD search, FI-MAP-Elites [12] hybridizes FI-2Pop GA [4] and MAP-Elites [8] by maintaining two archives (one with feasible elites and one with infeasible elites). Parent selection alternates between the two archives, while mutated offspring can change archives based on their feasibility.

Despite its high suitability for design problems, QD in a Mixed-Initiative setting [15] is relatively under-researched. In the work of Alvarez et al. [1, 2], the designer can control the MAP-Elites

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3590616

algorithm's parameters and thus illuminate the design space in different ways. Moreover, the designer can manually intervene on the generated designs, or design their own from scratch and then use them as seeds for MAP-Elites. It is important to emphasize that while these modes of operation rely on a user's initiative, they are not matching the narrow definition [14] of IEC (see Section 1).

3 USER-CONTROLLABLE MAP-ELITES

We introduce the User Controllable MAP-Elites (UC-ME) algorithm as a way to endow a user with control over the direction and computational resources of QD exploration. The algorithm operates by allowing parent selection only from a small window of the archive of elites, and moving this *selection window* according to the user's selections. The user can select one favorite solution among a small number of alternatives (four in this paper) that have been sampled from within the current window. The general process of the UC-ME algorithm, and different methods for sampling design alternatives to show to the user, are described below.

Algorithm initialization. UC-ME first produces a number of solutions through a random initialization method and places them in the MAP-Elites archive according to their BCs. This step seeds the archive to enable interaction with the human user. The initial selection window of size $w \times w$ is centered at the cell with the mean BC values of existing elites, or the nearest elite if that cell is unoccupied. The window size (w) is a parameter of UC-ME which should be much smaller than the resolution of the feature map.

Interactive Operation. After initialization, the interactive session can begin. During an interactive session, the following steps are repeated indefinitely, until the designer decides to end it.

(1) **Design Alternatives Sampling:** The algorithm samples D design alternatives, from within the selection window, to be shown to the designer as options to select from.
(2) **Designer Input:** The designer selects one preferred design.
(3) **Selection Window Placement:** The selection window is centered at the coordinates of the designer's last selection.
(4) **Windowed Archive Expansion:** The algorithm operates for N_e evaluations, selecting parents from within the window. The mutated offspring are evaluated and placed at their corresponding archive cell, based on their Behavioral Characterization coordinates, without being constrained by the window. In case an offspring lands on an already occupied cell, the individual with the highest fitness survives.

Sampling methods for design alternatives. UC-ME samples a number of design alternatives to present to the user from within the selection window. We only test UC-ME with four design alternatives in this paper, and implement two semi-stochastic methods for design alternatives sampling (DAS). **Corners (A_C)** samples one individual per corner of the window, or the nearest individual to that corner. For **Medoids (A_M)**, the coordinates of the individuals within the selection window are used as data points in a k-medoids clustering algorithm, where $k = 4$ in this paper. The four medoids of these clusters are shown to the user.

4 USE CASE: LAYOUT GENERATION

We test UC-ME on the subjective, complex and constrained problem of generating architectural layouts. We follow the methodology of [12], where the problem definition is a set of topological and other constraints, and the output is a geometrical solution that respects these constraints. We summarize the process for this use case below; more details can be found in [12, 13].

Layout Representation. The representation of the architectural layout has two facets: a Design Specification (DS) and a Design Implementation (DI). The DS is a user-defined description of the problem at hand in terms of its space units (rooms or other regions). It consists of a connectivity graph of the layout, the desired area per space-unit, the number of doors to the exterior and windows in each space-unit, and whether it is an indoors or outdoors space. The DI is the geometric implementation of a DS, where every space unit occupies a specific region of the plane along with the precise location of doors and windows. To avoid constraining designs, we implement a system based on a Voronoi-tessellation of the plane. The generated space units can be placed at specific regions of this tessellation. Both the space-units' placement and the underlying structure are mutated during the algorithm's operation.

Constraints. Generated layouts have to satisfy many constraints, due to physical requirements (e.g. layout connectedness) and design constraints (from the DS). Details on the constraints of this problem are provided in [12]. If an individual fails any of these constraints, it is assigned a *feasibility score* proportionate to the number of constraints passed, or how close they are to passing (if failed).

Fitness. In this problem, quality is mainly ensured by the satisfaction of constraints. In order to guide the QD algorithm, we use the adherence to the DS as our main quality criterion in the feasible population and optimize how close the areas of the space units are to the specified ones. We define mean area precision ($\bar{P}_s$) as the average difference between specified (A_t) and actual (A) area of each space unit. For each space unit in the DS, its area precision (P_s) is $P_s = min(A, A_t)/max(A, A_t)$, or 0 if it is missing from the DI. Note that $\bar{P}_s$ is also a criterion for feasibility: if $\bar{P}_s < 0.6$ then the individual is infeasible. If $\bar{P}_s \geq 0.6$, this metric is treated as quality characterization for the feasible archive.

Behavioral Characterizations. We use two BCs as measures of diversity of generated DIs in both the feasible and infeasible archive:

Mean Space Units' Compactness ($\bar{C}_s$) is based on the notion of compactness, a unit-less measure that expresses the relation between a shape's perimeter and its area [5]. Compactness of a space unit in the DS (C_s) is calculated as $C_s = 2\pi A/\Pi^2$ where A is its area and Π its perimeter, or 0 if it is missing from the DI. Finally, $\bar{C}_s$ measures the mean compactness of all space units in the DS.

Plan Orthogonality ($\bar{O}_\theta$) is calculated as the mean orthogonality of all angles (θ) between connected walls in the layout. A single angle's orthogonality (O_θ) is calculated as shown in Eq. (1), penalizing angles between walls that are not at 90°or 180°.

$$O_\theta = \begin{cases} 2\theta/\pi & 0 \leq \theta < \pi/2 \\ 2 - 2\theta/\pi & \pi/2 \leq \theta < 3\pi/4 \\ 2\theta/\pi - 1 & 3\pi/4 \leq \theta \leq \pi \end{cases} \tag{1}$$

where $\theta \in [0, \pi]$ is the unsigned angle between two continuous wall segments.

Initial generation. Layouts are initialized in a semi-stochastic manner that does not guarantee feasibility, by filling in a random Voronoi tessellation with each space-unit iteratively.

Genetic Operators. Mutation of layouts occurs in two stages, stochastic *destruction* and scripted *repair*. This approach produces offspring that have partial similarities with the parents (as only parts of the layout are destroyed), while the repair functions aims to decrease the chance of producing infeasible offspring. Details of these operators are found in [12].

Constrained QD process. For this problem, we adapt UC-ME (Section 3) to the constrained FI-MAP-Elites [12] algorithm. Constrained UC-ME works on a two-archives approach: one archive for the feasible elites and one for infeasible. We use our two BCs for both archives. Quality is the feasibility score for the infeasible archive, and $\bar{P}_s$ for the feasible archive. For initialization, 100 individuals are generated as described above and assigned to the two archives according to their feasibility. Afterwards, we run FI-MAP-Elites QD search until the feasible archive has at least 1% coverage. This provides us with enough of a seed to run the interactive operation of the algorithm. In terms of the interactive operation, the only changes are that (a) DAS methods are applied only on the feasible archive, (b) the selection window is applied to both the feasible and the infeasible archive and (c) parents are selected in an alternating fashion between feasible and infeasible archive. Offspring are tested for feasibility and their BCs and placed in the appropriate archive in the appropriate cell, replacing any worse elite there.

5 EXPERIMENT PROTOCOL

As a specific case study for architectural layout generation, we use an ad-hoc DS for a medium-size apartment with 7 interior and 3 exterior space-units; details are in [13]. For this experiment, each archive is split into 4,096 cells (64×64). We produce an initial population by running FI-MAP-Elites on 100 randomly initialized individuals until 1% of the feasible archive is covered; all runs of all experiments use the same initial popluation. Between user selections, UC-ME iteratively selects 10,000 parents within the selection window (alternating between feasible and infeasible archives) before the next batch of design alternatives are shown to the user.

We use controllable, artificial users (U_i) with their own user selection criterion (USC) to test the algorithm [6]. All agents select the individual with the highest USC (maximization problem) and all USC$\in [0, 1]$. Eight artificial agents (U_1 to U_8) have a consistent USC throughout evolution, while four artificial agents (U_9 to U_{12}) change their USC after 5 selections in order to test how the algorithm adapts to a shifting user taste. U_1 maximizes BC1, U_2 maximizes BC2, U_3 maximizes the average of BC1 and BC2, U_4 maximizes the highest of BC1 or BC2. U_5 to U_8 reverse the heuristics of the U_1 to U_4 (e.g. U_5 choosing the lowest BC1). U_9 maximizes BC1, then minimizes it after 5 selections. U_{10} maximizes BC2, then minimizes it. U_{11} maximizes BC1, then maximizes BC2. U_{12} maximizes BC2, then maximizes BC1. We have chosen USCs that are captured in the two BCs of our case study, as UC-ME operates best when the user's taste is not orthogonal to the dimensions of QD explored.

Parameter	MAP-Elites	A_C	MAP-Elites	A_M
Coverage	12	0	12	0
Max Fitness	0	0	0	0
QD Score	12	0	12	0
Max USC	1	9	1	4
Mean USC	0	12	0	10

Table 1: Experiments with artificial users, showing which experiment had significantly higher scores between baseline MAP-Elites and UC-ME for different DAS methods.

We follow the literature and measure Coverage (percentage of occupied cells), Maximum Fitness (highest fitness among elites) and QD-Score (total fitness of all elites) of the feasible archive [8]. Since we want to achieve a user-controllable exploration of the problem space, we use the user selection criterion (USC) to assess how the algorithm caters for a user's tastes. The following metrics capture whether the elites match the user's selection criteria: the maximum and average value of the USC of elites in the archive (Max USC, Mean USC). We examine the Area Under Curve (AUC) of these metrics from the start of evolution, thus measuring performance during the entirety of the run—not just the final state.

6 RESULTS

Using the experimental setup of Section 4, we compare UC-ME with different DAS methods against MAP-Elites without user control (but with two archives). Results are collected from 10 independent runs; significance is established via Student's t-test with $p < 0.05$.

Table 1 shows the pairwise comparison between the unguided MAP-Elites baseline and each UC-ME variant. Results show the number of experiments in which the baseline or the UC-ME variant had superior performance in terms of the chosen metric (out of a total of 12 experiments), after 10 user selections. It is evident that unguided MAP-Elites has better coverage of the problem space and thus a higher QD score, across all experiments. This is not surprising, as UC-ME drives search towards specific parts of the problem space (and regions of the feature map), while MAP-Elites covers as much of the feature map as possible. We also note that there are no differences in terms of maximum fitness. This is somewhat surprising, since different parts of the feature map (targeted by different users) may not have equally good fitnesses. It seems that finding a highly fit individual is not challenging in this use case. As expected, the unguided exploration of the baseline MAP-Elites performs worse than both UC-ME versions for maximum and mean USC score of all elites in the archive. The A_M method is less efficient at reaching very high USC scores, compared to A_C; this is not surprising since the latter moves the selection window toward regions of the problem space with high USC faster.

Figure 1 shows how coverage changes after each user selection (or the same evaluation threshold for MAP-Elites). In addition, the figures show in red the selection window of UC-ME as it moves towards higher USC scores (in this case that of U_3). We focus on the A_C method, as the most efficient. The top row of images in Fig. 1 illustrates the differences between UC-ME and MAP-Elites exploration patterns: in gray we see the common cells discovered by both methods, in magenta we see the cells discovered only by

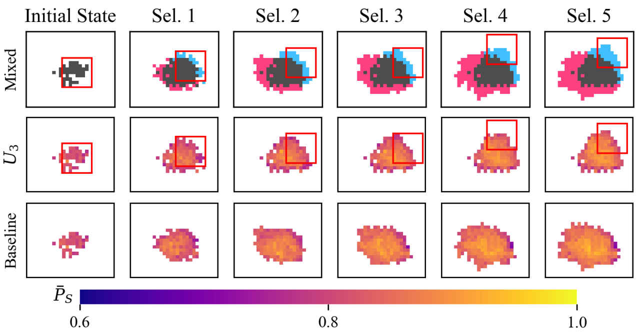
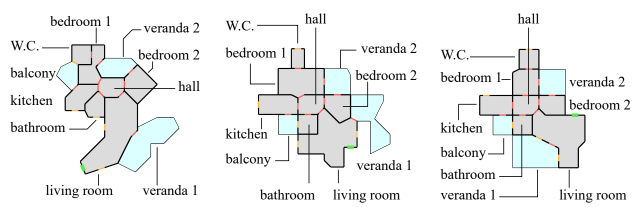


Figure 1: Behavioral space exploration for the baseline MAP-Elites (bottom row) and UC-ME with A_C DAS method guided by U_3 (middle row), for the first 5 selections. Their shared color scale (bottom) is the feasible quality. The top row shows coverage differences: red cells are discovered only by the baseline, blue cells are discovered only by UC-ME and gray cells are common. In these figures the x axis is $\bar{C}_s \in [0.44, 0.86]$ and the y axis is $\bar{O}_\theta \in [0.61, 0.97]$.



(a) Selection 1 **(b) Selection 10** **(c) Selection 50**

Figure 2: Indicative individuals selected by U_3, using the A_C DAS method, after 1, 10 and 50 selections with UC-ME. Exterior space-units are in cyan, external doors in green, internal doors in red and windows in yellow.

MAP-Elites and in blue we see the cells discovered only by UC-ME. We see that cells at higher USC values exclusively belong to UC-ME. The higher coverage of MAP-Elites is due to most cells occupying lower $\bar{C}_s$ and $\bar{O}_\theta$ values, which are undesirable for U_3. Figure 1 also shows how the selection window moves first towards a higher $\bar{C}_s$; once it reaches the edge of the feasible space and can not find individuals with higher scores in that direction, it moves towards higher $\bar{O}_\theta$ scores. We also see that within the first 3 selections, UC-ME with A_C has found the edges of the feasible space with the highest USC scores and starts moving around fairly haphazardly in that vicinity, leading to more selections and improved quality of individuals in that specific region of the problem space.

Finally, we show what U_3 selected in an indicative run of UC-ME with A_C in Fig. 2. We observe that initial individuals do not have a good USC score as the shown selection has many acute angles and complex shapes in the rooms. After 10 selections, the user has found an individual with mostly compact and square rooms. After 50 selections, the results are not much different than with 10 selections; thus 10 selections are usually enough for this problem and would not be overly fatiguing to the user.

7 CONCLUSION

In this paper, we propose a way of controlling the direction of exploration and the computational budget of Quality Diversity search with little cognitive load. The User Controlled MAP-Elites algorithm (UC-ME) is the first instance of the interactive evolution paradigm (in its narrow sense) applied to MAP-Elites, unlike past work [2]. The UC-ME algorithm operates by focusing parent selection in a smaller window of the feature map, which allows it to operate in both unconstrained and constrained problems via two archives [12]. In our experiments on a complex and heavily constrained problem, we observe that UC-ME can focus on interesting parts of the problem space according to the user's selection criterion, at the cost of lower coverage and fewer elites in total. The proposed method shows potential, but important next steps include testing its efficiency and impact on user fatigue with human users, as well as expanding the work in more domains, with more BC dimensions, and alongside constantly updated models of the user's taste [6].

ACKNOWLEDGMENTS

This work has been supported by the European Union's Horizon 2020 research and innovation programme from the PrismArch (Grant Agreement No. 952002) and AI4media (Grant Agreement No. 951911) projects.

REFERENCES

[1] Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius. 2019. Empowering quality diversity in dungeon design with interactive constrained MAP-Elites. In *Proc. IEEE Conf. on Games*.
[2] Alberto Alvarez, Jose Maria Maria Font Fernandez, Steve Dahlskog, and Julian Togelius. 2020. Interactive constrained MAP-Elites: Analysis and evaluation of the expressiveness of the feature dimensions. *IEEE Trans. on Games* 14, 2 (2020).
[3] Ahmed Khalifa, Scott Lee, Andy Nealen, and Julian Togelius. 2018. Talakat: Bullet hell generation through constrained MAP-Elites. In *Proc. Genetic and Evolutionary Computation Conf.* 1047–1054.
[4] Steven Orla Kimbrough, Gary J Koehler, Ming Lu, and David Harlan Wood. 2008. On a feasible-infeasible two-population (FI-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch. *European Journal of Operational Research* 190, 2 (2008), 310–327.
[5] Petros Koutsolampros, Kerstin Sailer, Tasos Varoudis, and Rosie Haslem. 2019. Dissecting Visibility Graph Analysis: The metrics and their role in understanding workplace human behaviour. In *Proc. International Space Syntax Symposium*.
[6] Antonios Liapis, Héctor P. Martínez, Julian Togelius, and Georgios N. Yannakakis. 2013. Adaptive game level creation through rank-based interactive evolution. In *Proc. IEEE Conf. on Computational Intelligence and Games*.
[7] Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. 2015. Constrained novelty search: A study on game content generation. *Evolutionary computation* 23, 1 (2015), 101–129.
[8] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).
[9] Yan Pei and Hideyuki Takagi. 2018. Research progress survey on interactive evolutionary computation. *Journal on Ambient Intelligence and Humanized Computing* (2018).
[10] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. 2016. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI* (2016).
[11] Jimmy Secretan, Nicholas Beato, David B. D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T. Folsom-Kovarik, and Kenneth O. Stanley. 2011. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation* 19, 3 (2011), 373–403.
[12] Konstantinos Sfikas, Antonios Liapis, and Georgios N. Yannakakis. 2022. A general-purpose expressive algorithm for room-based environments. In *Proc. FDG workshop on Procedural Content Generation in Games*.
[13] Konstantinos Sfikas, Antonios Liapis, and Georgios N. Yannakakis. 2023. Controllable exploration of a design space via interactive quality diversity. *arXiv preprint arXiv:2304.01642* (2023).
[14] Hideyuki Takagi. 2001. Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* 89, 9 (2001).
[15] Georgios N. Yannakakis, Antonios Liapis, and Constantine Alexopoulos. 2014. Mixed-initiative co-creativity. In *Proc. Foundations of Digital Games Conf.*