Preliminary Analysis of Simple Novelty Search

R. Paul Wiegand wiegandrp@winthrop.edu

Department of Computer Science & Quantitative Methods

Winthrop University, Rock Hill, SC, 29733, United States

Abstract

Novelty search is a powerful tool for finding diverse sets of objects in complicated spaces. Recent experiments on simplified versions of novelty search introduce the idea that novelty search happens at the level of the archive space, rather than individual points. The sparseness measure and archive update criterion create a process that is driven by a two measures: 1) spread out to cover the space while trying to remain as efficiently packed as possible, and 2) metrics inspired by k Nearest Neighbor theory.

In this paper, we generalize previous simplifications of novelty search to include traditional population (μ, λ) dynamics for generating new search points, where the population and the archive are updated separately. We provide some theoretical guidance regarding balancing mutation and sparseness criteria and introduce the concept of *saturation* as a way of talking about fully covered spaces. We show empirically that claims that novelty search is *inherently* objectiveless are incorrect. We leverage the understanding of novelty search as an optimizer of archive coverage suggest several ways to improve the search, and we demonstrate one simple improvement — generate some new points directly from the archive rather than the parent population.

Keywords

novelty search, quality diversity, convergence, k nearest neighbors.

1 Introduction

Though evolutionary computation can be an effective problem solver (De Jong, 2006), these biologically-inspired approaches are also applied to tasks other than traditional optimization. Evolutionary algorithms (EAs) are now used for a variety of creative endeavors from evolving computer programs (Langdon and Poli, 2002) to discovering faults (Hanes and Wiegand, 2019).

Quality diversity algorithms (Fontaine and Nikolaidis, 2021; Choi and Togelius, 2021; Fontaine et al., 2019; Nordmoen et al., 2020; Cully and Demiris, 2017; Mouret and Clune, 2015) use the innovating mechanisms in evolutionary methods to try to saturate a space by spreading out to find a variety of solutions. One well-known method is so-called, *Novelty Search* (Stanley and Lehman, 2015; Lehman et al., 2013; Lehman and Stanley, 2011b, 2008), which attempts to produce *different kinds* of results (in general), rather than results that optimize an external objective function. The motivation behind novelty search is that for certain complex spaces, it may be better to *ignore* the optimization objective and instead *explore* the space by finding objects that are increasingly "different" from those that the search process has encountered before. Though perhaps counter intuitive, novelty search has been surprisingly effective at finding good solutions to a number of challenging problems without even looking for those solutions explicitly. Examples of novelty search successes include discovering effective grasping behaviors for highly articulated robotic arms (Huang et al., 2014), unsupervised feature learning for deep networks (Szerlip et al., 2015), discovery of sophisticated gaits for quadruped locomotion (Morse et al., 2013), and goal-seeking in complex multiagent simulations (Lehman and Stanley, 2011a).

Advocates of these methods have proposed three critical ideas for *why* novelty search can succeed in such spaces. The first premise is that novelty search avoids being

deceived by local optima since it ignores an external objective in favor of a novelty metric (Lehman and Stanley, 2011a). This is predicated on the claim that novelty search "*diverges*" rather than converges Lehman et al. (2016); Lehman and Miikkulainen (2015); Stanley and Lehman (2015). Second, novelty search tends to be applied when using generative representations (Lehman and Stanley, 2008) to encourage a process referred to in the literature as "complexification" (Stanley and Miikkulainen, 2004). Complexification is the systematic and steady increase of complex representations via evolutionary search, and this in conjunction with a divergent search is expected to lead to surprising and nuanced discoveries. Third, using diversity measures within the true solution space (e.g., agent behaviors) rather than the genotype space allows novelty search to discover "interesting" areas of that behavior space.

While sometimes effective, there's little foundational understanding for whether or how these speculations (or others) lead to success nor much in the way of guidance for how to *improve* novelty search. Lehman and Stanley (2013) provided empirical evidence that even simple models of evolution can increase so-called "evolvability" without any explicit pressure to do so by essentially *filtering out* things that change more slowly. This is relevant to novelty search since it provides some experimental justification for the idea that novel solutions can be discovered through an evolutionary process without the external optimization objective being encoded into the algorithm. Very recently Doncieux et al. (2020, 2019) have presented a reinforcement learning based formal framework for approaching questions about novelty search. They use their formalism to suggest hypotheses, which they test empirically and conclude that novelty search mechanisms can be effective at spreading out in behavior space, suggesting that they may be capable of "uniform sampling" of that space. A slightly older empirical study of novelty search methods by Gomes et al. (2015) examines several key

parameters of novelty search including the types of archives used (none, novelty, and random), size of the neighborhood used for sparseness calculations, several methods for how fitness and novelty metrics are used during selection, and several variations for the underlying evolutionary algorithm used. This work included complex algorithms on a maze navigation task, and while it *does* provide real-world advice about how mutation can relate to these choices, it doesn't really address the core behaviors of the algorithm nor whether or not fundamental claims about the algorithm are true.

Quality diversity (QD) methods include different approaches than novelty search. One well-established alternative are so-called *MAP-Elite* methods (Choi and Togelius, 2021; Nordmoen et al., 2020; Flageat and Cully, 2020; Mouret and Clune, 2015). There are two relevant differences between canonical novelty search and MAP-Elite methods. The first is that unlike traditional novelty search, MAP-Elites *does* use the external objective function to make selection decisions. Diversity measures are used to make sure the population / archive spreads out in the space. The second is that in most applications of MAP-Elites, the population itself is the archive unlike typical applications of novelty search. Part of our analysis in Section 4.2 will address this point.

There are many other types of QD methods, as well. Some rely on the use of second order information about the problem (e.g., derivatives or gradients) (Fontaine and Nikolaidis, 2021), while others represent diversity maintenance mechanisms in traditional evolutionary algorithms, such as crowding (Goldberg and Richardson, 1987). Cully and Demiris (2017) attempt to provide a general framework to place most, or all of QD methods. Chiefly, there appears to be a substantive difference in philosophy between approaches in which the external fitness objective is used to guide search dynamics (such as MAP-Elites) and those in which the external fitness objective is ignored for the purposes of search behavior (such as traditional novelty search). Our focus is

on the latter group of methods. Are the general claims about novelty search true?

In this paper, we begin at the top by simplifying the algorithm and problem domain then looking more closely at the claim that novelty search has no objective, and that it is a "divergent" search process (Lehman et al., 2016; Lehman and Miikkulainen, 2015; Stanley and Lehman, 2015). This paper isolates the claim as to whether novelty search is inherently objectiveless, and we say nothing at all about complexification or discovery in behavioral spaces. Moreover, our results do not in any way invalidate the success of novelty search, rather we seek to complement existing research by providing more insight into how these searchers work and how to make principled choices regarding some search parameters. Our analysis is specific to the original *Novelty Search* algorithm and not newer variants; however, the overall goal is to build *better* novelty search algorithms.

We propose an alternative view of novelty search where the space being searched is not the solution space (e.g., behaviors) but is rather a space of *archives* of potential solutions. We take inspiration from k nearest neighbor theory to provide a formalization of this idea, conduct empirical analyses using measures based on that formalization, and provide a preliminary theoretical framework for understanding how such a search process works. Our analysis begins with an extremely simple formulation of novelty search that consists only of the so-called *sparseness* criterion and an archive, and we then relax this by adding a population separate from the archive. Some preliminary empirical results of this work appear in Wiegand (2020) and Wiegand (2021), though this article expands on these empirical results and also adds some theoretical guidance. All experiments in this paper are unique to the present work, though a few replicate experiments from those previous works. We will be clear where there are overlaps.

We borrow and revise well-known metrics from k-NN theory (Clarkson, 1999) to

define ϵ-coverage and ϵ-packing over sets (See Section 2.5 for the formal definition). We find that novelty search drives the archive toward increasingly better ϵ-covers of the solution space while also trying to optimize the ϵ-packing of the archive. Simple variants of novelty search that employ the traditional mechanisms can and do converge, whether or not the space is bounded. Moreover, local convergence to a suboptimal archive is possible. Understanding this allows us to give some constructive suggestions for how to apply novelty search more effectively. First, it is essential to balance the mutation rate and the minimum sparseness criterion appropriately. Second, there may be value in selecting some parents directly from the archive, and it may be useful to use convex hull estimates of the archive from which to draw future search points.

In part, much of what any quality diversity method does can be seen through the lens of random sampling theory. Standard evolutionary algorithms use initialization and genetic operators such as mutation to sample *different* parts of the space (De Jong, 2006), and Monte Carlo sampling methods are often used to reduce uncertainty. Indeed, traditional reinforcement learning methods have been successfully combined with stochastic local search mechanisms by leveraging the fact that these local search methods provide different kinds of sampling of the local space (i.e., a kind of novelty) (Boyan and Moore, 1998). Moreover, depending on the *way* novelty search is applied to a problem, one can see the archive update mechanism as a way to bias sampling to spread out in a space as a means of increasing the likelihood of getting "close" to whatever types of solutions one is trying to find. In that sense, novelty search can be seen as a specialized form of sampling, the goal of which is to reduce the uncertainty of finding or characterizing particular parts of a space (Grinstead and Snell, 2003; Sfikas et al., 2021). This also aligns well with the view point about evolvability discussed in Lehman and Stanley (2013). Our measure of *coverage* defined in Section 2.5 provides

some insight into this view since it can be seen as a rough metaphor for something like confidence interval: The better a space is covered, the more confident of sampling near *some* target, whatever is desired by the engineer applying novelty search.

2 Technical Approach

2.1 Sparseness and Archives

The concept of "novelty" in traditional novelty search is enforced by a metric designed to measure the *sparseness score* of a point with respect to a group of points. Specifically, the sparseness of some candidate object y, ρ_y, is computed as the average distance to the k nearest neighbors in some set A, $\rho_y := \frac{1}{k}\sum_{i=1}^{k} \delta\left(x_i^A, y\right)$, where x_i^A is the i^{th} closest point in set A to y and δ is some kind of distance calculation. In other words, ρ_y is the average distance between some point y and the k nearest neighbors to that point in some set A. Though perhaps a misnomer, we keep the term sparseness score. For traditional novelty search, sparseness is used in two ways.

First the sparseness score is used as a type of fitness value for evolution. Points that have larger scores — more distant from nearby points in some set — are more fit. For selection, this set is typically the population, and it can be used for parent and/or survival selection. It is also possible to explore the idea of using the archive as the baseline set for selection, and we explore this idea in Section 4.2.

The second way the score is used is as an update mechanism for an archive of points the search is maintaining: Novelty search adds individuals to the archive only if their sparseness over that archive is above a certain threshold, ρ_{min}. In this way, the archive grows and begins to contain more and more "novel" candidate solutions in the sense that they are *different* from one another as defined by the sparseness score.

Though there are many choices for search operators and selection, the rest of the

method is essentially an evolutionary algorithm (De Jong, 2006). The idea that the archive continues to expand with novel search points as search proceeds is what motivates the claim that novelty search diverges. To be clear, this paper takes no position as to whether or not divergence in novelty search is an appropriate goal; we merely explore an existing claim to determine whether or not it is generally true that novelty search inherently diverges rather than converges. It is very clear that the original researchers who composed novelty search intended the search to diverge (Lehman et al., 2016; Lehman and Miikkulainen, 2015; Stanley and Lehman, 2015).

2.2 Archive-Based Searches

In traditional uses of an evolutionary algorithm, individuals in the population represent candidate solutions to a problem. The search space is the set of candidate solutions to the problem, population members represent potential solutions, and the EA searches that space to find a good solution. Given this perspective, it is easy to understand why novelty search *appears* to be objectiveless: though individuals may be encoding candidate solutions, the algorithms completely ignore the objective function associated with traditional optimization. Imagine search diverging in the sense that increasingly different candidate solutions are progressively proposed in an ever-growing archive, and the direction of that growth is not at all influenced by the problem's objective.

However, there are alternative ways to understand archive-based searches. Specifically, some search methods are designed to search a *space of archives*, not individual points. The multiobjective optimization problem is typically formulated to find an approximation of the Pareto non-dominating *set* for some solution space (Zitzler, 2012). Further, *cooptimization* problems, to which coevolutionary algorithms are often applied, are supersets of multiobjective optimization problems. In cooptimization, the goal is

to simultaneously identify the set of underlying objectives and produce a set obeying some solution concept (Ficici and Pollack, 2001; Ficici, 2008). That solution concept may be Pareto-based, but there are other concepts (e.g., sets that approximate mixed Nash equilibria).

We take the position that novelty search is, in fact, *not* searching the space of candidate problem solutions being represented by individuals in the population. Instead, it is searching the space of *novel archives*. The dynamics of this search are such that it is driven to develop an archive that spreads out in some space while also keeping the points in that archive not too close together. These two notions can be formalized by using slightly modified versions of the concepts of *coverage* and *packing* from k nearest neighbor (k-NN) theory. In Section 13 we will define these terms more formally and describe some heuristic approximations for measuring these in empirical studies.

In this article, we will consider two variations of novelty search: *simple novelty search* and *population-based simple novelty search*. For the latter, we will consider both when population selection uses the population for the baseline sparseness measure, as well as when it uses the archive. In the former, the population and the archive is the same. We will also provide some simple formalisms regarding simple novelty search in bounded Hamming spaces.

2.3 The Simple Novelty Search EA

We start with a minimalist understanding of novelty search that has only three major components: a genetic operator, an archive, and an update rule. While the archive is a set of points, this minimalist algorithm cannot be said to have a population *per se*. Or, perhaps more precisely, the population *is* the archive (and vice-versa). The purpose of considering this minimalist approach is to understand the fundamental role of the

archive's update mechanism. Informally, the algorithm works as follows. A parent is selected uniformly at random from within the archive itself, then duplicated (copied), and then mutated to produce a child. The child's sparseness is measured against the archive. If that value is greater than some selected ρ_{min}, then the child is added to the archive. This is repeated until some termination criterion is met. The pseudocode for this is replicated below. This code first appeared in Wiegand (2020), though Cully and Demiris (2017) also discuss methods in which the population is the archive.

Algorithm 1: Simple Novelty Search Evolutionary Algorithm (SNSEA)

input : $\rho_{min}, k,$ termination criterion

output: $archive$

initialize individual x

$archive$ = $\{x\}$

while *not reached termination condition* **do**

  draw $parent$ uniformly at random from $archive$

  $child$ = mutate(copy of $parent$)

  $\rho =$ sparseness$(child, archive, k)$

  if $\rho \geq \rho_{min}$ **then**

    $archive = archive \cup \{child\}$

  end

end

For this article, we consider $k = 3$ for the sparseness computation. The termination criterion is simply a maximum number of generations (500 in most cases). We will demonstrate the algorithm in several spaces: a discrete binary space, a bounded real

valued space, and an unbounded real valued space.

In the case of the discrete space experiments, the simple novelty search evolutionary algorithm (SNSEA) will use a binary representation where individual points exist in the space $\{0,1\}^n$, where n is the length of the binary string. Mutation is performed by independently flipping each bit with probability $1/n$. Since there is no external problem-oriented objective, without loss of generality we initialize the first point at 0^n. Distance is computed in this space using Hamming distance, thus we refer to it as a *Hamming Space*. For experiments in this paper, we consider $n \in \{6, 8, 10, 12, 14\}$.

In the case of the Euclidean space experiments, the SNSEA will use a real valued representation where the individual points exist in the space $\mathbb{R}^d$, where d is the dimension of the space. Mutation is performed by independently adding an offset to every value in the vector according to $N(0, \sigma)$ (fixed σ Gaussian mutation). Again, without loss of generality we initialize the first point at 0^d. Distance is computed in this space using the L_2 norm Euclidean distance, thus we refer to it as a *Euclidean Space*. In the experiment where the space is bounded, we restrict mutation so that it cannot produce child gene values outside of $[0, 1]$. In those experiments, this rule is enforced by redrawing the mutation offset until the mutated gene is inside this region. For these experiments, we consider $d = 5$

It is worth noting that conflating the population and the archive is a substantive change to the way novelty search typically works. Another difference is that all our distance calculations are in the genotype space; however, this is not an important departure since we are only evaluating the claim about whether an objective exists, not what it looks like for a given problem instance. That is: If an objective exists in a genotype space and/or if there is a way to characterize *some* novelty searches as "converging" then the *general* claims that novelty search *must* diverge and *doesn't* have an objective

are false. Note again that we are not arguing that novelty search *cannot* diverge in some cases, nor that novelty search is ineffective. We are just trying to understand the nature of archives, updates, and the type of space novelty search EAs traverse. For this, a genotypic representation is sufficient.

2.4 The Population-Based Simple Novelty Search Evolutionary Algorithm

Perhaps one possible argument for why novelty search *should* and typically *does* separate the archive and the population is the clean separation of roles: a population contains the current state of the exploration aspects of the search, while the archive contains the potential solution set at that moment of the search. To explore this role separation, we also implemented a population-based version of the algorithm. Our population-based simple novelty search EA works as follows. Each generation, λ children are produced from the μ parents, and the children are added to the archive if they meet the sparseness criterion as computed over the archive. Additionally, sparseness of the children with respect to the set is computed, and that the μ individuals with the highest sparseness values are selected to be in the parent set for the next generation. This is repeated until some termination criteria is met. This algorithm represents a slight generalization of the one presented in Wiegand (2021).

Again, we consider $k = 3$ for the sparseness computation. In most cases, the termination criterion is simple a maximum number of generations. We will consider a real-valued representations, where again $d = 5$ dimensions. Mutation works by independently adding a value drawn from $\mathcal{N}(0, \sigma)$. This allows the possibility for the archive to grow in an unbounded way by adding individuals that are "novel" in the sense that they are quite distant from individuals seen before. Again, distance is computed using the L_2 norm, Euclidean distance.

For reasons we clarify when we discuss the results of the SNSEA studies, we do not consider unbounded spaces for the population-based simple novelty search evolutionary algorithm (PSNSEA). As before, distances are in the genotype space.

Experiments were replicated for 50 independent trials and run to 500 generations. All experiments in this paper use $\sigma \in \{0.1, 0.2, 0.3\}$ and $\rho_{min} \in \{0.2, 0.4, 0.6\}$.

2.5 Packing and Coverage

In order to find bounds on how efficiently distance-based, lazy-evaluation methods like k nearest neighbor algorithms are able to develop hypotheses for a given space, the theory community for that field have developed several formal definitions (Clarkson, 1999). These ideas are useful here because novelty search's sparseness metric is explicitly based on k nearest neighbor calculations. Our formulations below rely on the observation that both the Hamming and Euclidean space distance measures we've discussed imply the spaces we are considering are *metric spaces* (Bryant, 1985).

The first concept we consider is the ϵ_c-coverage of a set. Coverage tries to address the question of how well a set spreads out over some space. A set that *covers* a space well is one that is spread out over that space such that no point in the space is too far from at least one point in the set.

Definition 1 *An* $\boldsymbol{\epsilon_c}$***-coverage*** *of some space* $Z = \langle U, \delta \rangle$*, where* $\delta : U \times U \mapsto \mathbb{R}$ *is a distance measure over* U*, is a set* $A \subset U$ *such that* $\forall x \in U, \exists a \in A$ *with* $\delta(x, a) \leq \epsilon_c$*.*

The second concept we consider is ϵ_p-packing of a set: Packing tries to address the question of how efficiently a set of points represents a space. A set that *packs* a space well is one in which points inside the set aren't too close together

Definition 2 *Given the space* $Z = \langle U, \delta \rangle$*, where* $\delta : U \times U \mapsto \mathbb{R}$ *is a distance measure over* U*, a set* $A \subset U$ *is an* $\boldsymbol{\epsilon_p}$***-packing*** *iff* $\delta(a, b) \geq 2\epsilon_p \ \forall a, b \in A$*.*

In the case of coverage, smaller is better (points in U are closer to points in A). That is, we are really examining how far our representative points are from other parts of the space — the smaller that distance, the better our representation. However, for packing, larger is better (points in A are further apart). Here we are examining efficiency by seeing how crowded our representatives are — the further apart the are, the more efficient. These obviously trade off of one another, and where coverage and packing meet, we have a *net*.

Definition 3 *An* ***ϵ-net*** *$A \subset U$ is a set that is an ϵ-coverage of U and an $(\epsilon/2)$-packing.*

Computing packing is straightforward even in an empirical setting: half the maximum pairwise distance between every point in the archive. Packing in novelty search, thus, starts small and grows as points are added to the archive.

To directly compute the coverage is computationally infeasible, so we must approximate it. We assume that the search will (with high probability) remain within the bound $\pm\sigma \cdot maxGen$ in all dimensions. This was confirmed experimentally: in all runs of all experimental groups, no points were generated outside that region. Second, we sample points uniformly at random from inside that bounded hypercube. Finally, we find the closest point in the archive to each sample point. The maximum of these distances is reported as our estimate for ϵ-coverage. Coverage estimates in novelty search will tend to start large, then drop as it becomes increasingly less probable that the algorithm will select new points that fill in the gaps of the space searched so far.

3 The SNSEA Optimization Process

The SNSEA does not separate the population from the archive as most applications novelty search do; however, as stated above another perspective is that the archive *is* the population. When seen this way, the SNSEA is analogous to a $(\mu + 1)$-EA (De

Jong, 2006), where parent selection occurs randomly from the population (archive) and truncation survival selection is being used. Of course, it differs from this in that the fitness of an individual depends on the current population (archive), which grows. This is an important distinction since an individual discovered early in the search process will typically have a different sparseness measure than that same individual discovered later after the archive has developed more.

The algorithm has selective pressure and progresses in a particular direction; it is not random search. Generally, the algorithm gradually grows the size of the archive, adding new points as they are discovered. So, as a measure of archive size, the algorithm is monotonically non-decreasing.

Packing in the population is also monotonically non-decreasing since the furthest pair of the points in the archive always remain in the archive. Moreover, it can (and does) increase since new points added to the archive can create a new distance pair in the archive that is even larger.

The true coverage of the archive is monotonically non-increasing. Again, the furthest distance from any point in the space to its closest archive point cannot grow any larger since the archive does not lose points. True coverage *can* decrease because the SNSEA can add a new point that is closer to the furthest point in the space. Our approximation metric estimates coverage by sampling, so it will tend to decrease (on average).

So, the SNSEA will steadily increase packing and archive size, and it will tend to decrease coverage. These are clear and well-defined objectives that are being optimized. Since the SNSEA is not guaranteed to optimally pack an archive (the points may not necessarily be spaced efficiently), it can overshoot the ϵ-net in terms of packing packing. However, as well will see, there is a natural point in bounded spaces that will limit the coverage.

3.1 Saturation of Bounded Spaces

For bounded spaces, the SNSEA archive will eventually *saturate* the space. That is, the space will be fully covered and packed and no more points can be admitted.

Let's start with a simple example to build intuition for this. The SNSEA in Hamming space will add new points to the archive until there are no points left in $\{0,1\}^n$ that can be added that will meet the sparseness criterion. It's easy to see this for a small example, where $n = 3$ and $\rho_{min} = 2$: start with the 000, then adding the 111 point, now all other points are at most Hamming distance 2 from one of these two points, so no new points can be added. This is true for all Hamming spaces for $k > 1$: eventually there will be an archive that fully covers the space with $\epsilon \sim \rho_{min}$.

Actually, it is generally true that the SNSEA on any bounded metric space will eventually saturate the space, as long as mutation is ergotic.

Theorem 1 *The SNSEA operating in a bounded space will converge in the sense that improvements in* coverage *will eventually stop or asymptote toward a fixed value.*

Proof: We start with the observation that an archive, A, can be seen as a set of centroids that define a Vornoi tessellation over a bounded space, U. The maximum distance between the centers of two Vornoi regions is not more than the sum of the maximum distance of points in the first region to its centroid and the maximum distance of points in the adjacent region to its centroid.

Let's consider the situation where we are adding an external point $\hat{x}$ to an archive that forms an ϵ-coverage and *at most* an ϵ-packing. That is, the maximum distance between any pair of points in the archive is at most 2ϵ and the maximum distance between any point in the space and its closest archive point is at most ϵ. Then the worst case situation is when the Vornoi regions are stacked in such a way that the second

closest archive point is directly behind the first, and so on. So the distance between $\hat{x}$ and the closest archive point is at most ϵ, the distance to the next closest point is at most $\epsilon + 2\epsilon$, and the series continues in this way.

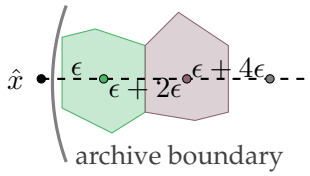

Figure 1: Vornoi illustration of worst-case archive layout for adding point $\hat{x}$.

The sparseness of this point with respect to the $A'_{\hat{x}} \subset A$ closest archive points is:

$$\rho_{\hat{x}} \quad = \quad \frac{1}{k} \sum_{a \in A'_{\hat{x}}} \delta(\hat{x}, a) \leq \frac{1}{k} \cdot \left(\epsilon + \sum_{j=1}^{k-1} (\epsilon + 2j\epsilon) \right) = \frac{\epsilon + \epsilon \left(k^2 + 1\right)}{k} = k\epsilon$$

There are two situations to consider. The first case is when the coverage of the algorithm during run time reaches a point small enough that $k\epsilon < \rho_{min}$. In which case, no new points in U can be added to the archive and the coverage measure will no longer improve. In the second case, the coverage never gets small enough for this to occur, which by definition implies that the algorithm is converging on a limit point for ϵ. In either event, the SNSEA in bounded spaces converges and saturates the space. ■

As straightforward as this is, it is also instructive. In a perhaps unsatisfying way, this is sufficient to dismiss the claim that novelty search *always* diverges. Simple novelty search in bounded spaces *must* converge in the sense that the archive will saturate and no new points can be added. We will discuss informally why this kind of stalling can still happen in unbounded spaces, below.

Note what we call "saturation" is related to what Doncieux et al. (2019) call "uniform sampling": A bounded space will eventually fill up so that all points are *covered* in some sense. However, there's no evidence that novelty search does this efficiently

nor that it cannot get temporarily focused on one part of the space at the expense of exploring some other part. Indeed, Doncieux et al. (2019) discuss the idea of "reachability" and show that not all parts of the space being explored are equally reachable. Saying that a bounded space will eventually be saturated by an archive is, in principle, no different than the idea that a genetic algorithm working on a pseudo-Boolean problem using ergodic mutation will *eventually* find the solution — but in this case the case, the "solution" is a minimal coverage archive that saturates the bounded space.

Like many MAP-Elite approaches, Fontaine and Nikolaidis (2021) use a gridding formalism for characterizing how quality diversity algorithms explore space (similar to our idea of "coverage") and show that incorporating some problem-specific knowledge (e.g., a differentiable gradient) can greatly improve an algorithm's ability to explore the space efficiently. So it is a mistake to assume that because the archive will eventually cover the space that novelty search cannot be mislead during search or is particularly efficient. Regardless, novelty search clearly converges in such spaces. This seems to be implicitly accepted by Doncieux et al. (2019) since they use the phrase "probability of convergence" to talk about the criteria for when uniform sampling of the space has been reached.

3.2 Mutation & Minimum Sparseness

One important observation to make about the SNSEA is that it is clear that the minimum sparseness criterion and the magnitude of the mutation are deeply related.

We start with some intuition. Consider the first SNSEA step in a Hamming space of dimensionality n and a bit-flip mutation rate of $1/n$. If ρ_{min} is established as a function that grows quickly with n (say $\sqrt{n}$), then we will expect to wait many steps just to see a point far enough away to add to the archive. As the archive expands, the prob-

lem gets worse because we must select a random archive point near the surface of the archive *and* generate a sufficiently far away point. In the Euclidean space, this problem is magnified since the relative ratio of the volume of the interior to the convex hull of the archive increases quickly as the archive grows. The curse of dimensionality leads to further complications with highly dimensional spaces.

On the other hand, if ρ_{min} grows very little (e.g., constant) relative to the dimensionality of the space, then the SNSEA will spread out very slowly in the space — indeed, exponentially slowly as n increases for Hamming space.

Below we formalize this idea for Hamming spaces with two proofs. In the first case, we examine the waiting time until a new point can be admitted into the archive and bound the *expected admission time*, $E[A]$, when ρ_{min} is too large.

Theorem 2 *The expected admission time into the archive for the SNSEA with mutation rate* $1/n$ *operating in a Hamming space where* $\rho_{min} = \sqrt{n}$ *is* $E[A] = \Omega(\sqrt{n}^{\sqrt{n}})$.

Proof: We will consider only the first step of the algorithm where the initial point in the archive is the all zero bit string. In order to admit *any* new point into the archive, it must be at least $\sqrt{n}$ distant from the original point. Though there are many such points, it remains extremely unlikely when the mutation rate is $1/n$.

We use the Hoeffding bound for binomial distributions (Hoeffding, 1963):

$$\Pr\left[\text{binom}\left(n,p\right) \geq k\right] \leq e^{-n\,\mathrm{KL}\left(\frac{k}{n}\,\|\,p\right)},$$

when $k > np$ and where KL is the *Kullback-Leibler divergence*. This provides a bound on the cumulative probability of the binomial random variable deviating too far above the expected value. In our case, we interested in the situation where $k := \sqrt{n}$ and $p := 1/n$.

First, via substitution, we see that:

$$\begin{aligned} n\,\mathrm{KL}\left(\sqrt{n}/n \,\|\, 1/n\right) &= n\left[\left(\frac{\sqrt{n}}{n}\right)\log\frac{\sqrt{n}/n}{1/n} \;+\; \left(1-\frac{\sqrt{n}}{n}\right)\log\frac{1-\sqrt{n}/n}{1-1/n}\right] \\ &= n\left[\left(\frac{\sqrt{n}}{n}\right)\log\sqrt{n} \;+\; \left(\frac{n-\sqrt{n}}{n}\right)\log\frac{n-\sqrt{n}}{n-1}\right] \\ &= \sqrt{n}\log\sqrt{n} \;+\; \left(n-\sqrt{n}\right)\log\frac{n-\sqrt{n}}{n-1} \end{aligned}$$

$$\begin{aligned} \Pr\left[\mathrm{binm}\left(n,\frac{1}{n}\right)\leq\sqrt{n}\right] &\leq e^{-n\,\mathrm{KL}\left(\sqrt{n}/n\,\|\,1/n\right)} = e^{-\sqrt{n}\log\sqrt{n}\,+\,\left(n-\sqrt{n}\right)\log\frac{n-\sqrt{n}}{n-1}} \\ &= \frac{1}{e^{\left(\log\sqrt{n}\right)^{\sqrt{n}}}\cdot e^{\left(\log\frac{n-\sqrt{n}}{n-1}\right)^{n-\sqrt{n}}}} = \frac{1}{\sqrt{n}^{\sqrt{n}}\cdot\left(\frac{n-\sqrt{n}}{n-1}\right)^{n-\sqrt{n}}} \end{aligned}$$

So the expected waiting time for such an event is in $\Omega\left(\sqrt{n}^{\sqrt{n}}\right)$. ■

However, we also consider the opposite extreme when the archive ϵ-covers the space and ϵ is at least some constant fraction of ρ_{min}, $\Omega(\rho_{min})$. In such a case, the closest archive point to any point in the space is still $\Omega(\rho_{min})$, and again the expected waiting time to admit a new point is at least $\Omega(\sqrt{n}^{\sqrt{n}})$. In general, since we know the algorithm will tend to add points at distances on average at least $\sqrt{n}$ away from their k closest archive points, we can expect most waiting times to admit points to be exponential.

In the second case, we examine the waiting time for a specific Hamming level to be saturated, the *expected level saturation time*, $E[L]$, when ρ_{min} is too small.

Theorem 3 *Expected time for SNSEA to saturate the $n/2$ Hamming level with $k > 2$ and mutation rate $1/n$ operating in a Hamming space where $\rho_{min} = O(c)$ is $E[L] = 2^{\Omega(n)}$ steps.*

Proof: The $n/2$ Hamming level contains all the binary strings with precisely $n/2$ bits. There are $2^{n/2}$ such strings. To pack the level, our archive will need $2^{\frac{n}{2c}}$ points. We make the optimistic assumption that on every step the SNSEA flips precisely c bits to produce a child that will definitely be c distant from all neighbors and that our process is able to pack that space perfectly uniformly. Since the SNSEA can only add one point

to the archive in a given step, it will take $2^{\Omega(n)}$ steps to do that. Any relaxation of those assumptions will force the algorithm to take longer. ■

Despite their narrow circumstances, these proofs emphasize the importance of balancing the mutation rate and the sparseness criterion. For Hamming spaces, we suggest setting the mutation rate to $1/n$ and the sparseness criterion to $c \lg n$, where $c \in (0, 1)$. We do not have a recommendation yet for Euclidean spaces, though it is clear that this tradeoff exists there, as well. In that case, there is a strong relationship between σ and ρ_{min}. Nevertheless, it is important that practitioners give careful consideration to these parameters.

3.3 SNSEA Converges in Hamming Space

Empirically, it is easy to construct an example demonstrating saturation. We consider $n = 10$, $\rho_{min} = \frac{3}{4} \lg 10 = 2.491$, and a max generation of 200. We ran 50 independent trials and reported the mean values for our packing and coverage approximations for each generation. For comparison purposes, we also compute the average minimum sparseness for the archive in each generation — that is, we find the minimum sparseness of the archive at each step for each independent trial, and report the average over all trials. This gives us a sense for how the internal SNSEA mechanics will relate to the packing and coverage measure. This in turn provides insight into the convergence properties of the SNSEA.

We can see from Figure 2 that the coverage of the Hamming space decreases steadily and the packing increases steadily. This experiment clearly shows SNSEA converging to an ϵ-net. The internal sparseness measure also levels off with the other two curves. These results are independent but consistent with (Wiegand, 2020).

Now let's see what coverage convergence looks like over different values of n

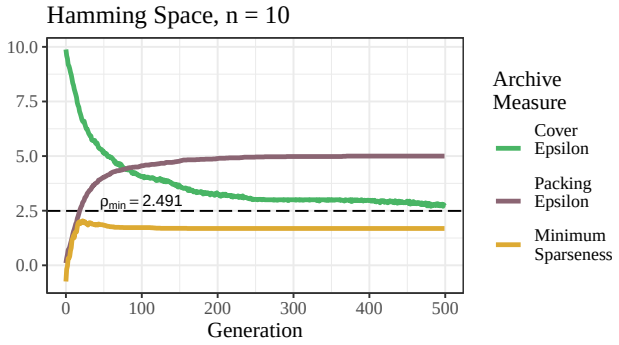

Figure 2: SNSEA applied to a Hamming space with $n = 10$ and $\rho_{min} = 2.491$ averaged over 50 independent trials.

($n \in \{6, 8, 10, 12, 14\}$). Figure 3 below shows that in all cases, the algorithm appears to converge in terms of the coverage epsilon measure.

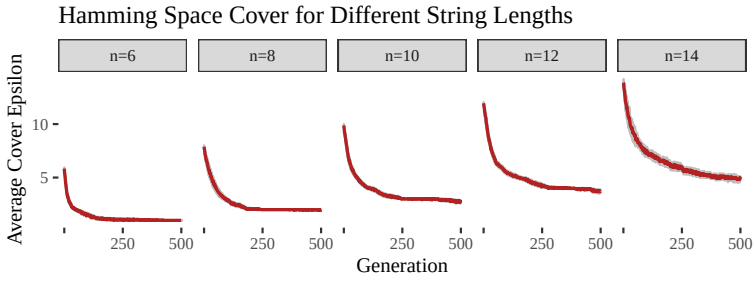

Figure 3: Coverage values for SNSEA applied to a Hamming space with n ($n \in \{6, 8, 10, 12, 14\}$ averaged over 50 independent trials. The 95% CIs are represented as gray bars.

To determine whether these algorithms are indeed converging, we conducted the following experiment. For each trial, we kept track of the lowest coverage value found and insisted that the algorithm must conduct at least 50 generations, and further that it must continue for at least as many generations as it took to first uncover that value without finding a lower value. In other words, if it took 121 generations to find the

lowest coverage, then we continued the algorithm until *at least* generation 242 unless a new coverage value was found. Moreover, we *also* required that no additional points were added to the archive during the last half of the trial.

Our maximum number of generations was set to 5,000. Anytime the coverage value found in a trial was discovered after generation 2,500 *or* any point was added to the archive after that point, we considered that trial to have NOT CONVERGED, otherwise the algorithm clearly CONVERGED since *doubling* the generation count to find a lower value did not help.

In all trials of all values of n for the Hamming space experiments, the SNSEA converged in the sense we just described.

While from the perspective of individual points in the space, one might be led to believe anecdotally that the process is divergent (the archive grows steadily until the space is filled), empirically we can see that the search is occurring at the level of the *archive* space, not the individual point space in the sense that the *dynamics* of the search process are governed by the increasing inability of the algorithm to continue packing and covering. In that sense, the overall dynamical process is clearly convergent — the SNSEA's archive converges in terms of *coverage* in Hamming space. Indeed, progress *slows down* as saturation is approached.

3.4 SNSEA Converges in Bounded Euclidean Space

Again, it is easy to construct an example illustrating that bounded Euclidean spaces can converge. We consider $d = 5$, $\rho_{min} = 0.2$, $\sigma = 0.1$, and a max generation of 500. We ran 50 independent trials and again reported the mean values for our packing, coverage, and min sparseness approximations for each generation.

Figure 4 clearly shows an example where the SNSEA on a bounded 5D Euclidean

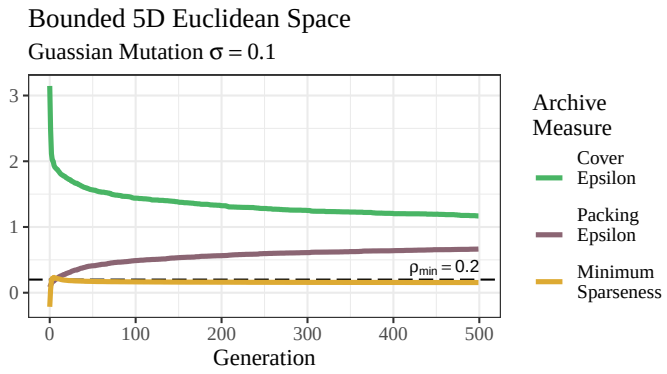

Figure 4: SNSEA applied to a bounded 5D Euclidean space averaged over 50 trials.

space converges *both* in terms of coverage and packing. Indeed, the algorithm appears to be converging toward an archive that approximates an ϵ-net over the bounded space.

Now let's see what coverage convergence looks like over different values of σ and ρ_{min}. Figure 5 below shows that in all but one case, the algorithm appears to converge in terms of the coverage epsilon measure. When $\sigma = 0.1$ and $\rho_{min} = 0.6$ the algorithm never improves at all. This is because the radius of likely changes due to mutation is too small to generate sufficiently different individuals to meet the minimum sparseness conditions. When the ratio between σ and ρ_{min} is around 1:3 or 1:4, the algorithm appears to perform the best in terms of making steady progress in coverage. This again highlights the fact that there is an important balance to be reached between mutation and the sparseness criterion.

Across a wide range of parameterizations of the algorithm in all cases the coverage can be observed to level off eventually (or fail to start). To quantify this, we used the procedure described above: each trial was allowed to continue until it took at least as long after finding the smallest coverage value or adding a new point to the archive. If the algorithm accomplished these both before 2,500 generations, we considered it to

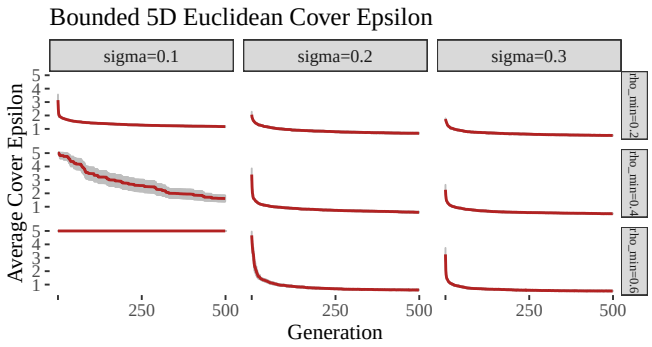

Figure 5: Coverage values for SNSEA applied to a bounded Euclidean space with $\sigma \in \{0.1, 0.2, 0.3\}$ and $\rho_{min} \in \{0.2, 0.4, 0.6\}$ averaged over 50 independent trials. The 95% CIs are represented as gray bars.

have CONVERGED. Figure 6 below shows that in almost all cases, these conditions were met. Even in the handful of cases where it did not meet, there were always more than 1,500 generations of stagnation before we gave up at generation 5,000.

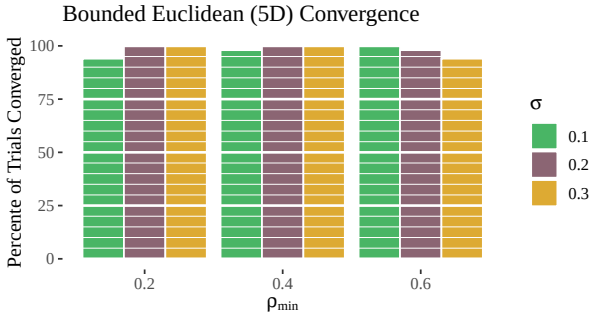

Figure 6: Convergence percentages over 50 trials for the SNSEA operating over different σ and ρ_{min} values on a bounded Euclidean space.

Knowing that the SNSEA converges in bounded spaces is useful! We can implement approximations of coverage and packing in our algorithm, and when we see that an ϵ-net is formed either stop the search or reset the archive to enable the search to

explore new archive configurations.

3.5 SNSEA Can Converge in Unbounded Spaces

Unbounded spaces present a particular challenge to novelty search algorithms (Lehman and Stanley, 2008; Doncieux et al., 2019). Perhaps it is obvious that any bounded space will eventually be saturated by the archive? What the above examples do, though, is illustrate that if your understanding is that novelty search is *diverging until* it runs into the "walls" of the bounded space, then you may be looking at the process incorrectly: The processes are *converging* right from the beginning of the optimization, and the resulting archive is implicitly a local optimum (an ϵ-net).

Still, it is constructive to look at unbounded spaces because we may be able to disentangle some misunderstandings. For one, there appears to be the claim that because the space cannot be saturated, novelty search can have problems (Lehman and Stanley, 2008; Doncieux et al., 2019). But saturation is not the issue. Even in the unbounded case, there are parameterizations where the algorithm will converge in every sense of the word. Figure 7 illustrates such an example, where $\sigma = 0.1$, $\rho_{min} = 0.2$, $d = 5$, and the algorithm is run to 500 generations. The space is completely unbounded. While packing can still be estimated as it was above, coverage cannot be since the space is infinitely large. To address this, we assume that the search will (with high probability) remain within the bound $\pm\sigma \cdot maxGen$ in all dimensions, and we estimate coverage as above but inside that region. We also confirmed that though any 5D point was *possible* in principle, in all runs none were generated outside that region. Again, 50 independent trials were performed.

This process asymptotes toward an ϵ-net just as in the bounded case. Unbounded Euclidean spaces do offer an opportunity that bounded spaces do not: The ability to

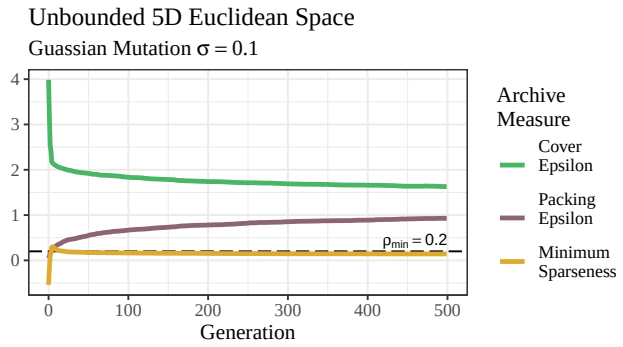

Figure 7: SNSEA applied to an unbounded 5D Euclidean space averaged over 50 independent trials when $\sigma = 0.1$ and $\rho_{min} = 0.2$.

keep adding new points on the *outside* of the known, explored region. Consider the following more specific situation when the mutation σ and ρ_{min} are relatively far apart but ρ_{min} is not so large that no progress can occur. Under such circumstances, we can exacerbate the problem that occurs where large archive sizes make the algorithm increasingly unlikely to select points close enough to the surface of the archive that mutation has a reasonable probability to generate a point that will meet the sparseness criterion. That is, we can create a scenario where new points on the edges of the known, explored space are very occasionally produced, and though they do not fundamentally change the ratio of the space that is *covered* they do increase the largest pair-wise distance of points in the archive.

In Figure 8 we see a situation in which coverage is converging but packing appears to be steadily increasing. Indeed, though coverage appears to converge in most or all parameterizations, some parameterizations will lead to an apparent continual increase in the packing. Though once the ϵ-net is formed, the algorithm continues to occasionally add new points, these points are not substantively improving how the archive *represents* the space — we're not really any closer to potential important points (e.g., a

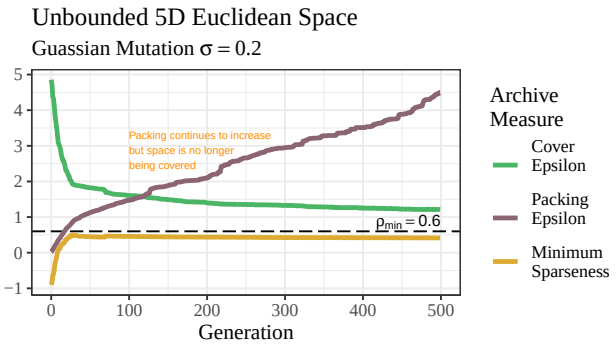

Figure 8: SNSEA applied to an unbounded 5D Euclidean space averaged over 50 independent trials when $\sigma = 0.2$ and $\rho_{min} = 0.6$.

solution). A local optimum has been identified by the algorithm, it has converged into that local optimum, and now it is making small fine-tuning changes to move closer to true locally optimal archive configuration strictly in terms of packing. This is analogous to a traditional real-valued EA maximizing a function like $f(x) = 3x - 2$: once the top of the ridge is found, the EA will continue to produce new x values that give larger $f(x)$. Whether one calls this "convergence" or "divergence" depends on one's point of view; however, the SNSEA process *is* optimizing against the packing/coverage objectives, regardless.

Again, we examine coverage convergence across different values of σ and ρ_{min}. Figure 9 below shows that (again) in all but one case, the algorithm appears to converge in terms of the coverage epsilon measure. As in the bounded case, when the ratio between σ and ρ_{min} is around 1:3 or 1:4, the algorithm appears to perform the best in terms of making steady progress in coverage.

Figure 10 below shows that in almost all cases, convergence conditions were met. As before, even in the few of cases where it did not meet, there were always more than 1,500 generations of stagnation before we gave up at generation 5,000.

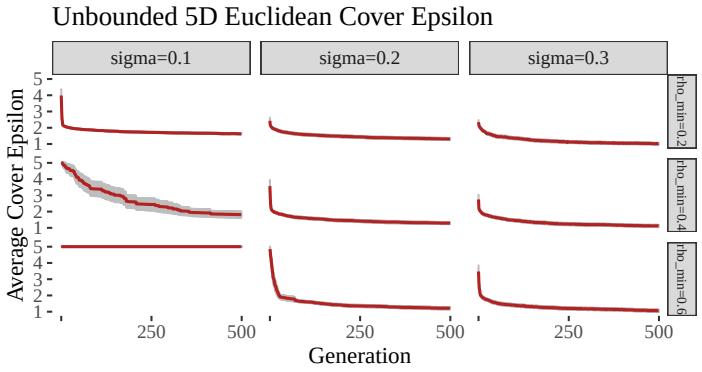

Figure 9: Coverage values for SNSEA applied to an unbounded Euclidean space with $\sigma \in \{0.1, 0.2, 0.3\}$ and $\rho_{min} \in \{0.2, 0.4, 0.6\}$ averaged over 50 independent trials. The 95% CIs are represented as gray bars.

To be clear: We *do not claim* that the SNSEA *never* diverges, nor do we ascribe any quality (good or bad) to whether it *should* converge or diverge. We merely note that it *does* converge in many cases. More importantly, there are constructive things to learn about this observation. For example, it's clear that balancing mutation and the sparseness criterion is very important.

4 The PSNSEA Optimization Process

Our population-based SNSEA is a lot more like a traditional novelty search algorithm: Child individuals are selected from among the parent population individuals and the *most novel* survive, while we also continue to update an ever-growing archive of individuals that meet our sparseness criterion. In this section, we consider empirical results of such an algorithm.

Wiegand (2021) demonstrated that population-based SNSEAs can converge in the sense that we defined above; however, that work had a couple of limitations. First, the paper considered unbounded spaces, spaces for which novelty search enthusiasts have

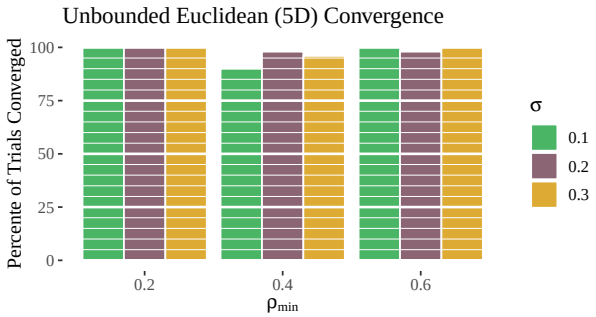

Figure 10: Convergence percentages over 50 trials for the SNSEA operating over different σ and ρ_{min} values on an unbounded Euclidean space.

already indicated may be less suitable for such methods (Lehman and Stanley, 2008). Second, the algorithm used $(\mu + \lambda)$ selection dynamics in order to drive home the point that convergence was at least *possible*. For this paper, we consider bounded spaces.

4.1 PSNSEA Converges Quickly in Bounded Spaces

Figure 11 illustrates such an example, where $\sigma = 0.1$, $\rho_{min} = 0.2$, $d = 5$, and the algorithm is run to 500 generations. In this case, $\mu = 2$ and $\lambda = 8$. Again, 50 independent trials were performed. As with previous examples, the algorithm minimizes coverage and maximizes packing. Indeed, this appears to happen much faster than for the SNSEA, which is consistent with the prior study.

For consistency, we examined the same parameter set of σ and ρ_{min} as in the previous section using $(2, 8)$ selection dyanmics, Figure 12. In all cases, coverage is minimized. Convergence is slowest when σ and ρ are not to far apart.

4.2 Improving the PSNSEA

One curious difference in terms of traditional dynamics is that the *ratio* of child-to-parent population sizes doesn't seem to affect convergence speed. There appears to

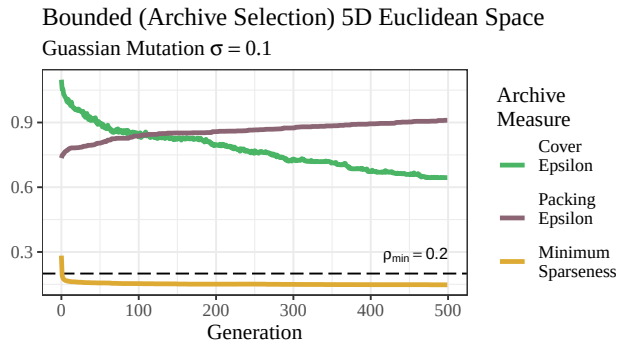

Figure 11: PSNSEA applied to a bounded 5D Euclidean space averaged over 50 independent trials when $\sigma = 0.1$, $\rho_{min} = 0.2$, $\mu = 2$, and $\lambda = 8$.

be more going on than simple selection pressure. In Wiegand (2021), it was noted that the current populations in novelty search represent a current estimate for position in archive space in the sense that it is a sample estimator for the leading edge of the archive. It turns out that in some cases the PSNSEA populations do not represent the archive's leading edge very well. In that study, the severity of the $(\mu + \lambda)$ dynamics may have given a false sense of this property, so it's valuable to re-examine this under different conditions.

Recall the relationship between mutation and the sparseness metric discussed in Section 3.1. From that analysis, we know that we can greatly slow down the convergence of the archive to saturation by setting ρ_{min} much lower than σ. We will do this for our next experiment in order to better visualize the dynamics between the population and the archive in the PSNSEA. Specifically, we use $\sigma = 0.1$ and $\rho_{min} = 0.001$ and apply $(2, 8)$ selection dynamics. The top plot in Figure 13 illustrates a typical run over time. We see that the child population (the points in red) are not particularly representative of the archive. The point of novelty search is to spread out in the space, so this poor characterization of the archive slows down convergence.

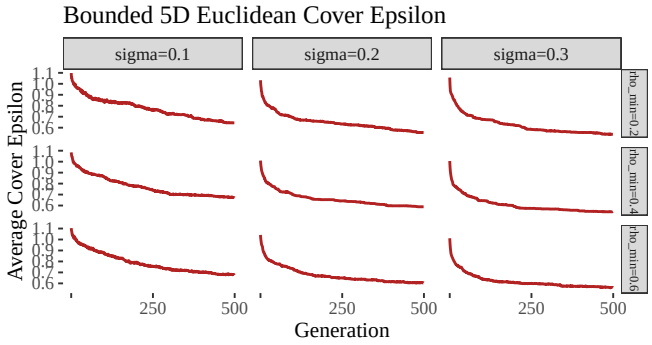

Figure 12: Coverage values for PSNSEA applied to a bounded Euclidean space with $\sigma \in \{0., 0.2, 0.3\}$ and $\rho_{min} \in \{0.2, 0.4, 0.6\}$ using $(2, 8)$ selection dynamics averaged over 50 independent trials.

Understanding novelty search as a search through *archive space* (rather than individual points) is key to addressing this issue. The current population is a representation of archive space (as a whole), so it should be the case that improving this representation will speed up convergence — meaning the algorithm will cover the space faster. If selecting points from the parent population leads to degenerate child population representations of the archive, then an easy way to improve that representation is to draw at least *some* (say 75% of) points directly from the archive itself, bottom plot in Figure 13.

This appears visually to be a big improvement. But let's see how this affects one of our previous cases. Figure 14 illustrates the identical case as the first one shown in this section: a $(2, 8)$-PSNSEA with $\sigma = 0.1$ and $\rho_{min} = 0.2$. Except in this case, when selecting points to mutate to produce a child, we selected a point uniformly at random directly from the archive with 75% probability. If not drawn from the archive, the parent population was used as in all previous cases.

Archive selection drives the algorithm to a lower coverage, and it does so faster as well as more smoothly. At least in this case, selecting individuals from the archive

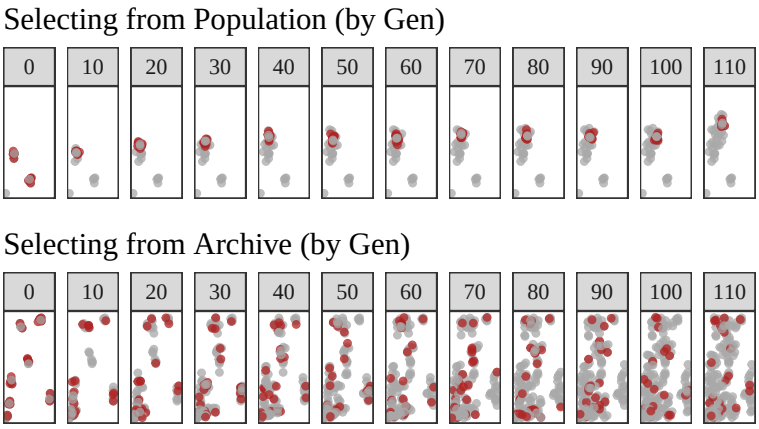

Figure 13: Visualizing a $(2, 8)$-PSNSEA applied to a bounded 2D Euclidean space with $\sigma = 0.1$, $\rho_{min} = 0.001$ over 110 generations. The gray dots represent archive points, and the red dots represent the current children in that generation. The top plot illustrates select from the population, the bottom selection directly from the archive.

improves performance. Indeed, treating the archive as the population itself is precisely the mechanism that the MAP-Elites method uses. So our results represent an independent confirmation of this idea, as it applies to traditional novelty search. That is, we now have a better understanding for *why* selecting from the archive can improve performance.

In general, we believe that finding ways for child samples to better represent the archive in terms of *coverage* should improve novelty search performance. Another example would be to maintain a subset of the archive that form a convex hull over the whole archive, and select points from that subset. Monitoring coverage differences between the parent population and the archive may also provide triggers for when to make changes to selection methods.

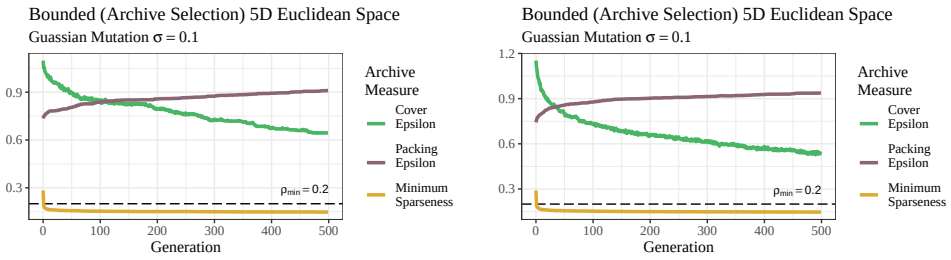

Figure 14: PSNSEA applied to a bounded 5D unbounded Euclidean space averaged over 50 independent trials when $\sigma = 0.1$, $\rho_{min} = 0.2$, $\mu = 2$, and $\lambda = 8$. On the left, all children come from parents drawn from the parent population. On the right, there is 75% chance each child comes from a parent drawn from the archive.

4.3 Simple Behavior Mapping & the PSNSEA

Our study deliberately focuses on a simplification of novelty search in order to better understand the method and the claims about the method. One simplification that we made was to use the explicit genotype in our sparseness calculations.

It should be clear that as long as there is a *metric* space in which a valid distance measure exists between points in the archive, the results we describe above hold: simple novelty search methods can and do converge, regardless of behavior metrics. To be sure, the *relationship* between mutation and the final observed archive will be radically different in a more realistic application that includes a mapping from the genotype to the phenotype to behavior; however, as long as novelty search is adding points to an archive using the sparseness criteria we discuss above via the calculation of *some* distance, the algorithm will attempt to find an archive that tries to cover and pack.

Not only will this happen, regardless of the mapping, but it is clear that local (premature) convergence can occur. We provide a simple example below to illustrate that convergence can and does happen even when evaluating behaviors, and that local con-

vergence does occur.

Consider a genotype consisting of 11 real-valued genes that are interpreted to control an agent as follows. The first value will be used to determine the initial orientation of the robot (in radians). The remaining values will turn the robot by the specified number of radians. The "*behavior*" of the agent is then the path it takes from an originating point given this angular specifications. As in Lehman and Stanley (2008), we use Euclidean distance between final destinations of agents to measure behavior dissimilarity. All other representation, algorithmic, and measure differences are consistent with the previous real-valued experiments above. Here $\sigma = 0.1$, $\rho_{min} = 0.2$, $\mu = 2$, and $\lambda = 4$.

Running the convergence test we described in Section 4.1, we found that in 39 out of 50 runs the system converged. That is, in 78% of the independent trials, the archive reached a point of coverage saturation and no further progress was made in terms of discovering novel individuals. When it did so, it took an average of 130 generations (361 generations in the worst case). To emphasize: *some* runs produced an archive that stagnated, while *others* continued to make progress. This is a classic sign of premature convergence.

We repeated the experiment with the following exceptions. We ran 100 independent trials for three different experiments in which the algorithm was run until 50 generations, 100 generations, and 200 generations at which point the entire archive contents of final end points of the agents were dumped. Below we see the results of every point in every archive for those three runs. We provide 2D statistical kernel density plot to show that the behaviors do not uniformly saturate the space. These are shown in the top plot of Figure 15.

Below that graph we examine runs that *converged*. Indeed, some of the runs continued to make progress, but as before many runs stagnated. Here used the simple

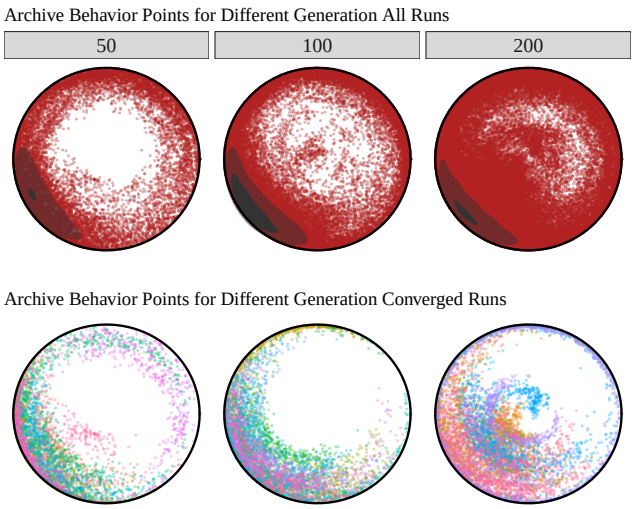

Figure 15: Archive of behavioral path end points learned by PSNSEA after 100 independent trials for runs to 50, 100, and 200 generations. Top: *all* archive points with statistical density. Bottom: only those from *coverged* runs. Trials are colored differently.

convergence criteria of whether the *final* coverage of the archive represents an improvement over the archive at the halfway point of the run (25 generations, 50 generations, and 100 generations, respectively). We filtered out all runs that *did not* converge and highlighted each separate remaining trial with a different color.

This illustrates clearly several things. First, even applying a genotype to phenotype mapping does not change the fact that the algorithm can and does converge in the sense that it will no longer make substantive progress toward covering the space. Moreover, the algorithm can and does (in a specific trial) get overly focused on one area of the space over another. This is precisely local, premature convergence.

5 Discussion

Novelty search can be a helpful algorithm for discovering complex objects. Proponents espouse three critical components: a sparseness metric and archive, generative repre-

sentations, and distance measures in the actual solution space (e.g., behaviors) rather than genotype space. The foundational work of understanding how novelty search works is only begun, though. Since there are several claims about how novelty search explores spaces that exist in the literature, it is important to begin to address this. The first idea to consider is the claim that novelty search is "divergent" and "objectiveless".

In this paper, we consider only the first piece of the puzzle: The sparseness metric and archive. We use simplified methods both with and without populations in order to concentrate on how the sparseness metric and the archive update criterion affect search dynamics. We find that the most constructive view is that novelty search happens at the level of the *archive space*, not the individual point space. Looking at it this way is helpful because we can start to answer questions and make improvements to the algorithm and how we apply it. Our analysis is specific to the original novelty search algorithms; we do not consider newer variants.

First: Is novelty search inherently "objectiveless", does it "diverge"? The answer to this question is very clearly, "no." At least some novelty search methods are driven to spread out to *coverage* the space, while trying to remain as efficiently *packed* as possible. Novelty search is driven to converge to an ϵ-net in the sense defined by k-NN theory. We do not claim that this is true of *any* novelty search algorithm, just that the mere act of abandoning an external objective function in favor of a sparseness criteria does not make the search inherently divergent. This is actually good news, since it means we can analyze the search process in similar ways as other optimization processes.

Second: We showed simple novelty search methods can converge to local optima even when a behavior mapping is used. Note that we are not claiming *all* novelty search methods *always* converge, we are simply refuting the claim that others have made that novelty search *cannot* get stuck in a local optima (Lehman et al., 2016; Lehman and Mi-

ikkulainen, 2015; Stanley and Lehman, 2015). This is very important to the application of novelty search to real problems since it has an immediate implication about whether to use techniques such as *restarts* or methods to try to combat archive stagnation.

Third: Given it is optimizing, are there reasons to believe the algorithm will be particularly sensitive to certain parameter choices? Indeed, there is an intrinsic relationship between mutation and the sparseness minimum. For some problems, obtaining a good balance may be difficult. Understanding the precise nature of this balance will be particularly difficult in true applications of novelty search since distance calculations will tend to be in *behavior space*, not genome space. This only exacerbates the problem that is inherent to the process. Engineers that apply novelty search are warned that establishing a good balance between mutation and the sparseness minimum may well be challenging, and they should concentrate their attention on this during parameterization studies. While the idea that mutation rate is important is certainly not surprising, what we show is that balancing it can be *extremely* important and difficult to do in novelty search — if mutation does not provide enough diversity in behavior space, the algorithm can wait (possibly exponential time) for sufficiently sparse point to be generated; however, if mutation leads to copious new points (in whatever space) then managing archive growth becomes vitally important. Since estimating coverage and packing occurs in behavior space, it is possible to add these metrics to existing novelty search methods. Knowing how rapidly the space is being covered and packed can help provide engineers with insights about mutation and archive update mechanisms to keep search moving forward.

Fourth: If novelty search is optimizing for coverage, can we *improve* that search? We saw above, that at least in some cases it may be beneficial to draw points directly from the archive to improve population representation of the search space. This is use-

ful advice and provides a direct tie between common practices in MAP-elite methods and traditional novelty search. Selecting *uniformly at random* from the archive can be problematic, and there are several existing methods within quality diversity algorithms for addressing this, including the use of Monte Carlo selection methods (Sfikas et al., 2021) and preferring newer archive members to older (Coninx and Doncieux, 2021); however, our analysis suggests a more directed approach: finding ways of ensuring the population remains near the *boundaries* of the space will even further improve results by speeding up exploration of new portions of the space.

This instead considers the perspective that the archive is the object of search. Rather than the external optimization function being the objective, *coverage* is the objective; however, the results we discuss are *not subjective* experiences based on that viewpoint. Rather, changing perspectives allows us to examine the *dynamics* of the algorithm, regardless of the viewpoint. Whether one sees the individual as the object of search or not, simple novelty search can and does stagnate, and it can get stuck in a local optima. The idea that the algorithm implicitly "diverges" (Lehman et al., 2016; Lehman and Miikkulainen, 2015; Stanley and Lehman, 2015) is simply not a good description for the dynamics of the algorithm, and that viewpoint may interfere with improving the method. Far from being a philosophical difference, understanding that novelty search can and does converge has very practical implications about its use. The simplest of these being that it is useful to try to detect stagnation and (potentially) reset the algorithm to continue effective production of novel individuals. Therefore, it is our firm position that the *outcome* of traditional novelty search is to produce an archive that effectively covers and packs a space, be it behavior or otherwise.

References

Boyan, J. A. and Moore, A. W. (1998). Learning evaluation functions for global optimization and boolean satisfiability. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, page 3–10, USA. American Association for Artificial Intelligence.

Bryant, V. (1985). *Metric Spaces: Iteration and Application*. Cambridge University Press.

Choi, T. J. and Togelius, J. (2021). Self-referential quality diversity through differential map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '21, page 502–509, New York, NY, USA. Association for Computing Machinery.

Clarkson, K. L. (1999). Nearest neighbor queries in metric spaces. *Discrete & Computational Geometry*, 22(1):63–93.

Coninx, A. and Doncieux, S. (2021). Younger is better: A simple and efficient selection strategy for map-elites. In *Proceedings of the 2021 Genetic and Evolutionary Computation Conference Companion*, GECCO '21, page 87–88, New York, NY, USA. Association for Computing Machinery.

Cully, A. and Demiris, Y. (2017). Quality and diversity optimization: A unifying modular framework. *CoRR*, abs/1708.09251.

De Jong, K. A. (2006). *Evolutionary Computation: A Unified Approach*. MIT Press.

Doncieux, S., Laflaquière, A., and Coninx, A. (2019). Novelty search: A theoretical perspective. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, page 99–106, New York, NY, USA. Association for Computing Machinery.

Doncieux, S., Paolo, G., Laflaquière, A., and Coninx, A. (2020). Novelty search makes evolvability inevitable. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, page 85–93, New York, NY, USA. Association for Computing Machinery.

Ficici, S. G. (2008). Multiobjective optimization and coevolution. In *Multiobjective Problem Solving from Nature*, pages 31–52. Springer.

Ficici, S. G. and Pollack, J. B. (2001). Pareto optimality in coevolutionary learning. In *Advances in Artificial Life, 6th European Conference*, pages 316–325.

Flageat, M. and Cully, A. (2020). Fast and stable map-elites in noisy domains using deep grids. *CoRR*, abs/2006.14253.

Fontaine, M. C. and Nikolaidis, S. (2021). Differentiable quality diversity. *Robotics: Science & Systems*, XVII4.

Fontaine, M. C., Togelius, J., Nikolaidis, S., and Hoover, A. K. (2019). Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2019 IEEE Conference on Games*.

Goldberg, D. E. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, page 41–49, USA. L. Erlbaum Associates Inc.

Gomes, J., Mariano, P., and Christensen, A. L. (2015). Devising effective novelty search algorithms: A comprehensive empirical study. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, page 943–950, New York, NY, USA. Association for Computing Machinery.

Grinstead, C. M. and Snell, J. L. (2003). *Introduction to Probability*. AMS.

Hanes, J. and Wiegand, R. P. (2019). Analytical and evolutionary methods for finding cut volumes in fault trees constrained by location. *EEE Transactions on Reliability*.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:12–30.

Huang, P., Lehman, J., Mok, A. K., Miikkulainen, R., and Sentis, L. (2014). Grasping novel objects with a dexterous robotic hand through neuroevolution. In *2014 IEEE Symposium on Computational Intelligence in Control and Automation*, pages 125–132.

Langdon, W. and Poli, R. (2002). *Foundations of Genetic Programming*. Springer.

Lehman, J. and Miikkulainen, R. (2015). Enhancing divergent search through extinction events. In *Proc. of the Genetic and Evolutionary Computation Conference*, pages 951–958.

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on the Synthesis and Simulation of Living Systems*, pages 329–336.

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

Lehman, J. and Stanley, K. O. (2011b). Improving evolvability through novelty search and self-adaptation. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 2693–2700.

Lehman, J. and Stanley, K. O. (2013). Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PLoS One*, 8(4).

Lehman, J., Stanley, K. O., and Miikkulainen, R. (2013). Effective diversity maintenance in deceptive domains. In *Proc. of the 15th Annual Conference on Genetic and Evolutionary Computation*, pages 215–222. ACM.

Lehman, J., Wilder, B., and Stanley, K. O. (2016). On the critical role of divergent selection in evolvability. *Frontiers in Robotics and AI*.

Morse, G., Risi, S., Snyder, C. R., and Stanley, K. O. (2013). Single-unit pattern generators for quadruped locomotion. In *Proceedings of the 15th Genetic and and Evolutionary Computation Conference*, GECCO '13, pages 719–726.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv e-prints*, page arXiv:1504.04909.

Nordmoen, J., Veenstra, F., Ellefsen, K. O., and Glette, K. (2020). Map-elites enables powerful stepping stones and diversity for modular robotics. *CoRR*, abs/2012.04375.

Sfikas, K., Liapis, A., and Yannakakis, G. N. (2021). Monte carlo elites: Quality-diversity selection as a multi-armed bandit problem. In *Proceedings of the 2021 Genetic and Evolutionary Computation Conference*, GECCO '21, page 180–188, New York, NY, USA. Association for Computing Machinery.

Stanley, K. O. and Lehman, J. (2015). *Why Greatness Cannot Be Planned - The Myth of the Objective*. Springer.

Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100.

Szerlip, P. A., Morse, G., Pugh, J. K., and Stanley, K. O. (2015). Unsupervised feature learning through divergent discriminative feature accumulation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2979–2985.

Wiegand, R. (2020). The objective of simple novelty search. In *Proceedings from the 2020 Florida Artificial Intelligence Research Symposium Conference*.

Wiegand, R. (2021). Population-based novelty searches can converge. In *Proceedings from the 2021 Florida Artificial Intelligence Research Symposium Conference*.

Zitzler, E. (2012). Evolutionary multiobjective optimization. In *Handbook of Natural Computing*, pages 871–904. Springer.

Algorithm 2: Population-Based Simple Novelty Search Evolutionary Algorithm (PSNSEA). Algorithm can be configured s.t. $S^\star$ is either C or $archive$.

input : ρ_{min}, k, termination criterion, μ, λ

output: $archive$

initialize individual $x = 0^n$

$archive$ = $\{x\}$

$P = \{\}$

for $i \in \{1 \dots \mu\}$ **do**

  initialize parent $x \in \mathbb{R}^n$ uniformly at random, i.i.d

  $P = P \cup \{x\}$

end

while *not reached termination condition* **do**

  $C = \{\}$

  for $j \in \{1 \dots \lambda\}$ **do**

    $j = j\%\mu$

    draw x from the j^{th} individual in P

    y = mutate(copy of x)

    $C = C \cup \{y\}$

    ρ = sparseness($S^\star, y, k$)

    if $\rho \geq \rho_{min}$ **then**

      $archive = archive \cup \{y\}$

    end

  end

  compute sparseness for each individuals in C over $S^\star$

  set P to contain the μ individuals with the highest sparseness in C

end