Surprise Search: Beyond Objectives and Novelty

Daniele Gravina
Institute of Digital Games
University of Malta
Msida 2080, Malta
daniele.gravina@um.edu.mt

Antonios Liapis
Institute of Digital Games
University of Malta
Msida 2080, Malta
antonios.liapis@um.edu.mt

Georgios N. Yannakakis
Institute of Digital Games
University of Malta
Msida 2080, Malta
georgios.yannakakis@um.edu.mt

ABSTRACT

Grounded in the divergent search paradigm and inspired by the principle of *surprise* for unconventional discovery in computational creativity, this paper introduces *surprise search* as a new method of evolutionary divergent search. Surprise search is tested in two robot navigation tasks and compared against objective-based evolutionary search and novelty search. The key findings of this paper reveal that surprise search is advantageous compared to the other two search processes. It outperforms objective search and it is as efficient as novelty search in both tasks examined. Most importantly, surprise search is, on average, faster and more robust in solving the navigation problem compared to objective and novelty search. Our analysis reveals that surprise search explores the behavioral space more extensively and yields higher population diversity compared to novelty search.

CCS Concepts

•Computing methodologies → Search methodologies; Genetic algorithms; *Evolutionary robotics;*

Keywords

Surprise search; novelty search; divergent search; deception; fitness-based evolution; map navigation; NEAT

1. INTRODUCTION

The widely accepted approach in search is to design a function that will reward solutions with respect to a particular objective that characterizes their goodness: better solutions have higher values with respect to that objective. In evolutionary computation (EC) the *fitness function* [9] encapsulates the principle of evolutionary pressure for fitting (adapting) within the environment. While it is natural to think that measuring progress in terms of fitness [9, 22] is the most appropriate approach towards finding a high-fit solution, recent findings from evolutionary divergent search [14, 17] suggest that explicit *objective* (fitness) design can be detrimental to evolutionary search. Instead, aspects of divergent search beyond objectives — such as *novelty* — have proven more efficient in a number of tasks such as robot navigation [14] and locomotion [15].

As a general principle, more *deceptive* [9, 29] problems challenge the design of a corresponding objective function; in this paper we follow [31] and view deception as the *intuitive definition of problem hardness*. The effectiveness of a fitness function in EC is largely affected by the multimodality of the search space and the local optima that may exist in the fitness landscape. In turn, a fitness function attributes deceptive characteristics to the search space [9] such as roughness and epistasis. On that basis, an ill-posed fitness acts against the problem's objective as it drives the algorithm to undesired directions in the search space.

While search towards a particular objective is a dominant practice within EC and machine learning at large, no explicit objectives are considered in open-ended evolution studies within artificial life [4]. Instead, it is typical to consider open-ended search for e.g. survival [33, 1]. Most importantly for this paper, a large body of research within computational creativity and generative systems [3, 26, 32] focuses on the creative capacity of search rather than on the objectives, since creativity is a human-centric and highly subjective notion that cannot be easily formalized. Instead of objectives, particular dimensions of creativity such as *value* and *novelty* [26] define dominant directions for the search towards creative outcomes or unconventional solutions to problems. According to Ritchie [26], value is the degree to which a solution (or generated output from a computational creator) is of high quality whereas novelty is the degree to which a solution (or output) is dissimilar to existing examples. While searching for value can be viewed as the metaphor of fitness-based EC, searching for novelty can be viewed as the metaphor of divergent evolutionary search towards novelty [14, 17]. Both can be used effectively for the evolutionary generation of highly novel, yet valuable, creative outcomes [18].

According to other perspectives within computational creativity, however, novelty and value are not sufficient for the discovery of unconventional solutions to problems [11]. As novelty does not cater for the temporal aspects of discovery, it is suggested that *surprise* is included as a core assessment dimension of a generated solution [11]. The notion of surprise is built on literature from cognitive science suggesting that not only are humans capable of self-surprise but, most importantly, that surprise is a core internal driver of

GECCO '16, July 20-24, 2016, Denver, CO, USA
© 2016 ACM. ISBN 978-1-4503-4206-3/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2908812.2908817

creativity and its final outcomes [10]. Surprise constitutes a powerful drive for computational discovery as it incorporates predictions of an expected behavior that it attempts to deviate from; these predictions may be based on behavioral relationships in the solution space as well as historical trends derived from the algorithm's sampling of the domain.

We draw inspirations from the above perspectives in computational creativity and we propose the use of *surprise* as a new form of evolutionary divergent search. Our hypothesis is that the search for surprise (i.e. *surprise search*) is beneficial to EC as it complements our search capacities with highly efficient and robust algorithms beyond the search for objectives or mere novelty. To test our hypothesis, we introduce the idea of surprise search and propose an evolutionary algorithm that realizes it. Using the methodology and testbeds proposed in [14], we evolve robot controllers employing neuroevolution of augmenting topologies (NEAT) [28] for two maze navigation tasks and we compare the performance of surprise search against fitness-based evolution and novelty search. We use two performance measures for this comparison: efficiency (i.e. maximum fitness obtained) and robustness (i.e. the number of times a problem is solved). The key findings of the paper suggest that surprise search is as efficient as novelty search and both algorithms, unsurprisingly, outperform fitness-based search. Furthermore, surprise search appears to be the most robust algorithm in the two test-bed tasks. While both novelty and surprise search converge to the objective significantly faster than fitness-based search, surprise search solves the navigation problem faster, on average, and more often than novelty search. The experiments of this paper validate our hypothesis that surprise can be beneficial as a divergent search approach and provide evidence for its supremacy over novelty search in the tasks examined.

2. BACKGROUND

The term *deception* in the context of EC was introduced by Goldberg [8] to describe instances where highly-fit building blocks, when recombined, may guide search away from the global optimum. Since that first mention, the notion of deception has been refined and expanded to describe several problems which challenge evolutionary search for a solution; indeed Whitley argues that "the only challenging problems are deceptive" [31]. EC-hardness is often attributed to deception, as well as sampling error [19] and a rugged fitness landscape [13]. In combinatorial optimization problems, the fitness landscape can affect optimization when performing neighborhood search, which is usually the case [25]. Such a search process assumes that there is a high correlation between the fitness of neighboring points in the search space, and that genes in the chromosome are independent of each other. The latter assumption refers to *epistasis* [5] which is a factor of GA-hardness: when epistasis is high (i.e. where too many genes are dependent on other genes in the chromosome), the algorithm searches for a unique optimal combination of genes but no substantial fitness improvements are noticed during this search [5].

Deception actively leads search away from the global optimum — often by converging prematurely to local optima in the search space. To discourage this behavior, numerous approaches have been proposed as surveyed by [17]. Many diversity maintenance techniques, such as speciation [28] and niching [30] enforce local competition among similar solutions. Localized competition allows the population to explore multiple promising directions, discouraging premature convergence. An alternative way of exploring the search is coevolution, where the calculation of fitness is dependent on the current population [2]; competition between individuals in the same population ideally leads to an arms race towards better solutions and finds a better gradient for search. However, coevolution runs the risk of causing mediocre stalemates where all competitors perform poorly and cannot improve, or that one competitor is so much better than the others that no gradient can be found [7].

Novelty search [14] differs from previous approaches at handling deceptive problems as it explicitly ignores the objective of the problem it attempts to solve. The search methods described above provide control mechanisms, modifiers or alternate objectives which complement the gradient search towards better solutions; in contrast, novelty search motivates exploration of the search space by rewarding individuals which are different in the phenotypic (or behavioral) space without considering whether they are objectively 'better' than others. Novelty search is different than a random walk, however, as it explicitly provides higher rewards to more diverse solutions (hence promoting exploration of the search space) and also because it maintains a memory of the areas of the search space that it has previously explored; the latter is achieved with a *novel archive* of past novel individuals, with highly novel individuals being constantly added to this archive. Each individual's novelty score is the average distance from a number of closest neighbors in the behavioral space; neighbors can be members of the current population or the novel archive. The distance measure is problem-dependent and can also bias the search [14] and thus affect the performance and behavior of the novelty search algorithm: examples include the agents' final positions in a two-dimensional maze solving task (as the one in Section 5), the position of a robot's center of mass [14], or properties of images such as brightness and symmetry [16].

3. THE NOTION OF SURPRISE SEARCH

This section discusses the notion of surprise as a form of divergent search. For that purpose we first attempt to define surprise, we then compare it against the notion of novelty and finally we discuss the dissimilarities between novelty search and surprise search as evolutionary search methods.

3.1 What is Surprise?

The study of surprise has been central in neuroscience, psychology, cognitive science, and to a lesser degree in computational creativity and computational search. In psychology and emotive modeling studies, surprise defines one of Ekman's six basic emotions [6]. Within cognitive science, surprise has been defined as a temporal-based cognitive process of the unexpected [20], a violation of a belief [23], a reaction to a mismatch [20], or a response to novelty [32]. In computational creativity, surprise has been attributed to the creative output of a computational process [11, 32].

While variant types and taxonomies of surprise have been suggested in the literature — such as aggressive versus passive surprise [11] — we can safely derive a definition of surprise that is general across all disciplines that study surprise as a phenomenon. For the purposes of this paper we define surprise as the *deviation from the expected* and we use the notions *surprise* and *unexpectedness* interchangeably due to their highly interwoven nature.

3.2 Novelty vs. Surprise

Novelty and surprise are different notions by definition as it is possible for a solution to be both novel and/or expected to variant degrees. Following the core principles of Lehman and Stanley [14] and Grace et al. [11], novelty is defined as the degree to which a solution is *different from prior* solutions to a particular problem. On the other hand, surprise is the degree to which a solution is *different from the expected* solution to a particular problem.

Expectations are naturally based on inference from past experiences; analogously surprise is built on the temporal model of past behaviors. Surprise is a temporal notion as expectations are by nature temporal. Prior information is required to predict what is expected; hence a *prediction of the expected* is a necessary component for modeling surprise computationally. By that logic, surprise can be viewed as a *temporal novelty* process. Another interesting temporal metaphor of the relationship between surprise and novelty is that the first can be viewed as the *time derivative* of the latter — e.g. position (novelty) and velocity (surprise).

3.3 Novelty Search vs. Surprise Search

According to Grace et al. [11], novelty and value (i.e. objective in the context of EC) are not sufficient for the discovery of unconventional solutions to problems (or creative outputs) as novelty does not cater for the temporal aspects of discovery. Novelty search rewards divergence from *prior* behaviors [14] and provides the necessary stepping stones toward achieving an objective. Surprise, on the other hand, complements the search for novelty as it rewards exploration and divergence from the *expected* behavior. In other words while novelty search attempts to discover new solutions, surprise search attempts to deviate from expected solutions.

Highly relevant to this study is the work on computational models of surprise for artificial agents [21]. However, that work does not consider using a computational model of surprise for search. Other aspects of unexpectedness such as intrinsic motivation [24] and artificial curiosity [27] have also been modeled. The concepts of novelty within reinforcement learning research are also interlinked to the idea of surprise search [12, 24]. Artificial curiosity and intrinsic motivation, however, are not resembling the search for surprise which, similarly to novelty search, is based on evolutionary divergent search and motivated by open-ended evolution.

Inspired by the above arguments and findings in computational creativity, we view surprise for computational search as the degree to which expectations about a solution are violated through observation [11]. Our hypothesis is that if modeled appropriately, surprise may enhance divergent search and complement or even surpass the performance of traditional forms of divergent search such as novelty. The main findings of this paper validate our hypothesis.

4. THE SURPRISE SEARCH ALGORITHM

To realize surprise as a search mechanism we need to model a process that rewards *deviation from the expected.* We can decompose the task by first defining what an *expected behavior* is within search and then quantifying *deviation* from it. These two processes express the two main components of the surprise search algorithm described in this section. This process yields an individual's *surprise score* which replaces any objective-driven fitness score and puts pressure on unexpected solutions in the behavioral space.

4.1 Expected Behavior

Modeling an expected behavior requires a predictive model built on prior behaviors of an individual or a set of individuals in the population. In that process, three core questions need to be addressed with respect to the amount of history, model type, and locality of prediction considered.

How much history of prior behaviors (h) should surprise search consider? We consider this to be a domain-dependent parameter for the algorithm. Similarly to a novelty archive [14], behaviors that have performed well in the past could be included in a *surprise archive.*

What predictive model (m) should surprise search use? Any predictive modeling approach can be used to predict a future behavior, such as a simple linear regression of a number of points in the behavioral space, non-linear extrapolations, or machine learned models. Again, we consider the predictive model, m, to be problem-dependent.

How local (k) are the behaviors surprise search needs to consider to make a prediction? Predictions can be based on prior compound positions in the behavioral space for the whole population (global information), or a number of population groups, or even considering only prior behaviors of an individual. A parameter k determines the level of *prediction locality* which can vary from 1 (all individuals are used to determine the expected behavior of the population as a whole) to P which is the population size.

In summary, the set of expected behaviors, $\mathbf{p}$, are derived from a predictive model, m, that considers a degree of local (or global) behavioral information (expressed by k) and depends on a history of prior behaviors (expressed by h). In its general form $\mathbf{p}$ is as $\mathbf{p} = m(h, k)$.

4.2 Deviation

To put pressure on unexpected behaviors, we need an estimate of the deviation of a behavior from the expected. Following the principles of novelty search [14], this estimate is derived from the *behavior space* as the average distance to the n-nearest expected behaviors (prediction points). We thus define that estimate as the *surprise value*, s for an individual i in the population as follows:

$$s(i) = \frac{1}{n} \sum_{j=0}^{n} d_s(i, p_{i,j}) \qquad (1)$$

where d_s is the domain-dependent measure of behavioral difference between an individual and its expected behavior, $p_{i,j}$ is the j-closest prediction point (expected behavior) to individual i and n is the number of prediction points considered; n is a problem-dependent parameter determined empirically.

4.3 Important notes

Surprise search operates similarly to novelty search with respect to evolutionary dynamics. As surprise search considers the current generation of solutions and a set of prior behaviors (expressed by h) to make predictions of expected behavior, it maintains a temporal window of where search has been. However, surprise search operates differently to novelty search with respect to the goal: surprise maximizes deviation from the expected behaviors whereas novelty moves the search towards new behaviors. This evidently creates a new form of divergent search that considers prior behaviors *indirectly* to make predictions to deviate from.

As surprise search ignores objectives, a concern could be whether it is merely a version of random walk. Surprise search is not a random walk as it explicitly maximizes unexpectedness: surprise search allows for a temporal archive of behaviors that accumulates a record of earlier positions in the behavioral space (similarly to novelty search). Comparative experiments with variant random benchmark algorithms in Section 5 show that surprise search traverses the search space in a different and far more effective manner.

5. MAZE NAVIGATION TEST BED

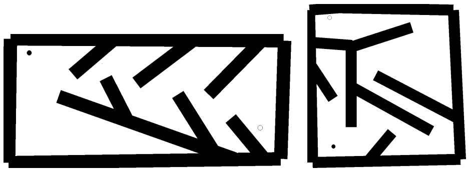
An appropriate domain to test the performance of surprise search is one that yields a deceptive fitness landscape, where search for surprising behaviors can be evaluated against novelty and objective search. Inspired by the comparison of novelty versus objective search in [14], we consider the two-dimensional maze navigation task as our test bed in this paper. The maze navigation task is a representative deceptive problem due to the presence of dead-ends acting as local optima which do not bring search closer to finding the goal in the maze. In this section we describe the maze navigation problem briefly and the two test bed mazes considered.

In the maze navigation task a robot (agent) controlled by an artificial neural network (ANN) has to find a path from a starting position to a goal position in a fixed time in a human-authored maze. The navigation task is made harder with the presence of more dead-ends, and by placing the starting position far away from the goal position. Following the perception system of [14], the robot has six range sensors measuring the distance to the nearest obstacle and four range radars that fire when the goal is within their range. This results in 10 inputs for the robot's ANN controller (6 range sensors and 4 radar sensors). The robot has two effectors (ANN outputs) which control the robot's movement, i.e. whether to turn and speed up or slow down. More details about the agent and its ANN can be found in [14].

The two mazes named *medium* and *hard* used in [14] are considered for testing the performance of surprise search. The "medium" maze (see Figure 1a) is moderately difficult as an algorithm should evolve a robot able to avoid the dead-ends existent within the maze towards finding the goal. The "hard" maze (see Figure 1b) is more deceptive, primarily due to the dead-end at the leftmost part of the maze; an algorithm must search in less promising (lower-fit) areas of the maze towards finding the global optimum. We consider a robot successful if it manages to reach the goal within a radius of five units at the end of an evaluation of 400 simulation steps.

6. EXPERIMENTS

The maze navigation problem is used to compare the performance of surprise, novelty and objective search. Section 6.1 provides the details and parameters for all the algorithms compared. We then compare the algorithms' efficiency and robustness and finally we analyze some typical generated solutions on the behavioral and the genotypical space.



(a) Medium maze (b) Hard maze

Figure 1: The robot maze navigation tests that appear in [14] as "medium" and "hard". For comparative purposes the same mazes are used for all experiments in this paper. The filled and empty circle represent the robot's starting position and the goal, respectively.

6.1 Algorithm parameters

All three algorithms use NEAT to evolve a robot controller with the same parameters as in [14], where the maze navigation task and the mazes of Fig. 1 were introduced. Evolution is carried on a population of 250 individuals for a maximum of 300 generations in all experiments presented in this paper. The NEAT algorithm uses speciation and recombination, as described in [28]. The specific parameters used for each of the three main algorithms compared are detailed below.

Objective search uses the agent's proximity to the goal as a measure of its fitness. Following [14], proximity is measured as the Euclidean distance between the goal and the position of the robot at the end of the evaluation.

Novelty search uses the same novelty metric and parameter values as presented in [14]. In particular, the novelty metric is the average distance of the robot from the 15 nearest neighboring robots among those in the current population and in a novel archive. *Distance* in this case is the Euclidean distance between two robot positions at the end of the evaluation period; this rewards robots ending in positions that no other robot has explored yet.

Surprise search uses the surprise metric of Equation (1) to motivate the generation of *unexpected behaviors*. As with other algorithms compared, *behavior* in the maze navigation domain is expressed as the position of the robot at the end of an evaluation. The behavioral difference d_s in Equation (1) is the Euclidean distance between the final position of the robot and a considered prediction point, p. Note that in our experiments we only consider the closest prediction point to each robot position to deviate from, i.e. $n = 1$ in Equation (1). Following the general formulation of surprise in Section 4.1, the prediction points are a function of a model m that considers k local behaviors of h prior generations. In this initial study we use the simplest possible prediction model (m) which is a one-step linear regression of two points ($h = 2$) in the behavioral space. Thus, only the two previous generations are considered when creating prediction points to deviate from in the current generation; in the first two generations the algorithm performs mere random search due to lack of prediction points. The locality (k) of behaviors is determined by the number of behavioral clusters in the population which is obtained by running k-means on the final robot positions. The surprise search algorithm applies k-means at each generation by seeding the initial configuration of the k centroids with the centroids obtained in the previous generation (this seeding process is skipped only in the first generation). This way the algorithm is able to pair centroids in subsequent generations and track their behavioral history. Using the k pairs of centroids of the last two generations we create k prediction points for the current generation through a simple linear projection. As mentioned in Section 4.1, the number of clusters k is a problem-dependent parameter obtained empirically by varying k between 10 and P in increments of 10 and selecting the k that yields the most maze solutions (successes); k is 200 and 100 for the medium and hard maze experiments, respectively. The impact of history, h, and the prediction model, m, on the algorithm's performance is not examined in this paper and remains open to future studies (see Section 7).

Two more **baseline algorithms** are included for comparative purposes. *Random search* is a baseline proposed in [14] which uses a uniformly-distributed random value as the fitness function of an individual. The second baseline algorithm is a variant of surprise search with random prediction points, identified as *surprise search (random)*. In this variant we substitute the prediction points obtained via surprise search with a uniform random distribution, which provides k prediction points within the maze.

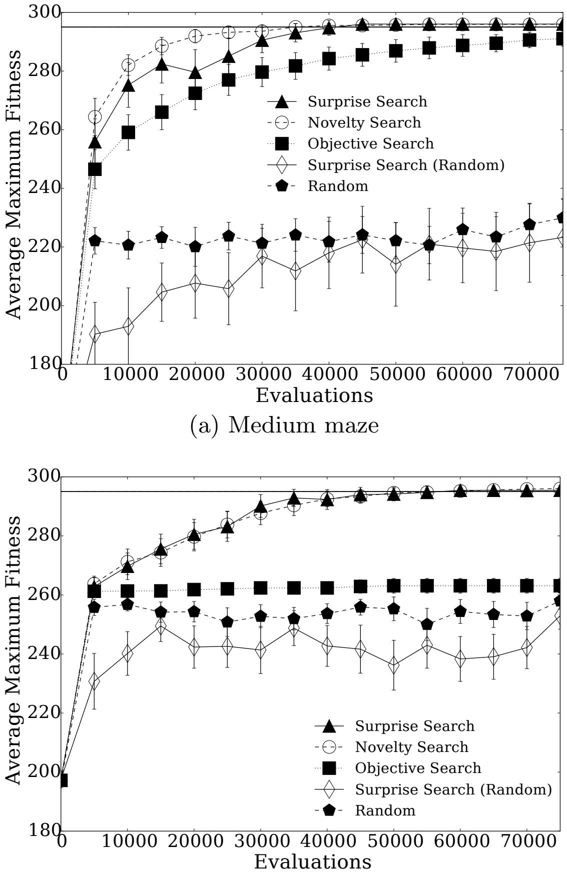
To test the performance of all above mentioned algorithms we follow the approach proposed in [34] and compare both their efficiency and robustness in both test bed mazes. Results are collected from 50 independent evolutionary runs; reported significance (and p values) in this paper is obtained via two-tailed Student's t-tests; significance is 5%.

6.2 Efficiency

Following the analysis in [14], efficiency is defined as the *maximum fitness over time*; fitness is 300 minus the Euclidean distance between the final position of the robot and the goal. Figure 2 shows the average maximum fitness across evaluations for each algorithm for the two mazes, averaged from 50 independent runs.

Observing the efficiency of the different algorithms in the medium maze in Fig. 2a, it is obvious that both surprise search and novelty search converge to the absolute maximum fitness after approximately 35,000 evaluations. While novelty search appears to yield higher average maximum fitness values than surprise search, this difference tends to be insignificant. On average, novelty search obtains a final maximum fitness of 296.18 ($\sigma = 1.05$) while surprise search obtains a fitness of 296.03 ($\sigma = 0.92$); $p > 0.05$. From the derived 95% confidence intervals there appears to be a significant difference only between 19,000 and 25,000 evaluations; novelty search yields higher average maximum fitness during this time interval. This difference is due to the predictions of surprise search, as two consecutive generations may yield very distant cluster centroids which in turn create predictions which are even further away. Eventually, the population converges and consecutive generations' centroids (and predictions) are closer to each other, allowing surprise search to solve the maze. Objective search seems moderately efficient in approaching the goal, although it is unable to find it in all runs; in contrast, the two baseline algorithms have a poor efficiency and show very little improvement as evolution progresses. The poor efficiency of the two baselines demonstrates that surprise search is different from random search and moreover that its predictive model positively affects the search.

For the more deceptive hard maze, Fig. 2b shows that there is no significant difference across evaluations between surprise search and novelty search, while the other algorithms' efficiency is far lower. On average, novelty search obtains a final maximum fitness of 296.03 ($\sigma = 1.55$) while surprise search obtains 295.46 ($\sigma = 5.12$); $p > 0.05$. In such a deceptive maze, surprise and novelty search find the solution in 49 and 48 out of 50 runs, respectively, while the other algorithms have a much lower success rate. Of particular interest is the performance of objective search, which reaches a high fitness score around 260 quickly and then fails to improve it; this is due to the deceptive nature of the maze, where the local optima of 260 is in the left-most dead-end in Fig. 1b which is close to the goal. In order to solve the maze, however, the algorithm must explore far less fit areas of the search space and thus objective search remains stuck in the local optimum.

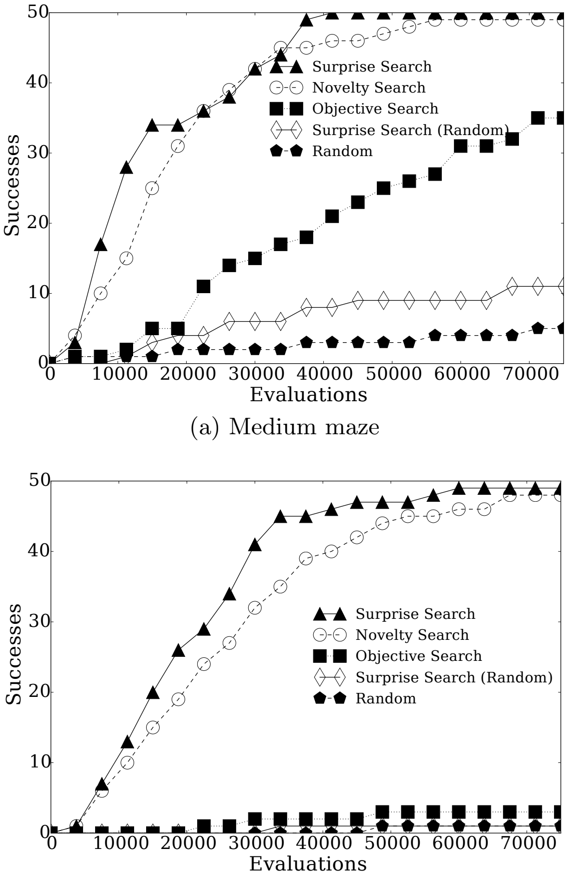


(a) Medium maze

(b) Hard maze

Figure 2: **Efficiency** (average maximum fitness) comparison for the two mazes of [14]. The graphs depict the evolution of fitness over the number of evaluations. Values are averaged across 50 runs of each algorithm and the error bars represent the 95% confidence interval of the average.

It is worth noting that if efficiency is alternatively viewed as the effort it takes an algorithm to find a solution then surprise search has a clear benefit over novelty and objective search. In the medium maze surprise search manages to find the goal, on average, in 14,897 evaluations (σ = 11,544) which is faster than novelty (18,993; σ = 14,526) and significantly faster than objective search (48,266; σ = 23,728). Surprise search is similarly advantageous in the hard maze as it solves the problem in 21,390 evaluations (σ = 14,519) which is faster than novelty (27,167; σ = 18,510) and significantly faster than objective search (72,427; σ = 10,587).



(a) Medium maze

(b) Hard maze

Figure 3: **Robustness** comparison for the two mazes of [14]. The graphs depict the evolution of algorithm successes in solving the maze problem over the number of evaluations.

While the analysis of computational effort is beyond the scope of this paper it is worth noting that, on average, surprise search yields a significantly lower average CPU cost per generation (0.285 s) compared to novelty search (0.353 s); all experiments run in a 3 GHz workstation.

The findings from the above experiments indicate that, in terms of maximum fitness obtained, surprise search is comparable to novelty search and far more efficient than objective search in deceptive domains. It is also clear that, on average, it finds the solution faster than any other algorithm.

6.3 Robustness

In this section we compare the algorithms' *robustness* defined as the number of successes obtained by the algorithm across time (i.e. evaluations). Figure 3 shows the robustness of the different algorithms for the two mazes, collected from 50 runs. Observing the successes of each algorithm on the medium maze (see Fig. 3a), surprise search finds more solutions in the first 20,000 evaluations than novelty search. Moreover, surprise search manages to find all 50 solutions faster than novelty search; in approx. 40,000 evaluations versus in 55,000 evaluations, respectively. Novelty search fails to find the solution in 1 out of the 50 runs. Interestingly, while surprise search finds more solutions in the first 20,000 evaluations, novelty search has a comparable or higher maximum fitness in Fig. 2a, on average; this points to the fact that while some individuals in surprise search manage to reach the goal, others do not get as close to it as in novelty search. As noted in Section 6.2, objective search does not manage to find solutions in 15 runs despite attaining a relatively high maximum fitness. Finally, the two random baseline algorithms have a similar robustness and manage, by chance, to discover a few solutions.

On the hard maze, Fig. 3b shows that while surprise and novelty search attain an equal number of successes in the first 10,000 evaluations, surprise search systematically solves the maze more times when more evaluations are considered. Once again, the superior robustness of surprise search after 10,000 evaluations is not captured in the efficiency graph of Fig. 2b, leading us to assume that during surprise search individuals' behavior changes from inferior to optimal (i.e. solving the maze) more abruptly than in novelty search, where the improvement in individuals' performance is smoother. Unlike the medium maze, objective search performs almost as poorly as the two random baseline algorithms, since the deceptive fitness drives individuals to the dead-ends of the maze.

6.4 Analysis

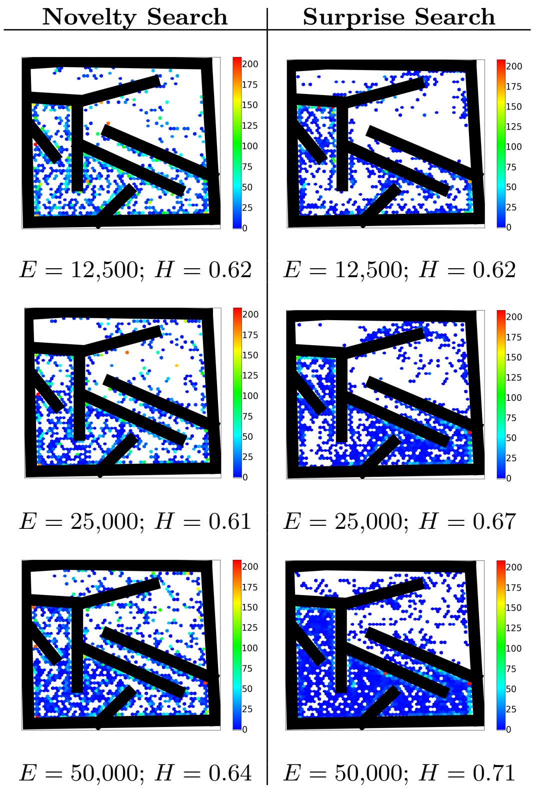
To get further insight on the behavior of surprise search and its strengths over novelty search in the more deceptive problem (i.e. hard maze), we first discuss a number of typical examples in the behavioral space and then present some statistics derived from the genotypic space. Objective search is not further analyzed in this section given its evident disadvantages with respect to both efficiency and robustness.

6.4.1 Behavioral Space: Typical Examples

Table 1 shows pairs of typical runs (for novelty and surprise search) in the hard maze which are solved after 12,500, 25,000, and 50,000 evaluations. Typical runs are shown as heatmaps which represent the aggregated distribution of the robots' final positions throughout all evaluations. Moreover, we report the entropy (H) of those positions as a measure of the population' spatial diversity in the maze. The heatmaps illustrate that surprise search results in a more sparse distribution of final robot positions. The corresponding H values indicate that surprise search explores the maze better than novelty search which is more evident in the longest runs. While surprise search is built on the principles of novelty search, it augments it by exploring the prediction space, which means that it considers the predicted behaviors as the points it needs to deviate from. This allows surprise search to explore the space in an orthogonal way to novelty search, and to diverge not from current and previously found behaviors (in the current population and novel archive, respectively) but from expected behaviors (which may not have been exhibited, or may never will). In the case of the deceptive hard maze, this deviation from predicted behaviors shows a better exploratory capability.

6.4.2 Genotypic Space

Table 2 contains a set of metrics that characterize the final ANNs evolved by surprise and novelty search, which quantify aspects of genomic *complexity* and genomic *diversity*. For genomic complexity we consider the number of connections and the number of hidden nodes of the final



$E = 12{,}500$; $H = 0.62$ | $E = 12{,}500$; $H = 0.62$

$E = 25{,}000$; $H = 0.61$ | $E = 25{,}000$; $H = 0.67$

$E = 50{,}000$; $H = 0.64$ | $E = 50{,}000$; $H = 0.71$

Table 1: **Behavioral Space.** Three typical successful runs solved after 12,500, 25,000 and 50,000 evaluations (E) obtained by novelty search and surprise search. Heatmaps illustrate the aggregated numbers of final robot positions across all evaluations. Note that white space in the maze indicates that no robot visited that position. The entropy ($H \in [0, 1]$) of visited positions is also reported and is calculated as follows: $H = (1/logC)\sum_i\{(v_i/V)log(v_i/V)\}$; where v_i is the number of robot visits in a position i, V is the total number of visits and C is the total number of discretized positions (cells) considered in the maze.

Algorithm	**Complexity**		**Diversity**
	Connections	Nodes	Compatibility
Novelty	29.75 (8.4)	2.30 (1.1)	34.35 (12.2)
Surprise	44.74 (20.4)	3.32 (2.1)	58.30 (29.0)

Table 2: **Genotypic Space**. Metrics of genomic complexity and diversity of the final ANNs evolved using NEAT, averaged across successful runs. Values in parentheses denote standard deviations.

ANNs evolved, while genomic diversity is measured by the compatibility metric defined in [28]. Surprise search generates significantly more densely connected ANNs than novelty search and it also evolves, on average, significantly larger ANNs (in terms of hidden nodes) than novelty search. Most importantly, surprise search yields a significantly higher population diversity, expressed by the compatibility metric [14], than novelty search.

7. DISCUSSION AND FUTURE WORK

The key findings of this paper suggest that surprise search yields comparable efficiency to novelty search and it outperforms objective search. Moreover it finds the solution faster and more often than any other algorithm considered. The comparative advantages of surprise search over novelty are inherent to the way the algorithm searches, attempting to deviate from predicted *unseen* behaviors instead of prior *seen* behaviors. The difference between the two algorithms is manifested in both the behavioral and the genotypic space. Surprise search is more exploratory than novelty search in the deceptive mazes examined as it leads to higher spatial diversity. Spatial exploration in surprise search increases over time, gradually increasing the search capacity of the algorithm. Furthermore surprise search yields larger and denser ANN controllers while diversifying the population more than novelty search. In summary, the combination of higher population diversity, ANN connectivity and exploratory capacity seems beneficial for surprise over novelty search.

The comparative analysis of surprise search against the two random search variants investigated suggests that surprise search is not random search. Clearly it outperforms random search in efficiency and robustness. Further, the poor performance of the surprise search variant with random prediction points suggests that the prediction of expected behavior is beneficial for divergent search.

While this study already offers evidence for the advantages of surprise as a form of divergent search, further work needs to be performed. We need to further test the algorithm's potential within the maze navigation domain (through more deceptive and complex environments) and in other domains such as robot locomotion or procedural content generation. In terms of the predictive model of expected behavior, this paper uses a simple model of 1-step predictions via linear regression; we can only envision that better results can be achieved if machine learned or non-linear predicted models are built on more prior information. In terms of deviation from predicted points, we use the distance from the closest prediction as the surprise score, but more nearest neighbors can be considered in future work; moreover, deviation from predictions can use non-linear and probabilistic methods for a deviation function (as e.g. in [11]). Additionally, surprise search allows for variant degrees of *prediction locality*, i.e. the amount of local information considered by surprise search to make a prediction (k in this paper). Prediction locality can be derived from the behavioral space (as in this paper) but also on the genotypic space. Future work will need to focus on the effect of locality on surprise search. Finally, surprise search as presented here requires some form of clustering of behaviors: while k-means was employed for its simplicity and popularity, any clustering algorithm is applicable and comparative studies between approaches can be conducted.

8. CONCLUSIONS

This paper introduced the notion of surprise for search, provided a general algorithm that follows the principles of searching for surprise and tested the idea in a maze navigation problem. Evidently, surprise search shows advantages over other forms of evolutionary divergent search such as novelty search and outperforms traditional fitness-based evolution (i.e. objective search) in deceptive problems. While it yields comparable efficiency to novelty search, it tends to find solutions faster and more often. Detailed analysis on the behavioral and genomic properties of surprise search showcase that deviation from the expected behavior in the search space results in higher exploratory capacity and behavior diversity. These properties, in turn, appear to be the key benefits of surprise over novelty or objective search.

9. ACKNOWLEDGMENTS

This work has been supported in part by the FP7 Marie Curie CIG project AutoGameDesign (project no: 630665). The authors would also like to thank Dora Lee Borg for initial implementations of the algorithm.

10. REFERENCES

[1] C. Adami, C. Ofria, and T. C. Collier. Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97(9), 2000.

[2] P. J. Angeline and J. B. Pollack. Competitive environments evolve better solutions for complex tasks. In *Proceedings of the International Conference on Genetic Algorithms*, 1994.

[3] M. A. Boden. *The Creative Mind: Myths and Mechanisms*. Routledge, 2004.

[4] A. Channon. Passing the alife test: Activity statistics classify evolution in geb as unbounded. In *Advances in Artificial Life*. Springer, 2001.

[5] Y. Davidor. Epistasis variance: A viewpoint on ga-hardness. In *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991.

[6] P. Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4), 1992.

[7] S. Ficici and J. B. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In *Proceedings of the International Conference on Artificial Life*, 1998.

[8] D. E. Goldberg. Simple genetic algorithms and the minimal deceptive problem. In *Genetic Algorithms and Simulated Annealing, Research Notes in Artificial Intelligence*. Morgan Kaufmann, 1987.

[9] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2), 1988.

[10] K. Grace and M. L. Maher. Specific curiosity as a cause and consequence of transformational creativity. *Proceedings of the International Conference on Computational Creativity June*, 2015.

[11] K. Grace, M. L. Maher, D. Fisher, and K. Brady. Modeling expectation for evaluating surprise in design creativity. In *Design Computing and Cognition*. 2014.

[12] F. Kaplan and V. V. Hafner. Information-theoretic framework for unsupervised activity classification. *Advanced Robotics*, 20(10), 2006.

[13] S. A. Kauffman. Adaptation on rugged fitness landscapes. In *Lectures in the Sciences of Complexity*. Addison-Wesley, 1989.

[14] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2), 2011.

[15] J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2011.

[16] J. Lehman and K. O. Stanley. Beyond open-endedness: Quantifying impressiveness. In *Proceedings of the International Conference on Artificial Life*, 2012.

[17] J. Lehman, K. O. Stanley, and R. Miikkulainen. Effective diversity maintenance in deceptive domains. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2013.

[18] A. Liapis, H. P. Martínez, J. Togelius, and G. N. Yannakakis. Transforming exploratory creativity with DeLeNoX. In *Proceedings of the International Conference on Computational Creativity*, 2013.

[19] G. E. Liepins and M. D. Vose. Representational issues in genetic optimization. *Journal of Experimental and Theoretical Artificial Intelligence*, 2(101), 1990.

[20] E. Lorini and C. Castelfranchi. The cognitive structure of surprise: looking for basic principles. *Topoi*, 26(1), 2007.

[21] L. Macedo and A. Cardoso. Modeling forms of surprise in an artificial agent. In *Proceedings of the nnual Conference of the Cognitive Science Society*, 2001.

[22] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[23] A. Ortony and D. Partridge. Surprisingness and expectation failure: what's the difference? In *Proceedings of the Joint conference on Artificial intelligence*, 1987.

[24] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2), 2007.

[25] C. R. Reeves. Fitness landscapes. In *Search Methodologies*. Springer, 2005.

[26] G. Ritchie. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, 17(1), 2007.

[27] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3), 2010.

[28] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2), 2002.

[29] T. Weise, R. Chiong, and K. Tang. Evolutionary optimization: Pitfalls and booby traps. *Journal of Computer Science and Technology*, 27(5):907–936, 2012.

[30] S. Wessing, M. Preuss, and G. Rudolph. Niching by multiobjectivization with neighbor information: Trade-offs and benefits. In *Proceedings of the Evolutionary Computation Congress*, 2013.

[31] L. D. Whitley. Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991.

[32] G. A. Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, 19(7), 2006.

[33] L. Yaeger. Poly world: Life in a new context. *Proc. Artificial Life*, 3, 1994.

[34] G. N. Yannakakis, J. Levine, J. Hallam, and M. Papageorgiou. Performance, robustness and effort cost comparison of machine learning mechanisms in flatland. *Proceedings of the MED*, 2003.