Speciation in Evolutionary Algorithms: Adaptive Species Discovery

Antonio Della Cioppa
Natural Computation Lab,
DIEII
Università di Salerno, I–84084
Fisciano (SA), ITALY
adellacioppa@unisa.it

Angelo Marcelli
Natural Computation Lab,
DIEII
Università di Salerno, I–84084
Fisciano (SA), ITALY
amarcelli@unisa.it

Prisco Napoli
Natural Computation Lab,
DIIIE
Università di Salerno, I–84084
Fisciano (SA), ITALY
prisco.napoli@alice.it

ABSTRACT

The use of niching methods for solving real world optimization problems is limited by the difficulty to obtain a proper setting of the speciation parameters without any *a priori* information about the fitness landscape. To avoid such a difficulty, we propose a novel method, called *Adaptive Species Discovery*, that removes the basic assumption of *perfect discrimination* among peaks underlying Fitness Sharing and, consequently, allows to overcome the drawbacks of the most performing sharing-based methods. This is achieved through an explicit mechanism able to discover the species in the population during the evolution. The method does not require any *a priori* knowledge, in that it makes no assumption about the location and the shape of the peaks, while it exploits information about the *ruggedness* of the fitness landscape, dynamically acquired at each generation. The proposed method has been evaluated on a set of standard functions largely adopted in the literature to assess the performance of niching methods. The experimental results show that our method has a better ability to discover and maintain all the peaks with respect to other methods proposed so far.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: I.2.8—*Problem Solving, Control Methods, and Search*

General Terms

Algorithms

1. INTRODUCTION

The need for adequate models describing the evolutionary dynamics that governs the process of transformation of living beings has always been a main goal of the Evolutionary Computation community. In this context, one of the main interest is devoted to the factors governing the emergence of speciation phenomena, with the aim to develop methods able to simulate such phenomena in an artificial environment. The rationale behind such interest is that the expected behavior of an Evolutionary Algorithm (EA) may differ significantly from the natural evolutionary process. Indeed, natural evolution leads to the formation of a wide variety of species co-existing within an ecosystem, wherein such a multitude of different species evolve at the same time and compete for using the environmental resources. On the contrary, a canonical EA inevitably converges towards a population composed by just one fit individual and its close mutants [5, 15]. The loss of genetic diversity is mainly due to the *genetic drift* resulting from both the selection process and the working mechanisms on which EAs are based upon [7, 9, 13]. To overcome such a difficulty, a relevant research activity has been devoted to Niching [15] which is able to trigger in EAs a process similar to *natural speciation.* As a consequence, Niching is essential to counteract the tendency of a canonical EA to converge towards only one optimal solution, hopefully the best, even when dealing with a multimodal domain made of many different peaks in term of size and height. The basic idea most of the niching methods are based upon draws from the analogy with natural ecosystems. In Nature, an ecosystem is composed by different physical spaces (niches) exhibiting different features and allows both the formation and the maintenance of different types of life that compete to survive (species). For each niche, the physical resources are finite and must be shared among the individuals belonging to it [4, 16]. By analogy, Niching promotes genetic diversity by encouraging the emergence of species in different sub-spaces of the environment (niches), each commonly referred to as an optimum of the domain. The fitness represents the resources (carrying capacity) of the niche, while species can be defined as similar individuals in terms of a similarity metrics. With the aim to promote genetic diversity, speciation is achieved by dividing, either implicitly or explicitly, the population into different subpopulations that are prone to occupy distinct areas of the search space [8, 15].

A large variety of niching methods have been proposed in the literature [8, 15, 17, 20, 23, 24]. Generally, they differ for the speciation parameters needed, e.g., niche radius, population size, etc., and for the assumptions made on the fitness landscape, i.e., perfect discrimination hypothesis. Unfortunately, the use of such methods in solving real world optimization problems is limited by the difficulty to obtain a proper setting of the speciation parameters without any

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'11, July 12–16, 2011, Dublin, Ireland.
Copyright 2011 ACM 978-1-4503-0557-0/11/07 ...$10.00.

a priori information about the fitness landscape [12, 15]. Speciation parameters are typically related to topological information of the fitness landscape corresponding to the problem at hand, such as the location, the shape and the distance of the peaks in the landscape. As a consequence, making a proper setting is not trivial, and the use of niching methods in real applications typically require an expert that, through repeated executions, establishes the best configuration of the parameters to be applied [1, 18, 21]. In addition, the expected behavior of these methods is strongly sensitive to slight changes of the values of these parameters: small changes can lead to discover an inadequate number of optimal solutions, or even to the loss of the best solutions.

In this paper, we propose a novel niching method, called *Adaptive Species Discovery* (ASD), that removes the basic assumption of *perfect discrimination* underlying the Fitness Sharing (FS). Consequently, unlike many sharing-based methods, it is more effective in dealing with irregular and unevenly spaced peaks in a multimodal landscape. ASD is based on an explicit mechanism to discover the species in the population during the evolution that does not require any *a priori* knowledge and makes no assumption about the fitness landscape. The sharing mechanism is applied separately to each niche at the end of each generation. Every niche, moreover, is identified explicitly by a representative individual, i.e., the *species master*, which is chosen among its members. The explicit knowledge about the number of niches and their location in the search space at each generation allows to encapsulate a *k-elitist* strategy to copy the species masters from one generation to the next, so improving the ability of maintaining a given number of niches. The method has been evaluated on a set of standard functions largely adopted to assess the performance of niching techniques on multimodal functions. The experimental results show that ASD has a better ability to find and to maintain all peaks with respect to other niching methods proposed in the literature.

The remainder of the paper is organized as follows: Section 2 describes the background of the FS and some of the best sharing-based methods, while Section 3 gives the fundamentals of Adaptive Species Discovery. The experimental results are illustrated and discussed in Section 4, while some conclusions are reported in Section 5.

2. NICHING METHODS

Fitness Sharing [8] is one of the most well-known and widely used niching methods. It modifies the fitness landscape by reducing the payoff in densely populated regions to counteract the genetic drift. The basic idea is to treat the fitness as a resource so as to force all individuals within a niche to share the fitness among them. This is achieved by reducing the fitness of each individual by an amount nearly equal to the number of similar individuals in the population. The adjusted fitness is called *shared fitness* and represents the fitness landscape the selection mechanism works with. Formally, at time t, the shared fitness $\phi_{sh,t}(\boldsymbol{\tau})$ of an individual $\boldsymbol{\tau}$ is computed according to:

$$\phi_{sh,t}(\boldsymbol{\tau}) = \frac{\phi(\boldsymbol{\tau})}{m_t(\boldsymbol{\tau})}, \tag{1}$$

where $\phi(\boldsymbol{\tau})$ is the raw fitness and $m_t(\boldsymbol{\tau})$ is the *niche count* providing information about the crowding in the neighborhood of $\boldsymbol{\tau}$. In other words, $m_t(\boldsymbol{\tau})$ is an estimate of the number of individuals belonging to the same niche: and it is computed as follows:

$$m_t(\boldsymbol{\tau}) = \sum_{\boldsymbol{\omega} \in P_t} \mathrm{Sh}(\boldsymbol{\tau}, \boldsymbol{\omega}) \tag{2}$$

$$\mathrm{Sh}(\tau, w) = \begin{cases} 1 - \left(\frac{d(\boldsymbol{\tau},\boldsymbol{\omega})}{\sigma_{\mathrm{sh}}}\right)^{\alpha} & \text{if } d < \sigma_{\mathrm{sh}} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

$\mathrm{Sh}(\cdot,\cdot)$ is the *triangular sharing function* which measures the similarity among individuals, $d(\boldsymbol{\tau}, \boldsymbol{\omega})$ is the distance between $\boldsymbol{\tau}$ and $\boldsymbol{\omega}$ and α is a control parameter that determines the shape of $\mathrm{Sh}(\cdot,\cdot)$ and it is typically set to 1. Eq. (3) shows that the triangular sharing function relies on the niche radius σ_{sh}, which is the maximum distance between two individuals so that they can be considered similar and, therefore, belonging to the same niche. The niche radius and the population size play a key role for the discovery and the maintenance of all the niches in the search space. In fact, it has been proved that, when the peaks are perfectly discriminable (all niches can be represented by an hypersphere with the same radius), the population size is large enough and the niche radius is properly set, FS provides as many species in the population as the number of peaks in the fitness landscape, thus populating all the niches [14]. Unfortunately, setting proper values for the above parameters involves the need of some *a priori* knowledge about the fitness landscape. In order to reduce such a difficulty, various techniques have been introduced to obtain a reliable estimate of the speciation parameters. The method proposed by Deb et al. [12] expect the number of peaks and the dimensionality of the problem to set a proper value for the niche radius. Mahfoud [15] suggests, instead, a method for determining a lower bound for the population size, which requires an estimate of the number of niches to be maintained and their heights. However, is unlikely that in real problems the assumptions made by these two techniques are satisfied, making FS very hard to apply. In [1], Della Cioppa et al. have introduced an iterative approach for learning the optimal niche radius and population size without *a priori* information on the width, number, height, and position of the peaks in the fitness landscape. Nevertheless, the method still requires that the perfect discrimination hypothesis is verified, in that the niches are obtained by using a single niche radius.

Many other sharing based methods have been proposed for improving the ability of FS to find and maintain all peaks in hard domains. The Dynamic Niche Sharing (DNS) of Miller et al. [3] was the first method to introduce an explicit mechanism to dynamically identify the species in the population during the evolution. The method, however, is based upon two assumptions which are difficult to satisfy in practice: the number of peaks in the fitness landscape is known and the perfect discrimination hypothesis holds. At each generation, DNS identifies all niches in the population and uses this information to classify individuals as belonging to one of the identified niches or as belonging to a unique class, called *nonspecies*. This is achieved through an algorithm called Dynamic Peak Identification (DPI). After the niche identification phase, FS is applied by using two different sharing mechanisms. The shared fitness of an individual belonging to a niche is computed by dividing its raw fitness by the number of individuals populating his same niche (*occupation number*), while standard FS is applied to the individuals belonging to the *nonspecies* class. The authors advocate the use of the occupation number as niche count because it reduces the tendency of FS to keep individuals too spread within the niche. Moreover, this leads to a benefit in terms of computational cost, in that the occupation number is computed only once for all the members belonging to a niche, while the niche count has to be computed for each individual in the population. It is obvious that DNS suffers the same drawbacks of FS about the population size and niche radius.

Dynamic Fitness Sharing (DFS) was suggested by Della Cioppa et al. [2] to further improve DNS. The method adopts a method (Dynamic Species Identification) to identify the species in every generation that does not require to know the number of peaks in the landscape, and uses FS to ensure that each niche will be populated proportionally to the fitness of its peak. Unlike DNS, moreover, DFS adopts a more biological plausible definition of species, because it is assumed that a species consists of a subpopulation of at least two individuals and is represented by its species master, i.e. the most representative individual. The only assumption made is the perfect discrimination hypothesis. Consequently, it is unable to deal with irregular unevenly spaced peaks in the search space and, as authors notice, DFS suffers the same drawbacks as FS when wrong values for either the niche radius or the population size are chosen.

Dynamic Niche Clustering (DNC), introduced by Gan et al. [10], tries to solve this latter problem by using a set of overlapping fuzzy niches with independent radius (*nicheset*). Each niche is described by some parameters that vary during evolution: the current midpoint in decoded parameter space, the current niche radius, the original midpoint where the niche was spawned, the generation at which the niche was spawned, and a list of references to the individuals that are currently members of the niche. DNC allows for an individual to be a member of more than one niche, so the niches themselves may overlap to a certain degree in the search space. To counteract the excessive growth or decrease of the number of the niches discovered during evolution, both a minimum and a maximum value for the niche radius are required. The possibility of allowing an individual to belong to more than one niche leads to an effect called *striation*. Striation occurs when two niches, traversing up the slope on either side of the peak, start to overlap but don't converge due to the heavy penalization incurred by individuals from being a member of two niches. In order to solve this problem, Gan et al. have introduced in [11] the *outer niche radius*, defined as twice the current niche radius. Every niche maintains a separate list of references to individuals that lie within its outer radius, but outside the niche radius. When two niches overlap, individuals belonging to both of them are penalized less than in the previous case. DNC is based on the assumption that the search space must have bounds of the same magnitude, hence, when this condition does not hold, a wrong value for the niche radius could be selected. The initial population size is another important issue, because it determines the initial value, the upper and the lower bounds of the niche radius. Moreover, DNC spreads the individuals in a niche around the top of a peak, rather than cluster them tightly at the very apex. Consequently, the average niche fitness is slightly lower when compared to other niching techniques.

As regards to the last version of DNC [11], a very interesting improvement is the introduction of a fitness topology function, which is used together with the upper and the lower bound of the niche radius, to decide whether to merge and/or to split niches during the evolution. In fact, this function is useful to determine if two points in the fitness landscape belong to the same peak by checking the fitness of a fixed number of inner points which lie on the line between them. If none of the inner points has a lower fitness than the points at the ends, it is assumed that there is no valley between end ponts and, hence, they belong to the same peak. In our opinion, the major criticism about the methods employing a topology function is the introduction of a fixed threshold on the niche size (in DNC a niche could be split only if it has a population size greater then 10% of the total population size) as well as the use of a fixed number of inner points to evaluate. About this latter, is obvious that a wrong setting of the number of inner points can reduce the effectiveness of this method: low values can lead to an insufficient exploration, while high values can lead to a high computational cost.

Recently, niching methods have also been investigated in the Evolution Strategies (ES) framework [5]. Very interesting is the proposal of Shir et al. [18], which introduces the new concepts of adaptive niche radius and niche shapes into the framework of niching with the covariance matrix adaptation evolution strategy ((M-)S-CMA). This is done through the definition of a self-adaptive niche radius and the application of the Mahalanobis distance for the adaptation of the niche shapes. The major feature of this method is the relaxation of speciation parameters in the fixed-radius CMA niching algorithm [17], which permits to make less assumptions on the topology of the fitness landscape. More specifically, CMA embeds the DPI routine for classifying the entire population into niches. However, it requires to set the number of desired niches. Shir et al. try to reduce this drawback by adopting a self-adaptive approach without any influence on the algorithmic behavior [18]. Each individual, that is a feasible solution, carries and adapts a niche radius along with its adaptive strategy parameters at every generation [19]. Niche radius is coupled to the global step size σ through an *individual learning coefficient*, and the indirect selection of the niche radius is governed by the objective that every niche should ideally consist of λ individuals. This is implemented by means of a *quasi-dynamic* fitness sharing mechanism. This latter permits to enforce the requirement of having a fixed resource of λ individuals for every niche. The main drawback is that the function of the learning coefficients has to be tuned through two parameters α and γ, which seem to be dependent on the the strategy adopted and the fitness landscape. As the authors notice, this is an undesirable situation, but they stress that the proposed setting for α and γ applies to a wide range of practically relevant landscapes.

Finally, along the line of thought adopted by us, Stoean et al. [22] have proposed a speciation method integrating the conservation of the best species masters with a topological subpopulations separation mechanism. ASD, however, dynamically adapts the number of inner points needed to check if two individuals lie or not on the same peak, adopts an explicit method for merging subspecies and does not require any explicit limit on the number of species to discover and maintain.

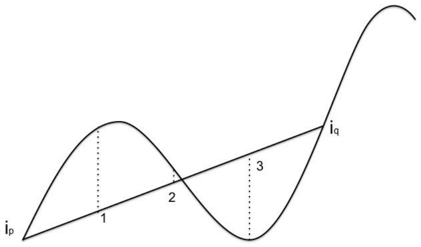

Figure 1: Hill-Valley Function in 1-D.

3. ADAPTIVE SPECIES DISCOVERY

The analysis of the main niching methods indicates that we find ourselves once again facing the problem of a proper speciation parameters setting. Moving from such a consideration, we propose a novel niching method, called Adaptive Species Discovery, developed for overcoming the main difficulties of the methods introduced so far. ASD removes the perfect discrimination hypothesis and does not require any kind of *a priori* knowledge about the fitness landscape. Furthermore, differently from many other sharing-based methods, it does not require to set the values of the speciation parameters. Its most prominent feature is the presence of an explicit mechanism for discovering the species in the population, so that the species naturally emerge during the evolution. This is possible because ASD does not require a parameter like the niche radius to describe a niche, nor tries to estimate it dynamically during the evolution: simply it does not need any niche radius.

The discovery mechanism is based upon a limited set of information referring to the ruggedness of the fitness landscape. This information is dynamically acquired at each generation through an Hill-Valley function. This function was firstly proposed by Ursem [23] as a method to verify whether two points i_p and i_q lie or not on the same peak in the landscape. This is done by checking several inner sampling points on the line between i_p and i_q. If none of the inner points has a lower fitness than both i_p and i_q, it is concluded that there is no valley between them and, hence, they belong to the same peak (Fig. 1). Hill-Valley function allows to detect genetic divergence phenomena[16] (which are the principal promoters of speciation in Nature) that occur within an artificial population. The basic idea for adopting the Hill-Valley function is to prevent individuals with very different phenotypic (or genotypic) features from being considered as belonging to the same species. Only to bootstrap ASD, we perform a partitioning of the initial population (for example, by using a clustering method) in order to identify the initial set of species. After the current population has been partitioned taking into account the set of species, the Hill-Valley function is used to check whether a species can be *separated* in two new species or it can be *merged* with another species. During the evolution, the sharing mechanism is applied separately to each niche by using the occupation number as niche count. Every species is identified explicitly by a representative individual, i.e., the *species master*, which is chosen among its members similarly to DFS. No assumption on the minimum number of individuals constituting a species is adopted. The explicit knowledge about the number of species and their location in the search space at each generation, allows to easily realize a *k-elitist* strategy. The species set discovered is preserved from one generation to the next one by copying the species masters into the new population. Such a choice improves the ability of maintaining all the species and the computational efficiency as well. The elitist strategy, moreover, reduces the lower limit on the population size needed to maintain a given number of species. This also allows for defining a simple and efficient mechanism to dynamically adapt the number of inner points to check with the Hill-Valley function during the fusion phase. The skeleton of ASD is outlined in **Alg. 1**, while in the following we describe the main algorithm steps.

Algorithm 1 Adaptive Species Discovery

```
Require: $S$: the species set, $P$: the population
  repeat
    {Determine the species individuals belong to}
    for all $\boldsymbol{\tau}$ in $P$ do
      determine the species $\boldsymbol{\tau}$ belongs to
    end for
    {Find the masters of the species}
    for all $\boldsymbol{s}$ in S do
      find the master of the species $\boldsymbol{s}$
    end for
    {Separation of the species}
    for all $\boldsymbol{s}$ in S do
      check if species $\boldsymbol{s}$ can be separated into two new species
    end for
  until (there was a separation)
  {Fusion of the species}
  for all $\boldsymbol{s}_i$,$\boldsymbol{s}_j$ in S $\wedge$ $i \neq j$ do
    check if $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ can be merged together
  end for
```

Determine the species individuals belong to. At time t, for each individual $\boldsymbol{\tau}$ in the population P, the Euclidean distance between $\boldsymbol{\tau}$ and all the species masters found at $t-1$ is computed. The individual $\boldsymbol{\tau}$ is then marked as belonging to the species which corresponds the minimum distance $d(\cdot, \cdot)$ according to:

$$d(\boldsymbol{\tau}, \boldsymbol{s}_i) = \underset{s_i \in S}{min}\{d(\boldsymbol{\tau}, \boldsymbol{s}_i)\} \rightarrow \boldsymbol{\tau} \in M(\boldsymbol{s}_i), \qquad (4)$$

where $d(\boldsymbol{\tau}, \boldsymbol{s}_i)$ is the Euclidean distance between $\boldsymbol{\tau}$ and the species master of $\boldsymbol{s}_i$, while $M(\boldsymbol{s}_i)$ are the members of $\boldsymbol{s}_i$.

Find the masters of the species. For each species $\boldsymbol{s}_i$, the master is chosen as the best individual belonging to $\boldsymbol{s}_i$.

Separation of the species. For each species $\boldsymbol{s}_i$ at time t, the following steps are applied:

1. $\boldsymbol{k}_{\boldsymbol{s}_i}$ pairs of individuals are randomly chosen in $M(\boldsymbol{s}_i)$;

2. for each pair of individuals $(\boldsymbol{\mu}$,$\boldsymbol{\nu}) \in \boldsymbol{k}_{\boldsymbol{s}_i}$, the Hill-Valley function is applied to $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$;

3. if a valley is detected between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, the species $\boldsymbol{s}_i$ is separated into two new species. In particular, the pair of individuals $(\boldsymbol{\mu}$,$\boldsymbol{\nu})$ corresponding to the deepest valley is chosen, and two new species with $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ as species masters are inserted into the species set $\boldsymbol{S}$. Subsequently, all members of $\boldsymbol{s}_i$ are reallocated with reference to the new species. Finally, the masters of the new species are recomputed, and the species $\boldsymbol{s}_i$ is removed from $\boldsymbol{S}$.

As regards the above separation mechanism, it should be noted that our strategy is based on the idea that a crowded niche can potentially cover more than one peak

Algorithm 2 EA with Adaptive Species Discovery

```
$t \leftarrow 0$
*randomly initialize* a population $P_t$ of $N$ individuals
*randomly initialize* the species set S choosing $m$ distinct individuals from $P_t$
**while** (termination condition is not satisfied) **do**
    *evaluate* the raw fitness of each individual
    *apply* the Adaptive Species Discovery algorithm
    **for all** $\boldsymbol{s}$ **in** S **do**
        *apply* the FS among the individuals belonging to $\boldsymbol{s}$
    **end for**
    *copy* the species masters in $P_{t+1}$
    *apply* the selection mechanism
    *apply* the crossover operator
    *perform* mutation on the offsprings
    $t \leftarrow t + 1$
**end while**
```

of the fitness landscape. Hence, it is more promising to apply the Hill-Valley function to such a niche. Keeping the mechanism independent of any form of foreknowledge of the problem and considering that the sharing mechanism does not provide any information to understand when a niche is overcrowded, we have decided to use only the information available, i.e. the number of species m maintained by the algorithm during the evolution. At time t, a species $\boldsymbol{s}_i \in \boldsymbol{S}$ is considered by the separation mechanism only if $n(\boldsymbol{s}_i) > \left\lfloor \frac{N}{m} \right\rfloor$, where N is the population size and $n(\boldsymbol{s}_i)$ is the occupation number of $\boldsymbol{s}_i$. This means that ASD uses the most simple assumption that we can do about a totally unknown fitness landscape. i.e., all the peaks carry the same amount of resources and, consequently, all individuals should be equally distributed among the niches. If $\boldsymbol{s}_i$ satisfies the above equation, the number of pairs $\boldsymbol{k}_{\boldsymbol{s}i}$ is set to $\lfloor \frac{n(\boldsymbol{s}_i)}{2} \rfloor$, with the restriction that each individual can be selected at most once. For each pair $(\boldsymbol{\mu}, \boldsymbol{\nu}) \in \boldsymbol{k}_{\boldsymbol{s}i}$, only one inner point is evaluated with the Hill-Valley function. Such a point is not the middlepoint, but it is generated randomly on the line between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$.

Fusion of the species. Assuming that, after the separation step, m species have been discovered and $\boldsymbol{S} = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_m\}$. For each pair ($\boldsymbol{s}_i$,$\boldsymbol{s}_j$) with $i, j \in [1, m]$ and $i \neq j$, the following steps are applied:

1. the Hill-Valley function is applied to the masters of $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$;
2. if no valley is detected between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, the two species are merged. All the members of $\boldsymbol{s}_j$ are marked as members of $\boldsymbol{s}_i$. Finally, the master of $\boldsymbol{s}_i$ is recomputed and the species $\boldsymbol{s}_j$ is removed from $\boldsymbol{S}$.

To counteract generic drift the sharing mechanism is adopted. The shared fitness $\phi_{sh,t}(\boldsymbol{\tau})$ of an individual $\boldsymbol{\tau}$ is computed by using the occupation number $n(\boldsymbol{\tau})$ as in DNS:

$$\phi_{sh,t}(\boldsymbol{\tau}) = \frac{\phi(\boldsymbol{\tau})}{n(\boldsymbol{\tau})}. \tag{5}$$

The skeleton of an EA with ASD is outlined in **Alg. 2**. The initialization of the species set $\boldsymbol{S}$ is made either by randomly choosing m members from $P_{t=0}$ or by using a clustering method as the k-means algorithm. Obviously, a proper value for m and the strategy to be used for the selection of individuals can be defined according to the degree of *a priori* knowledge, if this latter is available. For example, an estimate of the number of peaks in the fitness landscape can be very useful to set m. Similarly, it can be chosen a selection mechanism for $\boldsymbol{S}$ favoring either the fittest individuals or individuals lying in certain areas of the search space. It should be remarked here that the choice effected for the initialization method of $\boldsymbol{S}$ is not critical for our method. However, a proper choice certainly has a beneficial influence on the convergence time.

3.1 Hill-Valley Function

The ability of our method to discover and maintain different species during the evolution depends on how effectively the phases of separation and fusion are accomplished. These two phases are based upon the extraction of reliable information from the fitness landscape. So, the effectiveness of Hill-Valley function becomes critical for the success of ASD. Given that the Hill-Valley function works on the fitness landscape, if the cost of decoding parameters is expensive, the computational effort may be high. Moreover, it requires to set the number of inner sampling points h to check. Setting a proper value for this latter is not trivial because the ability to take right decisions is highly dependent on h. In fact, Hill-Valley function is effective only when it detects a valley between two points. If no valley is detected, there is no confidence that there isn't an actual valley there. For example, given two points i_p and i_q, and $h = 3$, Fig. 1 shows a typical problem of Hill-Valley function. The fitness of i_p and i_q is not lower than the fitness of all inner points, so no valley is detected. If i_p and i_q were two species masters, such undesirable behavior would lead to merge the two species.

To assess the effectiveness of the Hill-Valley function, we have chosen to adopt two different algorithms. During the fusion phase the original algorithm proposed by Ursem is used, whilst Gar's algorithm is used for the separation phase [11]. The reasons are related to the type of information necessary for these two phases. In the fusion phase, the relevant information is just the presence of a valley between two niches, not its depth. Hence, Ursem's algorithm is more suitable, because it returns its decision when the first point with lower fitness is discovered. Conversely, in the separation phase, the relevant information is the maximum depth of the valley detected. In such a case, Gar's algorithm is more suitable, because all the h inner points are checked and the point with the lowest fitness is returned. It is then evident that by adopting a fixed value for h is not certainly the best solution, since it is necessary to have some *a priori* knowledge. As a consequence, we propose a new method to dynamically adapt the number of inner points in the fusion phase through the explicit knowledge about the location and the number of niches during the evolution. It is based upon a very simple concept: the optimal number of inner points to be used for the Hill-Valley function should be computed according to the distance between points i_p and i_q, and the value of h must be sufficient to ensure that these points are spaced by a value not exceeding $d_{\min}$, where $d_{\min}$ is the minimum distance between two species of the species set $\boldsymbol{S}$. By assuming that, at time t, $\boldsymbol{S} = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_m\}$ species have been found and that it is necessary to check if a valley lies between the species $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, h is computed as:

$$h = \left\lfloor \frac{d(\boldsymbol{s}_i, \boldsymbol{s}_j)}{d_{\min}} \right\rceil, \tag{6}$$

where $d_{\min} = \min_{s_m, s_n \in S, m \neq n} \{d(\boldsymbol{s}_m, \boldsymbol{s}_n)\}$.

4. EXPERIMENTAL RESULTS

The proposed method has been tested on a wide set of multimodal functions with increasing degree of complexity, largely adopted in the literature for performance assessment of niching methods. However, here we present only the results achieved on the most significant functions. In Table 1 are summarized the unconstrained multimodal test functions as well as their main features. The evolutionary algorithm adopted is a Genetic Algorithm in which the genotype has been encoded by using a Gray code of 30 bits for each variable. As genetic operators we have adopted the one-point crossover and the bit-flip mutation, with $p_c = 0.8$ and $p_m = \frac{1}{\ell}$, respectively, where ℓ is the total length of the genome. Finally, the selection mechanism adopted is the *tournament selection with continuously update sharing* [6].

As performance criteria we report the effective number of peaks maintained (ENPM), the maximum peak ratio (MPR) [4] and the generations after which the 99% of their saturation values are achieved (G_E and G_M). For each function we have executed 50 runs with different randomly-generated initial populations. In addition, to evaluate the performance of our method in the worst conditions, we have chosen to initialize the size of $\boldsymbol{S}$ to 1 and to choose randomly the corresponding master among the individuals in the initial population.

In Table 2 we report the values of ENPM and MPR at the end of evolution, averaged over the 50 runs, and their standard deviations, together with G_E and G_M. The population size, the number of generations and the number of evaluations used for each simulation are also reported. The values given in brackets refer to the best results obtained by (M-)S-CMA as reported in [18, 19].

The first function we have faced is the $\mathcal{M}$ function, which consists of 16 non-uniform peaks with identical fitness. The distance among peaks exponentially decreases along the horizontal axis, with the minimum and the maximum distance among peaks equal to 0.01 and 32.0 respectively. Therefore, this is a challenging test case, since it is expected that when the perfect discrimination hypothesis is not verified, our method outperforms all the sharing-based methods which use a single fixed niche radius. Figure 2 shows the distribution of the species masters detected (diamonds). It is evident that the proposed method discovers all the peaks, although this is not trivial because of their particular location on fitness landscape.

Similarly to $\mathcal{M}$ function, in the $\mathcal{V}$ function there are many peaks that are neither equidistant nor distinguishable. The number of global optima, moreover, grows exponentially with the dimensionality of the function. Table 2 shows the results obtained using different values for n, while Figure 3 shows the distribution of the species masters detected in the 1-D case. It is very interesting to note that for the $\mathcal{V}$ function, our method is able to discover and maintain all the peaks in a relatively small number of generations. In addition, the values of $\langle MPR \rangle$ and σ_{MPR} confirm that the masters species match with the top of the peaks.

The $\mathcal{G}$ function has one global optima at the origin $\vec{\mathbf{0}}$ and several thousands local optima. In our tests we have considered only the discovery of 5 peaks: the global maximum and four suboptimal located at $\vec{x} \approx (\pm\pi, \pm\pi\sqrt{2}, 0, ..., 0)$. Table 2 shows the results obtained on $\mathcal{G}$ function using two different values for n. In both cases ASD discovers all the desired peaks using the same population size, without any significant loss of performance as n varies between 3 and 10.

The $\mathcal{S}$ function is characterized by the presence of many local and global optima unevenly distributed in the search space. There is one especially hard peak to deal with: the third one from the left in Figure 4. It is a very small irregularity within the slope of a larger peak: this makes this function very interesting in order to describe the behavior of our method. From the results shown in Table 2, it can be argued that for $n = 1$ and $n = 2$ ASD does not take into account such a spurious peak. To explain the underlying dynamics, we have reported in Figure 5 the average number of species discovered during the evolution after the separation and fusion phases.

The Figures 4 and 5 show that the spurious peak is indeed discovered, but because of the distribution of the individuals associated to it, that very likely will have a higher fitness, it disappears during the merging. In our opinion, this is a desirable effect, in that it performs a sort of noise removal automatically, without any other some of information than the fitness landscape. Methods, as for example (M-)S-CMA, that attempt to discover and maintain such kind of peaks, on the other hand, discover also non-desirable peaks in the landscape with a useless waste of resources. Moreover, such a behavior of ASD is close to the biological speciation process, because the spurious peak (species) could be considered a sub-species of the larger peak.

Finally, we report the results concerning the $\mathcal{W}$ function. It is a non-separable function, which contains 10 insidious and asymmetric peaks, some located in the center and others at the bounds of the search space. Figure 6 shows the species masters discovered after 200 generations by using a population size of 100 individuals. The circles are the actual peaks, while the diamonds are the species masters discovered. It is evident that, while all the niching methods using a unique niche radius fails, ASD is effective and very efficient, in that it is able to discover all the relevant peaks with a minimum amount of resources.

As regards the performance of ASD with reference to other methods, let us note that we have considered only (M-)S-CMA, because it is one of the most performing among those with less assumptions on the search landscape. Except for $\mathcal{M}$ and $\mathcal{W}$ and $\mathcal{V}$:n=3, which are not faced by (M-)S-CMA, ASD outperforms (M-)S-CMA on all the function considered in terms of $\langle MPR \rangle$. The reference values for (M-)S-CMA are reported in Table 2 in brackets. Note that for the $\mathcal{S}$ function, when $\langle MPR \rangle$ is computed we do not consider the spurious peak.

5. CONCLUSIONS

We have proposed the Adaptive Species Discovery, a novel niching method that does not require the basic assumption of *perfect discrimination* underlying the FS and, consequently, allows to overcome the drawbacks of the most performing sharing-based niching methods. ASD achieves niching through an explicit mechanism for discovering the species in the population during the evolution that does not require any *a priori* knowledge about the problem and that makes no assumptions about the location and shape of the peaks in the landscape. The discovery mechanism uses information about the *ruggedness* of the landscape, dynamically acquired at each generation, through the Hill-Valley function. To this aim, a simple and efficient

Table 1: The Test functions adopted.

Name	Function	Range	Peaks
$\mathcal{M}$	$\mathcal{M}(\vec{x}) = sin^6(log_{1.2}(x))$	$x \in [0.01, 100]^n$	16^n
$\mathcal{V}$[Vincent]	$\mathcal{V}(\vec{x}) = \frac{1}{n}\sum_{i=1}^{n} sin(10 \cdot log(x_i))$	$x \in [0.25, 10]^n$	6^n
$\mathcal{G}$[Griewank]	$\mathcal{G}(\vec{x}) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}})$	$x \in [-10, 10]^n$	5
$\mathcal{S}$[Shekel]	$\mathcal{S}(\vec{x}) = -\sum_{i=1}^{10} \frac{1}{(k_i(x-a_i))^2+c_i}$	$x \in [0, 10]^n$	8
$\mathcal{W}$[Waves]	$\mathcal{W}(x, y) = (0.3x)^3 - (y^2 - 4.5y^2)xy+$ $-4.7cos(3x - y^2(2 + x))sin(2.5\pi x)$	$x \in [-0.9, 1.2]$, $y \in [-1.2, 1.2]$	10

Table 2: Experimental findings: ENPM (Effective Number of Peaks Maintained), MPR (Maximum Peak Ratio), G_E and G_M (Generations after which the 99% of ENPM and MPR saturation values are achieved).

Function	N	Gen.	# of evaluations	$\langle ENPM \rangle$	σ_{ENPM}	G_E	$\langle MPR \rangle$	σ_{MPR}	G_M
$\mathcal{M} : n = 1$	250	200	$5 \cdot 10^4(-)$	16.000 $(-)$	0.000	81	1.000 $(-)$	0.000	99
$\mathcal{V} : n = 1$	200	50	$10^4(7 \cdot 10^6)$	6.000 (5.050)	0.000	41	1.0 (0.839)	0.000	48
$\mathcal{V} : n = 2$	200	150	$3 \cdot 10^4(3.6 \cdot 10^7)$	36.000 (17.860)	0.000	36	0.998 (0.806)	0.002	67
$\mathcal{V} : n = 3$	200	300	$6 \cdot 10^4(-)$	216.000 $(-)$	0.000	21	0.994 $(-)$	0.001	100
$\mathcal{G} : n = 3$	150	500	$7.5 \cdot 10^4(1.5 \cdot 10^5)$	5.000 $(-)$	0.000	1	1.000 (0.494)	0.000	20
$\mathcal{G} : n = 10$	150	500	$7.5 \cdot 10^4(6 \cdot 10^6)$	5.000 (2.200)	0.000	1	0.988 (0.398)	0.005	218
$\mathcal{S} : n = 1$	50	200	$10^4(9 \cdot 10^6)$	7.000 (7.833)	0.000	8	1.000 (0.968)	0.046	9
$\mathcal{S} : n = 2$	200	500	$10^5(9 \cdot 10^6)$	6.320 (6.330)	0.587	52	0.903 (0.806)	0.049	110
$\mathcal{W} : n = 2$	100	200	$2 \cdot 10^4(-)$	10.000 $(-)$	0.000	18	0.999 $(-)$	0.001	29

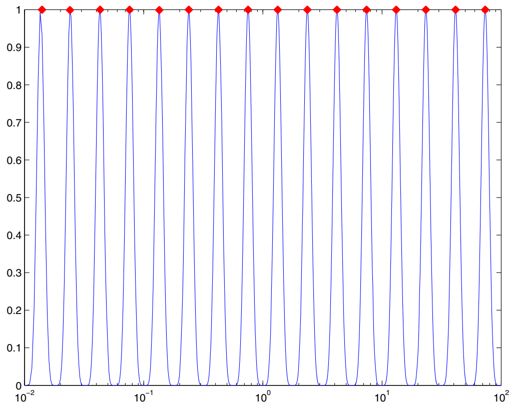

Figure 2: $\mathcal{M}$ function: species masters.

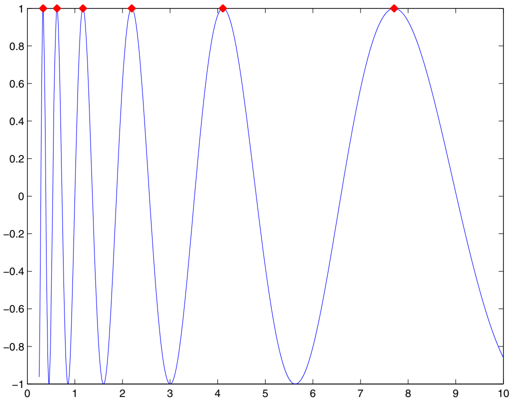

Figure 3: 1-D $\mathcal{V}$ function: species masters.

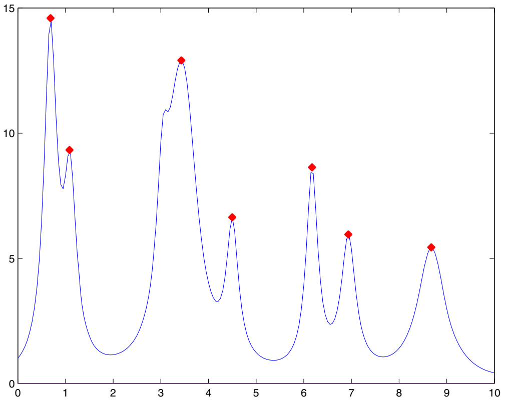

Figure 4: 1-D $\mathcal{S}$ function: species masters.

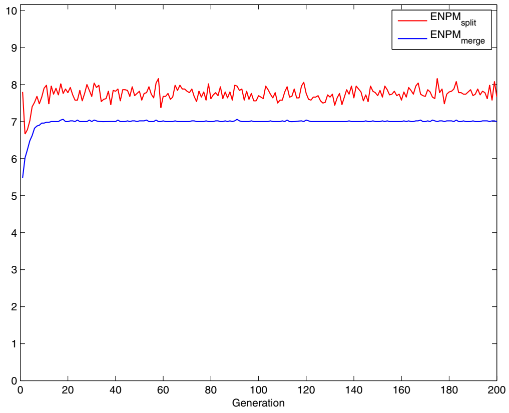

Figure 5: 1-D $\mathcal{S}$ function: ENPM.

mechanism to dynamically adapt the number of inner points to check with the Hill-Valley function during the fusion phase is proposed. From a speciation point of view, the use of such a function during the separation phase is aimed at discovering whether genetic divergence inside a species is occurring, while, during the fusion phase, Hill-Valley verifies

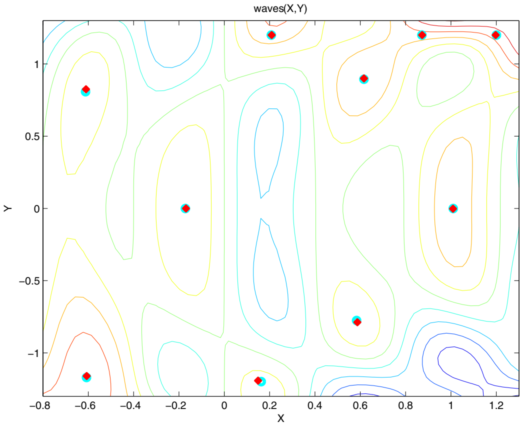

Figure 6: $\mathcal{W}$ function: species masters.

whether there is a reproductive compatibility between two different species and, therefore, they should be merged.

ASD has been tested on a set of problems with increasing degree of complexity, largely adopted in the literature to assess the performance of niching methods. The experimental results show that our method has a better ability to find all peaks with respect to other techniques proposed so far. The performance exhibited by ASD is very interesting when solving problems in which the perfect discrimination hypothesis is not verified and conventional sharing-based methods fail, due to the limitation of adopting a unique niche radius for all the niches. Eventually, the advantages of adopting ASD with respect to other niching methods become more evident when dealing with problems where the fitness landscape is highly irregular.

6. REFERENCES

[1] A. Della Cioppa, C. De Stefano, A. Marcelli. On the role of population size and niche radius in fitness sharing. *IEEE Transactions Evolutionary Computation*, 8 (6):580–592, 2004.

[2] A. Della Cioppa, C. De Stefano, A. Marcelli. Where are the niches? dynamic fitness sharing. *IEEE Transactions on Evolutionary Computation*, 11(4):453–465, 2007.

[3] B. Miller, M. J. Shaw . Genetic algorithms with dynamic niche sharing for multimodal function optimization. In *IEEE International Conference on Evolutionary Computation*, pages 786–791, 1996.

[4] B. Sareni, L. Krähenbühl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2 (3), 1998.

[5] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms.* Oxford University Press, Oxford, UK, 1996.

[6] C. K. Oei, D. E. Goldberg, S. J. Chang. Tournament selection, niching, and the preservation of diversity. *IlliGAL Report 91011*, 1991.

[7] D. Futuyma. *Evolutionary Biology.* Sinauer Associates, 1998.

[8] D.E. Goldberg, J.Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the 2nd International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[9] K. A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1975.

[10] J. Gan, K. Warwick. A genetic algorithm with dynamic niche clustering for multimodal function optimisation. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 248–255, 1998.

[11] J. Gan, K. Warwick. Dynamic niche clustering: A fuzzy variable radius niching technique for multimodal optimisation in gas. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 215–222, 2001.

[12] K.Deb, D. E. Goldberg. An investigation of niche and species formation in genetic function optimization. *ICGA 1989:42-50*, 1989.

[13] S. W. Mahfoud. Genetic drift in sharing methods. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, pages 67–72, 1994.

[14] S. W. Mahfoud. Population size and genetic drift in fitness sharing. In *FOGA*, pages 185–223, 1994.

[15] S. W. Mahfoud. *Niching Methods for Genetic Algorithms.* PhD thesis, University of Alabama, 1995.

[16] E. Mayr. *Systematics and the Origin of Species from the Viewpoint of a Zoologist.* Harvard University Press, October 1999.

[17] O. M. Shir. Dynamic niching in evolution strategies with covariance matrix adaptation. In *In Proceedings of the Congress on Evolutionary Computation, Piscataway*, pages 2584–2591. IEEE Press, 2005.

[18] O. M. Shir, M. Emmerich, T. Bäck. Adaptive niche radii and niche shapes approaches for niching with the cma-es. *Evol. Comp.*, 18:97–126, March 2010.

[19] O. M. Shir, T. Bäck. Niche radius adaptation in the cma-es niching algorithm. In *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature*, pages 142–151. Springer, 2006.

[20] A. Petrowski. A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 798–803, May 1996.

[21] P.J. Darwen, X. Yao. Every niching method has its niche: Fitness sharing and implicit sharing compared. In *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 398–407, London, UK, 1996. Springer-Verlag.

[22] C. Stoean, M. Preuss, R. Stoean, and D. Dumitrescu. Multimodal optimization by means of a topological species conservation algorithm. *IEEE Transactions on Evolutionary Computation*, 14(6):842–864, 2010.

[23] R. K. Ursem. Multinational evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation*, pages 1633–1640. IEEE Press, 1999.

[24] X. Yin, N. Germay. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In *Proceedings of International Conference on artificial neural networks and genetic algorithms*, pages 450–457, 1993.