Interactive Constrained MAP-Elites:
Analysis and Evaluation of the Expressiveness of the Feature Dimensions

Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius, *Member, IEEE*

Abstract—We propose the Interactive Constrained MAP-Elites, a quality-diversity solution for game content generation, implemented as a new feature of the Evolutionary Dungeon Designer (EDD): a mixed-initiative co-creativity tool for designing dungeons. The feature uses the MAP-Elites algorithm, an illumination algorithm that segregates the population among several cells depending on their scores with respect to different behavioral dimensions. Users can flexibly and dynamically alternate between these dimensions anytime, thus guiding the evolutionary process in an intuitive way, and then incorporate suggestions produced by the algorithm in their room designs. At the same time, any modifications performed by the human user will feed back into MAP-Elites, closing a circular workflow of constant mutual inspiration. This paper presents the algorithm followed by an in-depth evaluation of the expressive range of all possible dimension combinations in several scenarios, and discusses their influence in the fitness landscape and in the overall performance of the procedural content generation in EDD.

Index Terms—Procedural Content Generation, Evolutionary Algorithms, Mixed-Initiative Co-Creativity, Evaluation Methods.

I. INTRODUCTION

Procedural Content Generation (PCG) refers to the generation of game content with none or limited human input [1], where game content could be anything from rules and narrative, to levels, items, and music. While PCG has been a factor in game development since trailblazing games like *Rogue* [2] and *Elite* [3], it has only been a popular academic research topic for little more than a decade. Search-based PCG designates the use of a global search algorithm, such as an evolutionary algorithm to search content space [4].

Part of PCG's appeal is the promise to produce game art and content faster and at a lower cost, as well as enabling innovative content creation processes such as player-adaptive games [5]–[7], data-driven content generation [8], [9], and mixed-initiative co-creativity [10]. Mixed-initiative co-creativity (MI-CC), a concept introduced by Yannakakis et al. [11], refers to the approach of using a creation process through which a computer and a human user provide and inspire each other in the form of iterative reciprocal stimuli. Examples of MI-CC systems are *Pitako* [12], *Ropossum* [13], *Tanagra* [14], *CICERO* [15], and *Sentient Sketchbook* [16].

MI-CC aligns with the principles of lateral thinking and creative emotive reasoning: the processes of solving seemingly unsolvable problems or tackling non-trivial tasks through an indirect, non-linear, creative approach [10]. Additionally, MI-CC provides insight on the affordances and constraints of the human process for creating and designing games [1].

A key mechanism in MI-CC approaches is to present suggestions to users, and these suggestions must be of high quality but also be sufficiently diverse. So-called quality-diversity algorithms [17] are very well suited for this, as they find solutions that have high quality according to some measure but are also diverse according to other measures [18]. The Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [19] is a suitable algorithm for this kind of problem. Khalifa et al. [8] presented constrained MAP-Elites, a combination MAP-Elites with the feasible-infeasible concept from the FI2Pop genetic algorithm [20], and applied this to procedurally generating levels for bullet hell games. Another recent implementation of MAP-Elites has been used to produce small sections of Super Mario Bros levels called *scenes*, addressing specific game mechanics [21].

The Evolutionary Dungeon Designer (EDD) is a MI-CC tool for generating dungeons for adventure games using a FI2Pop evolutionary approach [22]–[25]. This research was presented in [26], introducing *Interactive Constrained MAP-Elites* (IC MAP-Elites), a combination of Constrained MAP-Elites with interactive evolution, in the shape of a continuous evolutionary process that takes advantage of MAP-Elites' multidimensional discretization of the search space into cells. In [26], we analyzed the effects of using quality-diversity in procedurally generating dungeons, as well as the effects of continuous evolution and dimension customization.

This paper contributes with a thorough evaluation of the expressive range of EDD's MAP-Elites through all possible dimension combinations in several scenarios, as well as by extending the feature dimensions to two new dimensions, *Inner similarity*, which adds a similarity measure independent of the aesthetics, and *Leniency*, which strives to measure the subjective challenge score of a level. Following recent research on how to evaluate procedural content generators and, in particular, quality-diversity approaches [27]–[29], we present the results from new experiments with the objective to evaluate the expressive range of all dimensions in pairs, as well as to analyze how the generated and unique solutions relate to all the dimensions included in the search space, and assess IC MAP-Elites feasibility and adaptability to create dungeons and adventure levels.

A. Alvarez, S. Dahlskog, and J. Font are with the Department of Computer Science and Media Technology (DVMT), Malmö University, Malmö, Sweden (e-mail: alberto.alvarez@mau.se; steve.dahlskog@mau.se; jose.font@mau.se).

J. Togelius is with the Department of Computer Science and Engineering, New York University, New York, NY 11201 USA (e-mail: julian@togelius.com).

II. PREVIOUS WORK

A. Map-Elites for illuminating search spaces

Quality-diversity algorithms are algorithms which search a solution space, not just for the single best solution, but for

a set of diverse solutions which are high performing. MAP-Elites maintains of map of good solutions [19] and is a well-known quality-diversity algorithm. The map is divided into a number of cells according to one or more feature dimensions. In each cell, a single solution is kept. At every update, an offspring is generated based on one or more existing solutions. That offspring is then assigned to a cell based on its feature dimensions, which might or might not be the same as the cell(s) its parent(s) occupy. If the new offspring has a higher fitness than the existing solution in that cell, it replaces the previous item in the cell. This process results in a map of solutions where each cell contains the best found solution for those particular feature dimensions.

B. Evaluation of Procedural Content Generators

Shaker, et al. [30], argues that ultimately the evaluation of content generators is to verify that they fulfil their design goals. In order to be able to understand or modify a generator it is important to visualize its content space. However, it is seldom enough to look at a single individual piece of content, but rather it is vital to examine the frequency the different features appear, or the amount of variety the features demonstrate. Previous attempts of doing this are termed expressivity measures [31], [32] and have, for instance, explored difficulty measures [33]. Other approaches incorporate tool assisted parameter exploration due to its effect on the content space [27], [34].

C. Evolving Dungeons as a Whole, Room by Room

EDD is a MI-CC tool that allows a human designer to create a 2D dungeon and the rooms it is composed of. The designer is able to manually edit both the dungeon by placing and removing rooms, and the individual rooms by editing the tiles that each room consists of. EDD's underlying evolutionary algorithm provides procedurally generated suggestions, and is driven through the use of game design micro- and meso-patterns. A detailed description of all EDD's features, including the use of game design patterns, can be found in [22]–[25], where [22], [24] analyze and discuss the mixed-initiative system in EDD and [23], [25] focus on the procedural content generator.

In this section we present the latest version of EDD[1], which includes significant improvements based on the outcomes from the qualitative analysis discussed in [22]. The dungeon is now represented as a graph of interconnected rooms of any given size between 3×3 and 20×20 tiles. The smallest allowed dungeon is composed by two rooms with one connection to each other. Rooms and connections can now be added and removed at any time. Connections are marked with door tiles.

The designer marks one room as the *initial room* for feasibility calculation: a dungeon is feasible when there is at least one path between the *initial room* and any other passable tile in the dungeon. Rooms and doors that are unreachable from the *initial room* are highlighted in red, so that they can be easily identified by the designer.

[1]Available for download at https://github.com/mau-games/eddy

Fig. 1. The room editor screen in EDD. The left pane contains all the options for manually editing the room displayed at the center-left of the screen. The right section displays the procedurally generated suggestions.

The starting screen in EDD is the dungeon editor screen. Every new room is empty (composed solely of floor tiles) when created and is placed detached from the dungeon graph. After manually connecting the room to the dungeon with at least one connection, the designer has the option to populate the room using the room editor screen (Figure 1). This screen can be reached by either double-clicking on the room or by clicking on the "Start with our suggestions" button, which will present six procedurally generated suggestions to the designer to start editing from.

Figure 1 shows the room editor screen displaying a sample room with the dimensions 7×13 tiles. The left pane lists all the available options for manually editing the room by brush painting with one of the available tile types (floor, wall, treasure, or enemy) and one of the two brush sizes (single tile and five-tile cross shape). Control-clicking allows the designer to bucket paint all adjacent tiles of the same type. Brush painting with the lock button on preserves selected tiles in all the procedurally generated suggestions. A detailed description of all the options in this pane is included in [22], [23].

The right side of the screen displays the procedurally generated suggestions, by means of the Interactive Constrained MAP-Elites (IC MAP-Elites) genetic algorithm (see Section III). When the designer accesses the room editor screen, IC MAP-Elites starts and continuously populates the suggestions pane with elites. The evolutionary process is fed with the manually edited room (i.e. target room), so that every change in the room affects the generated suggestions. By clicking on "Apply Suggestion", the manually edited room is replaced by the selected suggestion, thus affecting the upcoming procedural suggestions. "Restart" restarts evolution, and "Go To World Grid" takes the user back to the dungeon editor screen.

III. INTERACTIVE CONSTRAINED MAP-ELITES

The overarching goal of MI-CC is to collaborate with the user to produce content, either to optimize (i.e., exploit) their current design towards some goal or to foster (i.e., explore) their creativity by surprising them with diverse proposals.

By implementing MAP-Elites [19] and continuous evolution into EDD, our algorithm can (1) account for the multiple dimensions that a user can be interested in, (2) explore multiple areas of the search space and produce a diverse amount of high-quality suggestions to the user, and (3) still evaluate how interesting and useful the tile distribution is within a specific room. Henceforth, we name the presented approach **Interactive Constrained MAP-Elites** (IC MAP-Elites).

EDD uses a single-objective fitness function (shown in Equation (1)) with a FI2Pop genetic algorithm, where fitness is a weighted sum divided equally between (1) the inventorial aspect of the rooms and (2) the spatial distribution of the design. $f_{inventorial}$ is the evaluation of the aggregated and normalized quality of treasures, enemies, and doors (inventorial patterns). $f_{spatial}$ refers to the quality and distribution of chambers, i.e. open areas in the room and corridors (both categorized as spatial patterns), and the meso-patterns that are created within chambers and their quality. Quality refers to the positioning, safety, composition, and relation between patterns. The fitness adapts to the user's current design, automatically informing target ratios and distributions to be used as targets. In-depth evaluation of EDD's fitness function, as well as discussion and explanation of the quality of each inventorial, spatial, and meso pattern, can be found in [23]–[25].

$$f_{fitness}(r) = \frac{1}{2}f_{inventorial}(r) + \frac{1}{2}f_{spatial}(r) \quad (1)$$

Furthermore, IC MAP-Elites adds interactive and continuous evolution to the Constrained MAP-Elites presented by Khalifa et al. [8]. This is done through an adaptive fitness function (based on the designer's design) that adapts the content generation, and by enabling the designer to flexibly change the feature dimensions and the granularity of the cells. It also adapts the usability of MAP-Elites to generate dungeon and adventure levels in an MI-CC system, which gives more control to the designers over non-intuitive parameters and aspects of MAP-Elites, while providing a richer set of high-performing and diverse suggestions.

A. Illuminating Dungeon Populations with MAP-Elites

MAP-Elites explores the search space more vastly by separating interesting feature dimensions, that affect different aspects of the room, such as playability or visual aesthetics, from the fitness function, to categorize rooms into cells.

In this subsection, we firstly present all the current feature dimensions identified and implemented in EDD, secondly, we explain the transition from fixed evolution to continuous evolution, and lastly, we introduce and outline the IC MAP-Elites algorithm.

1) Dimensions: Dimensions in MAP-Elites are identified as those aspects of the individuals that can be calculated in the behavioral space, and that are independent of the fitness calculation. These are crucial in MAP-Elites, as they represent the discretized dimensions where individuals will be retained as the space is explored. EDD offers the designer the possibility to choose among the following dimensions, two at a time (examples of rooms generated using the dimensions are shown in Figure 2c):

Symmetry (Sym) and Similarity (Sim). We choose Symmetry as a consideration of the aesthetic aspects of the edited room since symmetric structures tend to be more visually pleasing for the user and relate to fairness distribution and human-made structures [23], [35], [36]. Similarity is used to present the user variations of their design but still preserving their aesthetical edits. Symmetry is assessed using only impassable tiles (i.e., walls) and evaluated along the X and Y axes and diagonals. The highest value is normalized by the total amount of walls and used as the symmetry score (Equation (2)). Similarity is calculated by comparing tile by tile with the target room (Equation (3)). In-depth descriptions and evaluations can be found in [23], where both dimensions were used as aesthetic fitness evaluations.

$$D_{sym} = \frac{highestSymmetricValue}{totalWalls} \quad (2)$$

$$D_{sim} = \frac{totalTiles - notSimilarTiles}{totalTiles} \quad (3)$$

Number of Meso-patterns (NMP). The number of meso-patterns correlates to the type and amount of encounters the designer wants the user to have in the room in a more structured manner. The considered patterns are the treasure room (tr), guard rooms (gr), and ambushes (amb). Meso-patterns associate utility to a set of tiles in the room, for instance, a long chamber filled with enemies and treasures could be divided into 2 chambers, the first one with enemies and the second one with treasures so the risk-reward encounter is more understandable for the player. Since we already analyze the rooms for all possible patterns, the number of meso-patterns is simply $\#MesoPat = tr, gr, amb \in AllPatterns$. Equation (4) presents the dimensional value, and since the used meso-patterns can only exist in a chamber, we normalize by the maximum amount of chambers in a room, which are of a minimum size of 3×3, and results in $Max_{chambers} = \lfloor Cols/3 \rfloor \cdot \lfloor Rows/3 \rfloor$.

$$D_{NMP} = \min\left\{\frac{\#MesoPat}{Max_{chambers}}, 1.0\right\} \quad (4)$$

Number of Spatial-patterns (NSP). By spatial-patterns we mean chambers (c), corridors (cor), connectors (con), and nothing (n). We identify the number of spatial-pattern relates to how individual tiles group (or not) together to form spatial structures in the room. The higher the amount of spatial-patterns the lesser tiles will be group together in favor of more individualism. For instance, a room with one spatial-pattern can be one with no walls and just an open chamber, while a room with a higher number of spatial-patterns would subdivide the space with walls, using tiles for more specific patterns. Equation (5) presents how we calculate the value for such a dimension. The number of spatial patterns is simply $\#SpatialPat = c, n, cor, con \in AllPatterns$, we then normalize it by the largest side of the room and multiply it by a constant value, determined as $K = 4.0$ through a process of experimentation.

$$D_{NSP} = \min\left\{\frac{\#SpatialPat}{\max\{Cols, Rows\} \cdot K}, 1.0\right\} \quad (5)$$

Linearity (Lin). Linearity represents the number of paths that exist between the doors in the room. This relates to the type of gameplay the designer would like the room to have by the distributions of walls among the room. Having high linearity in a room does not need to only be by having a narrow corridor between doors but could also be generated by having all doors in the same open space (i.e. the user would not need to traverse other areas) or by simply disconnecting all paths between doors. Equation (6) shows the linearity calculation. Due to the use of patterns, we calculate the paths between doors as the number of paths that exist from a spatial-pattern containing a door to another. Finally, this is normalized by the number of spatial patterns in combination with the number of doors and their possible neighbors.

$$D_{lin} = 1 - \frac{AllPathsBetweenDoors}{\#spatialPat + \#NeighborsPerDoor} \quad (6)$$

Inner Similarity (IS). Inner similarity compares the target room to one generated, considering only the distribution and ratios of micro-patterns in both rooms rather than any aesthetic criteria. Specifically, we look into the density (den) and sparsity (spa) of enemies (en), treasures (tre), and walls (wal) in the target room (R_{tg}) in comparison with the generated rooms (R_{gen}). To calculate the density and sparsity of each micro-pattern, we first clustered all micro-patterns of the same kind based on the distance within the room (i.e. $distance = 1$). We then use these clusters to calculate the density (Equation (7)) using as density threshold $\theta = 4$ for treasures and enemies, and $\theta = 6$ for walls, and calculate the sparsity (Equation (8)). Finally, we calculate the difference between each distribution in R_{tg} and R_{gen}, linearly combining all the values into the IS measure as in Equation (9).

$$den(x) = \frac{\sum_{i=1}^{|clusts|} \min\left\{1.0, \frac{|clust_i|}{\theta_x}\right\}}{|clusts|} \quad (7)$$

$$spa(x) = \frac{\sum_{i=1}^{|clusts|} \sum_{j=1, j\neq i}^{|clusts|} \frac{Dist(clust_i, clust_j)}{|room|}}{|clusts| \cdot (|clusts| - 1)} \quad (8)$$

$$D_{IS} = \sum_{i=1}^{micropats} |den(R_{gen}) - den(R_{tg})| + \\ |spa(R_{gen}) - spa(R_{tg})| \quad (9)$$

Leniency (Len). Leniency calculates how challenging a room is at any given point. It is based on the amount of enemies and treasures that are in a room, their density and sparsity calculated as in eq. (7) and (8), respectively, and how safe the doors are (i.e. entry/exit points) calculated as in [24]. We base our calculation in the idea that rooms are less lenient the more enemies they contain as well as how they are distributed, counterbalanced by the number of treasures as they reward players. We calculate the dimension value for each room as shown in (Equation (12)), which uses a combination of precomputed non lenient (Equation (10)) and lenient (Equation (11)) values.

$$nonLenientValues = w_0 \cdot \log_{10}(|en| \cdot spa(en)) + \\ w_1 \cdot \log_{10}(|en| \cdot den(en)) + w_2 \cdot (1.0 - door_{safety}) \quad (10)$$

$$LenientValues = {}^1\!/_2 \log_{10}(|tre| \cdot spa(tre)) + \\ {}^1\!/_2 \log_{10}(|tre| \cdot den(tre)) \quad (11)$$

$$D_{len} = 1.0 - (nonLenientValues - ({}^1\!/_2 \cdot LenientValues)) \quad (12)$$

(a) Basic room (b) Complex room
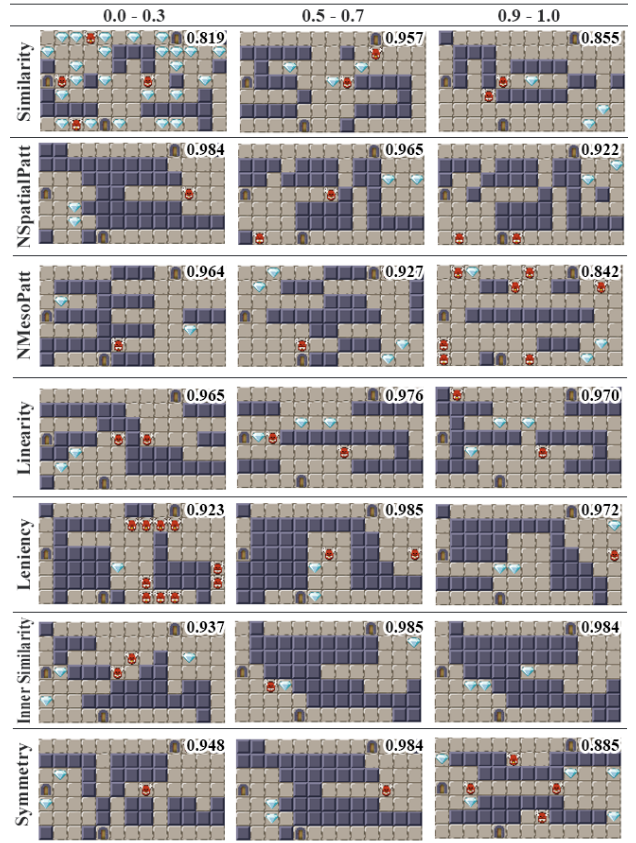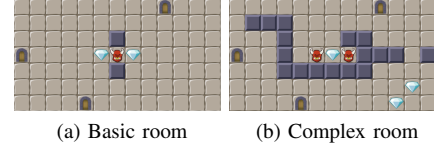
(c) Example Elites generated using IC MAP-Elites with (b) as target.

Fig. 2. (a) and (b) represent target rooms used in the experiments. Each row in (c) represents an independent run of the algorithm using the dimension specified to the left. Each column splits the dimension score into three intervals. Each cell displays (top-right) the fitness of the optimal individual in its related interval.

2) Continuous and Interactive Evolution: Since EDD already uses a FI2Pop, we took the Constrained MAP-Elites, presented by Khalifa et al. [8], as a starting point. The illuminating capabilities of MAP-Elites explore the search space with the constraints aspects of FI2Pop. This approach manages two different populations, a feasible and an infeasible one, within each cell. Individuals move across cells when

their dimension values change, or between the feasible and infeasible population according to their fulfillment of the feasibility constraint.

As discussed by Takagi, interactive evolution is a way to improve the capabilities of Evolutionary Algorithms (EA) by having humans in-the-loop to subjectively evaluate individuals. This hybrid approach has proven to reach better and more adaptive results but at the expenses of user's fatigue and user's understanding of the EA and the given problem [37]. However, in EDD and the IC MAP-Elites, the user does not directly evaluate individuals; instead, IC MAP-Elites adapts the fitness and search based on different interactions the designer has with the algorithm.

The designer can interact with the crossover step by locking tiles [23], and with the dimensions and cells by changing the dimensions and granularity for the MAP-Elites, enabling IC MAP-Elites to focus on different regions of the generative space. Furthermore, IC MAP-Elites constantly updates the target room and configuration with the most recent version of the designer's design. Once the suggestions are broadcasted, that room is incorporated without changes to the population of individuals in the corresponding cell.

This adaptability feature and different designer's indirect interactions with IC MAP-Elites, enables the implementation of continuous and interactive evolution, as well as allowing the designer to focus solely in the design of the room, while IC MAP-Elites adapts to the new designs.

Algorithm 1 Interactive Constrained MAP-Elites

1: **procedure** IC MAP-ELITES($[\{d_1, v_1\}, ..., \{d_n, v_n\}]$)
2:   $target \leftarrow curEditRoom$ ▷ Always in background
3:   createCells($[\{d_1, v_1\}, ..., \{d_n, v_n\}]$)
4:   **for** $i \leftarrow 1$ to $PopSize$ **do**
5:     add mutate($target$) to $population$
6:   CheckAndAssignToCell($population$)
7:   **while** true **do**   ▷ start continuous evo
8:     **for** $generation \leftarrow 1$ to $publishGen$ **do**
9:       **if** $dimensionsChanged$ **then**
10:         $previousPop \leftarrow cells_{pop}$
11:         createCells($newDimensions$)
12:         checkAndAssignToCell($previousPop$)
13:       **repeat** [for feasible & infeasible pop.]
14:         **for** $i \leftarrow 1$ to $ParentIteration$ **do**
15:           $curCell \leftarrow$ rndCell($cells$)
16:           add tournament($curCell$) to $parent$
17:         $offspring \leftarrow$ crossover($Parent$)
18:         checkAndAssignToCell($offspring$)
19:       sortAndTrim($cells$)
20:     broadcastElites()   ▷ render elites
21:     $pop' \leftarrow cells_{population}$
22:     add mutate($cells_{pop}$) to pop'
23:     add $target$ to pop'
24:     checkAndAssignToCell (pop')
25:     sortAndTrim($cells$)
26: **procedure** CREATECELLS(DIMENSIONS)
27:   **for each** $dim \in dimensions$ **do**
28:     add newCell(dim_d, dim_v) to $cells$
29: **procedure** CHECK&ASSIGNTOCELL($curPopulation$)
30:   **for each** $individual \in curPopulation$ **do**
31:     $individual_f \leftarrow$ evaluate($individual$)
32:     $individual_d \leftarrow$ dim($individual$)
33:     add $individual$ to $cell_{pop}(individual_d)$

---

3) Algorithm: IC-MAP-Elites is depicted in Algorithm 1. Cells are first created based on the dimensions selected by

the user and proceed to initialize the population based on the user's design, evaluate it and assign each individual to the corresponding cell. Before starting each generation, we check if the dimensions have changed, and if so, recreate the cells and populate them with the previous individuals, and proceed through the evolutionary strategies. We first select uniformly random which cell to choose parents from, and then we select 5 parents through tournament-selection. Offspring are produced through a two-point uniform crossover operation with a 30% chance of mutation. Offspring are placed in the correct cell and population after calculating their fitness and dimension's information. Finally, cells eliminate the low-performing individuals that over-cap their maximum capacity. Since interbreeding is not allowed, and can only happen indirectly (i.e. the offspring changing population and then used for breeding in consequent generations), the strategies are repeated for each of the populations.

IC MAP-Elites runs for n generations, and once it reaches the specified limit, it broadcasts the found elites. In order to foster the exploration, we first mutate all the individuals from all the populations and cells (while retaining the previous population), and add them into the same pool together with the current edited room without changes. Finally, we evaluate and assign all the individuals to the correct cells, and cells that are over maximum capacity eliminates low-performing individuals. This procedure is repeated until the user decides to stop the algorithm.

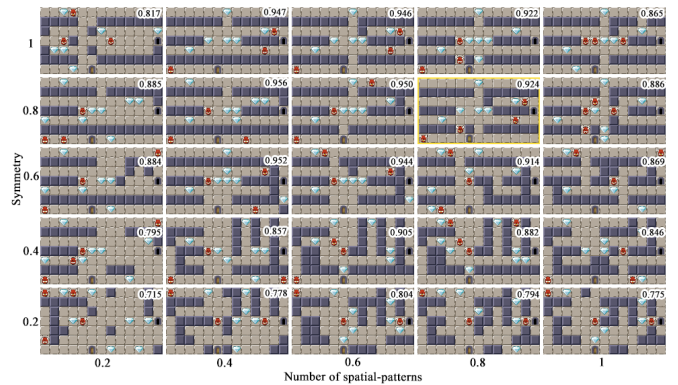## IV. PERFORMANCE EVALUATION ACROSS DIMENSIONS

Fig. 3. Rooms at generation 2090 targeting Number of spatial-patterns (X) and Symmetry (Y). Each cell displays (top-right) the fitness of the optimal individual in its related feasible population.

First, we ran a set of experiments to test the results from the IC MAP-Elites using all possible combinations of the available dimensions using two dimensions at a time. All experiments were run using 13×7 rooms, the same room size as in *The Binding of Isaac* [38], a representative example of a dungeon-based adventure game. In each experiment, the initial population was set to 1000 mutated individuals distributed in feasible and infeasible populations in all cells which were set to a maximum capacity of 25 individuals each. IC MAP-Elites ran continuously, and every 100 generations rendered the elites of each cell. At each generation, it selected 5 parents per population among uniformly random chosen cells. Offspring

Evaluation	◇	△	◯
IC MAP-Elites: Avg. dimensions in pairs	0.89	36.46%	52.4%
IC MAP-Elites: All Dimensions	0.78	51.7%	N/A
Objective-based EA	0.92	22.48%	N/A

◇ Avg. Fitness   △ Avg. coverage across all dimensions
◯ Avg. coverage in respective dimension pair

were always produced through a two-point crossover and had a 30% chance of being mutated, which would randomly alter one tile in the level.

Figure 3 shows a grid containing the best found suggestions at generation 2090, while aiming for number of spatial-patterns at the X-axis and symmetry at the Y-axis with a granularity of 5. Each cell displays the optimal individual of the feasible population under a given pair of dimension values. The fitness score is displayed on the cells' top-right corner.

The fitness evaluation in IC MAP-Elites is quite lightweight in terms of computational cost, which enables the grid of suggestions to be completed in a matter of seconds. This is of principal importance for successfully implementing continuous evolution, so that the influence of each manual change in the edited rooms is reflected in the suggestions almost instantly.

Results in Figure 3 are representative of the good quality diversity solutions produced by EDD. The average fitness across elites is 0.872, and the highest fitness is 0.956 (cell $[0.4, 0.8]$). No two rooms are the same. As intended, high levels of symmetry are displayed in the upper rows, gradually decreasing towards the bottom row. Similarly, rooms in the leftmost column contain lower amounts of spatial patterns, increasing towards the rightmost column. Lower amounts of spatial patterns translate into more open rooms with almost no corridors and one or two large adjacent chambers (as in cell $[0.2, 0.2]$), as opposed to highly pattern filled rooms that comprise intricate pathways converging at one or two small chambers (cell $[1, 0.2]$).

Fitness values show that some dimension combinations are harder to optimize than others, so that the whole grid depicts a gradient landscape of the compatibility between each pair of dimensions. The bottom-left corner shows difficulties producing symmetric rooms with low amounts of spatial patterns, as opposed to rooms with many corridors (upper-right corner), which seem to favor the generation of symmetrical structures.

Additional similar experiments can be found in [26], where we combined other pairs of dimensions and analyzed and discussed the correlation and limitations found in them. However, in the following section, we present an extension of those evaluations through a more exhaustive and in-depth assessment of IC MAP-Elites.

V. EXPRESSIVE RANGE ANALYSIS

We ran a second set of experiments analyzing the expressive range [31] of the IC MAP-Elites using the 21 possible combinations of dimension pairs. We followed the same setup as presented in the previous section, and every 100 generations,

we collected the unique generated individuals' data, which was aggregated in all the expressive ranges and individually plotted in Figure 7. Through this, between 150 to 2001 individuals were produced every 100 generations. By exploring the expressive range and conducting a comparative analysis among the various dimension combinations, we intend (1) to assess IC MAP-Elites, its exploration and exploitation capabilities, and the feature dimensions, (2) to analyze how different dimension combinations affect the search for QD content, and (3) to identify bias in the search space.

Moreover, in Table I we present a comparison of the avg. diversity and avg. quality of the generated individuals between different approaches to support our evaluation. The approaches consist of IC MAP-Elites picking dimensions in pairs (avg. of the aggregated results), IC MAP-Elites using all dimensions at the same time in the search, and the objective-based EA that was used in EDD previous to IC MAP-Elites. When using a pair of dimensions in IC MAP-Elites, the search explores in avg. 52.4% in the respective pair of dimensions (individual results are presented in Figure 4) and the individuals generated have an avg. fitness of 0.89. In comparison, when using all dimensions rather than only pairs, the search explores 51.7% across all dimensions, 15% more than when using pairs, but with a drop in the avg. fitness of the population (0.78), which is discussed further in Section V-B and shown in Figure 6. As expected and concluded before by Mouret and Clune [19], objective-based EA generated individuals with an avg. high fitness (0.92), but with low diversity (22.48%). This means that the search had a narrower focus, and the generated levels did not differ much from each other. Objective-based ran for 5000 generations, but it stagnated and stopped generating novel levels after 13 generations i.e., stopped exploring, generating a total of 1669 levels. Conversely, IC MAP-Elites kept generating novel levels until stopped, but considerably less after 1000 generations.

Figure 4 shows the expressive range of the IC MAP-Elites with each letter referring to a unique pair of dimensions tested, and the subfigure divided into two different plots. In the left plot, we evaluate the setup based on the used pair of dimensions with each hexagon placed in relation to their dimensions' score. The hue of each of them is connected to the number of unique suggestions generated. Likewise, in the right plot, we evaluated the setup based on its linearity and leniency score, which is used to compare the setups' expressiveness. All the setups were run for 5000 generations, using the same target room, which is shown as an orange marker (Figure 2a).

In Figure 4, it is shown that with IC MAP-Elites, it is explored a substantial area of the generative space (denoted in the figure with ◯ under each subfigure) rather than just exploiting the area around the target room, depicted as an orange marker. On average, in all the independent runs in their respective dimensions, the search explores around 52% of the space, filling at least half of the map with high-performing elites averaging 0.89 (as shown in Table I). It can also be observed that the dense areas of the search space (i.e., where the algorithm exploited the most) are distant from the target room's scores, and most of them are sparse throughout the search. This indicates that using IC MAP-Elites and a pair of
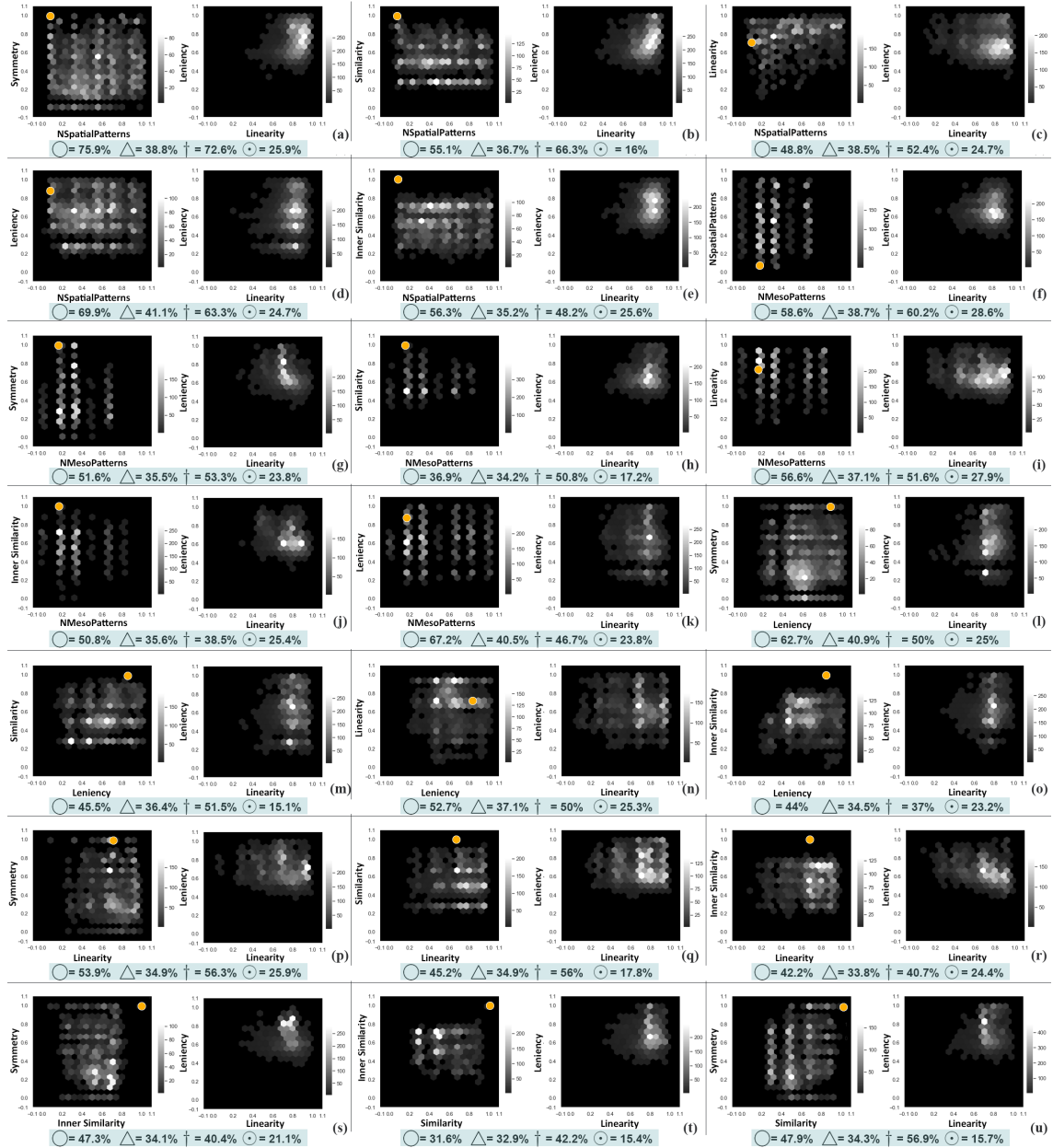
Fig. 4. Expressive range of the 21 possible combinations (from a to u) of dimensions picked in pairs. Each subfigure is composed of two plots: (left) an evaluation based on a given pair of dimensions; (right) the same pair of dimensions but evaluated in terms of Linearity and Leniency scores. Each hexagon relates to a dimensional score, whereas a lighter hue indicates a higher number of unique individuals generated under that particular score. All runs used Figure 2a as the target room, and its dimensional score is highlighted with an orange mark. Under each subfigure, we present data particular to the specific tested pair of dimensions. ◯, †, and ⊙ represent the coverage percentage in the respective pair of dimensions when running IC MAP-Elites with the subfigure's pair of dimensions (◯), IC MAP-Elites with all dimensions in the search (†), and when using objective-based EA (⊙). △ represents the avg. coverage percentage across all dimensions when running IC MAP-Elites with the subfigure's pair of dimensions.

dimensions at a time helps the distribution of the search while exploiting promising areas, and the search is less likely to be biased towards creating levels similar to the target.

Nevertheless, when using both leniency and linearity to compare the performance of the dimensions' pairs, it is shown that they are underexplored and within the same range (0.4-1.0) when not using them as dimensions. This points towards the search having difficulties getting out of other dimensions' local optima, especially since the densest search area is within the target room (this is denoted with △ under each subfigure).

A. Alternative Scenarios

In Figure 5, we examine how the algorithm would vary its dimensions' exploration and exploitation in two different scenarios. (a) Using the same target room as in all the cases in Figure 4 but using all the possible dimensions in the search space (i.e. 7), and (c) using NMP and NSP (see Section III-A1) as dimensions in the search but changing the target room to Figure 2b. To draw a better comparison, we added (b), which shows how all dimensions were explored when using Figure 4f (NMP-NSP) as pair of dimension.
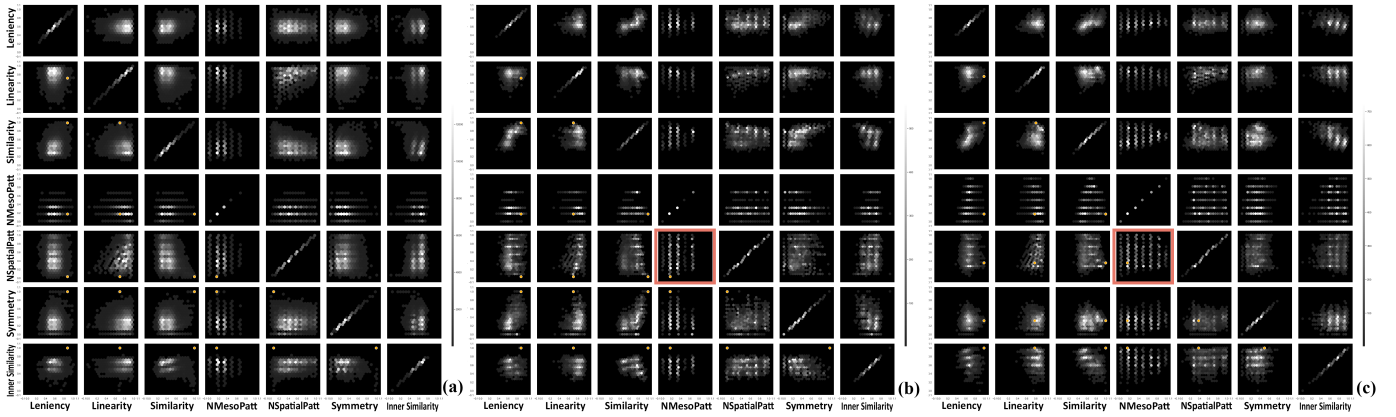
Fig. 5. In-detail run showing how all dimensions relate to each other in alternative scenarios. In (a) IC MAP-Elites ran for 2000 generations using the same target room as in Figure 4 (Figure 2a), but using all the dimensions in the search, which results in 78125 cells. (b) Shows how all dimensions were explored when using Figure 4f (NMP-NSP) as pair of dimension. In (c) we used the same dimensions as in (b) but we changed the target room to Figure 2b. The dimensional score of the respective target room in each subfigure (Figure 2a for a and b, and Figure 2b for c) is highlighted with an orange mark. In b and c, the dimensions used for the experiment are highlighted with a red border.
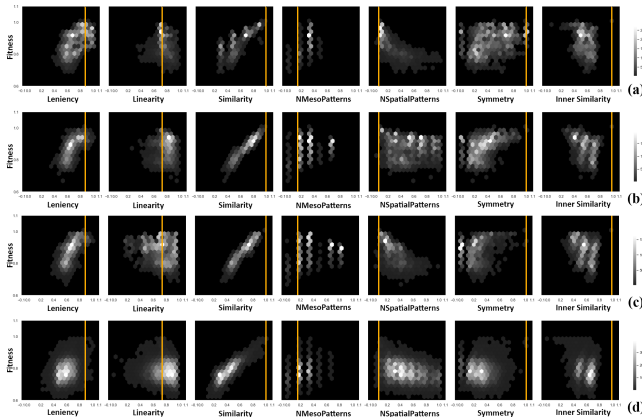

Fig. 6. Relation between dimension score and fitness score. (a), (b), and (c) show results using, respectively, Similarity and Symmetry, NMP and NSP, and NMP and Linearity. (d) was run using all the 7 dimensions. All runs used Figure 2a as target room, and it's dimensions score are denoted with an orange line across the different plots.

When using all the dimensions (a), IC MAP-Elites can explore a substantial area of the search space in each of the dimensions (51.7%), which in total is 15% more searched space than when using only a pair of dimensions on average. This is expected since all the dimensions are now acting as archives. However, as it is noted in Figure 4 under each subfigure, the actual explored space in the respective pairs (○) is most of the time greater than what using all dimensions explore in the respective pair (†).

It can also be observed that when using NMP and NSP as dimensions Figure 5b and c, regardless of the target room, IC MAP-Elites manages to still search a good amount of space (in avg. 38.7% and 43% among all dimensions, respectively). We suspect that this is because the range between low and high scores in the NMP or NSP dimensions produces very different rooms, as it can be seen in Figure 2c in their respective rows.

Moreover, when comparing (a) and (b), it is noticeable that while (a) explores a greater area than (b) (in general, 15.24% more), it seems to be recurrently generating the same type of

individuals (i.e. depicted with the hue of the hexagon) while in (b), the dense areas for most of the dimensions are sparser, especially when matching the pair of dimensions used for evaluation. Finally, it should be noted that while these three plots are comparable in their diversity search, they differ in the number of elites they store during the search, with an archive of 78125 cells for (a), and 25 cells for (b) and (c).

B. Fitness Evaluation

Figure 6 shows the relation of the fitness with the explored individuals in each dimension in 4 independent runs. (a, b, c) Were runs using dimensions in pairs with the same data and dimensions as in Figure 4 (u), (f), and (h), respectively. (d) Was run using all the dimensions in the search space with the same data as in Figure 5 (a). There is an evident high correlation between Similarity scores and fitness across all the subfigures, which is expected since our fitness value is highly dependant on the target's ratios. In contrast, IS is not even close to match the fitness curve of Similarity rather there are high-performing individuals along the dimension, even when IS calculates similarity using ratios, densities, and sparsities of the target's micropatterns to calculate the score of individuals.

Moreover, our experiment shows that when using specific dimensions (Figure 6a-c), we achieve a relatively better search (i.e. find more diverse and high-quality individuals) in those dimensions, while still being able to explore the rest of dimensions. For instance, when not using NSP as a feature dimension such as in (a) or (c), the NSP dimension is fully explored but generating no high-quality individuals, meanwhile, when using NSP as a feature dimension as in (b), the search can find individuals in the same range as in (a) or (c) but with a higher fitness. Similar results can be seen in the rest of the dimensions where the search uses specific dimensions, for instance, in (a) exploring diverse and higher quality individuals in Similarity, or in (c) in Linearity.

The most interesting result can be seen in (d), where we used all dimensions in the search. This allows for a vast search of diverse individuals in all dimensions, but at the same time

it seems to exploit sub-optimal areas. On the other hand, when using only a pair of dimensions as in (a), (b) and (c), the search remained dense in high-quality individuals in all dimensions.
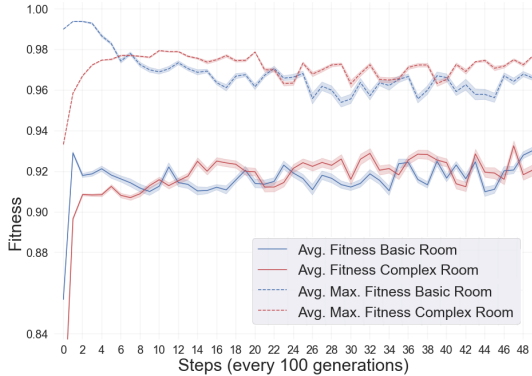

Fig. 7. Fitness evolution per 100 generations throughout all the runs. Blue represents the fitness of the basic room (Figure 2a), and Red represents the fitness of the complex room (Figure 2b).

Furthermore, figure 7 shows the aggregated fitness over time of all dimension pairs (21). The avg. max fitness and the avg. total fitness are depicted as a dashed line and a continuous line, respectively. Both are surrounded by a low opacity thicker line representing the confidence interval. The graph shows that across all pairs of dimensions, the avg. fitness of novel generated levels is very high regardless of the target room and the generation, with a minor fluctuation between 0.92 and 0.94. In addition, the graph furthers supports what is presented in Figure 6 a, b, and c, with most of the individuals generated in high score regions of the space in relation to fitness, but also showing that this is stable throughout the generations. The stable fitness

This stationary overall high fitness is expected from MAP-Elites as it aims at constantly generating high-performing diverse individuals. This diversity goal, combined with an adaptive fitness dependant on the target room, makes it harder to generate levels with maximum fitness. Figure 7 clearly shows this; depending on the fitness, there are scores in dimensions or pairs of dimensions that are less compatible with the fitness evaluation. Yet, what has been shown thus far in the multiple expressive range analysis and especially in Figure 6a-c, is that IC MAP-Elites is able to generate high-performing and diverse levels.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have done an in-depth evaluation of IC MAP-Elites by analyzing the expressive range of all the possible pairs of dimensions, their relation to other dimensions and the fitness. Our results indicate specific dimensions in level generation, such as when using NMP, NSP, LENIENCY, or Symmetry, that foster greater exploration. The exploration is not only fostered in their respective dimensions (in avg. 54%, 61%, 57%, 57%, respectively, when used) but also in all the others due to the diverse individuals generated within the respective dimensions as shown in Figure 2c. As observed on Figure 6, aesthetic feature dimensions such as *Similarity* and *Symmetry* have an impact when used or not

in the search. When not using *Symmetry*, the search does not explore high levels of symmetry, disregarding to some extent that aesthetic feature in favor of exploring the other feature dimensions. Moreover, *Similarity* has a high correlation with fitness as observed in Figure 6 and depending on the designer's objective, this might affect positively or negatively since there will be a strong bias towards highly similar individuals. In contrast, IS seems to be more robust in the fitness landscape and the exploration of other dimensions because it captures the properties of the target room rather than its aesthetics.

Regarding the bias in the exploration when using IC MAP-Elites together with continuous and adaptive evolution, our results show that the generated content is highly diverse with dense areas along most of the searched space, which is shown as well in Table I. This means that due to the diversity pressure imposed by IC MAP-Elites, the search is unlikely to be biased towards creating content that is similar in the feature dimensions' scores of the target room. Yet IC MAP-Elites adapts to the target room and generates high-performing individuals along the rest of the space in the other dimensions, especially in those explicitly used in the search.

To further assess the algorithm, we ran an experiment using all possible dimensions (Figure 5a) rather than specific pairs to observe the exploration and exploitation of the algorithm when not using specific pair of dimensions. As expected, it explores a substantial area of the search space in all dimensions (in avg. 51.7%) but surprisingly, the search results in the exploitation of sub-optimal individuals in all dimensions with an avg. fitness of 0.78 as shown in Figure 6d. While using pair of dimensions results on 15.24% reduced exploration, the exploitation seems to be fairly spread among the explored space, visible in Figure 5b and c, and the density related to fitness is focused on high-performing individuals Figure 6a-c. This points towards that there are difficulties in (1) **fully exploring** the space when using a pair of dimensions even when the exploitation is distributed and focused on high-performing individuals, and in (2) **exploiting the promising areas** of the search when using a higher range of dimensions even when the space is vastly explored.

In addition, based on our experiments, such difficulties are exacerbated since the exploration stagnates and keeps exploiting the same areas after ˜1000 generations, yet finding novel individuals. This happens regardless of the number of dimensions, which dimensions, or the target room. Our findings point to challenges in the *selection step* of IC MAP-Elites, which selects cells uniformly random. Exploring different methods for the selection of cells and individuals is a promising future step. For instance, Gravina et al. [29] explored how four divergent search algorithms guide cells' selection. benefit MAP-Elites standard selection method.

Furthermore, preliminary experiments seem to indicate that some dimensions, which are more explorative (e.g., NMP, NSP, Leniency, Symmetry), are more robust to changes in the target room, exploring similar areas of the search space regardless of the target, which can be observed in Figure 5 (b) and (c). This points towards dimensions that would make the algorithm more robust to changes, reinforcing its adaptability feature. Further experiments are needed to analyze how differ-

ent dimensions are better at adapting to continuous changes in the target room, which would also indicate better stability in the search. Along these lines, further evaluation is needed with human designers to assess and explore whether IC MAP-Elites is beneficial for the MI-CC workflow and interaction.

Currently, IC MAP-Elites is only used in EDD on a per room basis. Future work should consider the whole dungeon for the evolution procedure and further develop for the generation of complete dungeons using holistic metrics to evaluate the dungeon.

In conclusion, based on our results, IC MAP-Elites generate more diverse levels while retaining high-quality among them, which would result in richer and more options and suggestions to designers. Our experiments show that which dimensions are used significantly impact the search space, fostering the search of high-quality individuals within the selected dimensions while not discouraging exploration in other dimensions. This means that by editing their levels and choosing which dimension IC MAP-Elites should focus on, the designers will be given more meaningful choices and interactions.

ACKNOWLEDGMENT

REFERENCES

[1] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*. Springer, 2018, vol. 2.

[2] M. Toy, "*Rogue*," Game [PC], Epyx, 1980.

[3] D. Braben and I. Bell, "*Elite*," Game [PC], Acornsoft, 1984.

[4] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 172–186, Sep. 2011.

[5] N. Shaker, G. N. Yannakakis, J. Togelius, M. Nicolau, and M. O'neill, "Evolving personalized content for super mario bros using grammatical evolution." in *AIIDE*, 2012.

[6] E. J. Hastings, R. K. Guha, and K. O. Stanley, "Evolving content in the Galactic Arms Race video game," in *2009 IEEE Symposium on Computational Intelligence and Games*, Sep. 2009.

[7] J. Dormans, "*Unexplored*," Game [PC], 2017.

[8] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, "Talakat: Bullet hell generation through constrained map-elites," in *Proceedings of The Genetic and Evolutionary Computation Conference*. ACM, 2018.

[9] M. C. Green, G. A. B. Barros, A. Liapis, and J. Togelius, "DATA agent," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, ser. FDG '18. New York, NY, USA: ACM, 2018, pp. 1–10.

[10] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, "Can computers foster human users' creativity? theory and praxis of mixed-initiative co-creativity," *Digital Culture & Education*, vol. 8, no. 2, pp. 136–153, 2016.

[11] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity," in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014, pp. 1–8.

[12] T. Machado, D. Gopstein, A. Nealen, and J. Togelius, "Pitako-recommending game design elements in cicero," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019.

[13] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels." in *AIIDE*, 2013.

[14] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, Sep. 2011.

[15] T. Machado, A. Nealen, and J. Togelius, "Cicero: Computationally intelligent collaborative environment for game and level design," in *3rd workshop on Computational Creativity and Games (CCGW) at the 8th International Conference on Computational Creativity (ICCC'17)*, 2017, pp. 1–8.

[16] A. Liapis, G. N. Yannakakis, and J. Togelius, "Generating Map Sketches for Strategy Games," in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS. Springer, 2013.

[17] J. K. Pugh, L. B. Soros, and K. O. Stanley, "Quality diversity: A new frontier for evolutionary computation," *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016.

[18] D. Gravina, A. Khalifa, A. Liapis, J. Togelius, and G. N. Yannakakis, "Procedural content generation through quality diversity," in *2019 IEEE Conference on Games (CoG)*, 2019.

[19] J. Mouret and J. Clune, "Illuminating search spaces by mapping elites," 2015. [Online]. Available: http://arxiv.org/abs/1504.04909

[20] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, "On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch," *European Journal of Operational Research*, vol. 190, no. 2, pp. 310–327, 2008.

[21] A. Khalifa, M. C. Green, G. Barros, and J. Togelius, "Intentional computational level design," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '19. New York, NY, USA: ACM, 2019, pp. 796–803.

[22] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, "Fostering creativity in the mixed-initiative evolutionary dungeon designer," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, ser. FDG '18, 2018.

[23] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, "Assessing aesthetic criteria in the evolutionary dungeon designer," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, ser. FDG '18, 2018.

[24] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, "Towards pattern-based mixed-initiative dungeon generation," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, ser. FDG '17. New York, NY, USA: ACM, 2017.

[25] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, "Mixed-initiative procedural generation of dungeons using game design patterns," in *Proc. IEEE Conf. Computational Intelligence and Games (CIG)*, 2017.

[26] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, "Empowering quality diversity in dungeon design with interactive constrained map-elites," in *2019 IEEE Conference on Games (CoG)*, 2019.

[27] M. Cook, S. Colton, J. Gow, and G. Smith, "General analytical techniques for parameter-based procedural content generators," in *2019 IEEE Conference on Games (CoG)*, Aug 2019.

[28] M. Cook, J. Gow, and S. Colton, "Towards the automatic optimisation of procedural content generators," in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, Sep. 2016.

[29] D. Gravina, A. Liapis, and G. N. Yannakakis, "Blending notions of diversity for map-elites," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '19. New York, NY, USA: ACM, 2019, pp. 117–118.

[30] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[31] G. Smith and J. Whitehead, "Analyzing the expressive range of a level generator," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. New York, NY, USA: ACM, 2010.

[32] A. Summerville, "Expanding expressive range: Evaluation methodologies for procedural content generation," in *AIIDE*, 2018, pp. 116–122.

[33] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius, "A comparative evaluation of procedural level generators in the mario ai framework," in *Proceedings of the 9th International Conference on the Foundations of Digital Games*, ser. FDG '14, 2014, pp. 1–8.

[34] M. Cook, J. Gow, and S. Colton, "Danesh: Helping bridge the gap between procedural generators and their output," in *Proceedings of the 7th Workshop on Procedural Content Generation, FDG*, 2016, pp. 1–16.

[35] J. Mariño and L. Lelis, "A computational model based on symmetry for generating visually pleasing maps of platform games," in *AIIDE*, 2016.

[36] A. Liapis, G. N. Yannakakis, and J. Togelius, "Adapting models of visual aesthetics for personalized content creation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 3, pp. 213–228, 2012.

[37] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[38] E. McMillen and F. Himsl, "*The Binding of Isaac*," Game [PC], 2011.