We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Artificial Immune Systems, Dynamic Fitness Landscapes, and the Change Detection Problem

Hendrik Richter
HTWK Leipzig University of Applied Sciences
Germany

1. Introduction

To let biological processes, behaviors and structures inspire the design of problem solving algorithms and devices has been a prominent and persistent theme in engineering and applied sciences in the last few decades. Within this context, bio–inspired computing has taken a pioneering role. Fields such as evolutionary computing (1; 8; 25), artificial immune systems (4; 6; 43), membrane computing (29) or swarm systems (9; 22) have outgrown their infancy and found theoretical ground as well as important applications. The fact that and the way how these fields advanced into its current form is due to three major developments: (i) the upcoming of cheap, fast and reliable computational power in form of digital computers, (ii) the understanding that computational power in connection with implementing an algorithmic approach creates potent problem solvers, and (iii) the insight that biological systems can be fruitfully understood as information–processing units and can hence frequently be employed for computational and/or algorithmic proposes. This trend is of course not to be confused with computational biology, but it is highly related and probably unthinkable without the fundamental progress towards algorithmization and mathematization in biology, see e.g. (5; 16; 21; 38) for some recent discussion. Among the mentioned fields of bio–inspired computing, evolutionary algorithms and artificial immune systems play a unique role as their history is particularly long and the maturity reached is notably high. In this paper we will use both schemes in connection to solve the intertwined problem of maximum tracking and change detection in dynamic optimization.

For successfully solving dynamic optimization problems by evolutionary computation, there is a need for additions to the standard algorithmic structure, namely by operators maintaining and enhancing population diversity. Dynamic optimization here means that the topology of the associated fitness landscape changes with time. A considerable number of these operators for diversity management (for instance memory schemes, random immigrants or hyper–mutation (24; 26; 30; 33; 35; 39; 45)) can only be provoked and hence made to work properly if the points in time are known where the changes in the fitness landscape occur. So, the problem of change detection is of high practical relevance in solving dynamic optimization problems (3; 19; 27).

In principle, change detection is based on using information about the fitness values of points in the search space extracted from the fitness landscape. This extraction of information can be done in two ways. One is to use the fitness evaluations of the evolutionary algorithm's

population, which is called population–based detection, the other is to use additional measurement of the landscape's fitness on prescribed points (26), which is called sensor–based detection. Recently, a study (34) compared both types of methods. It has been shown that in populations–based detection there is no need for additional fitness function evaluation, but elaborate statistical tests have to be carried out. On the other hand, sensor–based detection can forgo these tests but at the cost of redoing measurements in the fitness landscape. Irrespective of the quality of the change detection, using statistical tests on population–based fitness data is sometimes generally objectionable. Although using non–parametric statistical tests fits the non-Gaussian nature of the fitness distribution, the tests require independent samples to be accurately employed. This independence might not be given for fitness distributions from sequential generations. The fitness values of next generation's population have their origin in the current generation and are only partly affected by the stochastic influence driving the evolutionary algorithm. This situation might be different if a (randomly induced) change in the fitness landscape has occurred, but again there is no guarantee that the resulting fitness distributions are statistically independent. Because of these reasons, it would be desirable to have alternative methods. A promising option is the use of methods from artificial immune systems, particularly negative selection (4; 7; 13; 18; 44). These algorithms have been successfully employed to solve similar problems, for instance network security, computer virus detection, network intrusion detection and fault diagnosis, see e.g. (13; 18; 43). In this paper, we present an immunological approach to change detection in dynamic fitness landscapes.

The paper is organized like this. In the next two sections, dynamic fitness landscapes are introduced and the change detection problem is defined. Then, in section 4, the immunological change detection scheme is given and its main components, shape space, affinity function, detector generation and detection processing, are described. In section 5, we present numerical experiments with the scheme and use receiver–operating characteristics (ROC) as well as the area under the ROC curve (AUC) as an analyzing tool. We end with summarizing the findings and pointing at future work.

2. Dynamics and fitness landscapes

The concept of fitness landscapes is an important approach to foster theoretical understanding in evolutionary computation (20; 40; 41). Such landscapes are traditionally considered to be static and can be obtained from either a genotype–to–fitness mapping or more generally by encoding all possible solutions of the optimization problem and giving a fitness value to each solution. All the possible solutions span a search space S, while a fitness function $f(s) : S \to \mathbb{R}$ provides every point $s \in S$ with a fitness value. In case of a genotype–to–fitness mapping, S coincides with the genotypical space. If the search space S is not metric, we must explain which solutions we would obtain if we were to slightly modify a possible solution $s \in S$ (and hence were to move it locally in the search space). This is done by a neighborhood structure $n(s)$ which gives every point in the search space a set of direct and possibly also more distant neighbors.

If the fitness landscape is dynamic all of its three defining ingredients – search space S, fitness function $f(s)$, neighborhood structure $n(s)$ – can, in principle, be changing with time. So, we additionally need for description a time set and mappings that tell how S, $f(s)$ and/or $n(s)$

evolve with time (31; 32; 37). Dynamic optimization problems considered in the literature so far address all these possibilities of change to some extend. Whereas a real alteration of the fundamental components of a search space such as dimensionality or representation (binary, integer, discrete, real, etc.) is really rare, a change in the feasibility of individuals is another and less substantial kind of a dynamic search space and is discussed within the problem setting of dynamic constraints (28; 36). The works on dynamic routing can partially be interpreted as a changing neighborhood structures (2; 15), while most of the work so far has been devoted to time–dependent fitness function (24; 26; 30; 33; 35; 39; 45), which will also be the focus of this paper.

In dynamic optimization problems (DOPs), the fitness landscape

$$f(x, k), k \geq 0 \qquad (1)$$

defined over a fixed bounded search space $M \subset \mathbb{R}^n$ with $x \in M$, changes with discrete time $k \in \mathbb{N}_0$. The DOP is solved by an evolutionary algorithm (EA) with population $P \in \mathbb{R}^n$ and generational time $t \in \mathbb{N}_0$. Its population dynamics can be described by the generation transition function

$$P(t+1) = \psi\left(P(t)\right), t \geq 0, \qquad (2)$$

which explains how a population $P(t+1)$ at generation $t+1$ originates from the population $P(t)$ at generation t. Both the time scales t and k work as a measuring and ordering tool for changes (t for changes in the population from one generation to the next, k for changes in the dynamic fitness landscape). As μ individuals $p_i(t) \in P(t)$, $i = 1, 2, \ldots, \mu$, populate the fitness landscape (1), they can be labeled with a fitness value $f(p_i(t), k)$. Both time scales are related in the solving process of the DOP by the change frequency $\gamma \in \mathbb{N}$ with

$$t = \gamma k. \qquad (3)$$

Usually, γ is considered to be constant for all generations t, but it might also be a function of k and even be different (for instance a positive integer realization of a random process) for every k. Note that we require more than one generation in between landscape changes, $\gamma > 1$, and hence $k = \lfloor \gamma^{-1} t \rfloor$.

3. The change detection problem

From (3), we see that the fitness landscape changes every γ generations. As the temporal patterns of these changes are assumed to be not explicitly known, our interest is now to infer from the fitness values

$$f_i(k) = f(p_i, k) \qquad (4)$$

of the individuals $p_i(t) \in P(t)|_{t=\gamma k}$ if a change in the fitness landscape has occurred or not. This we call the change detection problem in dynamic fitness landscapes using fitness data from the population. More explicitly, we want to detect the change point t_{cp} with the property $\exists x \in M$ for which

$$f(x, \lfloor \gamma^{-1}(t_{cp} - 1) \rfloor) \neq f(x, \lfloor \gamma^{-1} t_{cp} \rfloor). \qquad (5)$$

Our convention is to define the change point t_{cp} in the generational time scale t as we base the detection solely on the fitness values $f(x, \lfloor \gamma^{-1} t \rfloor)$ of the population $P(t)$. From (3) follows

$\lfloor \gamma^{-1}(t_{cp} - 1) \rfloor = k - 1$ and $\lfloor \gamma^{-1} t_{cp} \rfloor = k$, that is for every integer $\gamma^{-1} t$ there is a change in the fitness landscape (1).

The change point definition (5) says that a change in the fitness landscape has happened no matter how small and insignificant the alteration in the landscape's topology actually is. From a computational point of view this raises some problems regarding practical detectability. Change detection based on population data assumes that a change in the fitness landscape affects a substantial number of its points and makes them to increase or decrease their fitness values. Moreover, generally there is $P(t) \neq P(t - 1)$ so that we cannot check if $f(p_i, \lfloor \gamma^{-1}(t - 1) \rfloor) \neq f(p_i, \lfloor \gamma^{-1} t \rfloor)$ as $p_i(t - 1) \neq p_i(t)$.

Another aspect is that the given framework is unsuitable for discriminating between small but gradual changes and larger but abrupt changes. Such a distinction can only be made in terms of the fitness landscape considered. The treatment and discussion presented here is intended to apply for fitness landscapes that undergo abrupt and substantial changes in their topology. Hence, it can reasonably be assumed that these changes are practically detectable. We exclude small but gradual changes in the fitness landscape (for instance those resulting from the presence of noise and/or other perturbations in the fitness evaluation process). This is in line with the application context of the change detection scheme considered here. It should help to trigger and control the diversity enhancement and maintenance of the EA. For fitness landscapes with small but gradual changes an additional change–activated diversity management does not play a prominent role anyway; other types of EAs (particularly those emphasizing robustness such as self–adaption) are found to be more apt here.

As shown above it is generally not possible to verify condition (5) directly. The basic idea behind using the fitness values $f(p_i, k)|_{k = \lfloor \gamma^{-1} t \rfloor}$ of the population's individuals $p_i(t) \in P(t)$ is that these quantities form a fitness distribution

$$F(t) = (f_1(t), f_2(t), \ldots, f_\mu(t)) \qquad (6)$$

that can be analyzed by itself or compared to the ℓ preceding ones, that is creating a time window of width ℓ, $(F(t - 1), F(t - 2), \ldots, F(t - \ell))$. The fitness distributions can be regarded as a data stream and monitoring this data stream should make visible the normal optimum finding mode of the EA but also reflect that this normal mode is disrupted if a change in the fitness landscape has occurred (and hence results in a different pattern when evaluating $F(t)$ and $F(t - 1)$, respectively). Statistically speaking, the considered data set $F(t)$ can be regarded as coming from an unknown distribution $\mathcal{D}(t)$. This transforms the problem of change detection into the problem of testing whether the data sets $F(t)$ and $F(t - 1)$ or any data set created from any time window including $F(t)$ are coming from different distributions or not, which is known as statistical hypothesis testing. This connection is widely applied in solving change detection problems, e.g. (14; 23). The obvious question here is which test can tell us whether $\mathcal{D}(t)$ is different from $\mathcal{D}(t - 1)$ and if this difference necessarily and sufficiently implies that a change has occurred. In the language of statistical hypothesis testing, the test should ideally show only true changes, that is have no *false positives* and indicate all of them, that is have no *false negatives*. However, statistical hypothesis testing methods regularly require that the samples $F(t)$ are independent from $F(t - 1)$. This most likely is not the case if the samples come form a moving population of an evolutionary algorithm. The data set $F(t)$ includes the fitness value of an evolving population and represents two types of

interfering population dynamics. An evolving population moves (ideally and in the best case) monotonically towards the optima and in doing so changes its mean and standard deviation. Such a convergence behavior of the EA which is desired and the intended working mode is again a statistical phenomenon. Moreover, the fitness values of $F(t)$ are a direct result of the values of $F(t-1)$ and therefore can hardly be regarded as independent of each other. A second aspect is the reaction of a change in the fitness landscape, which more likely can be seen as independent if the dynamics is a stochastic process. For all these reasons we look for an alternative to statistical hypothesis testing. So, we intend to use ideas from artificial immunology for solving the change detection problem in dynamic fitness landscapes.

4. The immunological change detector

Artificial immune systems (AIS) are soft computing algorithms that take their inspiration from and mimicking working principles and functions of their biological counterparts (43; 44). AIS date back to the 80s (11) and were initiated by an increasing theoretical understanding of the natural immune system in connection with a strong interest in utilizing biological processes for computational proposes. These algorithms are capable of adaption, learning and memory and have been applied to problem solving in areas as different as classification, pattern recognition and data mining/analysis (6; 13; 18; 43). Among the different types of AIS, negative selection algorithms acclaimed a prominent role in solving so–called anomaly detection (7; 10; 13; 18; 42). Here, anomaly detection means to distinguish the normal behavior of a dynamic process, usually characterized by some (external) model, from anomalies defined by deviations from that model. In the following, we review negative selection and show how it can be used for detecting change in dynamic fitness landscapes.

Negative selection is anchored at the concept of a shape space that represents the observable features of the dynamic process for which a change in behavior needs to be detected. Within that shape space, we define a set of self elements that stem from the normal behavior. From these self elements, in turn, a set of detectors is derived that must not match any sample of the self set, usually by using some training data. Subsequently, the detectors are taken to decide if an incoming new feature data from the dynamic process is normal (self) or not (non–self). Thus, negative selection mimics the self/non–self discrimination of the natural immune system, see Fig. 1 which shows self and non–self elements in a two–dimensional shape space. We now describe the main components of the negative selection algorithm: shape space, affinity measure, detector generation and detection process.

i.) Shape space. The shape space of a negative selection algorithm is a representation of the data coming from the dynamic process under study and can be either a string over a finite alphabet (for instance a binary string, which has been used in a large number of previous works, see e.g. (18)) or real–valued (13). As the base for the change detection, the fitness distribution $F(t)$, is real–valued it seems straightforward to use a real–valued shape space $\mathcal{S} = [0,1]^m$ here, where m is its dimension. Dimensionality of the shape space is an important parameter influencing computational effort and performance of the detection scheme (42). The dimensionality of the fitness distribution $F(t)$ equals the number of individuals in the population μ. So, for the reason given above, it appears sensible to pre–process the data from the fitness distribution with the aims of both reducing dimensionality and extracting the most meaningful information about when the landscape has changed.
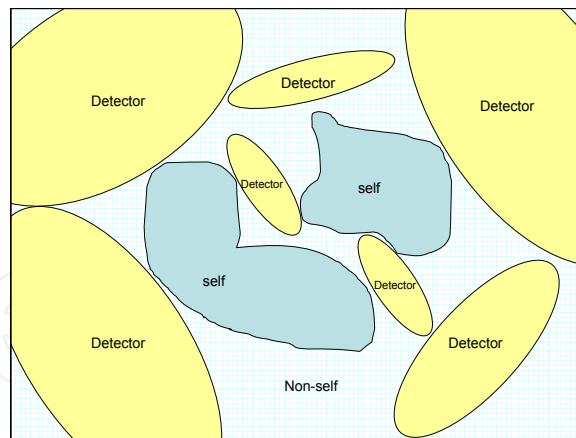
Fig. 1. Self and non–self elements and detectors for a two–dimensional shape space.

Among the several conceivable ways to do the pre–processing, we here consider the following scheme. The distributions $F(t)$ and $F(t-1)$ are independently sorted according to their fitness values. Then the $\frac{m}{2}$ highest and $\frac{m}{2}$ lowest ranked elements of both sorted distributions are taken, the elements coming from $F(t-1)$ are subtracted from the ones from $F(t)$ and finally these calculated quantities (which reflect the difference between two consecutive fitness distributions) are normalized to the interval $[0,1]$. Hence, the result is a point in the m–dimensional shape space. The given procedure of pre–processing the data is motivated by the common sense arguments that a change in the fitness landscape particularly affects the magnitude of the best and the worst fitness values and also that their relative difference from one generation to the subsequent one is telling if either a standard evolutionary search or a reaction to a landscape change has taken place. Note that by this pre–processing a metric on the fitness distributions is defined.

ii.) Affinity measure. The affinity measure states the degree of matching and recognition between elements (that are points) in the shape space. In other words, the affinity measure describes to what degree elements in the shape space differ. Every element in the shape space is defined by its center point $c \in [0,1]^m$ and a matching radius r. According to their function in the immunological detection and classification process, there are three types of elements in the shape space:

(a) self elements $se = (cs, rs)$ that are samples known to belong to the self space (usually from a training data set),

(b) detectors $dt = (cd, rd)$ that are derived from the self elements and must not match them, and

(c) incoming data samples $id = (ci, ri)$ that must be classified as belonging to either the self set or not.

Self elements, detectors and incoming data are also known by their immunological motivated terms self cells, antibodies and antigens, respectively. There is a large number of different affinity measures, see e.g. (18) for an overview. We here use Euclidean distance so that there is a match if $\|c_i - c_j\| < r_i + r_j$, with the indices i, j denoting different elements of the shape space.

iii.) Detector generation. In detector generation, self elements that come from training data with known self/non-self discrimination are used to calculate detector center points and radii. Similarly to the situation in computing the affinity measure, there is a multitude of different detector generation mechanisms, see e.g. (18). In some initial experiments (which are not reported here for sake of brevity), a scheme mainly using ideas from v–detectors (17) had shown best results and is considered here. This scheme has the advantage to address the problem of coverage of the non–self space in the generation process and to maximize the size of individual detectors to achieve a larger coverage. This comes at the cost of making the number of detectors actually created a (not predicable) result of the generation process and not a parameter to be set initially.

The scheme works like this. Input is a collection of self elements se_j, with $j = 1, 2, \ldots \#_{se}$, and $\#_{se}$ the total number of self elements. Further, we set a target coverage α and calculate a quantity $h = \frac{1}{1-\alpha}$. Then, the following steps are repeated. A candidate detector point and its radius are generated as a realization of a uniformly distributed random variable. It is tested if the detector matches any of the self elements. If so, its radius is shrunk so that any match is abolished. If not so, the radius is enhanced to the limit of any match. After that it is tested if the candidate detector with its updated radius is entirely covered by a detector that already passed this test. If so, the candidate is discarded, otherwise it is saved to the set of detectors that passed the coverage test. These steps are repeated until candidate detectors cannot pass the coverage test h times in a row. The saved detector candidates are accepted for the change detection process.

iv.) Detection process. After the training time in which detectors are generated as described above, the immunological change detector can be used for deciding if a change point according to eq. (5) has been reached or not by monitoring a metric defined on the fitness distributions $F(t)$ and $F(t-1)$. The necessary pre–processing is the same as the one given for the training phase above in Sec. 4.i. So, an incoming data sample $id(t) = (ci(t), ri)$ is produced every generation t, where the center point comes from the pre–processing and the self–radius ri is a quantity that defines the sensitivity of the detector, is to be set in initializing the immunological change detector and will be examined in the numerical experiments reported below. The affinity function is calculated by using a Euclidean distance measure

$$w(t) = \sum_{j=1}^{\#_{det}} \text{aff}(id(t), dt_j), \qquad (7)$$

where $\#_{det}$ is the total number of detectors dt_j, the individual affinities

$$\text{aff}(id(t), dt_j) = \begin{cases} \beta \|ci(t) - cd_j\| & \text{if} \quad \|ci(t) - cd_j\| < rd_j \\ |ri - rd_j| & \text{if} \quad \|ci(t) - cd_j\| < rd_j + ri \end{cases}$$

and β is a weighting factor. From the values $w(t)$, a change point can be concluded. Therefore, a threshold value $\bar{w}$ has to be set and a $w(t) > \bar{w}$ indicates a change.

v.) Performance evaluation. To evaluate the success and the quality of the change detection, the method of receiver–operating characteristics (ROC) curves can be used, e.g. (12). ROC curves are a tool for organizing and visualizing classifications together with their performances. So, they can be used to analyze and depict the relative trade–offs between

benefits of the schemes (correctly identified instances according to the classification) and costs (incorrect identifications). That makes them particularly useful to assess change detection schemes. The classification here is between positive and negative change detections. Hence, we can define the following performance metrics. If there is a positive detection and a change in the fitness landscape has happened, it is counted as *true positive* (tp), if a change happened but is not detected, it is a *negative positive* (np). If, on the other hand, no change has happened and the detection is negative, it is a *true negative* (tn), a positive detection in this situation yields a *false negative* (fn). For this two–by–two change classification, we obtain as the elements of performance metrics: the tp rate

$$tp \approx \frac{\text{correctly identified changes}}{\text{total changes}} \tag{8}$$

and $fn = 1 - tp$ as well as the fp rate

$$fp \approx \frac{\text{incorrectly identified changes}}{\text{total non changes}} \tag{9}$$

and $tn = 1 - fp$. In the ROC plot, the tp rate is given (on the ordinate) versus the fp rate (on the abscissa). Hence, the tp and fp rates for the immunological change detector for a given threshold value $\bar{w}$ give a point in the ROC space; ROC curves are obtained by plotting the rates for varying the threshold value $\bar{w}$.

5. Experimental results

In the following we report numerical experiments with the change detection schemes described above. In the experiments, we use as dynamic fitness landscape a "field of cones on a zero plane", where N cones with coordinates $c_i(k)$, $i = 1, \cdots, N$, are moving with discrete time $k \in \mathbb{N}_0$. These cones are distributed across the landscape and have randomly chosen initial coordinates $c_i(0)$, heights h_i, and slopes s_i, see Fig. 2. So, we get

$$f(x, k) = \max \left\{ 0, \max_{1 \leq i \leq N} [h_i - s_i \| x - c_i(k) \|] \right\}, \tag{10}$$

where the number of cones is $N = 50$ and its dimension $n = 2$. The dynamics of the moving sequence for the cones' coordinates is mostly normally random, that is each $c_i(k)$ for each k is an independent realization of a normally distributed random variable. In the last set of experiments we also consider different kinds of dynamics, namely regular, circle–like (and hence completely predictable) dynamics where the cones' coordinates form a circle and so return to the same place in search space after a certain amount of time, and chaotic dynamics where the cones' coordinates follow the trajectory of a nonlinear dynamical system with chaotic behavior, see (30) for details of this kind of setting of landscape dynamics.

Further, we employ an EA with a fixed number of $\lambda = 48$ individuals that uses tournament selection of tournament size 2, a fitness–related intermediate sexual recombination (which is operated λ times and works by choosing two individuals randomly to produce offspring that is the fitness–weighted arithmetic mean of both parents) and a standard mutation with the mutation rate 0.1. Note that the choice of the EA is of secondary importance as long as it
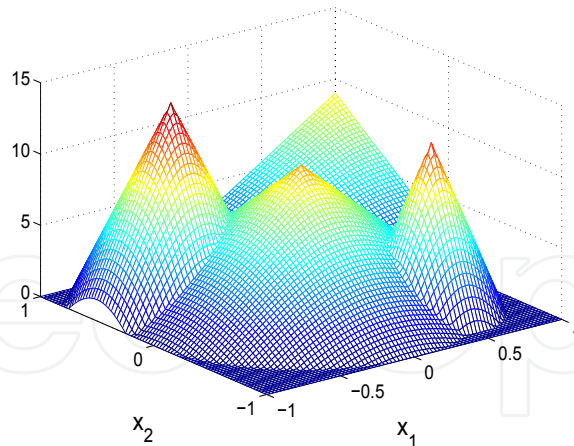
Fig. 2. Fitness landscape (10) for $n = 2$ and $N = 4$.

solves the DOP with some success. The immunological change detector was implemented as described in Sec. 4; we set $\beta = 5$.

Fig. 3 shows the detection process by monitoring the affinity function $w(t)$ calculated according to eq. (7), for shape space dimension $m = 8$, coverage $\alpha = 0.999$ and self radius $ri = 0.2$. We here use a training time of 20 generations. We see that even for this small training set, spikes in $w(t)$ can be used as indication for changes in the fitness landscape. As results of a second set of experiments, we give the ROC curves for varying dimension m and coverage α, see Fig. 4. Here, training time is 400 generations with total 1000 generations taken into account, self radius is $ri = 0.2$ and change frequency $\gamma = 20$. The tp and fp rates are calculated according to the eqs. (8) and (9) and are means over 100 repetitions. From the ROC curves we can deduce that the lower left point $(0, 0)$ represents a change detection that never produces any positive decision. It makes neither a false positive error nor yields any true positives. Likewise but opposite, a detection represented by the upper right point $(1, 1)$ only produces positive decisions with only true positives but also false positive errors in all cases. The line between these two points in the ROC space can be regarded as expressing a purely random guessing strategy to decide on whether or not a change has happened. Any classification that is represented by a point below that line is worse than random guessing, while classifications above are better, the more so if the point is more north–westwards of another, with the point $(0, 1)$ expressing perfect classification.

With this in mind, we see from Fig. 4 that good detection results are achieved as we obtain curves that climb from the point $(0, 0)$ vertically for a considerable amount of threshold values towards $(0, 1)$ before bending off to $(1, 1)$. Further, it can be seen in Fig. 4a that a higher coverage rate produces slightly better results, but the differences are not dramatic. Also, varying the shape space dimension leads to no substantial increasing in the detection success. The curve for $m = 8$ is even slightly lower than that for $m = 6$.

An important feature of the v–detector design used in this paper is that we get detectors with variable size but also that their exact number is not known beforehand. The number is a statistically varying result of the creating process and thus becomes a quantity that can be verified and studied experimentally. Fig. 5a shows the number of detectors $\#_{det}$ depending on the shape space dimension m, while Fig. 5b gives $\#_{det}$ over the average detector radii
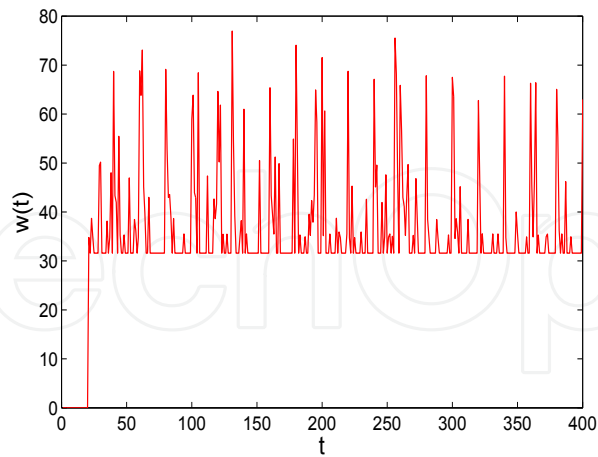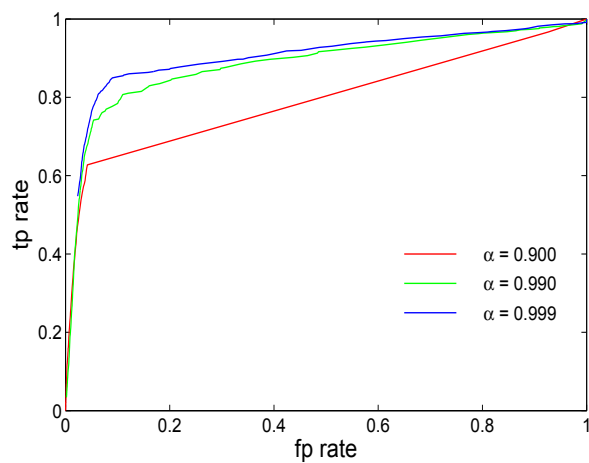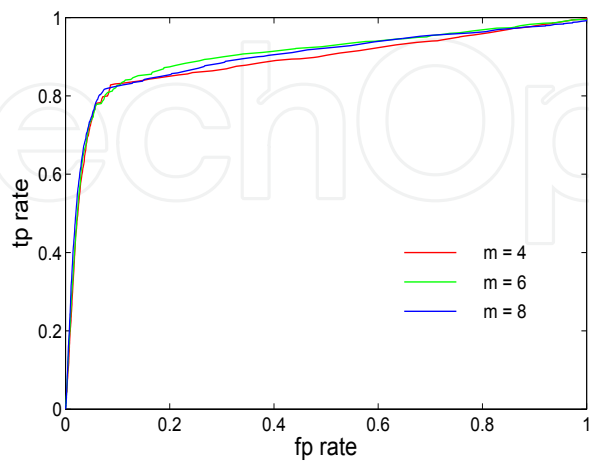
Fig. 3. Detection process for $\gamma = 20$, shape space dimension $m = 8$, coverage $\alpha = 0.999$ and self radius $ri = 0.2$.


a

b

Fig. 4. ROC curves for: a) varying coverage α; b) varying shape space dimension m.

r_d. Both figures show a scatter plot for 20 subsequent detector generations each. It can be seen that generally the larger the coverage α and the dimension m the more detectors are produced. Both facts appear quite logical as higher coverage requires more detectors, and higher dimension means larger generalized volume to be filled by the detectors. For dimensions becoming larger also the actual numbers of detectors are more spread, that is they increase in their range of variation. For $m = 8$, for instance, and $\alpha = 0.900$ the difference between the lowest value ($\#_{det} = 15$) and the highest ($\#_{det} = 33$) is 18, while for $m = 4$ the maximal difference is 8. Results that allow a similar interpretation can also be found for the number of detectors over the average detector radii r_d, see Fig. 5b. Smaller coverage α produces not only a smaller number of detectors but also detectors with smaller average radii, albeit the range of radii overlap. A possible explanation is that for a higher coverage a larger number of detectors candidates are produced and tested, and hence it becomes more likely that such with larger radii are finally found and selected.
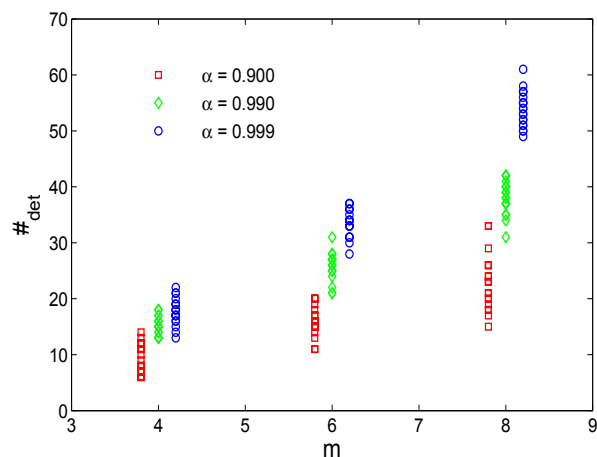

a
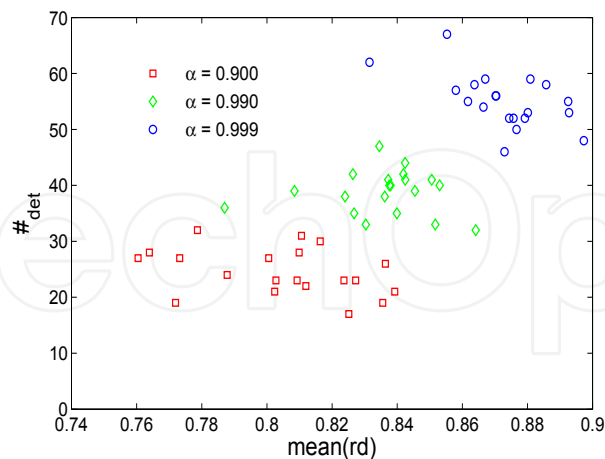
b

Fig. 5. Number of detectors $\#_{det}$ for different coverage α depending on: a) shape space dimension m; b) average detector radius mean(r_d) .

In a next experiment we study the effect of varying the self radius ri together with the influence of different kind of landscape dynamics. To get a numerical evaluation of the ROC curves, we calculate the area under the ROC curve (AUC), which is a measure for the detection
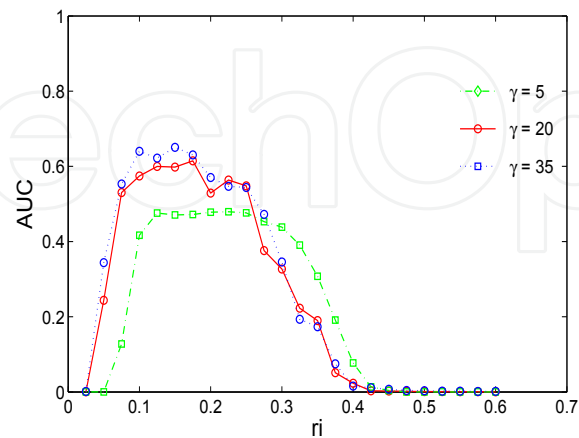
success (12). Since the AUC is a fraction of the unit square, its values are $0 \leq \text{AUC} \leq 1$. Moreover, since random guessing gives the diagonal line in the unit square, a well–working change detection should have values AUC > 0.5. Fig. 6 shows the AUC over the self radius
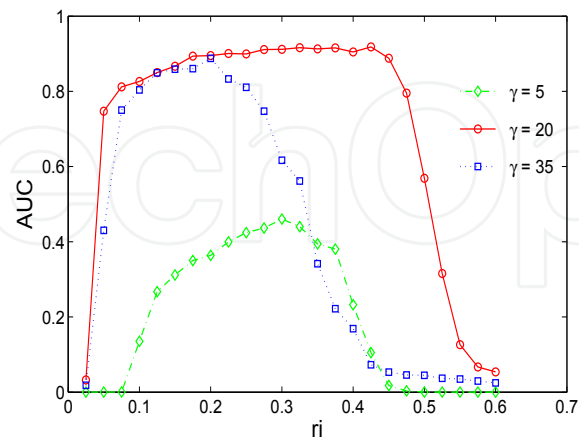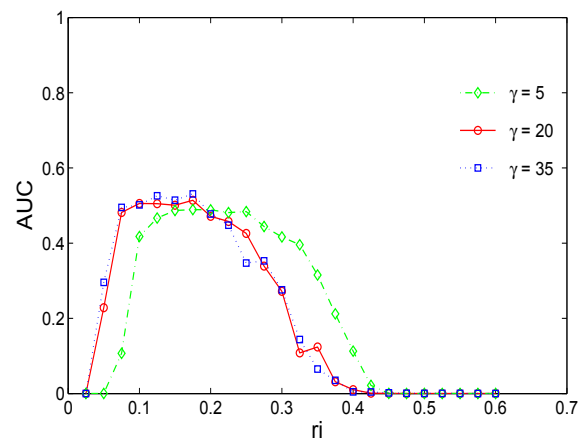
Fig. 6. AUC over self radius *ri* for different γ and different kinds of landscape dynamics, shape space dimension $m = 8$ and coverage $\alpha = 0.999$: a) random landscape dynamics; b) cyclic landscape dynamics; c) chaotic landscape dynamics.

ri for different change frequencies γ, shape space dimension $m = 8$ and coverage $\alpha = 0.999$, with Fig. 6a giving the results for random landscape dynamics, and Fig. 6b and Fig. 6c for regular dynamics (circle) and chaotic dynamics. The AUC is again the mean over 100 runs. We can generally see a reverse bath tube curve that indicates that a certain interval of *ri* is best. In some cases there is no significant differences in the performance within this interval, for instance for random dynamics and $\gamma = 20$, while for others, for instance random dynamics and $\gamma = 5$, a clear maximum can be observed. Further, the performance is generally better for random dynamics than for regular or chaotic, and a very fast landscape dynamics, $\gamma = 5$, may produce rather inferior detection results. These results appear to be a little bit surprising as it is know that for the optimization results for regular and chaotic dynamics we do not find such significant differences (33; 35). A possible explanation is that for random dynamics in the landscape the composition of the population is more homogeneous and the population dynamics more random–like. This in turn leads to better detector design and hence more effective change detection.

6. Conclusions

We have presented an immunological approach for solving the change detection problem in dynamic fitness landscape. A negative selection algorithm has been used to decide on whether or not the fitness landscape has changed. This is solely done with fitness information from the population on a sample base. Numerical experiments evaluated by receiver–operating characteristics (ROC) curves have shown the efficiency of the scheme. An important feature of the approach is that it does not directly uses any statistical test on which requirements could be imposed regarding the independence of the samples. In future work it would be interesting to compare and combine the immunological change detector with statistical tests. This could be connected with a study of dynamic fitness landscapes with higher dimension and complexity.

7. References

[1] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, New York Oxford, 1996.

[2] P. A. N. Bosman and H.Poutré. Computationally intelligent online dynamic vehicle routing by explicit load prediction in an evolutionary algorithm. In: T. P. Runarsson, H. G. Beyer, E. Burke, J. J. Merelo Guervós, L. D. Whitley, X. Yao (eds.), Parallel Problem Solving from Nature–PPSN IX, Springer–Verlag, Berlin Heidelberg New York, 312–321, 2006.

[3] J. Branke. *Evolutionary Optimization in Dynamic Environments*, Kluwer, Dordrecht, 2002.

[4] L. N. de Castro and J. Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer–Verlag, Berlin Heidelberg New York, 2002.

[5] J. E. Cohen. Mathematics is biology's next microscope, only better; Biology is mathematics' next physics, only better. PLoS Biol 2(12): e439. doi:10.1371/journal.pbio.0020439, 2004.

[6] D. Dasgupta. *Artificial Immune Systems and their Applications*, Springer–Verlag, Berlin Heidelberg New York, 1999.

[7] P. D'haeseleer, S. Forrest and P. Helman. An immunological approach to change detection: Algorithms, analysis and implications. In: *Proc. 1996 IEEE Symposium on Security and Privacy*, IEEE Computer Society, Los Alamitos, CA, 110–119, 1996.

[8] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing.* Springer–Verlag, Berlin Heidelberg New York, 2003.

[9] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*, Wiley, Chichester, 2006.

[10] F. Esponda, S. Forrest and P. Helman. A formal framework for positive and negative detection schemes. *IEEE Trans. Syst. Man Cybern.*, B34: 357–373, 2004.

[11] J. D. Farmer, N. H. Packard and A. S. Perelson. The immune system, adaption, and machine learning. *Physica*, D22: 187–204, 1986.

[12] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27: 861–874, 2006.

[13] F. Gonzalez and D. Dasgupta. Anomaly detection using real–valued negative selection. *Genetic Programming and Evolvable Machines*, 4: 383–403, 2003.

[14] J. G. De Gooijer. Detecting change–points in multidimensional stochastic processes. *Computational Statistics & Data Analysis*, 51: 1892–1903, 2006.

[15] J. I. van Hemert and J. A. La Poutré. Dynamic routing problems with fruitful regions: Models and evolutionary computation. In: X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo Guervós, J. A. Bullinaria, J. Rowe, P. Tino, A. Kabán, H. P. Schwefel (eds.), *Parallel Problem Solving from Nature–PPSN VIII*, Springer–Verlag, Berlin Heidelberg New York, 692–701, 2004.

[16] J. Leo van Hemmen. Biology and mathematics: A fruitful merger of two cultures. *Biol. Cybern.*, 97: 1–3, 2007.

[17] Z. Ji and D. Dasgupta. Estimating the detector coverage in a negative selection algorithm. In: H. G. Beyer et al. (eds.), *Proc. Congress on Genetic and Evolutionary Computation, GECCO 2005*, ACM Press, New York, 281–288, 2005.

[18] Z. Ji and D. Dasgupta. Revisiting negative selection algorithms. *Evolutionary Computation*, 15: 223–251, 2007.

[19] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments–A survey. *IEEE Trans. Evol. Comput.*, 9: 303–317, 2005.

[20] L. Kallel, B. Naudts and C. R. Reeves. Properties of fitness functions and search landscapes. In: L. Kallel, B. Naudts, A. Rogers (eds.), *Theoretical Aspects of Evolutionary Computing*, Springer–Verlag, Berlin Heidelberg New York, 177–208, 2001.

[21] E. F. Keller. A clash of two cultures. *Nature*, 445: 603, 2007.

[22] J. Kennedy, R. C. Eberhart and Y. Shi. *Swarm Intelligence*, Morgan Kaufmann, San Francisco, CA, 2001.

[23] D. Kifer, S. Ben-David and J. Gehrke. Detecting change in data streams. In: M. A. Nascimento et al. (eds.), *Proc. VLDB 2004*, Morgan Kaufmann, San Francisco, 180–191, 2004.

[24] R. Mendes and A. Mohais. DynDE: Differential Evolution for dynamic optimization problems. In D. Corne (ed.), *Proc. Congress on Evolutionary Computation, IEEE CEC 2005*, IEEE Press, Piscataway, NJ, 2808–2815, 2005.

[25] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer–Verlag, Berlin Heidelberg New York, 1996.

[26] R. W. Morrison and K. A. De Jong. Triggered hypermutation revisited. In: A. Zalzala et al. (eds.), *Proc. Congress on Evolutionary Computation, IEEE CEC 2000*, IEEE Press, Piscataway, NJ, 1025–1032, 2000.

[27] R. W. Morrison. *Designing Evolutionary Algorithms for Dynamic Environments*, Springer–Verlag, Berlin Heidelberg New York, 2004.

[28] T. T. Nguyen and X. Yao. Benchmarking and solving dynamic constrained problems. In: A. Tyrrell (ed.), *Proc. Congress on Evolutionary Computation, IEEE CEC 2009*, IEEE Press, Piscataway, NJ, 690–697, 2009.

[29] G. Paun, G. Rozenberg and A. Salomaa. *The Oxford Handbook of Membrane Computing*, Oxford University Press, Oxford, 2010.

[30] H. Richter. A study of dynamic severity in chaotic fitness landscapes. In: D. Corne (ed.), *Proc. Congress on Evolutionary Computation, IEEE CEC 2005*, IEEE Press, Piscataway, NJ, 2824–2831, 2005.

[31] H. Richter. Evolutionary optimization in spatio–temporal fitness landscapes. In: T. P. Runarsson, H. G. Beyer, E. Burke, J. J. Merelo Guervós, L. D. Whitley, X. Yao (eds.), *Parallel Problem Solving from Nature–PPSN IX*, Springer–Verlag, Berlin Heidelberg New York, 1–10, 2006.

[32] H. Richter. Coupled map lattices as spatio–temporal fitness functions: Landscape measures and evolutionary optimization. *Physica*, D237: 167–186, 2008.

[33] H. Richter and S. Yang. Memory based on abstraction for dynamic fitness functions. In: M. Giacobini et al. (eds.), *EvoWorkshops 2008: Applications of Evolutionary Computing*, Springer–Verlag, Berlin Heidelberg New York, 597–606, 2008.

[34] H. Richter. Detecting change in dynamic fitness landscapes. In A. Tyrrell (ed.), *Proc. Congress on Evolutionary Computation, IEEE CEC 2009*, IEEE Press, Piscataway, NJ, 1613–1620, 2009.

[35] H. Richter and S. Yang. Learning behavior in abstract memory schemes for dynamic optimization problems. *Soft Computing*, 13: 1163–1173, 2009.

[36] H. Richter. Memory design for constrained dynamic optimization problems. In: C. Di Chio et al. (eds.), *Applications of Evolutionary Computation - EvoApplications 2010*, Springer–Verlag, Berlin Heidelberg New York, 552–561, 2010.

[37] H. Richter. Evolutionary optimization and dynamic fitness landscapes: From reaction–diffusion systems to chaotic CML. In: I. Zelinka, S. Celikovsky, H. Richter, G. Chen (eds.), *Evolutionary Algorithms and Chaotic Systems*, Springer–Verlag, Berlin Heidelberg New York, 409–446, 2010.

[38] D. P. Rowbottom. Models in biology and physics: What's the difference? *Found. Sci.*, 14: 281–294, 2009.

[39] A. Simões and E. Costa. Variable–size memory evolutionary algorithm to deal with dynamic environments. In: M. Giacobini et al. (eds.), *EvoWorkshops 2007: Applications of Evolutionary Computing*, Springer–Verlag, Berlin Heidelberg New York, 617–626, 2007.

[40] T. Smith, P. Husbands, P. Layzell and M. O'Shea. Fitness landscapes and evolvability. *Evolut. Comput.*, 10: 1–34, 2002.

[41] P. F. Stadler and C. R. Stephens. Landscapes and effective fitness. *Comm. Theor. Biol.* 8: 389–431, 2003.

[42] T. Stibor, J. Timmis and C. Eckert. A comparative study of real–valued negative selection to statistical anomaly detection techniques. In: C. Jacob et al. (eds.) *Artificial Immune Systems: Proc. ICARIS-2005*, Springer–Verlag, Berlin Heidelberg New York, 262–275, 2005.

[43] J. Timmis, M. Neal and J. Hunt. An artificial immune system for data analysis. *Biosystems*, 55: 143–150, 2000.

[44] J. Timmis, A. Hone, T. Stibor and E. Clark. Theoretical advances in artificial immune systems. *Theoretical Computer Science*, 403: 11–32, 2008.

[45] R. Tinós and S. Yang. A self–organizing random immigrants genetic algorithm for dynamic optimization problems. *Genetic Programming and Evolvable Machines*, 8: 255–286, 2007.
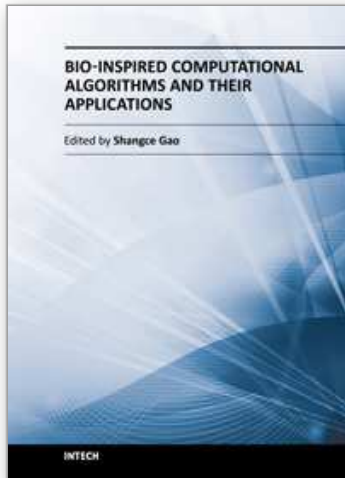
Bio-Inspired Computational Algorithms and Their Applications

Edited by Dr. Shangce Gao

ISBN 978-953-51-0214-4

Hard cover, 420 pages

Publisher InTech

Published online 07, March, 2012

Published in print edition March, 2012

Bio-inspired computational algorithms are always hot research topics in artificial intelligence communities. Biology is a bewildering source of inspiration for the design of intelligent artifacts that are capable of efficient and autonomous operation in unknown and changing environments. It is difficult to resist the fascination of creating artifacts that display elements of lifelike intelligence, thus needing techniques for control, optimization, prediction, security, design, and so on. Bio-Inspired Computational Algorithms and Their Applications is a compendium that addresses this need. It integrates contrasting techniques of genetic algorithms, artificial immune systems, particle swarm optimization, and hybrid models to solve many real-world problems. The works presented in this book give insights into the creation of innovative improvements over algorithm performance, potential applications on various practical tasks, and combination of different techniques. The book provides a reference to researchers, practitioners, and students in both artificial intelligence and engineering communities, forming a foundation for the development of the field.

INTECH

open science | open minds