

Devising Effective Novelty Search Algorithms: A Comprehensive Empirical Study

Jorge Gomes
BioMachines Lab &
Instituto de Telecomunicações
& Faculdade de Ciências,
Universidade de Lisboa, BiolSI
Lisbon, Portugal
jgomes@di.fc.ul.pt

Pedro Mariano
Faculdade de Ciências,
Universidade de Lisboa, BiolSI
Lisbon, Portugal
plmariano@fc.ul.pt

Anders Lyhne Christensen
BioMachines Lab &
Instituto de Telecomunicações
& Instituto Universitário de
Lisboa (ISCTE-IUL)
Lisbon, Portugal
anders.christensen@iscte.pt

ABSTRACT

Novelty search is a state-of-the-art evolutionary approach that promotes behavioural novelty instead of pursuing a static objective. Along with a large number of successful applications, many different variants of novelty search have been proposed. It is still unclear, however, how some key parameters and algorithmic components influence the evolutionary dynamics and performance of novelty search. In this paper, we conduct a comprehensive empirical study focused on novelty search's algorithmic components. We study the k parameter — the number of nearest neighbours used in the computation of novelty scores; the use and function of an archive; how to combine novelty search with fitness-based evolution; and how to configure the mutation rate of the underlying evolutionary algorithm. Our study is conducted in a simulated maze navigation task. Our results show that the configuration of novelty search can have a significant impact on performance and behaviour space exploration. We conclude with a number of guidelines for the implementation and configuration of novelty search, which should help future practitioners to apply novelty search more effectively.

Categories and Subject Descriptors

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

Keywords

Novelty search, evolutionary robotics, neuroevolution, premature convergence, empirical study

1. INTRODUCTION

Novelty search [15, 18] is an evolutionary technique that guides evolution towards behavioural novelty, in contrast with traditional evolutionary approaches where a static objective is pursued. In novelty search, individuals are re-

warded for being behaviourally different from the other individuals in the population, and optionally, from past individuals stored in an archive. The behavioural difference is given by a behaviour similarity measure provided by the experimenter for the specific task. The main rationale behind novelty search is to avoid deception and premature convergence [36]. By having a dynamic objective based on behavioural novelty, the evolutionary process avoids convergence to a single region of the search space.

Since its introduction in 2008 by Lehman and Stanley [15], novelty search has been applied to a wide range of problems in different domains with considerable success. Novelty search has, however, been predominantly studied in the evolutionary robotics domain, including the evolution of: (i) single-robot controllers [18, 26], (ii) controllers for homogeneous [11] and heterogeneous [7] multirobot systems, (iii) robot morphologies [19], and (iv) plastic neural networks [30]. A few applications outside robotics can also be found in, for instance, machine learning [27, 28] and game content generation [23]. Previous works have shown that novelty search can find good solutions faster and more consistently than fitness-based evolution in many different applications, especially in deceptive domains. It has also been shown that novelty search can discover a diverse set of solutions, as opposed to fitness-based evolution that typically converges to a single region in the solution space [11, 19].

Alongside the large number of applications of novelty search, a wide range of algorithmic variants have been proposed. Most previous works that study novelty search's evolutionary dynamics focus on the most crucial aspect of novelty search: the definition of a behavioural similarity measure [13, 26]. The remaining algorithmic components and parameters have not received much attention, and it is still unclear how some of them affect the novelty search process. For instance, there is no consensus on whether an archive of individuals from previous generations should be used in the computation of novelty scores, or how this archive should be composed. It is also unclear how many individuals should be used to obtain the novelty score of a new individual (the number k of nearest neighbours). Additionally, a number of techniques have been proposed to combine novelty search with fitness-driven evolution, but many of these techniques have not yet been compared.

Since novelty search only influences the scoring of individuals, another open question is how the underlying evolutionary algorithm and its genetic operators should be configured

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '15, July 11 - 15, 2015, Madrid, Spain

© 2015 ACM. ISBN 978-1-4503-3472-3/15/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2739480.2754736>

to facilitate an effective exploration of the behaviour space. The relation between the exploration of the genotype space and the exploration of the behaviour space should be further investigated: how the amount of genetic diversity generated by the evolutionary algorithm affects novelty search’s performance.

In this paper, we analyse the impact of the aforementioned algorithmic components. We conduct a comprehensive empirical study that encompasses the typical novelty search parameters, and most of the variants found in previous works. Our experiments are based on the maze navigation task, commonly used in novelty search works [18, 26]. The performance of the different novelty search variants is compared along two dimensions: we focus not only on the capacity of reaching a solution to the task, but also on the capacity to explore the behaviour space thoroughly and uniformly.

Our work sheds light on how the key algorithmic components influence novelty search’s performance. We do, however, acknowledge that some of these choices might be tied to the domain or task where novelty search is applied. We therefore do not seek to find universally optimal values for each parameter, but rather to gain insight into how changing certain parameters affects the evolutionary dynamics of novelty search. Based on our findings, we conclude with a list of guidelines that will aid experimenters in applying novelty search effectively.

2. RELATED WORK

2.1 Novelty Search

Implementing novelty search requires little change to any evolutionary algorithm aside from replacing the fitness function with a domain-dependent novelty metric. The metric measures how far an individual is from other individuals in behaviour space. The novelty of an individual is given by the mean behaviour distance to other individuals, including the individuals from the current population, and optionally, an archive of past individuals. The novelty metric thereby rewards individuals that appear in previously unexplored regions of the behaviour space, continuously guiding the evolutionary process towards behavioural novelty and diversity.

In order to apply novelty search, the experimenter has to provide an adequate behavioural similarity measure. To this end, the behaviour of each individual is typically characterised by a real-valued vector, and the behaviour distance between two individuals is then given by the distance between the corresponding vectors. The design of a behaviour characterisation has direct implications on the effectiveness of novelty search [13]. This topic has received considerable attention in previous works [4, 9, 13, 18, 26], and is beyond the scope of this paper. Originally, it was argued that the characterisation should be designed with task-specific knowledge [18]. Later works, however, have shown that it is possible to devise task-independent characterisations. Some of the possibilities include automatically extracting characterisations from the sensor-effector states of the agents [3, 6, 26], systematically crafting a characterisation based on a formal description of the task [7], and combining multiple different characterisations [4].

2.2 Novelty Search Variations

In this section, we review the different variants of novelty search that have been used in previous works. We consider

Table 1: Summary of the novelty search variants found in the current literature.

| Studies | Archive | | | k | | | Comb. | | | | Underlying EA | | | |
|-------------------------|---------|---|---|-----|---|------|-------|-------|---|---|---------------|----|-----|-----|
| | N | R | ∅ | 15 | ∅ | ... | ∅ | M | S | C | NEAT | FT | DNN | ... |
| [13–15, 18, 20, 30, 35] | • | | | • | | | • | | | | • | | | |
| [27, 28] | • | | | • | | | • | | | | | | | GP |
| [21] | • | | | • | | | • | • | | | • | | | |
| [1] | • | | | • | | | • | • | | | | • | | |
| [10] | • | | | • | | | • | • | • | | • | | | |
| [17] | • | | | • | | | • | | • | | • | | | |
| [12] | • | | | • | | | • | other | | | • | | | |
| [23] | • | | | | | 20 | • | | • | | | | | GA |
| [16] | • | | | • | | | • | | | | | | | GP |
| [19] | • | | | • | | | • | • | | | | | | ERO |
| [9] | • | | | • | | | • | | | | | • | | |
| [7] | • | | | • | | | | • | | | | • | | |
| [6] | • | | | • | | | | • | | | • | | | |
| [11] | • | | | • | | | • | • | • | | • | | | |
| [8] | • | | | • | | | • | | • | | | • | | |
| [29] | • | • | | • | • | | • | • | | | | • | | • |
| [24] | • | • | | • | • | | • | • | | | | | | • |
| [26] | | • | | • | | | • | | | | | • | | • |
| [22] | | • | | • | | | • | | | | • | | | |
| [3, 4, 25] | | • | | • | | | • | | | | | | • | |
| [33, 34] | | • | | | | 3–10 | • | | | | | | | GE |

From the left to the right: *Archive*: N – Novelty-based archive; R – Randomly composed archive; ∅: No archive. *k number of nearest neighbours*: $k=15$; ∅ – the entire population is used. *Technique to combine novelty and fitness*: ∅ – pure novelty search; M – Pareto-based multi-objectivisation; S – Linear scalarisation of novelty and fitness scores; C – Combination based on minimal criteria. *Underlying evolutionary algorithm*: NEAT [31]; FT – Fixed-topology neuroevolution; DNN – Direct Encoding of Neural Networks [26]. More than one dot in each column means that multiple variants were used/compared.

three key aspects in the algorithm’s configuration: (i) the novelty score computation; (ii) the archive storing individuals from previous generations; and (iii) the method used to combine novelty and fitness scores. The variants studied in previous works are summarised in Table 1.

2.2.1 Novelty score computation

In the original novelty search algorithm, the novelty score of each individual is computed as the mean distance to the k nearest neighbours in behaviour space:

$$\rho(x) = \frac{1}{k} \sum_{i=1}^k \text{dist}(x, \mu_i) , \quad (1)$$

where μ_i is the i th-nearest neighbour of x with respect to the behaviour distance metric dist .

The parameter k must be provided by the experimenter, but its effect on performance has not been studied in detail in previous works. In [15], a value of $k=15$ was proposed, and it was stated that the value is robust to moderate variation although no concrete results were provided. The vast majority of subsequent works used this value without further study (see Table 1). Slightly different values have been used in a few works [23, 33, 34], with the authors stating that the parameter was tuned for performance. A different approach

was taken in [3,4,25,26], which was named *behavioural diversity* instead of novelty search. In this approach, the novelty score of an individual is given by the mean behaviour distance to *all* other individuals in the current population, but without considering individuals from past generations.

In Section 4.1, we evaluate the influence of the k parameter in novelty search’s performance.

2.2.2 Archive

The purpose of the archive in novelty search is to encourage exploration of new behaviour regions, besides maintaining diversity in the population. Without the memory effect provided by the archive, it is argued [18] that evolution may cycle between behaviour regions, due to the lack of evolutionary pressure towards novel regions of the search space.

Originally, it was proposed [15] that the archive should be composed of individuals that have high novelty scores when they first appear. All individuals that receive a novelty score higher than ρ_{min} are added to the archive. This threshold is dynamic to achieve a constant flow of individuals into the archive: ρ_{min} is increased by a fixed percentage if more than n individuals were added to the archive in the previous g generations, and decreased if no individuals were added in that interval. This configuration has been used in the majority of previous works (see Table 1). A simpler version, with fewer parameters, was used in [23], where the n most novel individuals of each generation are added to the archive.

In [16], the authors argue that it might be preferable to abandon the novelty criterion for composing the archive, and instead rely on a purely stochastic criterion. Every generation, each individual has a fixed (small) probability of being added to the archive. Although no results were provided, it was argued that adding only highly novel individuals to the archive has the disadvantage of potentially penalising areas of the behaviour space that may merit further exploration.

The necessity of using an archive at all has also been disputed [24, 29]. A typical argument against the use of an archive is the increased complexity of the nearest neighbours computation. This increase can, however, be significantly reduced if space-partitioning data structures are used, such as KD-trees. As mentioned before, in the *behavioural diversity* approach [25], no archive is used. This approach is combined with fitness-based evolution, and compared with novelty search in [24, 29]. The results show that novelty search can reach higher fitness scores in fewer generations, but *behavioural diversity* still manages to achieve good solutions for the given tasks. In [33, 34], no archive is used, and the novelty score is computed based only on k nearest neighbours. Although no results are provided, the authors state that the archive was not used since it did not lead to any performance gains.

In Section 4.1, we study how the type of archive used influences novelty search. We evaluate performance when a randomly-composed archive is used, when a novelty-based archive is used, and when there is no archive.

2.2.3 Combination of novelty and fitness

It has been shown that novelty search can struggle to find good solutions when the behaviour space is vast [1, 17], as a great effort might be spent exploring regions that are irrelevant for the task objective. This problem is typically overcome by combining the exploratory pressure of novelty search with the exploitative character of fitness-based evo-

lution. Such combination can lead to a more effective evolutionary process [21], where solutions can be reached faster and more consistently, with a relatively low impact on the diversity of behaviours explored.

A number of techniques have been proposed to accomplish this combination. The first class of techniques relies on a minimal criterion that the individuals must meet in order to be considered viable for selection. This minimal criterion can either be static and provided by the experimenter (*MCNS* – Minimal Criteria Novelty Search) [17], or dynamic and calculated based on the fitness scores of the current population (*PMCNS* – Progressive MCNS) [10]. In [23], two populations are used: one contains feasible individuals, which are scored based on novelty, and the other contains infeasible individuals, which are scored based on their proximity to the feasibility threshold.

The second class of techniques base their selection process on novelty and fitness scores simultaneously. In [24], the task objective (fitness function) is combined with the novelty objective in a Pareto-based multi-objectivisation. A simpler multi-objectivisation is proposed in [1], where the score of each individual is based on a *linear scalarisation* of its novelty and fitness scores, allowing the experimenter to control the relative weight of the novelty and fitness scores. In [12], half of the population is subject to novelty-based selection, while the other half is subject to fitness-based selection.

Comparisons between different combination techniques can be found in [10, 11]. The authors compare PMCNS with a linear scalarisation of novelty and fitness scores, using two different behaviour spaces. The reported results show that the performance of the two techniques is very similar, with PMCNS pushing exploration slightly more towards high-fitness regions. In Section 4.2, we compare the techniques most commonly found in previous works: multi-objectivisation, linear scalarisation, and PMCNS.

2.2.4 Underlying evolutionary algorithm

Novelty search only influences the scoring of the individuals. The underlying evolutionary algorithm (including genetic operators, chromosome representation, etc.) is therefore independent from the novelty search algorithm itself. When implementing novelty search, there are thus additional choices to make: which evolutionary algorithm to use, and how to configure its parameters. Most previous novelty search studies rely on neuroevolution algorithms (see Table 1). While it was argued that the NEAT algorithm [31] was especially suited for novelty search in early studies (e.g. [18]), later works have shown that simpler direct-encoding genetic algorithms can yield equally good results [26].

A relatively unexplored dimension is how the parameters of the underlying evolutionary algorithm influence novelty search. Previous studies in evolutionary robotics have shown that the genotype distance between two individuals can be uncorrelated with their behavioural distance [32]: very different genotypes can result in similar behaviours, and very different behaviours can originate from similar genotypes. An open question is therefore how the degree of genetic diversity influences the behavioural diversity and novelty generated in the novelty search process. In Section 4.3, we focus on the most ubiquitous evolutionary parameter: the mutation rate, which is closely associated with the degree of genetic diversity. We study the mutation rate in NEAT and in a direct-encoding genetic algorithm.

3. EXPERIMENTAL SETUP

3.1 Maze Navigation Task

For our experiments, we use a maze navigation task, in which a simulated robot controlled by an artificial neural network must navigate from a starting-point to an end-point within a fixed time limit. Such tasks have been commonly used in novelty search studies [12, 13, 15–18, 20–22, 24, 26, 34, 35], as they facilitate the design of experimental setups with different levels of deception. The fitness function rewards the candidate solutions according to the distance between the final position of the robot and the end-point of the maze, which can potentially deceive evolution towards dead-ends. Maze navigation can be seen as an abstraction for more complex domains, as dead-ends in the path towards the end-point represent local optima in the search space.

In our experimental setup, the robot is equipped with six rangefinders that read the distance to the nearest obstacle, and four pie-slice radar sensors that detect when the goal is within the pie-slice (see Figure 1). The readings of these sensors are the input to the neural network, and the two outputs control respectively the linear speed (mapped to $[-2, 2]$ units/step) and the turning speed of the robot (mapped to $[-\pi/4, \pi/4]$ rad/step). The simulation terminates when the end-point is reached, or when 500 time steps elapse. The task was simulated in MASON¹.

The fitness function and behaviour characterisation are the same as used in previous works (e.g. [18]). The fitness function is given by the distance between the final position of the robot (p) and the end-point of the maze (e), according to: $F = 1500 - \text{dist}(p, e)$ (a constant is added to keep fitness scores positive). The behaviour characterisation of a robot is given by its final position: $\beta = (p_x, p_y)$.

We experiment with a set of mazes that explores different levels of deceptiveness and behaviour space difficulty. A behaviour space is difficult to explore when most of the behaviour space is *irrelevant*² for solving the task, which is frequent in more complex tasks [1, 7, 11, 26]. This difficulty can be further increased if the *relevant* behaviour regions are harder to explore than the irrelevant ones. The mazes are depicted in Figure 2 and described below:

Hard Deceptive maze with an easy behaviour space: discovering new regions is often a step towards the goal (only $\sim 30\%$ of the space is irrelevant).

Zigzag Non-deceptive maze with a harder behaviour space: $\sim 60\%$ of the space is irrelevant.

Star A maze where the fitness gradient is roughly correct, but there are multiple small deceptive regions along the path.

Subset Very hard behaviour space: $\sim 85\%$ of the space is irrelevant to reach the objective. The relevant space is furthermore hard to explore due to narrow corridors.

Multi A maze with four different paths to reach the goal, all of them with some degree of deceptiveness.

Open Unbounded behaviour space. The exploration of the behaviour space can lead to regions that are very far away from the objective. 99.5% of the reachable behaviour space is irrelevant.

¹<http://cs.gmu.edu/~eclab/projects/mason/>

²We consider as irrelevant the portions of the maze that do not need to be traversed in order to reach the goal.

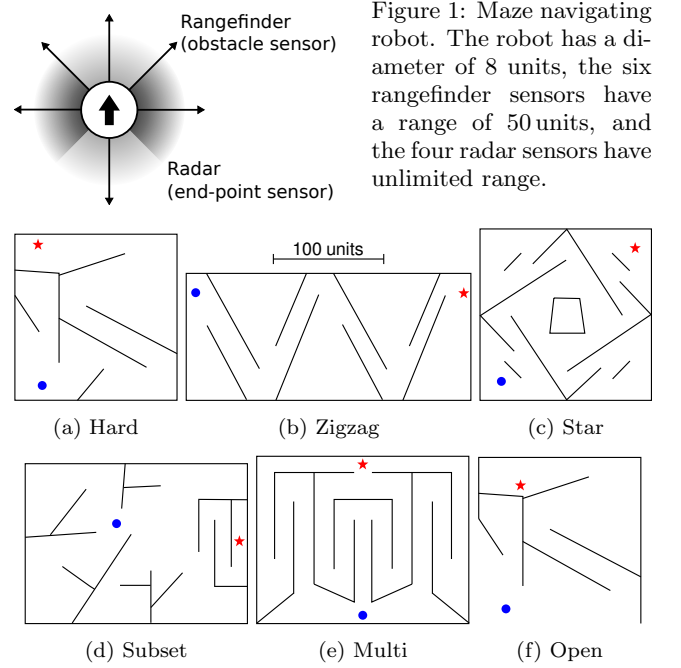


Figure 1: Maze navigating robot. The robot has a diameter of 8 units, the six rangefinder sensors have a range of 50 units, and the four radar sensors have unlimited range.

Figure 2: Mazes used in the experiments. The circle represents the initial position of the robot, scaled to match its size. The star marks the end-point.

3.2 Evolutionary Setup

The neural networks are evolved with NEAT [31], which evolves both the topology and the connection weights of the networks. The evolutionary algorithms are implemented on ECJ³ and using NEAT4J⁴. The NEAT parameter values are the default values of NEAT4J implementation: mutation probability is 0.25 (except when indicated otherwise), crossover probability is 0.2, and the probability of adding new connections and nodes is 0.05 and 0.03. Population size is 200 and the target number of NEAT species is 5.

Besides studying the quality of the solutions achieved, another important aspect is the degree of behaviour space exploration. Since novelty search's objective is to explore the behaviour space, the degree of exploration uniformity is a measure of novelty search's effectiveness, regardless of the task objective. To analyse exploration uniformity, we first divide the reachable behaviour space of each maze into 100 equal-sized regions. Based on this set of regions R , we compute the exploration uniformity $U(\varphi)$ in each evolutionary run. Let φ be the set of all individuals evolved in a run. $U(\varphi)$ is given by the distance between the distribution of φ over R , \mathbf{P}_φ , and the uniform distribution \mathbf{Q} . The distance metric is the *Jensen-Shannon distance* (JSD), a popular metric to measure distances between probability distributions [5]. Exploration uniformity is thus given by:

$$U(\varphi) = 1 - \text{JSD}(\mathbf{P}_\varphi, \mathbf{Q}), \text{ where:} \quad (2)$$

$$\mathbf{P}_\varphi = \left(\frac{|I_1|}{|\varphi|}, \dots, \frac{|I_{|R|}|}{|\varphi|} \right), \quad I_r = \{i \in \varphi \mid \text{region}(i) = r\}$$

$$\mathbf{Q} = \left(\frac{1}{|R|}, \dots, \frac{1}{|R|} \right)$$

³<http://cs.gmu.edu/~eclab/projects/ecj/>

⁴<http://neat4j.sourceforge.net/>

4. RESULTS

4.1 Novelty Search Parameters

We begin by studying the parameters that directly influence the computation of the novelty score. In these experiments, we use pure novelty search — the fitness function is not used at all. To understand how the archive configuration affects the optimal value for k , and vice-versa, we test multiple combinations of k values and archive types. Six values for k are used, ranging from 1 to 199. With $k=1$, only the distance to the closest individual is considered, and the maximum value of 199 was chosen as the population size is 200. We consider three types of archive, based on the configurations used in previous works:

None No archive is used, only the current population.

Novel Every generation the most novel λ individuals are added to the archive.

Random Every generation, λ randomly chosen individuals are added to the archive.

The archive (when used) has no size limit, to avoid additional parameters. The search for nearest neighbours was implemented with KD-trees, which allowed the use of large archives with an almost negligible computational cost.

Figure 3 shows the success ratio of each treatment (the proportion of evolutionary runs that evolved an effective solution), and the mean behaviour exploration uniformity, as defined in Section 3.2.⁵ Our results show that performance varies smoothly with the value of k , across all archive types and in both metrics. This result is consistent with previous works that argued for the robustness of this parameter [15, 18]. The results also show that the optimal value for k depends on the configuration of the archive. For all archive types, however, a value of $k=15$ yields either the best success ratio, or one that is not significantly different from it (Mann-Whitney test, $p > 0.25$).

When an archive is used, low to medium values of k yield the best performance. Regarding the novelty-based archive approach, the k values that yield the highest success ratio are in the interval [15, 100] ($p < 0.05$). The random-based archive approach is less sensitive to the k parameter, and the k values that yield the highest performance are in the interval [5, 100] ($p < 0.05$). When no archive is used, however, very low values of k are preferred, with the optimal values being in the interval [5, 15] ($p < 0.05$). When no archive is used, performance drops significantly as k increases.

Regarding archive types, our results show that a random-based archive is generally preferable: it yields the highest success ratios, it causes a more uniform exploration of the behaviour space, and it reduces the sensitivity to the parameter k . For all values of k , the random-based archive approach yields a significantly higher behaviour exploration uniformity than the novelty-based archive approach ($p < 0.001$). This result is consistent with [16], but in contrast to most previous works in which a novelty-based archive was used without justification (see Table 1). The use of no archive at all, however, should not be disregarded. When a low value of k is used (5), the success ratio achieved by this

⁵Due to space restrictions, we do not show the results obtained in each maze separately. Each data point was obtained with 30 runs for each of the six mazes, yielding a total of 180 runs for each treatment. As these distributions are not normal, error bars are not shown in any of the plots.

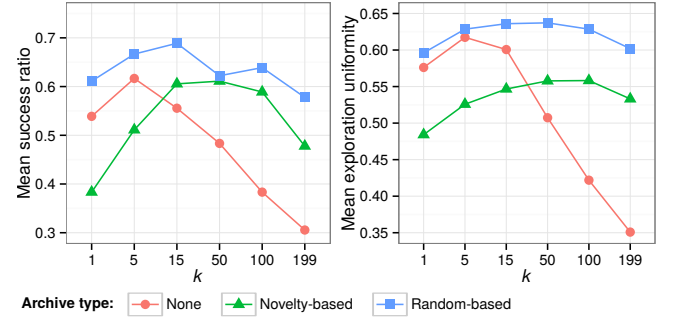


Figure 3: Performance of novelty search with multiple archive types, different values of k , and $\lambda=6$. Each point was obtained with 180 evolutionary runs (6 mazes \times 30 runs).

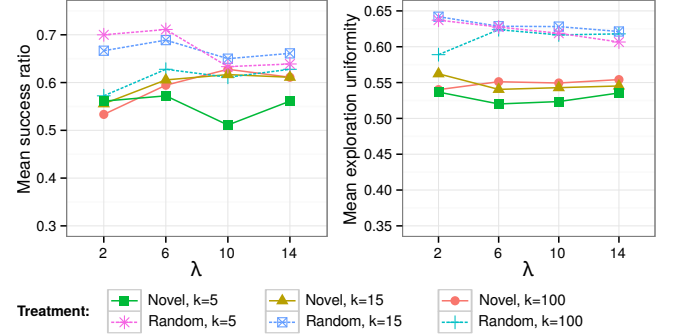


Figure 4: Performance impact of the archive growth rate λ on different novelty search treatments. Each point was obtained with 180 evolutionary runs (6 mazes \times 30 runs).

treatment is not significantly different to the ratio achieved by the best treatment with a novelty-based archive ($k=50$, $p=0.91$), or a random-based archive ($k=15$, $p=0.09$). This result is in line with [33, 34], where the authors use low values of k (3, 5, and 10) and state that the introduction of an archive did not bring performance improvements.

The use of the entire population and no archive to compute novelty scores, as done in the *behavioural diversity* approach [25], does not seem advantageous. This treatment ($k=199$ and no archive) actually yielded a significantly lower performance than all other evolutionary treatments. While the use of no archive does not seem deterrent, the use of the entire population to compute the novelty score consistently yielded significantly inferior results. It should be noted, however, that in previous works *behavioural diversity* was always used in combination with fitness-based evolution.

Finally, we analyse the influence of the archive growth rate λ . Figure 4 shows how performance varies with different growth rates, for different combinations of archive types and values of k . In all considered treatments (combinations of archive type and k), the growth rate has no significant impact on the success ratio (Kruskal-Wallis test, $p > 0.28$). The impact on exploration uniformity is also negligible. These results support previous works [18] in which it is argued that limiting the archive size has no significant performance impact when reasonable limits are used.

4.2 Combination of Novelty and Fitness

A significant point of divergence in the configuration of novelty-based search is if and how novelty scores should be combined with fitness scores. A number of techniques

have been proposed in previous works (see Section 2.2.3), but comparisons between them are scarce. In our study, we compare the techniques commonly used in previous works:

Fit Only the fitness scores are used.

LS-50 The score of an individual is given by the weighted sum of the normalised novelty and fitness scores [1]. Fitness and novelty have the same weight (50%).

LS-75 As the above, but novelty contributes to 75% of the score and fitness 25%.

NS Only the novelty scores are used.

NSGA Multiobjectivisation of novelty and fitness scores [24], using the NSGA-II algorithm [2]. As in previous works [21], the speciation mechanism of NEAT was disabled for compatibility with NSGA-II.

PMCNS Progressive minimal criteria novelty search [10]. The minimal criterion is given by the value of the 50th percentile of the fitness scores found in the current generation. Individuals above the criterion receive their normal novelty score, while individuals below receive a score of zero.

The results are shown in Figure 5. The computation of novelty scores is done with the parameters found in the previous section: $k=15$, and a randomly-composed archive with $\lambda=6$. Besides the metrics used in the previous section, we also measured the proportion of high-fitness individuals evolved: considering all individuals evolved in a given run, which percentage of them are solutions for the task.

The results show that *LS-50* and *NSGA* are the approaches with the highest success ratios (proportion of runs that evolved a solution for the task). No significant difference was found between these two ($p = 0.75$), and they are significantly superior to all other approaches ($p < 0.01$). *NSGA* yielded a significantly higher exploration uniformity than *LS-50* ($p < 0.001$). It is noteworthy that *NSGA*, when compared to pure novelty search, was considerably more biased towards high-fitness individuals, and reached solutions faster and more consistently. Nevertheless, the exploration uniformity in *NSGA* is just slightly inferior to that of pure novelty search, meaning that *NSGA* appears to balance exploration and exploitation exceptionally well. Linear scalarisation is considerably simpler to implement, and it is compatible with virtually any underlying evolutionary algorithm, but it should be used with caution in complex tasks or when the behaviour space is vast.

Previous works have shown that in linear scalarisation, a medium-high novelty weight (50–80%) yields the best results [1, 6, 7, 10, 11], hence the choice of 50% and 75% for this study. Our results confirm the importance of choosing an adequate novelty weight. In the maze navigation tasks used in study, a higher novelty weight (*LS-75*) offered little benefit over pure novelty search, and performed significantly worse than *LS-50* ($p < 0.01$). It should be noted, however, that the optimal novelty-fitness weight could be task-dependent.

PMCNS was generally ineffective, and did not offer significant advantages over pure novelty search, except in the amount of high-fitness individuals evolved ($p < 0.01$). This lack of effectiveness contrasts with previous works where PMCNS was successfully used [8, 10, 11]. Our results suggest that PMCNS’s low performance is due to the high degree of deception in most mazes. PMCNS performed better than pure novelty search in less deceptive mazes (*star* and *zigzag*), but worse in highly deceptive ones (*hard* and *multi*).

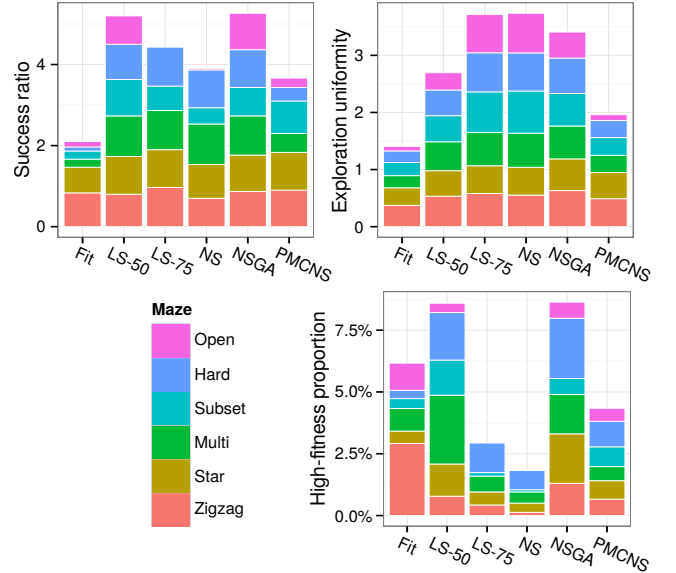


Figure 5: Comparison of multiple techniques for combining novelty search with fitness-based evolution. The high-fitness proportion is the proportion of evolved individuals that were able to solve the task.

4.3 Underlying Evolutionary Algorithm

While the underlying evolutionary algorithm typically has a large number of parameters, we only focus on the mutation rate in this study due to space restrictions. Mutation is a genetic operator present in all evolutionary algorithms, and the mutation rate directly influences the amount of genetic diversity in the population.

We study the mutation rate in two different neuroevolution algorithms, to cover most of the setups used in previous works (see Table 1): (i) the NEAT algorithm used in the previous sections, and (ii) a canonical genetic algorithm that evolves fixed-topology neural networks. For all mazes, the fixed-topology is a recurrent Elman network with 10 hidden neurons, resulting in a chromosome comprised of 232 weights. This topology was used in previous works [26] and was found to yield a good performance in preliminary experiments. The genetic algorithm uses tournament selection, an elite of size 5, per-gene Gaussian mutation, no crossover, and the population size is set to 200 (the same as in NEAT). The configuration of the NEAT algorithm is described in Section 3.2.

Novelty search is implemented over these two algorithms using the parameters found in Section 4.1: $k=15$ and a randomly-composed archive with $\lambda=6$. Fitness-based evolution is presented as a baseline in both cases. The results are depicted in Figure 6.

Overall, our results show that novelty search does not benefit from higher mutation rates, when compared to a traditional fitness-based algorithm. In the NEAT algorithm, the best mutation rates (with respect to the success ratio) were in the interval $[0.25, 0.8]$ for fitness-based evolution and in the interval $[0.1, 0.6]$ for novelty search ($p < 0.05$). In the genetic algorithm, the mutation rates that yielded the best results were in the interval $[0.025, 0.1]$ ($p < 0.05$) for both novelty search and fitness-based evolution.

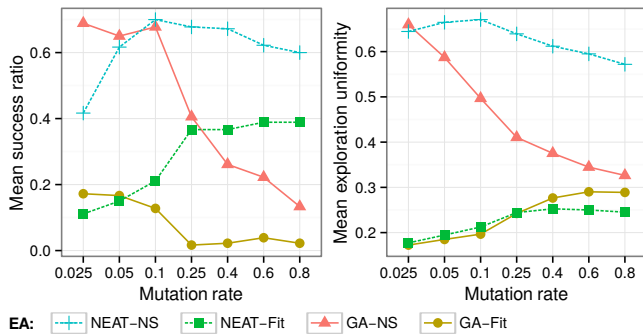


Figure 6: Success ratio and mean exploration uniformity of different techniques using different mutation rates. *GA-Fit*: Fitness-based evolution with a canonical genetic algorithm (no crossover); *GA-NS*: Pure novelty search with the same genetic algorithm; *NEAT-Fit*: Fitness-based evolution with NEAT; *NEAT-NS*: Novelty search with NEAT.

Regarding the behaviour exploration uniformity, the results show that novelty search does not benefit from higher mutation rates. In both NEAT and the genetic algorithm, the exploration uniformity of novelty search has a strong negative correlation with the mutation rate (Pearson’s correlation coefficient, $r = -0.96$ and $r = -0.88$ respectively).

These results highlight the substantial difference between genetic and behavioural diversity, and suggest that novelty search actually works better with lower mutation rates. Although additional experiments would be needed to fully understand this effect, our hypothesis is that low mutation rates do not lead to premature convergence due to the dynamic nature of the novelty objective, while they contribute to fine tuning the behaviours in the current population.

It is also noteworthy that the best *NEAT-NS* treatment (mutation rate = 0.1) was not significantly different from the best *GA-NS* treatment (mutation rate = 0.025), both in terms of the success ratio and exploration uniformity ($p < 0.01$). This confirms results from previous works [26] that show that NEAT is not a requirement for novelty search. The simpler genetic algorithm, however, has to be fine-tuned to use an adequate neural network topology.

5. CONCLUSION

We conducted a comprehensive empirical study on the parameters and configuration choices of novelty search, including: (i) the number k of nearest neighbours used in computation of novelty scores; (ii) the use of the archive, and how it should function; (iii) how novelty search can be combined with fitness-based evolution; and (iv) how the mutation rate of the underlying evolutionary algorithm affects the novelty search process. Our study is based on the simulated maze navigation task. We used a set of mazes that confront novelty search with different levels of deception and behaviour space difficulty. We analysed our results in two dimensions: (i) whether an evolutionary run is able to produce an effective solution or not; and (ii) whether novelty search is able to explore the behaviour space thoroughly and uniformly, regardless of the objective.

In summary, our findings can be distilled into the following guidelines:

- The parameter k is robust to moderate variation, but the optimal value depends on the type of archive used.
- Low to medium values of k are generally preferable. A value of $k=15$ yielded relatively good performance across all the tested archive types.
- A randomly composed archive is preferable over a novelty-based one, yielding better results across all the considered metrics.
- Novelty search is robust to moderate variations of the archive growth rate.
- The use of no archive at all should not be discarded: when used together with low values of k , it yielded satisfactory results.
- The highest performing methods for combining novelty and fitness were the multi-objectivisation of novelty and fitness scores, and the linear scalarisation with an equal weight for novelty and fitness. There were no major differences between these two.
- High mutation rates did not cause better exploration of the behaviour space. Novelty search actually benefited from lower mutation rates, when compared to fitness-based evolution.

Maze navigation tasks have been used in a large number of previous novelty search works, and it has been shown that most conclusions drawn in this domain generally transfer to more complex and realistic domains [11, 18, 26]. Nevertheless, in future work, we will extend our study and assess how the reported results generalise to other domains. In this work, we vary the behaviour space by varying the task environment. Another possibility would be to vary the behaviour characterisation. We intend to assess if certain features of the characterisation, such as its length, should influence the choice of novelty search parameters.

Acknowledgements

This research is supported by Fundação para a Ciência e Tecnologia (FCT), with grant SFRH/BD/89095/2012 and projects UID/EEA/50008/2013, UID/Multi/04046/2013, and EXPL/EEI-AUT/0329/2013.

6. REFERENCES

- [1] G. Cuccu and F. J. Gomez. When novelty is not enough. In *European Conference on the Applications of Evolutionary Computation (EvoApplications)*, volume 6624 of *LNCS*, pages 234–243. Springer, 2011.
- [2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [3] S. Doncieux and J.-B. Mouret. Behavioral diversity measures for evolutionary robotics. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE Press, 2010.
- [4] S. Doncieux and J.-B. Mouret. Behavioral diversity with multiple behavioral distances. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1427–1434. IEEE Press, 2013.
- [5] D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003.
- [6] J. Gomes and A. L. Christensen. Generic behaviour similarity measures for evolutionary swarm robotics.

- In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 199–206. ACM Press, 2013.
- [7] J. Gomes, P. Mariano, and A. L. Christensen. Avoiding convergence in cooperative coevolution with novelty search. In *International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pages 1149–1156. IFAAMAS, 2014.
 - [8] J. Gomes, P. Mariano, and A. L. Christensen. Novelty search in competitive coevolution. In *Parallel Problem Solving from Nature (PPSN)*, volume 8672 of *LNCS*, pages 233–242. Springer, 2014.
 - [9] J. Gomes, P. Mariano, and A. L. Christensen. Systematic derivation of behaviour characterisations in evolutionary robotics. In *International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 212–219. MIT Press, 2014.
 - [10] J. Gomes, P. Urbano, and A. L. Christensen. Progressive minimal criteria novelty search. In *Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, volume 7637 of *LNAI*, pages 281–290. Springer, 2012.
 - [11] J. Gomes, P. Urbano, and A. L. Christensen. Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2–3):115–144, 2013.
 - [12] B. Inden, Y. Jin, R. Haschke, H. Ritter, and B. Sendhoff. An examination of different fitness and novelty based selection methods for the evolution of neural networks. *Soft Computing*, 17(5):753–767, 2013.
 - [13] S. Kistemaker and S. Whiteson. Critical factors in the performance of novelty search. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 965–972. ACM Press, 2011.
 - [14] P. Krcah. Solving deceptive tasks in robot body-brain co-evolution by searching for behavioral novelty. In *Advances in Robotics and Virtual Reality*, volume 26 of *Intelligent Systems Reference Library*, pages 167–186. Springer, 2012.
 - [15] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *International Conference on Artificial Life (ALIFE XI)*, pages 329–336. MIT Press, 2008.
 - [16] J. Lehman and K. O. Stanley. Efficiently evolving programs through the search for novelty. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 837–844. ACM Press, 2010.
 - [17] J. Lehman and K. O. Stanley. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 103–110. ACM Press, 2010.
 - [18] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.
 - [19] J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 211–218. ACM Press, 2011.
 - [20] J. Lehman and K. O. Stanley. Improving evolvability through novelty search and self-adaptation. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 2693–2700. IEEE Press, 2011.
 - [21] J. Lehman, K. O. Stanley, and R. Miikkulainen. Effective diversity maintenance in deceptive domains. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 215–222. ACM Press, 2013.
 - [22] J. Li, J. Storie, and J. Clune. Encouraging creative thinking in robots improves their ability to solve challenging problems. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 193–200. ACM Press, 2014.
 - [23] A. Liapis, G. N. Yannakakis, and J. Togelius. Constrained novelty search: A study on game content generation. *Evolutionary Computation*, 2014. *In press*.
 - [24] J.-B. Mouret. Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*, volume 341 of *Studies in Computation Intelligence*, pages 139–154. Springer, 2011.
 - [25] J.-B. Mouret and S. Doncieux. Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1161–1168. IEEE Press, 2009.
 - [26] J.-B. Mouret and S. Doncieux. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary Computation*, 20(1):91–133, 2012.
 - [27] E. Naredo and L. Trujillo. Searching for novel clustering programs. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 1093–1100. ACM Press, 2013.
 - [28] E. Naredo, L. Trujillo, and Y. Martínez. Searching for novel classifiers. In *Genetic Programming (EuroGP)*, volume 7831 of *LNCS*, pages 145–156. Springer, 2013.
 - [29] C. Ollion and S. Doncieux. Why and how to measure exploration in behavioral space. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 267–274. ACM Press, 2011.
 - [30] S. Risi, C. E. Hughes, and K. O. Stanley. Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 18(6):470–491, 2010.
 - [31] K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.
 - [32] L. Trujillo, G. Olague, E. Lutton, F. F. De Vega, L. Dozal, and E. Clemente. Speciation in behavioral space for evolutionary robotics. *Journal of Intelligent & Robotic Systems*, 64(3-4):323–351, 2011.
 - [33] P. Urbano and L. Georgiou. Improving grammatical evolution in Santa Fé trail using novelty search. In *European Conference on Artificial Life (ECAL)*, pages 917–924. MIT Press, 2013.
 - [34] P. Urbano, E. Naredo, and L. Trujillo. Generalization in maze navigation using grammatical evolution and novelty search. In *Theory and Practice of Natural Computing*, volume 8890 of *LNCS*, pages 35–46. Springer, 2014.
 - [35] R. Velez and J. Clune. Novelty search creates robots with general skills for exploration. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 737–744. ACM Press, 2014.
 - [36] L. D. Whitley. Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms*, pages 221–241. Morgan Kaufmann, 1991.