Enhancements to Constrained Novelty Search

Two-Population Novelty Search for Generating Game Content

Antonios Liapis
Center for Computer Games Research
IT University of Copenhagen
Copenhagen, Denmark
anli@itu.dk

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

Julian Togelius
Center for Computer Games Research
IT University of Copenhagen
Copenhagen, Denmark
juto@itu.dk

ABSTRACT

Novelty search is a recent algorithm geared to explore search spaces without regard to objectives; minimal criteria novelty search is a variant of this algorithm for constrained search spaces. For large search spaces with multiple constraints, however, it is hard to find a set of feasible individuals that is both large and diverse. In this paper, we present two new methods of novelty search for constrained spaces, Feasible-Infeasible Novelty Search and Feasible-Infeasible Dual Novelty Search. Both algorithms keep separate populations of feasible and infeasible individuals, inspired by the FI-2pop genetic algorithm. These algorithms are applied to the problem of creating diverse and feasible game levels, representative of a large class of important problems in procedural content generation for games. Results show that the new algorithms under certain conditions can produce larger and more diverse sets of feasible strategy game maps than existing algorithms. However, the best algorithm is contingent on the particularities of the search space and the genetic operators used. It is also shown that the proposed enhancement of offspring boosting increases performance in all cases.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Constrained novelty search; Procedural content generation; Feasible-Infeasible two-population GA; Level design

1. INTRODUCTION

Evolutionary algorithms have a rich history of successful applications in solving numerical optimization problems. In the realm of constrained optimization, however, evolutionary algorithms have often faced challenges due to the very nature of stochastic, global optimization which is originally designed to react only to implicit constraints posed by the fitness function. As researchers attempted to tackle these challenges, a rich literature has arisen for handling constraints with evolutionary algorithms summarized by [4]. Even so, there has not been a consensus for a single best technique for constrained optimization; each technique has benefits and drawbacks, making the choice of one technique over another (and the details of its implementation) dependent on the problem and, admittedly, personal preference.

Novelty search is a recent departure from traditional evolutionary approaches, as it is driven by exploration of the search space rather than by an objective function [10]. As with objective-driven optimization in the nineties, handling constraints via novelty search is not straightforward; methods used for objective-driven constrained optimization, such as penalizing an individual's fitness score, are not directly transferable to constrained novelty search. This paper proposes two-population alternatives for searching for novelty in a constrained space, where feasible individuals are stored in a separate population and evolve towards optimizing their novelty while infeasible individuals evolve either towards novelty or towards minimizing their distance from feasibility.

This paper compares the proposed two-population novelty search methods with traditional novelty search and early attempts at constrained novelty search; comparisons are made on a case study of evolutionary design of game levels. Novelty search can be particularly beneficial to computer games, as it can generate diverse content which can potentially engage and surprise a player, increasing the game's replayability value. On the other hand, essential game content such as game levels or enemies must satisfy constraints of playability, competence, game balance or believability which can be as rigorous as those of real-world engineering problems.

The structure of this paper is as follows: Section 2 provides a short survey on constrained optimization, novelty search, and game content generation. Section 3 proposes the two-population enhancements to constrained novelty search. Section 4 presents a case study of constrained novelty search for the procedural generation of game levels and Section 5 presents a number of experiments which test different constrained novelty search problems. Section 6 discusses the larger points of constrained novelty search and suggests further improvements; the paper concludes with Section 7.

GECCO'13, July 6–10, 2013, Amsterdam, The Netherlands.

2. RELATED WORK

Novelty search is utilized in this paper for the constrained optimization of game content. A short survey of the relevant domains is presented below.

2.1 Constrained Optimization

While genetic algorithms have demonstrated, early on, their potential in numerical optimization, it has never been straightforward how constraints should be handled. Such constraints are of paramount importance to engineering problems [17], where solutions are often required to satisfy a minimal functional performance or safety and a maximal size or cost. The presence of constraints divides the search space, where optimization takes place, into a *feasible space* and an *infeasible space*. Depending on the problem, the feasible space can be fragmented, non-convex or simply much smaller than the infeasible space (see Fig. 1a). Several surveys have presented different methods for performing genetic optimization in such a divided search space, including [4, 16, 8]; this section covers some popular approaches. A straightforward method for handling constraints is either to assign a low fitness score to infeasible individuals or re-generate them until a feasible one is found. Such approaches amount to a "death penalty" for infeasible individuals, and have been argued against [15] since evolution does not exploit information stored within infeasible individuals. More sophisticated approaches reduce the fitness score of infeasible individuals by a penalty score, which can be a constant number, a measure of feasibility or a value dynamically adapted to the current state of search [4]. Although penalties have been generally successful in constrained optimization problems, their main drawback comes from the many parameters that need to be fine-tuned in order to avoid the detrimental effects of too low or too high penalties. Alternatively, infeasible individuals can be "repaired" to become feasible, often assigning a penalty to their fitness score proportionate to a repair "cost"; however, defining a repair function and a repair cost is not always straightforward and results are very sensitive to these design choices. Another approach is to use completely separate heuristics for the fitness scores of feasible and infeasible individuals; for feasible individuals the fitness score is in accordance with the objective of constrained optimization while for infeasible individuals the fitness score is the distance from feasibility. When selecting parents, the Two Sexes evolutionary strategy [8] requires that a feasible individual mates with an infeasible one, while in the Feasible-Infeasible two-population genetic algorithm (FI-2pop GA) [7] feasible individuals mate only with other feasible individuals, and similarly for infeasible individuals.

The FI-2pop GA maintains two populations: the *feasible* population containing individuals which satisfy all constraints and the *infeasible* population containing individuals which do not satisfy all constraints. Each population has its own objective function and selection scheme, with traditional FI-2pop GA approaches selecting infeasible individuals according to their proximity with the border of feasibility, where optimal solutions often lie [18]. Although each population (feasible or infeasible) selects parents only among its own members, offspring are tested for feasibility and may be transferred to another population. The migration of feasible offspring from infeasible parents to the feasible population (and vice versa) allows for a form of interbreeding which increases diversity in both populations.

2.2 Novelty Search

Novelty search is an objective-free method of genetic search which replaces optimization towards a fitness function approximating the quality of a solution with optimization towards the diversity of solutions. This algorithm can outperform objective-driven search when a fitness function is ill-defined, difficult to quantify, or subjective [10]. Novelty search prompts exploration of the search space by favoring individuals which are different from the current population as well as from previous "novel" content. Each individual i in the population is assigned a $\rho(i)$ value, which determines its preference for selection; $\rho(i)$, which is presented in Equation (1), is the average distance between individual i and its k closest individuals, either in the current population or in an archive of novel individuals. In every generation, the l most novel individuals are stored in this archive.

$$\rho(i) = \frac{1}{k} \sum_{j=1}^{k} dist(i, \mu_j) \tag{1}$$

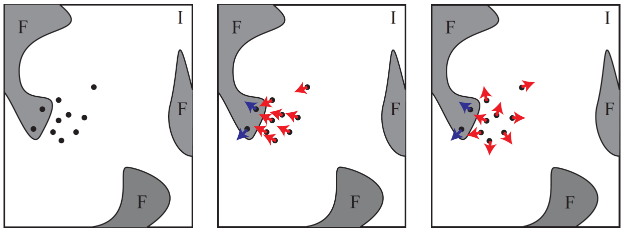
where μ_j is the j-th-nearest neighbor of i (within the population and in the archive of novel individuals); distance $dist(i, j)$ is a domain-dependent heuristic which evaluates the "difference" between individuals i and j.

In order to guide exploration towards valuable solutions, the notion of *minimal criteria novelty search* (MCNS) was introduced in [9]; MCNS assigns a fitness of 0 if an individual i does not satisfy certain criteria, and $\rho(i)$ as per Equation (1) if it does. By assigning the lowest possible fitness to infeasible individuals, MCNS severely limits their chances of reproducing. MCNS demonstrated an improvement over unconstrained novelty search and against objective-driven GAs with deceptive fitnesses in [9]. However, assigning the same low score for all infeasible individuals renders MCNS suboptimal in highly constrained problems, where discovery of feasible individuals is unlikely or where feasible parents are likely to generate infeasible offspring.

2.3 Procedural Generation of Game Content

While the game industry has been using procedural content generation (PCG) since the eighties to increase a game's unexpectedness and replayability, academic interest in PCG for games is relatively new [22]. Search-based approaches [21] — primarily genetic algorithms — have been very popular in generating content such as racing tracks [3], platformer levels [20], mazes [1], board games [2], weapons [6] or spaceships [14]. Many of these projects use objective-driven optimization, targeting navigational properties [1], game balance [2] or aesthetics [14], with interactive evolution being equally popular [6, 3], since the appraisal of game content is often subjective to a user's mercurial sense of taste.

Currently, commercial PCG ensures feasibility via tightly designed algorithms which limit the range of created content; a broader range of content however would increase unexpectedness and replayability, which is the goal of PCG in games. Many research projects opt for a larger expressivity of their algorithms, which necessitates feasibility testing of generated content. *Simulate-and-test* PCG approaches [21] usually discard and re-generate infeasible content [3]. More sophisticated approaches evolve both infeasible and feasible game content [13, 11, 20] via FI-2pop GA. Finally, declarative programming uses constraints to define the search space, ensuring the fast generation of feasible game content [19].



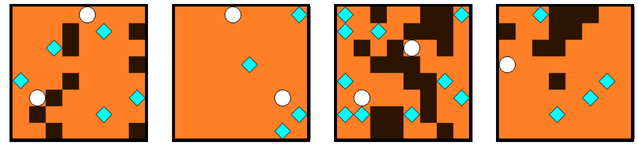
(a) Search space and population. (b) Search process of FINS. (c) Search process of FI2NS.

Figure 1: A visualization of two-population novelty search. Fig. 1a shows a possible search space, with infeasible space (I) and a fragmented feasible space (F); an initial population may contain feasible and infeasible individuals (black dots). With FINS, infeasible individuals move towards the closest border of feasible space as determined by their d_{inf}: in this case, they all move towards the same region of feasible space. With FI2NS, infeasible individuals move away from each other as they try to increase their ρ: while this may lead them away from feasible space, they may eventually discover other islands of feasible space. In both approaches feasible individuals optimize their novelty by moving away from each other, which may lead them to infeasible space.

3. TWO-POPULATION NOVELTY SEARCH

Although many prominent techniques for handling constraints with genetic algorithms penalize the fitness scores of infeasible individuals, applying penalties to novelty search is not straightforward considering the novelty metric of Equation (1). It is unclear, for instance, whether a penalty should be applied to $\rho(i)$ for infeasible i or to $dist(i,j)$ for feasible i but infeasible j. It is therefore preferable to avoid comparisons between infeasible and feasible individuals. The FI-2pop GA [7] presented in Section 2.1 maintains two populations so that infeasible individuals do not compete with feasible ones for the purposes of selection; feasible parents can thus be selected using a completely different criterion (i.e. novelty search) than infeasible ones. Additionally, feasible offspring of infeasible individuals migrate to the feasible population and increase its diversity, which coincides with the goals of novelty search among feasible individuals.

This paper presents two variations of this two-population approach, adapted to the purposes of novelty search. *Feasible-infeasible novelty search* (FINS) evolves feasible individuals, in their separate population, towards maximizing the novelty score $\rho(i)$ as per Equation (1) while infeasible individuals evolve towards minimizing a metric of their distance from feasibility d_{inf} (see Fig. 1b). In order to test the impact of the d_{inf} heuristic and do away with objective-driven optimization on both populations, the feasible-infeasible dual novelty search (FI2NS) performs novelty search on both the feasible and the infeasible population (see Fig. 1c). For FI2NS novelty search is carried out independently in each population, with two separate archives of feasible and infeasible novel individuals; while both populations use the same $\rho(i)$ metric, only the closest neighbors in the same population and archive are considered. Maintaining two populations for either FINS and FI2NS ensures that distances between feasible and infeasible individuals are not considered in the calculation of Equation (1).



(a) (b) (c) (d)

Figure 2: Sample levels with 64 tiles, demonstrating feasible and infeasible level layouts. Displayed tiles are passable (light), impassable (dark), bases (circles) and resources (rhombi). Levels 2a and 2b are feasible, with level 2b (which has no impassable tiles) being used to seed the initial population in certain experiments in Section 5. Level 2c is infeasible as it has no path between bases. Level 2d is infeasible as it has less than two bases; in this case, the repair mechanism is applied to add a base on a random passable tile.

In the FI-2pop GA paradigm, the number of offspring for each population is equal to the current population's size. However, previous experiments have indicated that an *offspring boost* on the feasible population was beneficial for enhancing optimization of feasible individuals. When the feasible population is smaller than the infeasible population, the offspring boost mechanism forces members of the feasible population to create a number of offspring equal to 50% of the total size of the two populations. The number of offspring in the infeasible population is reduced accordingly to keep the total population size steady. To showcase its effectiveness, experiments in Sections 5 include FINS and FI2NS approaches with and without the offspring boost.

4. GAME LEVEL GENERATION

A game level is a prime example where constraints are important for content generation. A Rogue-like dungeon must allow players to reach the exit from the entrance, a platformer level must have platforms at a height accessible to a jumping avatar, and a strategy game must allow players to reach their enemies. While game levels may have objectives such as fairness or challenge, failing such objectives renders the game unentertaining but not unplayable. Only the minimal criteria of playability will be considered in this paper.

The game levels optimized in this paper constitute map sketches, which are low-resolution abstractions of strategy game maps. The concept of map sketches is introduced in [11] where they are used as building blocks of a mixed-initiative tool; stochastic processes such as cellular automata can convert these sketches into large-scale maps appropriate for commercial strategy games such as *Starcraft* (Blizzard, 1998). Each level has a small number of tiles: tiles can be passable, impassable, resources or bases (see Fig. 2). A level is directly encoded in its genotype; each tile on the map is represented in the genotype as an integer indicating tile type. The level layout assumes that each player starts at a base and collects resources in order to build units; units travel through passable tiles in order to attack enemy bases.

A feasible map must have a number of bases and resources within a range specified by the designer, while passable paths must connect all bases and all resources. To alleviate some of these constraints, a repair mechanism transforms maps with excess or missing bases and resources into feasible: while repairing, excess bases and resources are replaced with passable tiles, while missing bases and resources are inserted on random passable tiles. With constraints on the number of bases and resources satisfied via the repair mechanism, the distance from feasibility for FINS approaches is:

$$d_{inf} = \frac{1}{2}\frac{2u_b}{b(b-1)} + \frac{1}{2}\frac{u_r}{rb} \tag{2}$$

where b and r are the number of bases and resources respectively, while u_b and u_r are the number of unconnected base pairs and base-resource pairs respectively.

The distance metric $dist(i, j)$ in Eq. (1) is the ratio of mismatched tiles at the same coordinates between maps i and j to the total number of map tiles.

5. EXPERIMENTS

The performance of the proposed methods of constrained novelty search will be assessed on a set of experiments, using the generation of strategy game levels as the case study. This paper compares different novelty search methods; a comparison of FINS and MCNS with objective-driven GAs (including FI-2pop GA) is included in [12]. The experiments' parameters, results and discussion will be covered below.

5.1 Parameters and Experimental Setup

The experiments presented below compare the proposed two-population novelty search methods of FINS and FI2NS with single-population novelty search methods, namely traditional novelty search (NS) and minimal criteria novelty search (MCNS). The benefits of boosting the offspring of the feasible population are shown by comparing FINS and FI2NS, which does not use the offspring boost, with FINS$_{ob}$ and FI2NS$_{ob}$ which do. All approaches in this paper calculate $\rho(i)$ from the average distance of the 20 closest individuals ($k = 20$), while the 5 highest scoring individuals in the population are inserted to the archive of novel individuals per generation ($l = 5$). These k and l parameters theoretically benefit the smaller feasible populations of FINS and FI2NS more than MCNS and NS. Different k and l configurations do not seem to affect the general performance of any approach according to preliminary experiments, although for very small k values all approaches perform poorly.

The introduction of two-population novelty search methods aims to address two important shortcomings of MCNS in constrained novelty search: a) the fact that MCNS performs random search when no feasible individuals exist in the population, which hinders the discovery of a feasible individual in tightly constrained problems, and b) the fact that MCNS kills individuals rendered infeasible during novelty search, reducing the population's diversity and the algorithm's efficiency at optimizing it. The performance metrics used in this paper to evaluate the algorithms' optimization behavior are therefore: a) the number of feasible individuals in the population, focusing on the discovery of the first feasible individual and b) the diversity of feasible individuals, which is quantified as the average $dist(i, j)$ of all pairs of feasible individuals i and j in the population.

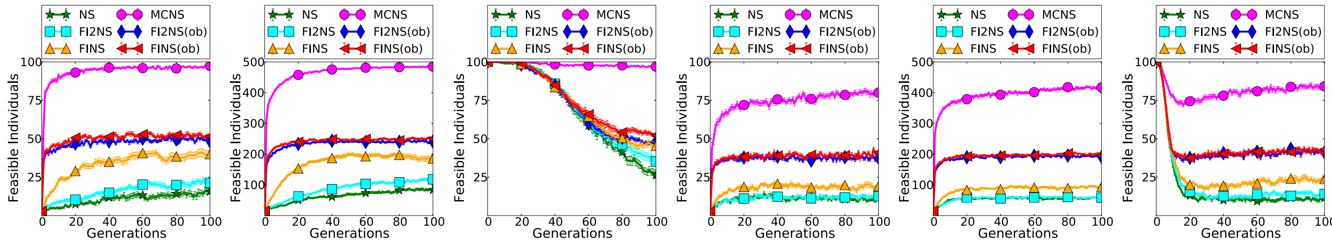
A number of different experiments test the optimization behavior of the different novelty search approaches:

- R, where offspring are primarily generated via the two-point crossover of two parents. There is a small chance of mutation (1%) for every offspring of two parents.
- M, where offspring are generated exclusively via mutation. Mutation is guaranteed to retain the number of bases and resources from parent to offspring and the repair mechanism is only used to create an initial population of maps with appropriate bases and resources.
- R_{500} and M_{500}, where the total population size is 500 (unlike the other experiments where the total population size is 100). The impact of a larger population is expected to influence the likelihood of discovering feasible individuals and may benefit NS and FI2NS approaches, which have a small number of feasible individuals compared to the number of infeasible ones. Experiments in R_{500} use recombination as per R and M_{500} exclusively use mutation as per M.
- R_s and M_s, where the initial population consists entirely of feasible individuals, which have the designer-specified number of bases and resources but no impassable tiles, thus guaranteeing connectivity (see Fig. 2b). As MCNS underperforms on highly constrained spaces, it is suggested in [9] that MCNS is initially seeded with feasible individuals; this experiment will test this hypothesis and evaluate the impact of a feasible initial population. Experiments in R_s use recombination as per R and M_s exclusively use mutation as per M.

All experiments use fitness-proportionate roulette wheel selection of parents. The same parent may be selected more than once and thus generate multiple offspring, which is necessary when the number of offspring is different than the number of parents (i.e. when applying the offspring boost). In each population the fittest individual is transferred to the next generation. Elitism does not seem valuable to novelty search, since the fittest individual in one generation is likely to have a low fitness in the next generation due to its presence in the archive of novel individuals. However, the "minimal" elitism used ensures the presence of at least one feasible individual in each generation for two-population novelty search approaches. Mutation in all experiments transforms a number of map tiles (between 5% and 20% of all tiles), either swapping them with adjacent tiles or converting an impassable tile into passable and vice versa; either tile transformation has an equal chance of occurring. This mutation scheme results in small changes in the map and is not likely to cause feasible individuals to become infeasible. Excluding R_{500} and M_{500}, the total population size is 100 which includes feasible and infeasible individuals. A population size of 100 individuals, even if separated into feasible and infeasible populations, is in tune with the k, l parameters used for novelty search, while the larger population used in R_{500} and M_{500} will also test the impact of k, l parameters.

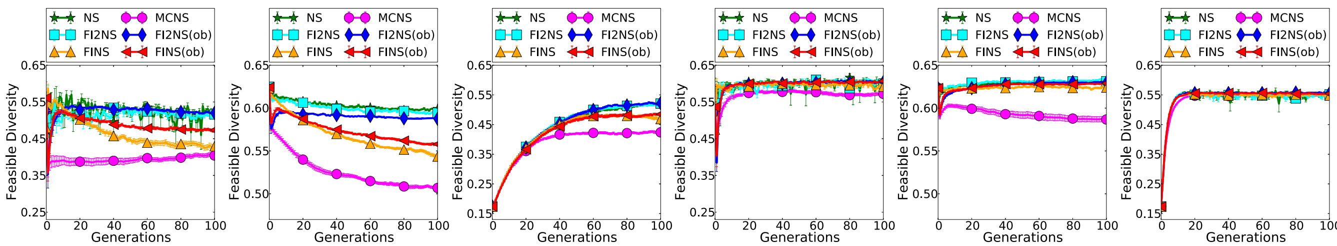
5.2 Benchmarks and Results

The performance of different novelty search approaches for generating feasible game levels is evaluated in two map setups: a small map of 64 tiles, with 2 bases and 4 to 10 resources, and a large map of 256 tiles, with 2 to 10 bases and 4 to 30 resources. These map setups vary in their difficulty at discovering feasible individuals; testing 10^6 randomly initialized maps, $3 \cdot 10^4$ were feasible for small maps while only 13 were feasible for large maps. Small maps have few tiles which must be connected (bases, resources), and combined with their small size are more likely to be feasible. Large maps can potentially have a large number of tiles which must be connected, and impassable tiles are more likely to block paths due to the maps' large size.



(a) R (b) R_{500} (c) R_s (d) M (e) M_{500} (f) M_s

Figure 3: Size of the feasible population when optimizing small maps, for experiments indicated in each figure's caption. Values are averaged across 20 independent runs, with error bars depicting standard error among runs.



(a) R (b) R_{500} (c) R_s (d) M (e) M_{500} (f) M_s

Figure 4: Average diversity of the feasible population when optimizing small maps, for experiments indicated in each figure's caption. Values are averaged across 20 independent runs, with error bars depicting standard error among runs.

The progress of the number and diversity of feasible individuals for small maps is shown in Figures 3 and 4 respectively, while Table 1 lists the final number and average diversity of feasible individuals after 100 generations. Large maps are unlikely to be feasible if initiated randomly; since the first feasible individual is discovered on different generations between runs, the progress of the number and diversity of feasible individuals can not be displayed as a figure; Table 2 lists for each approach the number of runs with feasible individuals and the generation where the first feasible individual occurred, along with the number and diversity of feasible individuals after 100 generations. Significance throughout this paper is $\alpha = 5\%$, measured via standard t-tests; the Bonferroni correction [5] is applied when comparing among the six constrained novelty search methods.

The different experimental setups have a clear impact on the number and diversity of feasible individuals. Experiments using recombination generally have a lower diversity than those using mutation, due to the tile-based similarity used as a distance metric and the direct mapping between genotype and phenotype. A map created via crossover inherits parts of the map from either parent, resulting in low distance between the offspring and either parent as well as between all offspring of the same parent. On the other hand, a map generated via mutation inherits parts of the map from a single parent, while the random mutation means that offspring of the same parent may still be dissimilar from each other. Observing the diversity in R_s and M_s experiments, the lack of impassable tiles in the initial feasible population results in low diversity values. Since impassable tiles can only be added via mutation, M_s increases the diversity of feasible maps much faster than R_s — although the diversity for M_s experiments is generally lower than for M. On the other hand, impassable tiles added via mutation are more likely to create disconnected paths between bases or resources, leading to fewer feasible individuals for M, M_{500} and M_s than for R, R_{500} and R_s respectively. Finally, a larger population (comparing R_{500} with R and M_{500} with M) is clearly beneficial for all approaches as it results in higher diversity values in both small and large maps; moreover, feasible individuals are discovered earlier and more often with the larger population when evolving large maps.

5.3 Performance of each approach

Observing the results of the different experiments on small and large maps, some conclusions can be drawn regarding the overall performance of each approach.

Unconstrained novelty search does not distinguish between feasible and infeasible individuals; it is as likely to select a feasible as an infeasible parent for generating offspring, resulting in few feasible individuals in all experiments. In large maps where the feasible space is very limited, NS primarily explores infeasible space and usually discovers feasible individuals late in the evolutionary process. Even when a feasible individual is found (or in R_s and M_s), no mechanism ensures that it will persist in the next generation; especially in experiments with recombination, most feasible individuals are rendered infeasible in the next generation after they are discovered. NS often has no feasible individuals in large maps at the end of evolution (excluding R_s), which explains their low diversity. While in small maps feasible individuals are still few, their number is sufficient to actually maintain a high diversity; NS, along with FI2NS, consistently has the highest diversity across experiments for small maps.

MCNS has the largest number of feasible individuals across all experiments, since infeasible individuals are as-

Metric	Approach	R	R_{500}	R_s	M	M_{500}	M_s
Final number of feasible individuals (p)	NS	16.0 (1.8)	**84.7 (4.4)**	25.7 (2.2)	11.8 (1.0)	54.9 (2.0)	11.2 (1.3)
	MCNS	**97.4 (0.3)**	**485.1 (0.9)**	**97.1 (0.4)**	**79.9 (0.9)**	**417 (3.3)**	**84.1 (1.1)**
	FI2NS	20.9 (1.7)	**117.2 (3.7)**	35.0 (2.5)	12.6 (0.9)	58.9 (1.5)	14.3 (1.2)
	FI2NS$_{ob}$	48.7 (0.9)	241 (2.2)	48.3 (0.6)	37.6 (1.0)	190.8 (2.2)	41.2 (1.2)
	FINS	40.0 (2.6)	**188 (8.0)**	45.5 (1.7)	**19.0 (1.3)**	**95.1 (3.2)**	**23.8 (1.2)**
	FINS$_{ob}$	51.5 (1.0)	250.1 (2.7)	52.3 (1.4)	41.3 (1.2)	196.8 (1.9)	42.8 (1.5)
Final average feasible diversity ($\bar{d}$)	NS	0.52 (0.01)	**0.60 (0.00)**	0.52 (0.00)	0.60 (0.00)	0.63 (0.00)	0.55 (0.00)
	MCNS	0.40 (0.01)	**0.51 (0.00)**	**0.42 (0.00)**	**0.57 (0.00)**	**0.59 (0.00)**	0.55 (0.00)
	FI2NS	0.52 (0.00)	**0.59 (0.00)**	0.52 (0.00)	0.60 (0.00)	0.63 (0.00)	0.55 (0.00)
	FI2NS$_{ob}$	0.52 (0.00)	**0.59 (0.00)**	0.52 (0.00)	0.60 (0.00)	0.63 (0.00)	0.55 (0.00)
	FINS	0.43 (0.01)	**0.54 (0.00)**	0.47 (0.01)	0.59 (0.00)	**0.62 (0.00)**	0.55 (0.00)
	FINS$_{ob}$	**0.47 (0.00)**	**0.56 (0.00)**	0.49 (0.00)	0.61 (0.00)	0.63 (0.00)	0.55 (0.00)

Table 1: The final number of feasible individuals (p) and their average diversity ($\bar{d}$) for the different novelty search approaches used on small maps; values are averaged across 20 independent runs, with the standard error included in parentheses. Values in bold are significantly different from *all* other methods in the same experiment.

signed a low fitness and are swiftly killed and replaced by feasible individuals — provided feasible individuals are present. When feasible individuals are not present, as is the case for most experiments with large maps (excluding R_s and M_s), MCNS performs random search since all infeasible individuals have the same fitness. When using recombination, the detrimental effects of random search are shown in the small number of runs where a feasible individual was discovered; when using mutation, however, MCNS does not perform much different than other approaches such as NS and FI2NS. Regarding the diversity of feasible solutions, MCNS underperforms compared to all other approaches in small maps (excluding M_s), although it sometimes reaches a higher diversity than other approaches for large maps.

Like unconstrained novelty search, FI2NS suffers from a small number of feasible individuals; in most experiments FI2NS has a marginally larger feasible population than NS. In large maps, FI2NS suffers from the extensive exploration of infeasible space; the "minimal" elitism used, however, ensures that once a feasible individual is discovered, at least one copy of it will persist in the feasible population. This makes FI2NS somewhat more robust than NS for large maps, although the diversity of its feasible individuals is usually very low. The offspring boost mechanism helps to maintain a sizable population, which is vital for large maps; FI2NS$_{ob}$ manages to achieve high diversity values for large maps, comparable to those of other approaches such as FINS and FINS$_{ob}$. In small maps, the larger feasible population of FI2NS$_{ob}$ does not seem to impact diversity, as the average diversity of feasible individuals is on par with those of FI2NS and NS despite the fewer feasible individuals of the latter.

Searching the infeasible space according to a measure of distance from feasibility gives FINS an advantage compared to other approaches in cases where discovery of feasible individuals is unlikely; FINS and FINS$_{ob}$ discover feasible large maps in earlier stages of evolution and more consistently (based on the standard error in the first generation of feasibility). For large maps, this allows more extensive novelty search on the feasible population for more generations, which results in high diversity values. In less constrained search spaces, such as those of small maps, FINS discovers more feasible individuals than FI2NS and NS since infeasible individuals closer to the border with feasibility are more likely to become feasible. Similarly, individuals evolving in the infeasible population before becoming feasible are unlikely to be very similar with those already in the feasible population; in small maps, this results in a larger diversity of feasible individuals compared to MCNS, although for experiments using recombination this diversity is significantly lower than that of FI2NS and NS. The offspring boost is beneficial in increasing the number of feasible individuals and their diversity; this is pronounced in small maps for R and R_{500} experiments and in large maps for R and M experiments. In experiments with large maps which exclusively use mutation, FINS$_{ob}$ results in much larger feasible populations than FINS (which has surprisingly few feasible individuals at the end of evolution) without a significant effect on the diversity of the feasible population.

6. DISCUSSION

Presented experiments evaluated the performance of different methods of novelty search in constrained problems. Results indicate that FI2NS (especially coupled with the offspring boost mechanism) can create a diverse set of feasible individuals in domains with easily satisfiable constraints. On the other hand, traditional novelty search can create equally diverse sets as FI2NS, but suffers from a small number of feasible individuals which is problematic in highly constrained spaces. MCNS achieves by far the highest number of feasible individuals which however can not be as diverse as other methods; even its most diverse individuals have a lower diversity than those in other approaches. Additionally, MCNS underperforms when no feasible individuals exist in the population. FINS is ideal for highly constrained spaces, as it discovers and diversifies feasible individuals more efficiently than other methods. As with objective-driven constrained optimization, each of the presented methods has its own benefits and drawbacks and choosing one over the other will depend on the task's intended outcomes, the parameters and genetic operators used and the topology of the feasible space.

It should be noted that while novelty search does away with objective functions, the quality of solutions is still affected by the distance heuristic used to evaluate the difference between two phenotypes. In this case study, game levels are compared on a tile-by-tile basis; however, if a map is the reflection of another on a Cartesian axis, then the two maps are identical for all gameplay purposes but are likely to have a large tile-based distance. The tile-based distance

Metric	Approach	R	R_{500}	R_s	M	M_{500}	M_s
Runs with feasible individuals (n)	NS	14	20	20	20	20	20
	MCNS	3	13	20	20	20	20
	FI2NS	14	20	20	20	20	20
	FI2NS$_{ob}$	11	20	20	20	20	20
	FINS	20	20	20	20	20	20
	FINS$_{ob}$	20	20	20	20	20	20
First generation of feasibility (g)	NS	63.3 (6.6)	32.4 (3.5)	0.0 (0.0)	20.5 (2.6)	9.9 (1.0)	0.0 (0.0)
	MCNS	56.0 (2.3)	47.8 (5.8)	0.0 (0.0)	19.5 (3.2)	9.0 (0.8)	0.0 (0.0)
	FI2NS	54.1 (4.9)	40.0 (4.0)	0.0 (0.0)	18.7 (2.4)	7.9 (1.0)	0.0 (0.0)
	FI2NS$_{ob}$	57.4 (6.2)	26.5 (3.6)	0.0 (0.0)	16.6 (2.1)	7.1 (0.6)	0.0 (0.0)
	FINS	8.2 (0.8)	4.0 (0.4)	0.0 (0.0)	8.0 (0.8)	4.8 (0.5)	0.0 (0.0)
	FINS$_{ob}$	7.8 (1.0)	4.0 (0.4)	0.0 (0.0)	6.6 (0.7)	4.7 (0.4)	0.0 (0.0)
Final number of feasible individuals (p)	NS	**0.3 (0.2)**	**0.6 (0.2)**	21.3 (2.3)	**0.2 (0.1)**	**0.6 (0.2)**	**0.8 (0.2)**
	MCNS	**97.7 (0.7)**	**473.6 (5.1)**	**96.3 (0.5)**	**33.7 (1.5)**	**185.5 (6.0)**	**69.4 (1.8)**
	FI2NS	**1.1 (0.1)**	**1.5 (0.2)**	31.8 (2.7)	**1.2 (0.1)**	**1.4 (0.1)**	**2.6 (0.5)**
	FI2NS$_{ob}$	**44.2 (0.8)**	**219.5 (1.9)**	46.7 (1.0)	15.4 (0.9)	**80.4 (2.0)**	36.3 (1.3)
	FINS	**29.1 (2.3)**	**130.2 (4.2)**	40.3 (2.0)	**3.1 (0.3)**	**9.1 (0.8)**	**6.9 (0.9)**
	FINS$_{ob}$	**48.9 (0.9)**	**244.9 (1.6)**	50.3 (0.8)	17.2 (0.8)	**92.4 (2.7)**	34.1 (1.7)
Final average feasible diversity ($\bar{d}$)	NS	0.04 (0.04)	0.11 (0.05)	0.42 (0.00)	0.00 (0.00)	0.06 (0.04)	**0.05 (0.03)**
	MCNS	0.15 (0.04)	0.28 (0.01)	**0.34 (0.00)**	0.54 (0.00)	**0.59 (0.00)**	0.50 (0.00)
	FI2NS	0.00 (0.00)	0.11 (0.05)	0.42 (0.00)	0.04 (0.03)	0.20 (0.06)	**0.25 (0.05)**
	FI2NS$_{ob}$	0.28 (0.05)	0.49 (0.01)	0.43 (0.00)	0.51 (0.01)	0.60 (0.00)	0.50 (0.00)
	FINS	0.26 (0.02)	**0.46 (0.00)**	0.38 (0.00)	0.47 (0.05)	0.60 (0.00)	0.47 (0.03)
	FINS$_{ob}$	0.36 (0.00)	0.47 (0.00)	0.38 (0.00)	0.54 (0.01)	0.60 (0.00)	0.50 (0.00)

Table 2: Performance metrics for different novelty search approaches used on large maps: number of runs (out of 20) where a feasible individual was discovered (n) and the first generation in which it occurred (g), as well as the final number of feasible individuals (p) and their average diversity ($\bar{d}$) of each experiment. Values (except n) are averaged across those runs where a feasible individual was found, with the standard error included in parentheses; for g, p, and $\bar{d}$, values in bold are significantly different from *all* other methods in the same experiment.

heuristic used also saturates the results of the feasible population's diversity — maps generated via mutation alone are more diverse in terms of the distance heuristic used, but arguably less useful as they are more prone to being infeasible. Novelty search is intended, in this line of research, to create interesting and engaging content for computer games; it would therefore be worthwhile to compare the methods in this paper on the usefulness or "interestingness" of the generated content rather than on their visual similarity. However, designing a more domain-specific distance heuristic (for instance, including independence to reflections or integrating features such as distance between bases) is not straightforward and may introduce additional bias to the search.

This paper focuses on constraint handling via novelty search and does not elaborate on other techniques that limit constraints or speed up constrained optimization. Such techniques may involve repairing infeasible individuals directly: an example of this technique is used to repair evolved game levels with excess or missing bases or resources. This repair mechanism automatically satisfies designer constraints on the number of bases or resources and reduces the size of the optimization problem; however, the stochasticity of the implemented repair mechanism reduces the locality of the genetic search. Additionally, experiments using recombination and mutation demonstrate that expert knowledge in the design of genetic operators appropriate to the problem can lead to a better optimization behavior. Similarly, the erratic discovery of feasible individuals can be alleviated by seeding the initial population with known feasible solutions. While the experiments presented cover the majority of these techniques, a larger number of experiments across representations and use cases should further evaluate the impact of smaller or larger population sizes, tailored genetic operators and novelty search parameters, different repair mechanisms and alternative methods of boosting feasible offspring.

While this paper tests constrained novelty search in the domain of game content generation, the presented methods can be used for any evolutionary optimization problem with clear and unavoidable constraints. The methods presented in this paper are tested on their diversity, as experiments exclusively compare novelty search methods; it is therefore assumed that the hypothesis that divergent search is preferable to objective search holds. As novelty search and MCNS have often proven more successful at reaching objectives than objective search, it is worthwhile to compare two-population novelty search methods against objective-driven genetic algorithms. Experiments reported in [12] compare FINS and MCNS with FI-2pop GA and a traditional GA in terms of diversity, although it may be worthwhile to test the performance of FINS and FI2NS in terns of a fitness function in constrained test problems such as those used in [8]. Additionally, the constrained nature of most industrial engineering problems such as those presented in [17] makes them ideal candidates for constrained novelty search.

7. CONCLUSION

This paper introduced two alternatives to current novelty search methods which are able to handle constraints. These approaches evolve two populations simultaneously, a feasible population to maximize novelty and an infeasible population either to minimize the distance from feasibility (FINS) or to maximize novelty (FI2NS). Comparing the behavior of these approaches in terms of discovering and diversifying feasible game content, results indicate that FINS is able to discover feasible solutions faster in cases where such solutions are rare. FI2NS is likely to evolve more diverse content, but underperforms in highly constrained search spaces as it struggles to discover and maintain a sizable feasible population unless the offspring boost mechanism is used. Traditional novelty search creates diverse sets of feasible individuals, but explores primarily infeasible space and finds even fewer feasible solutions than FI2NS. On the other hand, Minimal Criteria Novelty Search maintains a much larger number of feasible individuals than other approaches, which are however less diverse in most cases. Finally, boosting the number of offspring in the feasible population to match the size of the infeasible population is shown to be beneficial in increasing both the number and the diversity of feasible individuals in FINS and FI2NS. Both FINS and FI2NS seem particularly useful for the procedural generation of game content which requires its generated artifacts to be diverse yet functional, since the personal tastes of different players are difficult to capture via objectives. Depending on the computational requirements of the content generators, such as *offline* while the game loads or *online* while the game is played, the higher diversity of FI2NS or the speedy discovery of feasible content of FINS may be preferable.

8. ACKNOWLEDGMENTS

The research is supported, in part, by the FP7 ICT project SIREN (project no: 258453) and by the FP7 ICT project C2Learn (project no: 318480).

9. REFERENCES

[1] D. Ashlock, C. Lee, and C. McGuinness. Search-based procedural generation of maze-like levels. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):260–273, 2011.

[2] C. Browne and F. Maire. Evolutionary game design. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(1):1–16, 2010.

[3] L. Cardamone, D. Loiacono, and P. L. Lanzi. Interactive evolution for the procedural generation of tracks in a high-end racing game. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 395–402, 2011.

[4] C. A. Coello Coello. Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 2603–2624. ACM, 2010.

[5] O. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56:52–64, 2012.

[6] E. J. Hastings, R. K. Guha, and K. O. Stanley. Evolving content in the galactic arms race video game. In *Proceedings of IEEE Conference on Computational Intelligence and Games*, pages 241–248, 2009.

[7] 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*, 190(2):310–327, 2008.

[8] O. Kramer and H.-P. Schwefel. On three new approaches to handle constraints within evolution strategies. *Natural Computing: an international journal*, 5(4):363–385, 2006.

[9] J. Lehman and K. O. Stanley. Revising the evolutionary computation abstraction: Minimal criteria novelty search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2010.

[10] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[11] A. Liapis, G. Yannakakis, and J. Togelius. Generating map sketches for strategy games. In *Proceedings of Applications of Evolutionary Computation*, 2013.

[12] A. Liapis, G. Yannakakis, and J. Togelius. Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of ACM Conference on Foundations of Digital Games*, 2013. In Print.

[13] A. Liapis, G. N. Yannakakis, and J. Togelius. Neuroevolutionary constrained optimization for content creation. In *Proceedings of IEEE Conference on Computational Intelligence and Games*, pages 71–78, 2011.

[14] 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*, 4(3):213–228, 2012.

[15] Z. Michalewicz. Do not kill unfeasible individuals. In *Proceedings of the Fourth Intelligent Information Systems Workshop*, pages 110–123, 1995.

[16] Z. Michalewicz. A survey of constraint handling techniques in evolutionary computation methods. In *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pages 135–155, 1995.

[17] Z. Michalewicz, D. Dasgupta, R. Le Riche, and M. Schoenauer. Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering Journal*, 30:851–870, 1996.

[18] M. Schoenauer and Z. Michalewicz. Evolutionary computation at the edge of feasibility. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 245–254. Springer-Verlag, 1996.

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

[20] N. Sorenson, P. Pasquier, and S. DiPaola. A generic approach to challenge modeling for the procedural creation of video game levels. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):229–244, 2011.

[21] J. Togelius, G. Yannakakis, K. Stanley, and C. Browne. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, (99), 2011.

[22] G. N. Yannakakis. Game AI revisited. In *Proceedings of ACM Computing Frontiers Conference*, 2012.