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: 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 analysis of its behaviour, with the aims of evaluating the expressive range of all possible dimension combinations in several scenarios, as well as discussing their influence in the fitness landscape and in the overall performance of the mixed-initiative procedural content generation.**

***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 game rules, quests, and stories, to levels, maps, 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 during 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 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 [17]. Additionally, MI-CC provides insight and understanding 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 [18] 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 [19]. MAP-Elites [20] 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 [21], 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 [22].

The Evolutionary Dungeon Designer (EDD) is a MI-CC tool for generating dungeons for adventure games using a FI2Pop evolutionary approach [23]–[26]. This paper extends the previous research presented in [27], which introduced *Interactive Constrained MAP-Elites*, a combination of Constrained MAP-Elites with interactive evolution. The algorithm was implemented as a continuous evolution process that takes advantage of MAP-Elites' multidimensional discretization of the search space into cells. The previous paper [27] analyzed the effects of using quality-diversity in procedurally generating dungeons, as well as the effects of continuous evolution and dimension customization in a MI-CC approach. In the current work, we conduct a more in-depth analysis of the behaviour of the algorithm. Following recent research on how to evaluate procedural content generators and, in particular, quality-diversity approaches [28]–[30], we have extended EDD to include two further dimensions, and 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.

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 [20] and is a well-known quality-diversity algorithm. The map is divided into a

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).

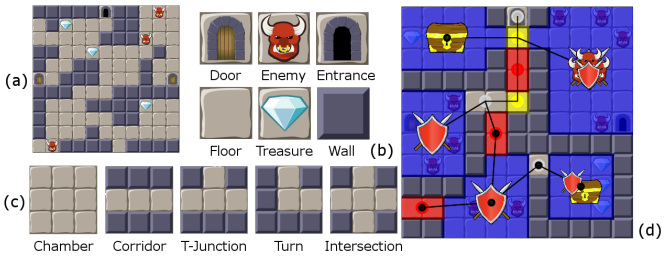

Fig. 1. The main components in EDD. (a) A basic room, (b) different placeable tiles, (c) micro-patterns and (d) meso-patterns [24].

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. [31], 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 expresivity measures [32] and have, for instance, explored difficulty measures [33]. Other approaches incorporates tool assisted parameter exploration due to its effect on the content space [28], [34].

C. Evolving Dungeons as a Whole, Room by Room

The Evolutionary Dungeon Designer (EDD) is a MI-CC tool that allows a human designer to create a 2D dungeon and the rooms it is composed of (Figure 1.a). The designer is able to manually edit both the dungeon by placing and removing rooms, and the individual rooms by editing the tiles (Figure 1.b) 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 (Figure 1.c and Figure 1.d). A detailed description of all EDD's features, including the use of game design patterns, can be found in [23]–[26].

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 [23]. The most significant upgrade is replacing the grid-based backbone that represented the dungeon by a more flexible graph-based representation. A dungeon is now 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. The designer can perform the following new actions:

- adding disconnected rooms to the dungeon. Rooms may also be removed at any time.
- connecting any pair of rooms by adding a new bi-directional connection to the graph. Rooms interconnect from and to passable border tiles (self-loops are not allowed). Both ends are marked with a door tile (Figure 1.b). A single border tile can only hold one connection, implying that a room can have as many connections as passable border tiles. Connections and rooms can be removed at any time, and their associated doors removed with them.
- calculating paths between any pair of passable tiles located in any connected room. Paths are automatically calculated according to one of the following heuristics: *fastest* returns the shortest path, *rewarding* returns the path that traverses the highest number of treasure tiles, *less danger* provides a path with the fewest number of enemies, whereas *more danger* does the opposite.

The designer is required to select one of the added rooms as the *initial room*, which is the first room the player meets when entering the dungeon. This selection can be modified unlimited times. The *initial room* is used by EDD to calculate the feasibility of the dungeon. A dungeon is considered feasible when there is at least one path between the *initial room* and any other passable tile in every room. Rooms and doors that are unreachable from the *initial room* are highlighted in red, so that they can be easily identified by the designer. This feasibility constraint ensures that all passable tiles are accessible, avoiding the possibility of accidentally creating unreachable areas.

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 2). This screen can be reached in two different ways:

1) directly: by double-clicking or zooming in (by using the mouse wheel or by pinching on the touchpad) on the room.
2) indirectly: by clicking on the "Start with our suggestions" button, six procedurally generated suggestions are displayed on a separate screen. The selected suggestion is then opened in the room editor screen.

Figure 2 shows the room editor screen displaying a sample room with the dimensions 7×5 tiles. The left pane lists all the available options for manually editing the room. Manual editing is carried out by brush painting over the room with one of the available tile types: floor, wall, treasure, or enemy. There are two brush sizes (single tile and five-tile cross shape), and control-clicking allows the designer to bucket paint all adjacent tiles of the same type. Brush painting with the lock button

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

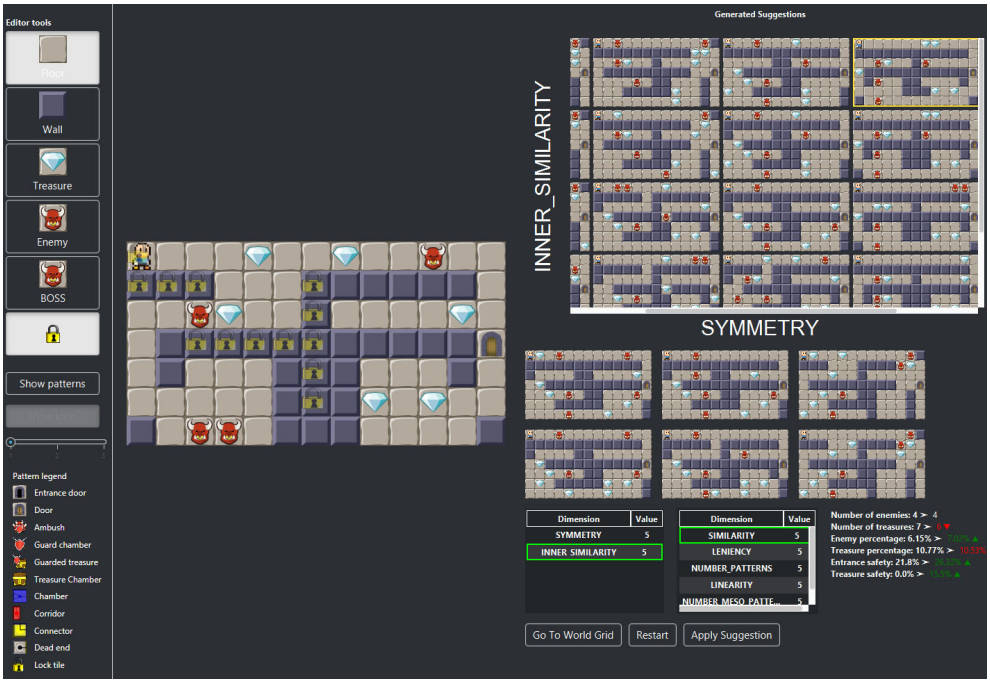

Fig. 2. The room editor screen in EDD. The left pane contains all the options for manually editing the room displayed at the center-left of the screen. The right section displays the procedurally generated suggestions.

on preserves selected tiles in all the procedurally generated suggestions. A detailed description of all the options in this pane is included in [23], [24].

The right side of the screen displays the procedurally generated suggestions, by means of the Interactive Constrained MAP-Elites genetic algorithm (see Section III). When the designer accesses the room editor screen, the EA 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 the EA, and ”Go To World Grid” takes the user back to the dungeon editor screen.

III. Interactive Constrained MAP-Elites

EDD uses a single-objective fitness function with a FI2Pop genetic algorithm where fitness is a weighted sum divided equally between (1) the inventorial aspect of the rooms, which relates to the placement of enemies and treasures in relation to doors and target ratios, and (2) the spatial distribution of the design patterns, which relates to the distribution between corridors and rooms, and the meso-patterns that those encompass. An in-depth explanation of EDD's fitness function can be found in [24], [25].

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 [20] and continuous evolution into EDD, our algorithm can (1) account for the multitude of 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).

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, using them to categorize rooms into niches (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 our current evolutionary algorithm, IC-MAP-Elites.

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. EDD offers the designer the possibility to choose among the following dimensions, two at a time:

Symmetry and Similarity. We choose Symmetry as a consideration of the aesthetic aspects of the edited room since symmetric structures tend to be more visually pleasing. Similarity is used to present the user variations of their design but still preserving their aesthetical edits. Symmetry is evaluated along the X and Y axes, backslash and front slash diagonal and the highest value is used as to how symmetric a room is. Similarity is calculated through comparing tile by tile with the target room. Formulas, information and support

for both evaluations are explained in greater details in [24], where both of them were used as aesthetic fitness evaluations.

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 $\#MesoPat = tr, gr, amb \in AllPatterns$. eq. (1) 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 $Max_{chambers} = \lfloor Cols/3 \rfloor \cdot \lfloor Rows/3 \rfloor$.

$$D_{NMP} = \min\left\{\frac{\#MesoPat}{Max_{chambers}}, 1.0\right\} \tag{1}$$

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. eq. (2) 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.

$$D_{NSP} = \min\left\{\frac{\#SpatialPat}{\max\{Cols, Rows\} \cdot K}, 1.0\right\} \tag{2}$$

Linearity. 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. eq. (3) 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.

$$D_{lin} = 1 - \frac{AllPathsBetweenDoors}{\#spatialPat + \#NeighborsPerDoor} \tag{3}$$

Inner Similarity (IS). Inner similarity compares the target room to one generated by the EA, 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 (eq. (4)) using as density threshold $\theta = 4$ for treasures and enemies, and $\theta = 6$ for walls, and calculate the sparsity (eq. (5)). 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 eq. (6).

$$den(x) = \frac{\sum_{i=1}^{|clusts|} \min\left\{1.0, \frac{|clust_i|}{\theta_x}\right\}}{|clusts|} \tag{4}$$

$$spa(x) = \frac{\sum_{i=1}^{|clusts|} \sum_{j=1, j\neq i}^{|clusts|} \frac{Dist(clust_i, clust_j)}{|room|}}{|clusts| \cdot (|clusts| - 1)} \tag{5}$$

$$D_{IS} = \sum_{i=1}^{micropats} |den(R_{gen}) - den(R_{tg})| + |spa(R_{gen}) - spa(R_{tg})| \tag{6}$$

Leniency. 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. (4) and (5), respectively, and how safe the doors are (i.e. entry/exit points) calculated as in [25]. 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 (eq. (9)), which uses a combination of precomputed non lenient (eq. (7)) and lenient (eq. (8)) values.

$$nonLenientValues = w_0 \cdot \log_{10}(|en| \cdot spa(en)) + w_1 \cdot \log_{10}(|en| \cdot den(en)) + w_2 \cdot (1.0 - door_{safety}) \tag{7}$$

$$LenientValues = 1/2 \log_{10}(|tre| \cdot spa(tre)) + 1/2 \log_{10}(|tre| \cdot den(tre)) \tag{8}$$

$$D_{len} = 1.0 - (nonLenientValues - (1/2 \cdot LenientValues)) \tag{9}$$

2) Continuous Evolution: EDD implements continuous evolution in two ways. First, the EA constantly updates the target room and configuration with the most recent version of the user's design, and once the suggestions are broadcasted, that room is incorporated without changes to the population of individuals in the corresponding cell. Secondly, by changing the dimension information and their granularity for the MAP-Elites, which can be done at any given time by the designer.

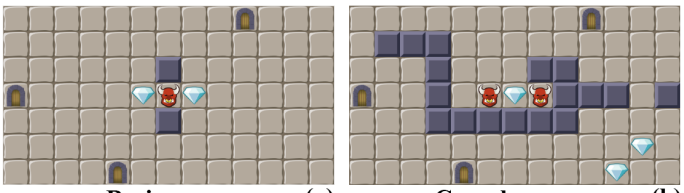

Fig. 3. Target Rooms used in the experiments.

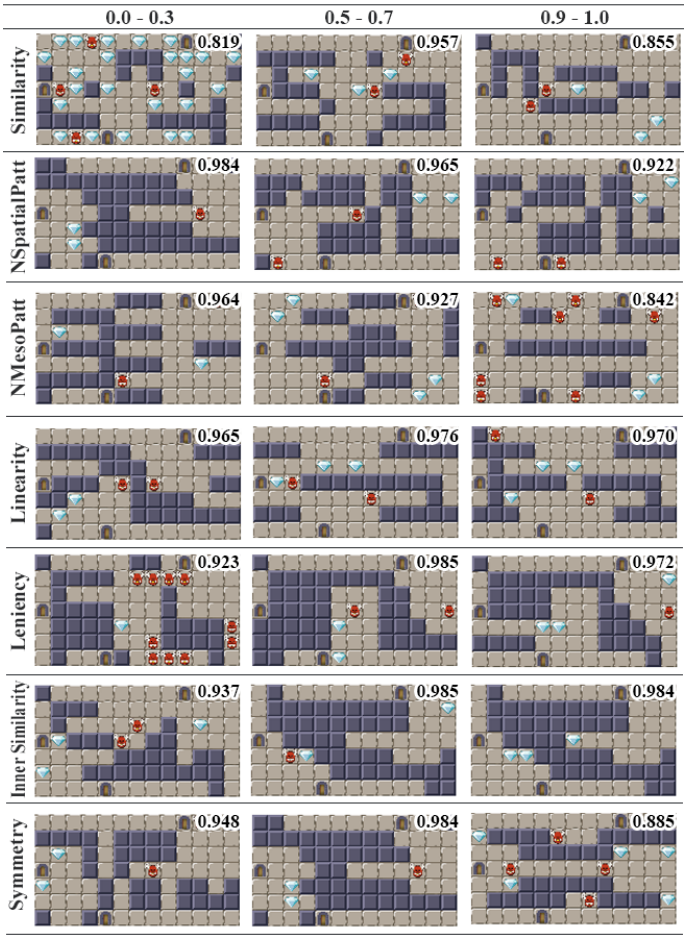

Fig. 4. Example of Elites generated using IC-MAP-Elites. Each row represents an independent run of the algorithm using the dimension specified to the left and fig. 3.b as target room. each column splits the dimension score into three intervals: 0.0-0.3 (low), 0.5-0.7 (medium), and 0.9-1.0 (high). Each cell displays (top-right) the fitness of the optimal individual in its related interval.

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.

3) Algorithm: The current evolutionary algorithm is depicted in Algorithm 1. Cells are first created based on the dimensions selected by the user and proceed to initialize the

Algorithm 1 Interactive Constrained MAP-Elites

procedure IC-MAP-ELITES($[\{d_1, v_1\}, ..., \{d_n, v_n\}]$)
  $target \leftarrow curEditRoom$ ▷ Always in background
  createCells($[\{d_1, v_1\}, ..., \{d_n, v_n\}]$)
  for $i \leftarrow 1$ to $PopSize$ **do**
    add mutate($target$) to $population$
  CheckAndAssignToCell($population$)
  while true **do** ▷ start continouous evo
    for $generation \leftarrow 1$ to $publishGen$ **do**
      if *dimensionsChanged* **then**
        $previousPop \leftarrow cells_{pop}$
        createCells($newDimensions$)
        checkAndAssignToCell($previousPop$)
      repeat [for feasible & infeasible pop.]
        for $i \leftarrow 1$ to $ParentIteration$ **do**
          $curCell \leftarrow$ rndCell($cells$)
          add tournament($curCell$) to $parent$
        $offspring \leftarrow$ crossover($Parent$)
        checkAndAssignToCell($offspring$)
      sortAndTrim($cells$)
    broadcastElites() ▷ render elites
    $pop' \leftarrow cells_{population}$
    add mutate($cells_{pop}$) to pop'
    add $target$ to pop'
    checkAndAssignToCell (pop')
    sortAndTrim($cells$)
procedure CREATECELLS(DIMENSIONS)
  for each $dim \in dimensions$ **do**
    add newCell(dim_d, dim_v) to $cells$
procedure CHECK&ASSIGNTOCELL($curPopulation$)
  for each $individual \in curPopulation$ **do**
    $individual_f \leftarrow evaluate(individual)$
    $individual_d \leftarrow dim(individual)$
    add $individual$ to $cell_{pop}(individual_d)$

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.

This procedure is repeated until the user decides to stop the algorithm. Meanwhile, the EA 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.

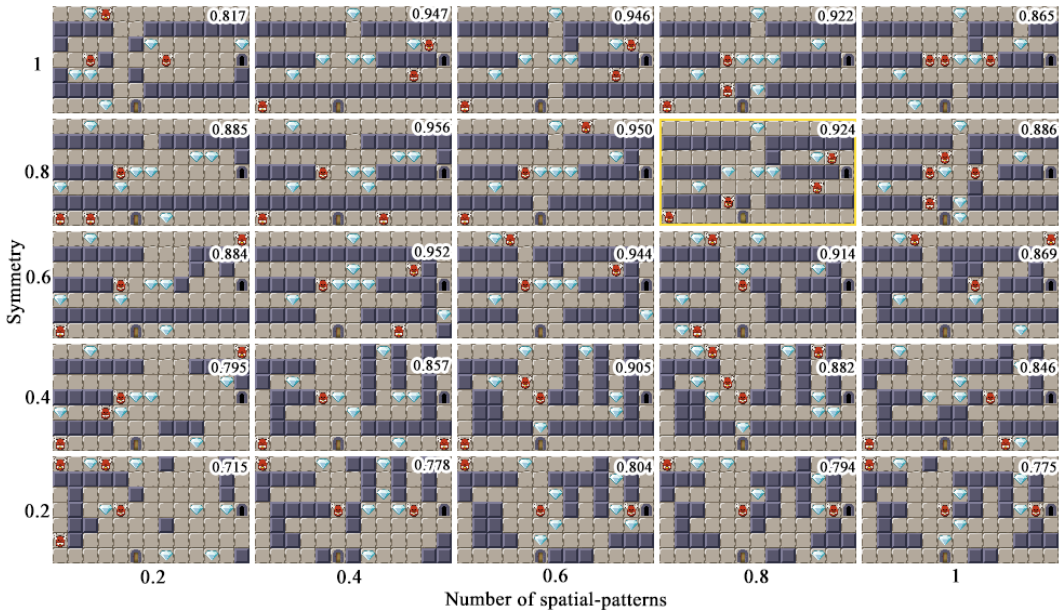

Fig. 5. Rooms at generation 2090 targeting Number of spatial-patterns (X) and Symmetry (Y). Each cell displays (top-right) the fitness of the optimal individual in its related feasible population.

IV. Performance Evaluation across Dimensions

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* [35], 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. The EA 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 were produced through a two-point crossover and were mutated with a 30% chance.

Figure 5 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. The feeling of immediacy is further increased through updating cells as soon as a new optimal individual is produced and incorporated to the cell's underlying feasible population.

Results in Figure 5 are representative of the good quality diversity solutions produced by EDD. The average fitness across cells 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. The bottom row shows that aiming for low symmetry generally produces slightly less optimal results, whereas the top row shows that corridors are the most favorable spatial pattern for building symmetric rectangular rooms.

Additional experiments were carried out combining other pairs of dimensions. The complete results can be found in [27], together with their analysis and discussion in terms of the existing correlations found between each pair of dimensions, as well as the effects of integrating the MAP-Elites approach into a continuously evolving environment.

V. Expressive Range Analysis

We ran a second set of experiments analyzing the expressive range [32] of the IC-MAP-Elites using the 21 possible combinations of dimension pairs. By exploring the expressive range, we are able (1) to analyze how and which different pairs of dimensions enable IC-MAP-Elites to explore a greater range

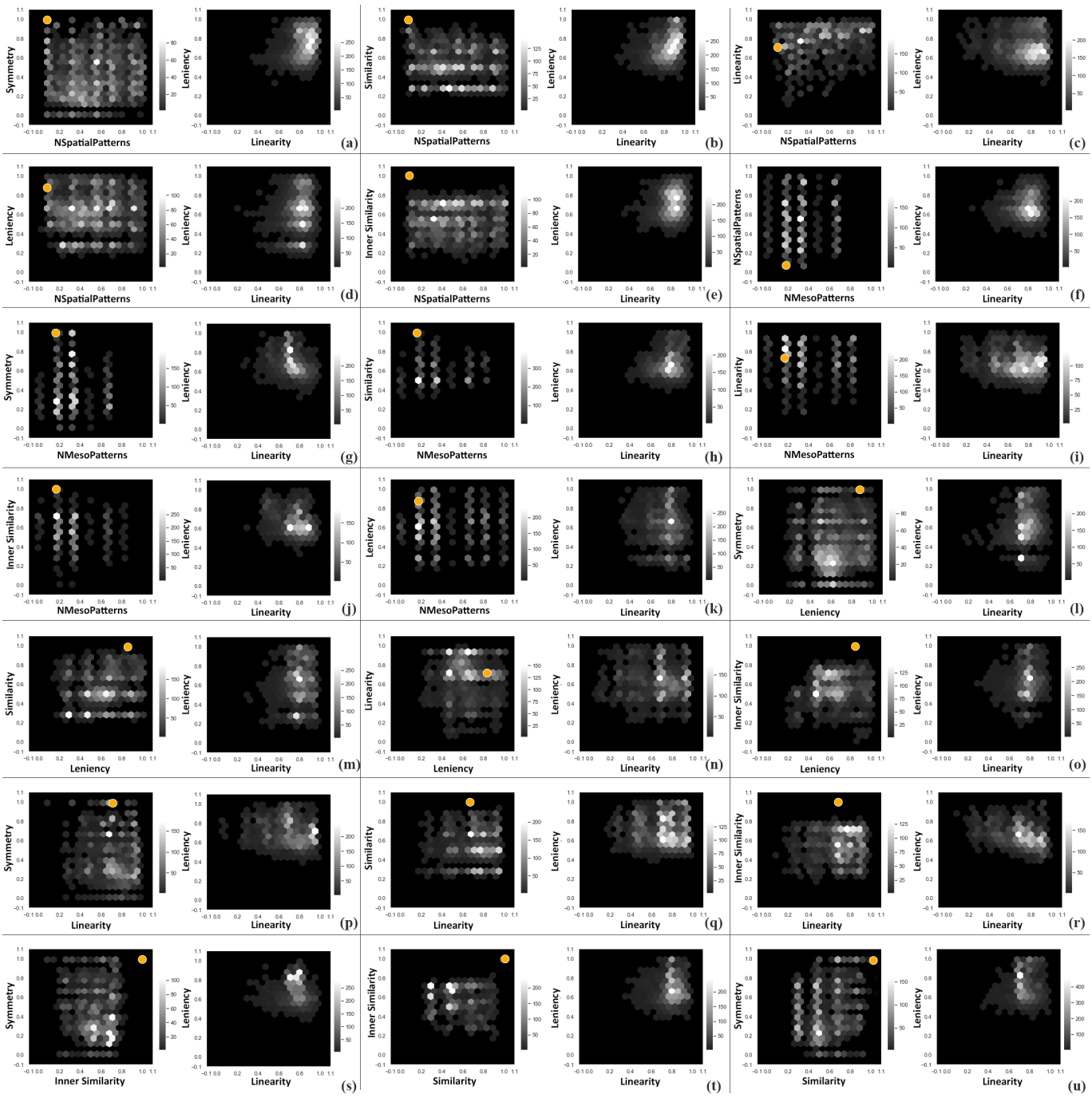

Fig. 6. Expressive range of the 21 possible combinations (from a to u) of dimensions picked in pairs. Each subfigure is composed of two plots: (left) an evaluation based on a given pair of dimensions; (right) the same pair of dimensions but evaluated in terms of Linearity and Leniency scores. Each hexagon relates to a dimensional score, whereas a lighter hue indicates a higher number of unique individuals generated under that particular score. All runs used fig. 3.b as target room and its dimensional score is highlighted with an orange mark.

of individuals while retaining high-quality, (2) to conduct a comparative analysis among various dimensions, and (3) to identify bias in the search space.

When Smith and Whitehead [32] introduced the expressive range analysis of generators as an evaluation tool to examine the variety of the generated artefacts and the impact of different setups. In the paper, they highlight the importance of using comparison metrics that can measure emergent properties of the generated content. Map-Elites dimensions are features of interest in the search space that are not used to evaluate the population, rather they define the feature space of interest [20]. Considering this and the previous work, we compare setups based on their behavior dimensions and on Leniency and Linearity, regardless of the used dimensions.

Figure 6 shows the expressive range of the IC-MAP-Elites with each letter referring to a unique pair of dimensions tested,

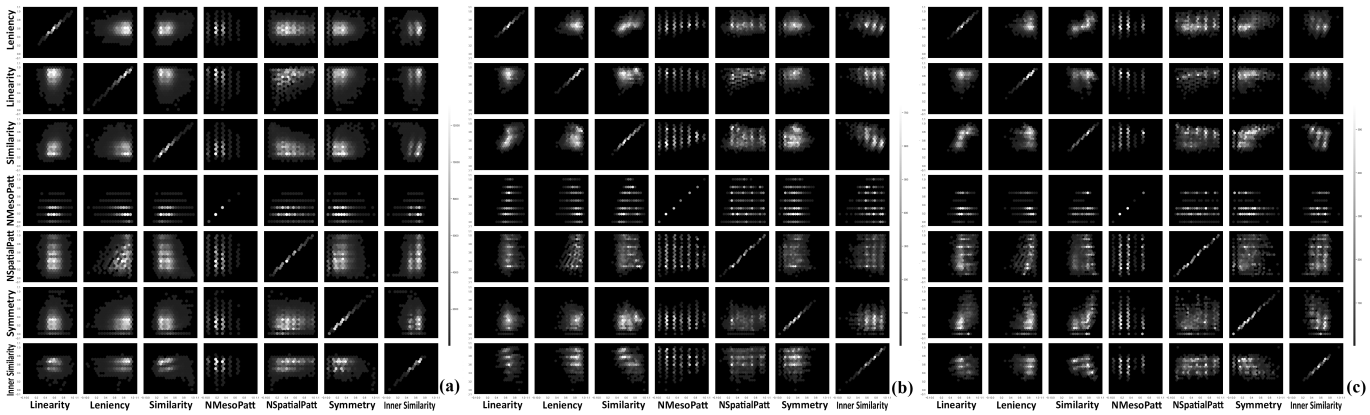

Fig. 7. In-detail run showing how all dimensions relate to each other in alternative scenarios. In (a) the EA ran for 2000 generations using the same target rooms as in Figure 6, but using all the dimensions in the search, which results in 78124 cells. In (b) we used the same dimensions as in Figure 6.f (NMP-NSP) but we changed the target room to fig. 3.b. (c) shows a detailed version of Figure 6.f

and the cell 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, and the hue of each of them, 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 expressiveness of the setups. An orange marker locates the target room on the tested dimensions. All the setups were run for 5000 generations, using the same target room.

We can observe that in most of the cases in Figure 6, regardless of what pair of dimensions is used, IC-MAP-Elites can explore a greater area of the search space rather than just around the target room, depicted as an orange marker in each plot, and even the dense areas of the search space (i.e. where the algorithm exploited the most) are distant from the target room's scores. Nevertheless, the expressive range shows that both leniency and linearity scores are explored within the same range (0.4-1.0) when not using any of them as dimensions, which points towards the search having difficulties getting out of other dimensions' local optima, especially since the densest search area is within the target room.

A. Alternative Scenarios

In Figure 7, 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 6 but using all the possible dimensions in the search space, and (b) using NMP and NSP (see Section III-A1) as dimensions in the search but changing the target room. To draw a better comparison, we added (c) which is a more detailed version of example (f) in Figure 6, using NMP and NSP as dimensions.

When using all the dimensions (a), IC-MAP-Elites explore

When using all the dimensions (a), IC-MAP-Elites can explore a substantial area of the search space in each of the dimensions, which is expected since all the dimensions are now acting as archives. It can also be observed that when using NMP and NSP as dimensions (b and c) regardless of the target room, the search space is greatly explored in most of the dimensions. 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 4 in their respective rows.

Moreover, when comparing (a) and (c), it is noticeable that while (a) can explore a greater area than (c), it seems to be recurrently generating the same type of individuals (i.e. depicted with the hue of the hexagon) while in (c), 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 these three plots, while comparable in their diversity search, differ quite drastically 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 8 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 6 (u), (f), and (h), respectively. (d) was run using all the dimensions in the search space with the same data as in Figure 7 (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, 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 behavior 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 behavior 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

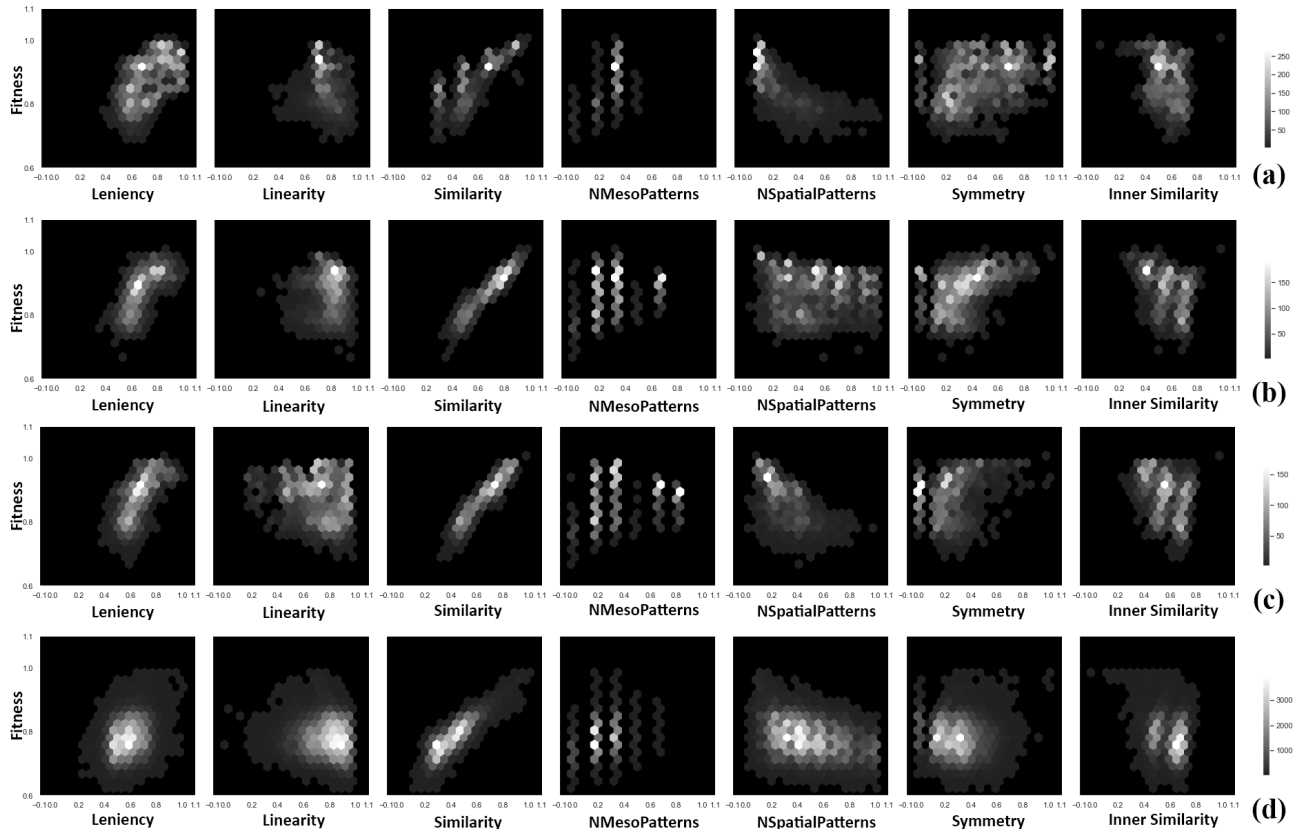

Fig. 8. Relation between dimension score and fitness score. (a), (b), and (c) show results using, respectively, Similarity and Symmetry, NMP and NSP, and NMP and Linearity. (d) was run using all the 7 dimensions. All runs used fig. 3.a as target room

instance, in (a) exploring diverse and higher quality individuals in Similarity, or in (c) in Linearity.

Finally, 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.

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 that there are specific dimensions in level generation such as when using NMP, NSP, LINEARITY, or Symmetry, that foster greater exploration not only in their respective dimensions but also in all the others due to the diverse individuals generated within the respective dimensions as shown in Figure 4. Moreover, Similarity has a high correlation with fitness, and depending on the objective of the designer, this might affect positively or negatively since there will be a strong bias towards highly similar individuals. In contrast, IS seems to be a more robust dimension in the fitness landscape and in the exploration of other dimensions because it captures the properties of the target room rather than its aesthetics.

To further assess the algorithm, we ran an experiment using all possible dimensions (Figure 7) 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 the search space in all dimensions but surprisingly, the search results in the exploitation of sub-optimal individuals in all dimensions as shown in Figure 8 (d). While using pair of dimensions results on a slightly reduced exploration, the exploitation seems to be fairly spread among the space, visible in Figure 7 (b and c), and the density related to fitness is focused on high-performing individuals. 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-quality 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, 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 deserves more attention and it is a promising future step. For instance, Gravina et al [30] explored how the selection of cells guided by four divergent search algorithms benefit MAP-Elites

standard selection method.

Furthermore, preliminary experiments seem to indicate that some dimensions, which are more explorative in concept (e.g. NMP, NSP, or Linearity), 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 7 (b) and (c). This points towards dimensions that would make the algorithm more robust to changes, reinforcing its adaptability features. Further experiments are needed to analyze how different dimensions are better at adapting to continuous changes in the target room, which would also indicate a better stability in the search.

In conclusion, our evaluation is aligned with our expectations when enabling designers to explore different pairs of dimensions. Individuals found in the search space are, in general, more diverse and high-performing, which results in a richer population to be suggested to the designer. Our experiments show that which dimensions are used have a big impact in the search space, fostering the search of high-quality individuals within the selected dimensions while not discouraging exploration in other dimensions. In other words, enabling the designers to proactively decide which dimensions should be used in the search, gives them a high level of controllability with minimal loss in the expressive range.

ACKNOWLEDGMENT

The Evolutionary Dungeon Designer is part of the project *The Evolutionary World Designer*, supported by The Crafoord Foundation.

REFERENCES

[1] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*. Springer, 2018, vol. 2.
[2] M. Toy, "Rogue," 1980.
[3] D. Braben and I. Bell, "Elite," 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," 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.
[10] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for sentient sketchbook," in *Proc. IEEE Conf. Computational Intelligence and Games*, Aug. 2014.
[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.
[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.
[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] 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.
[18] 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.
[19] 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.
[20] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.
[21] 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.
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] 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.
[27] 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.
[28] 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.
[29] 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.
[30] 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.
[31] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.
[32] 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.
[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.
[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.
[35] E. McMillen and F. Himsl, "The Binding of Isaac," 2011.