Characterizing the environmental change in robust optimization over time

Haobo Fu, Bernhard Sendhoff, Ke Tang

2012

Preprint:

This is an accepted article published in IEEE Congress on Evolutionary Computation. The final authenticated version is available online at:
https://doi.org/[DOI not available]

Characterizing Environmental Changes in Robust Optimization Over Time

Haobo Fu*, Bernhard Sendhoff†, Ke Tang‡ and Xin Yao*‡
*CERCIA, School of Computer Science, University of Birmingham, UK
†Honda Research Institute Europe, Offenbach, DE
‡Joint USTC-Birmingham Research Institute in Intelligent Computation and Its Applications,
School of Computer Science and Technology, University of Science and Technology of China, CN
Emails: hxf990@cs.bham.ac.uk, bernhard.sendhoff@honda-ri.de, ketang@ustc.edu.cn and x.yao@cs.bham.ac.uk

***Abstract*—Evolutionary dynamic optimization has been drawing more and more research attention, and yet most work in this area is focused on Tracking Moving Optimum (TMO), which is to optimize the current fitness function at any time point. Recently, we proposed a more practical way to solve dynamic optimization problems, which is referred to as Robust Optimization Over Time (ROOT). In ROOT, we are trying to find solutions whose performances are acceptable over more than one environmental state, i.e., fitness functions. Before any development of benchmarks or algorithms for ROOT, it is necessary to have some understanding of what aspects of an environment can change and more importantly how these changes influence the solving of ROOT problems. In this paper, we develop a number of measures which can be used to characterize and analyse the underlying changing environment in the framework of ROOT. We test these measures on several benchmark problem instances, and it is shown that these measures are able to differentiate different dynamics effectively and provide useful information about what kind of algorithms might or might not suit certain dynamic environments.**

I. Introduction

Most research in evolutionary dynamic optimization has been focusing on Tracking Moving Optimum (TMO), which is to optimize the current fitness function at any time point [1], [2], [3], [4]. However, in [5], we pointed out a probably more practical way to solve dynamic optimization problems, the corresponding methodology of which is referred to as Robust Optimization Over Time (ROOT). Consider a dynamic optimization problem, whose fitness function changes over time with stationary periods between two successive changes, i.e., the dynamic optimization problem can be represented as a sequence of fitness functions $(F_1, F_2, ..., F_N)$ during the investigated time interval $[t_0, t_{end})$. For TMO approaches, the goal is to find an optimal solution for each fitness function resulting a sequence of solutions $(X_1, X_2, ..., X_N)$, while in ROOT we are aiming to find solutions whose performances are satisfactory over more than one fitness function. In other words, the resulting solution sequence $(X'_1, X'_2, ..., X'_L)$ with $L \leq N$ in ROOT may have solutions which can be used for two or more successive fitness functions with solutions' performances subject to certain constraints. For instance, suppose solution X'_1 is used in the first two environmental states when the corresponding fitness functions are F_1 and F_2 respectively, and we determine the solution X'_1 before the first environmental state changes into the second one. Solutions in ROOT whose performances are desirable for two or more successive environmental states are termed *robust solutions over time*. For the sake of simplicity, we employ the matrix U to denote which fitness functions a solution is used for. The element in U is either 1 or 0 with $U_{ij} = 1$ meaning the ith solution is used for the jth fitness function. Since we solve dynamic optimization problems in an on-line manner, whenever we start to optimize the jth fitness function we check whether the solution used for $(j-1)$th fitness function is satisfying in the jth environmental state. If so, we do not need to provide any new solution for the jth fitness function. To make the objective clearer in ROOT, based on the notations mentioned above, we formulate the single objective ROOT problem as follows:

$$
\begin{aligned}
&\textbf{Min } L = |S|, \; S = (X'_1, X'_2, ..., X'_L). \\
&\textbf{S.t. } \forall\, i \,\&\, j, \; \left|\frac{F_j(X'_i) - opt_j}{opt_j}\right| \leq \delta_{app}, \; if \; U_{ij} = 1, \\
&\quad \forall\, i \,\&\, m, \; U_{im} = \begin{cases} 1, & \text{if } b_i \leq m \leq e_i \\ 0, & \text{otherwise} \end{cases}, \\
&\quad \forall\, k, \; b_{k+1} = e_k + 1, \; b_1 = 1, \; e_L = N, \\
&\quad 1 \leq i \leq L, \; 1 \leq j \leq N, \; 1 \leq k \leq L-1,
\end{aligned} \tag{1}
$$

where L is the cardinality of solution sequence S, and opt_j is the optimal solution's fitness in the jth environmental state. δ_{app} is the parameter which constrains the performances of solutions in their used environmental states. b_i and e_i denote respectively the indices of the first and last fitness functions solution X'_i is used for. It is noted that this problem definition requires the information of optimal solutions in each environmental state.

As argued in [6] and [7], the nature of environmental changes is crucial to the understanding and solving dynamic optimization problems. Therefore, a number of measures were proposed in [6] and [7] respectively, both of which can be used to determine what aspects of the underlying fitness landscapes change and more importantly what kind of TMO algorithms might be appropriate for the investigated problems. In our case, the underlying environmental change is also able to influence enormously the design of proper algorithms for ROOT problems due to the following reasons:

- If the environmental change is too 'severe', there might not exist any *robust solution over time* at all considering the setting of δ_{app}. As a result, the concept of ROOT makes no sense.
- It may happen that in some changing environments optimal solutions in previous environmental states tend to have good performances in later environmental states. In such dynamic environments, optimizing the current fitness function tends to have the same effect as finding *robust solutions over time*. Therefore, existing TMO approaches may succeed in solving ROOT problems, and there is no need to develop additional methods for ROOT problems.
- If the dynamic of environmental changes exhibits the pattern that the fitness changes of solutions tend to be stable across successive environmental changes, this information then can be exploited so that the search for *robust solutions over time* can be focused in some particular regions of solution space.

The aim of this paper thus is to characterize and analyse environmental changes for the purpose of solving ROOT problems using the notion of fitness landscape [8]. Accordingly, we propose a set of measures which can be used to understand what aspects of fitness landscapes change due to underlying environmental changes. For each measure, we discuss its implications for what properties those ROOT problems with certain measure values may have and what kind of methods might or might not be appropriate for solving such ROOT problems.

The remainder of the paper is structured as follows. We review some related work in Section II, which is about measuring and characterising environmental changes for the purpose of TMO. In Section III, a set of measures are proposed to analyse and characterize environmental changes when the objective for dynamic optimization is ROOT rather than TMO. Experimental studies are reported in Section IV to show whether these measures are able to differentiate environments with different dynamics and more importantly to provide some hints as to what kind of algorithms might or might not be appropriate for certain dynamic environments. Finally, conclusions and future work are discussed in Section V.

II. Related Work

Previous studies on classifying dynamic environments mainly serve to inspire the design of artificial dynamic optimization benchmark problems without necessarily quantifying those characteristics. Besides, all the work were carried out in the objective of TMO. De Jong [9] identified four main categories of dynamic optimization problems. The first category involves problems in which the optimal solution moves slightly and gradually due to environmental changes. The second category consists of problems where previous locally optimal solutions become optimal in new environments. In the third category, problems exhibit a cyclic pattern where the optimal solutions in the future return to one of those optimal solutions in the past, while in the fourth category problems undergo severe and abrupt environmental changes. In [6], Weicker introduced a mathematical framework to describe and categorize dynamic fitness functions. Within the framework, a dynamic fitness function consists of several static fitness functions, each of which dynamic rules (coordinate transformations and fitness rescalings) are given to. Based on the dynamic rules, several dynamic problem properties were discussed, and four severity measures were proposed. It was then showed experimentally, using a $(1, \lambda)$ Evolution Strategy for an abstract problem, that for dynamic optimization problems with different severities the proper algorithmic techniques and population sizes can vary significantly. Additionally, Branke [10] used the following four criteria to characterize changing environments: *frequency of change*, *severity of change*, *predictability of change* and *cycle length with accuracy*. The *frequency of change* measures how often a environment changes, which is often calculated using the number of fitness evaluations between two consecutive changes. The distance from the new optimum to the old one is often used as a measurement of *severity of change*. The *predictability of change* depends on whether the change follows a pattern or is completely random. If after a period of time, the environment returns to a previous state, then the measure of *cycle length with accuracy* is employed to quantify how long does it take and how accurate it returns to a former state.

Not until recently, Branke [7] proposed to develop a number of measures which serve to characterize and analyse the nature of changes of any dynamic optimization problem. In other words, given an artificial or real-world dynamic optimization problem, these measures are used to tell what aspects of fitness landscape change due to underlying environmental changes, and more importantly what kind of algorithms might be appropriate for dynamic optimization problems with such dynamics. These measures include *severity of change*, *fitness correlation*, *fitness change correlation* and *usefulness of previous good solutions*. The distance between the optimum before the environmental change and the optimum after the change is used to measure the *severity of change*. In cases where the optimum is impossible to determine, the *severity of change* is calculated using estimated optimum. As Branke said in [7], if the *severity of change* is low which means the new optimum is not far away from the old one, approaches which generate diversity through mutation after environmental change, like hypermutation [11], would have a good performance. Otherwise, such methods would definitely fail. The *fitness correlation* calculates the correlation coefficient of fitnesses or rankings of samples in the solution space before and after an environmental change. If the value of the measure is high which means previous good solutions tend to remain good in the future, memory mechanisms like [12] storing previous good solutions may improve the solving of the current fitness function. The *fitness change correlation* looks at correlation coefficient between fitness changes of similar points in search space. The high value of the measure simply means similar points undergo similar fitness changes and vice versa. Finally, the usefulness of good solutions found in previous environmental states, e.g., the best k local optimum in each environmental state, as the start points for local search in following environmental states are examined using the measure of *usefulness of previous good solutions*.

There are also some theoretical research on analysis of fitness landscapes for static optimization problems [13] and investigations on how dynamics of changing fitness landscapes influence the evolution process of evolutionary algorithms [14], [15].

III. Characterizing Environmental Changes for ROOT

In this section, we propose a number of measures which can be used to characterize the nature of environmental changes specially for the solving of dynamic optimization problems under the concept of ROOT. Within these measures, different aspects of environmental changes are quantified. What is more, we discuss some implications of those measures for whether existing TMO methods would succeed or what kind of algorithms might be successful in solving ROOT problems with certain characteristics.

Most of the measures require a large number of samples taken from solution space. To improve estimation accuracy, we employ stratified sampling as Branke did in [7]. In cases where correlation coefficient is calculated, we use the equation $R_{XY} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{(n-1)\sigma_x \sigma_y}$, where x_i and y_i ($i = 1, 2, ..., n$) are samples of two random variables X and Y respectively, with the corresponding sample means $\overline{x}$ and $\overline{y}$, and sample standard deviations σ_x and σ_y. When we talk about distance between solutions, we simply mean Euclidean distance since real-valued representations are employed in the experiment in Section IV. Besides, in cases where Local Hill Climbing (LHC) technique is required, we use steepest ascent climbing (maximization problem is considered in this paper) with the fixed step size equalling $\frac{1}{5}$ of the search range in each dimension.

A. Optimum Degradation

Since our objective in ROOT is finding *robust solutions over time* for dynamic optimization problems, a natural question would be whether existing TMO approaches would suffice for the task. The answer to the question is very much related to how optimum from previous particularly last environmental states performs in later environmental states. Therefore, we propose the measure of *optimum degradation* to quantify that performance. The measure is calculated using the current fitness of a previous optimum $F_c(X_p^*)$, $1 \leq p \leq c$, and the current optimal solution's fitness opt_c: $|\frac{F_c(X_p^*) - opt_c}{opt_c}|$. The higher the value of this measure is, the worse the previous optimum performs in the current environmental state, and the worse TMO approaches might perform in finding *robust solutions over time*.

B. Estimated Optimum Degradation

As we can see, the calculation of *optimum degradation* requires the information of optimum in each environmental state, which is often impossible to have in many real-world problems. Therefore, it would be helpful to have a measure to estimate the *optimum degradation* without knowing the exact optimum. The measure here is called *estimated optimum degradation*, for which we use the equation of *optimum degradation* but replace the true optimum with the solution obtained by conducting LHC on the best sample in corresponding environmental states.

C. Optimum Survival Length

The measure of *optimum degradation* does not really estimate how long a previous optimum can be used for. Given a value of δ_{app}, we are more interested in how many consecutive environmental states the performance of a previous optimum remains satisfactory. A solution X' survives in the ith environmental state if and only if it satisfies the equation $|\frac{F_i(X') - opt_i}{opt_i}| \leq \delta_{app}$. The number of successive environmental states a previous optimum survives is returned as the measure of *optimum survival length*. Clearly, the larger this measure is, the shorter the resulting solution sequence will be if a TMO approach is employed for the ROOT problem.

D. Estimated Optimum Survival Length

Like the measure of *optimum degradation*, the measure of *optimum survival length* also requires the knowing of optimum in each environmental state. In situations where the information about optimum is not available, we use the measure of *estimated optimum survival length*. The calculation of *estimated optimum survival length* is the same as *optimum survival length* except that the solution obtained by conducting LHC on the best sample is used as the replacement of real optimum in corresponding environmental states.

E. Survival Rate

While the above mentioned measures look at optimal solutions only, these might not be exact estimations of corresponding characteristics since often TMO approaches are not able to find optimal solutions but good and near optimal ones. Therefore, we propose to draw a set of good samples to estimate how many proportion of good solutions in last environmental state remain to be good in the next environmental state by the measure of *survival rate*. Given a value of δ_{app}, this measure is calculated first by drawing a number of samples before an environmental change. The best fitness of these samples is identified as f_{best} and all samples which have a fitness no smaller than $f_{best} * (1 - \delta_{app})$ form the set $A_{\delta_{app}}$. Then we identify the best fitness of newly generated samples as f'_{best} after the environmental change. Finally, the measure of *survival rate* is returned as the ratio of solutions in $A_{\delta_{app}}$, which have fitness better than $f'_{best} * (1 - \delta_{app})$ after the environmental change. The measure of *survival rate* is important in that it gives a general estimation of whether or how 'many' *robust solutions over time* exist in a changing environment.

F. Fitness Correlation

This measure investigates the correlation coefficient of fitnesses of samples before and after an environmental change. If the correlation coefficient is high, this means good solutions in the current environment will remain to be good in the next environment, and TMO approaches would still be a good choice for ROOT problems. However, the real difficulty for ROOT will be how to select, from those good solutions, a particular one which will survive for the most consecutive environmental states.

G. Fitness Change Correlation Over Time

In this measure, we look at the correlation coefficient of fitness changes from two successive environmental changes for the same sample set. If the measure is high, then the solutions' fitness in search space tends to undergo similar changes over time. In other words, in such dynamics, if a solution's fitness decreases due to one environmental change, it is very likely that the solution's fitness would keep going down in following environmental changes. The information about this measure is important in the sense that based on the measure we could focus our search in some particular regions over others. For instance, if we obtain several locally optimal solutions in different regions of search space, we would prefer the one whose performance will become better in the next environmental state, according to the prediction on the measure.

H. Fitness Change Correlation of Similar Points

In order to calculate the measure of *fitness change correlation over time*, some solutions need to be re-evaluated every time the environment changes. However, whether these solutions' fitness changes could represent those of similar (in terms of distance in solution space) solutions largely depends on whether the fitness landscape changes coherently. By coherence, we look into fitness changes of similar solutions. We calculate the measure of *fitness change correlation of similar points* as follows. We first obtain the fitness changes of n samples. After this, for each of these n generated samples, we pick a random point with distance d away as its neighbour. Finally, we return the correlation coefficient of fitness changes between these n samples and their corresponding neighbour points.

It might be worth noting that the set of measures proposed here are probably the most important ones for characterizing and analysing environmental changes in ROOT problems, but they may not be complete. Other measures could exist, which would help to understand the environmental dynamics in ROOT.

IV. Experimental Study

In this section, we first review the modified moving peaks benchmark proposed in [5]. Then we briefly discuss the influence of settings of some benchmark parameters on the underlying ROOT problems. Based on the discussion, we suggest three problem instances of the modified moving peaks benchmark, which are used as the test bed for the proposed measures in Section III. Finally, we report the calculated measures on these three problem instances and explain their meanings respectively.

A. Modified Moving Peaks Benchmark

The modified moving peaks benchmark is derived from Branke's moving peaks benchmark [16] by allowing each peak having its own change severities. Basically, the modified moving peaks benchmark consists of several peak functions whose height, width and center position change over time, which can be described as:

$$F_t(\vec{X}) = \max_{i=1}^{i=m}\{H_i(t) - W_i(t) * ||\vec{X} - \vec{C}_i(t)||_2\}, \quad (2)$$

where $H_i(t)$, $W_i(t)$ and $\vec{C}_i(t)$ denote the height, width and center of the ith peak function, $\vec{X}$ is the design variable, and m is the total number of peaks. Besides, the timer t is usually denoted as the index of environmental state. $H_i(t)$, $W_i(t)$ and $\vec{C}_i(t)$ change simultaneously after a certain period of time Δe which is usually measured by the number of fitness evaluations:

$$\begin{aligned} H_i(t+1) &= H_i(t) + height\_severity_i * N(0,1), \\ W_i(t+1) &= W_i(t) + width\_severity_i * N(0,1), \\ \vec{C}_i(t+1) &= \vec{C}_i(t) + \vec{v_i}(t+1), \\ \vec{v_i}(t+1) &= \frac{s_i * ((1-\lambda) * \vec{r} + \lambda * \vec{v_i}(t))}{\parallel (1-\lambda) * \vec{r} + \lambda * \vec{v_i}(t) \parallel}. \end{aligned} \quad (3)$$

N(0,1) means a random number drawn from the Gaussian distribution with zero mean and variance one. Each peak's height $H_i(t)$ and width $W_i(t)$ vary according to its own $height\_severity_i$ and $width\_severity_i$ respectively. The center $\vec{C}_i(t)$ is moved by a vector $\vec{v_i}$ of a length s_i in a random direction ($\lambda = 0$) or a direction exhibiting a trend ($\lambda > 0$). The random vector $\vec{r}$ is created by drawing random numbers in $[-0.5, 0.5]$ for each dimension and normalizing its length to s_i. The complexity of this benchmark can be varied by changing the number of peaks and the number of dimensions.

B. Parameter Settings

As we can imagine, different settings of benchmark parameters can result in different dynamics of environmental change, which would have a big influence on the underlying ROOT problems. For instance, each peak function changes according to its own change severities, and if the differences among these change severities are large, the fitnesses of some solutions in search space would change much less severely compared to others. Further, if those slowly or progressively changing solutions happen to have satisfactory performances according to the constraints mentioned in Equation 1, *robust solutions over time* can be found among them. On the other hand, if all peak functions have similar change severities, then it is either that there would exist no *robust solution over time* or that TMO approaches would suffice to solve ROOT problems. For

TABLE I: Parameter settings of three problem instances from the modified moving peaks benchmark. The differences among these problem instances only lie in the setting of s_i.

Number of changes	20
Number of peaks	5
Dimension	2
Search range	$[0, 50]$
Height range	$[30, 70]$
Initial height	50
Width range	$[1, 12]$
Initial width	6
Height_severity range	$[1, 10]$
Width_severity range	$[0.1, 1]$
s_i distribution	$[0, 0.5, 1, 1.5, 2.0]$: $1st$ instance $[0.5, 0.75, 1, 1.25, 1.5]$: $2nd$ instance $[1.5, 1.75, 2, 2.25, 2.5]$: $3rd$ instance
δ_{app}	0.2

the sake of simplicity, we create three problem instances from the modified moving peaks benchmark by varying only the setting of s_i parameter, which is used to change the position of each peak function center. The s_i parameter can influence the underlying ROOT problem to a large extent in the sense that it directly changes the position of optimal solutions, the extent of which depends on the mean and variance of s_i for all peaks. To be more specific, we decide the distributions of s_i (the number of peaks equals to 5) for the three problem instances as $[0, 0.5, 1, 1.5, 2.0]$, $[0.5, 0.75, 1, 1.25, 1.5]$ and $[2.5, 2.75, 3, 3.25, 3.5]$. The rest parameter settings are all the same for these three problem instances. We have 5 peaks and change the environment 20 times successively. The search space is $[0, 50]$ for each dimension, and we focus on the case of two dimensions for each problem instance. $height\_severity_i$ and $width\_severity_i$ are initialized at the beginning before any environmental change, drawn randomly from $[1, 10]$ and $[0.1, 1]$ respectively, and they stay unchanged through the whole period of time. Besides, $height\_i$ and $width\_i$ are initialized 50 and 6 respectively, and required to stay in the interval $[30, 70]$ and $[1, 12]$ respectively during the 20 consecutive environmental changes (if it hits the boundary, it just bounces back in the opposite direction). The parameter δ_{app} mentioned in Equation 1 is set to be 0.2. All these parameter settings are summarized in Table I.

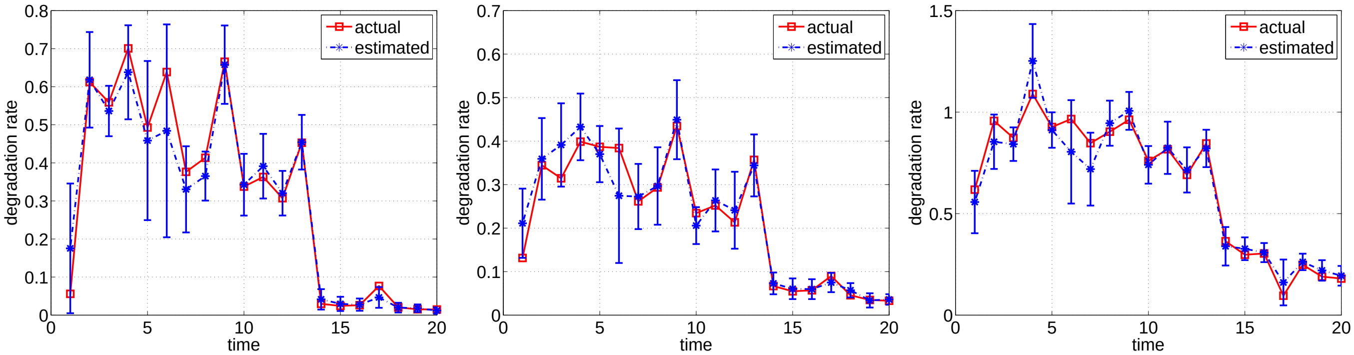
C. Simulation Results

In this section, we calculate the proposed measures for the three problem instances listed in Table I. The three problem instances are produced using the same set of random numbers through the whole time period. If samples need to be generated, we use 10000 samples in all cases. Besides, if some measured results are said to be significantly lower or higher than others, this is based on the 0.05 significance level Wilcoxon rank sum test. Finally, we rerun the calculation 20 times with samples generated differently each time on the same set of environmental changes, where repeated experiments are desired.

1) Optimum Degradation: The *optimum degradation* measures the performance degradation rate of previous optimum for a particular environmental change. Fig. 1 shows the actual and estimated optimum degradation rate for each of 20 successive environmental changes in the three problem instances. The *optimum degradation* here is based on $|\frac{F_c(X^*_{c-1}) - opt_c}{opt_c}|, 2 \leq c \leq 21$. For the first, second and third problem instances, the averaged *optimum degradation* over 20 environmental changes are 0.31, 0.22 and 0.65, with the corresponding standard deviation 0.25, 0.14 and 0.33. As we expect, the *optimum degradation* in the third problem instance is the highest, since its locations of optimal solutions change the most severely according to the setting of s_i. For the first two problem instances, the second one has an *optimum degradation* with larger standard deviation but the difference between both averaged *optimum degradation* is statistically tested insignificant. Finally, we may conclude that in all problem instances TMO approaches may perform better in later environmental states (from the 14th to 20th) than at the beginning for finding *robust solutions over time*, in that the *optimum degradation* in later environmental changes are much lower and most of them are below 0.2 ($\delta_{app} = 0.2$).

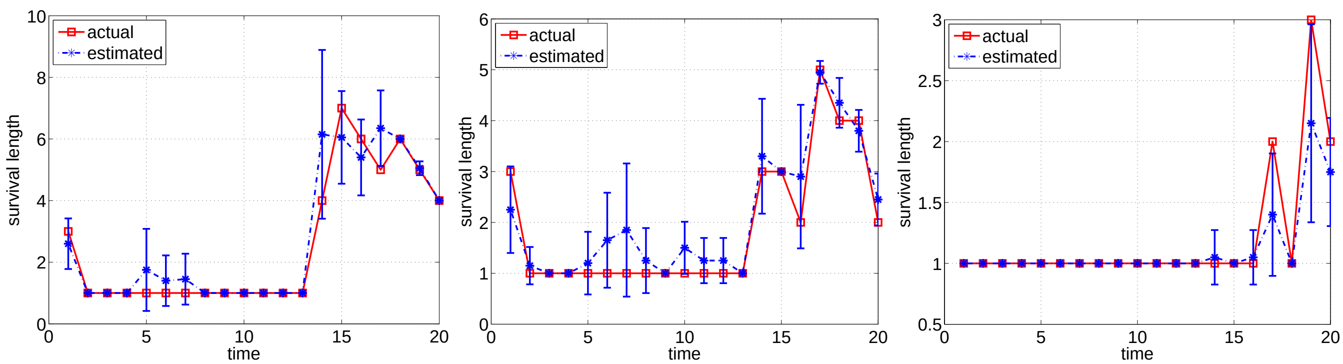
2) Optimum Survival Length: Fig. 2 shows the calculation of *optimum survival length*, which tells how long, in terms of the number of consecutive environmental states, a previous optimum can survive according to the constraints in Equation 1. Actual and estimated results are presented, and the data point in each sub-figure in Fig. 2 means how many environmental states a particular optimum can survive. For example, the point $(14, 4)$ for the actual case in Fig. 2a means the optimum for the 14th environmental state can also be used in the 15th, 16th and 17th environmental states considering its performance. The averaged *optimum survival length* over 20 environmental states for the three problem instances are 2.60, 1.90 and 1.20, with the corresponding standard deviation 2.16, 1.29 and 0.52. In accordance with the results showed in Fig. 1, the third problem instance has a significantly lower *optimum survival length* compared to the other two, and the second problem instance has a larger standard deviation of *optimum survival length* than what the first one has. Finally, we may conclude that TMO approaches will probably fail in finding *robust solutions over time* for all three problem instances during the first environmental state to approximately the 13th environmental state, since hardly any *optimum survival length* is more than 1 in that period.

3) Survival Rate: While the above mentioned measures look at how previous optimum performs in later environmental states, the measure *survival rate* asks generally whether there exists any *robust solution over time* and if so, what is the proportion. We calculate the measure of *survival rate* for each environmental change, and it is showed that approximately 41%, 42% and 14% (averaged over 20 consecutive environmental changes) currently satisfactory solutions continue to be satisfactory in the next environmental sate. By satisfactoriness, we mean the solution's fitness satisfies the constraints in Equation 1. Taking the data point in Fig. 3a at time 6 for example, it means around half satisfactory solutions in 6th environmental state being also satisfactory for the 7th



(a) 1st problem instance (b) 2nd problem instance (c) 3rd problem instance

Fig. 1: Actual and estimated optimum degradation for 20 successive environmental changes, with one standard deviation error bar for estimated cases (averaged over 20 independent runs). Time is the index of environmental change.



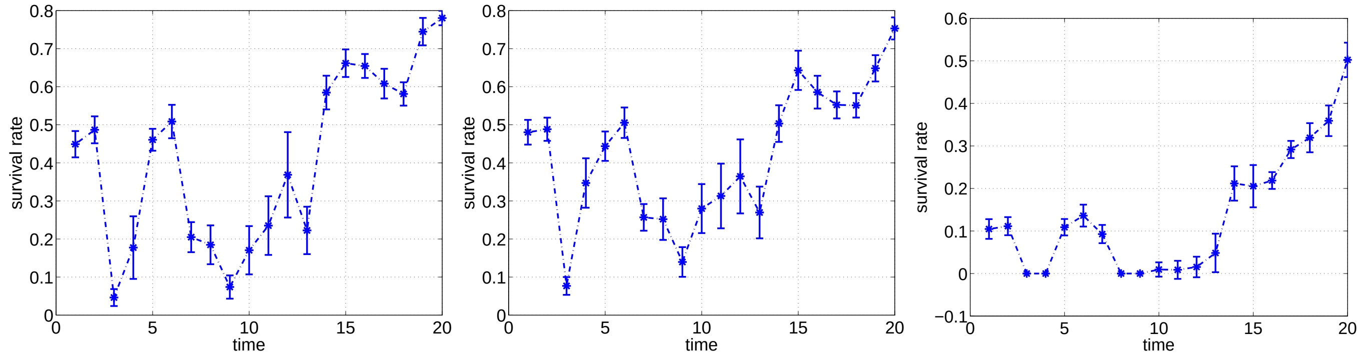
(a) 1st problem instance (b) 2nd problem instance (c) 3rd problem instance

Fig. 2: Actual and estimated optimum survival length for 20 successive environmental states, with one standard deviation error bar for estimated cases (averaged over 20 independent runs). Time is the index of environmental state.

environmental state. However, when we look at the data point $(6, 1)$ in Fig. 2a, the optimum at time 6 is unsatisfactory at time 7. This further demonstrates that TMO approaches can not find *robust solution over time* in some cases.

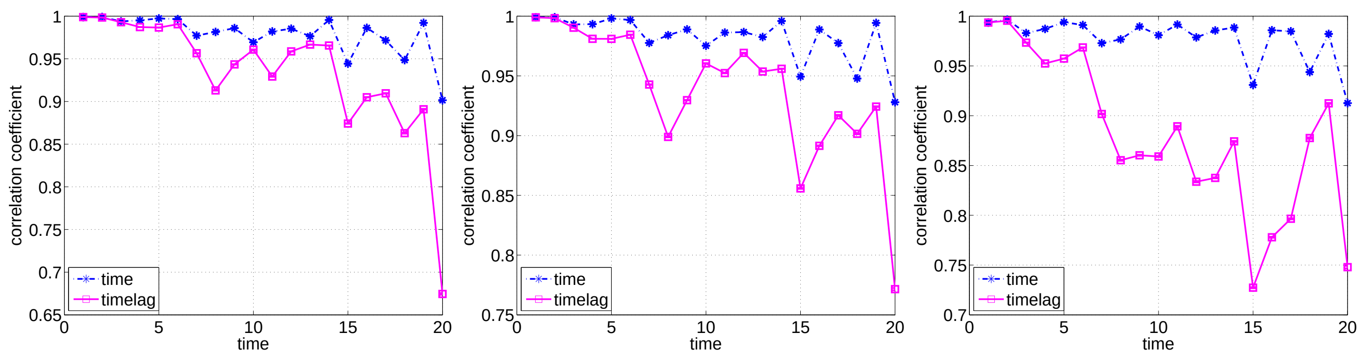
4) Fitness Correlation: The fitness correlation coefficient for each environmental change and fitness correlation coefficient depending on the time lag, i.e., after several environmental changes, are shown in Fig. 4. The fitness correlation coefficient for each environmental change in three problem instances are relatively high, which means good solutions tend to be good after an environmental change. However, this does not necessarily mean optimal solutions are *robust solutions over time*, comparing the Fig. 2c and Fig. 4c. As expected, the fitness correlation coefficient depending on the time lag deceases as the time lag increases in a general trend. The fitness correlation coefficient depending on the time lag is calculated based on the sample's fitness in the first environmental state and those in the ith $(2 \leq i \leq 21)$ environmental state.

5) Fitness Change Correlation Over Time: The *fitness change correlation over time* measures how strongly correlated a previous fitness change of a sample is to a later fitness change of that sample. This measure is calculated based on two consecutive fitness changes, and also on fitness changes with a time lag between each other. We can see from Fig. 5 that the *fitness change correlation over time* based on two consecutive fitness changes varies through time for all three problem instances with the interval $[0.81, -0.93]$, $[0.83, -0.94]$ and $[0.83, -0.88]$. The similar phenomenon can also be observed in the the *fitness change correlation over time* based on time lag. For positive *fitness change correlation over time*, it means if a solution's fitness increased due to a previous environmental change, the solution's fitness is likely to increase as well because of later environmental changes, and vice versa. The information of *fitness change correlation over time* is important in the sense that based on this measure we could tell which part of solution space is going to rise in the future, and algorithms which can successfully predict solution fitness change in the future may solve ROOT problems well. For instance, in the 10th environmental state in Fig. 5a, we could probably select a good solution whose fitness increased in the last environmental change as the *robust solution over time* when optimizing the fitness function in the 10th environmental



(a) 1st problem instance (b) 2nd problem instance (c) 3rd problem instance

Fig. 3: Survival rate for 20 successive environmental changes with one standard deviation error bar (averaged over 20 independent runs). Time is the index of environmental change.



(a) 1st problem instance (b) 2nd problem instance (c) 3rd problem instance

Fig. 4: Fitness correlation coefficient for each environmental change and fitness correlation coefficient depending on time lag over 20 successive environmental changes with one standard deviation error bar (averaged over 20 independent runs). Time is the index of environmental change.

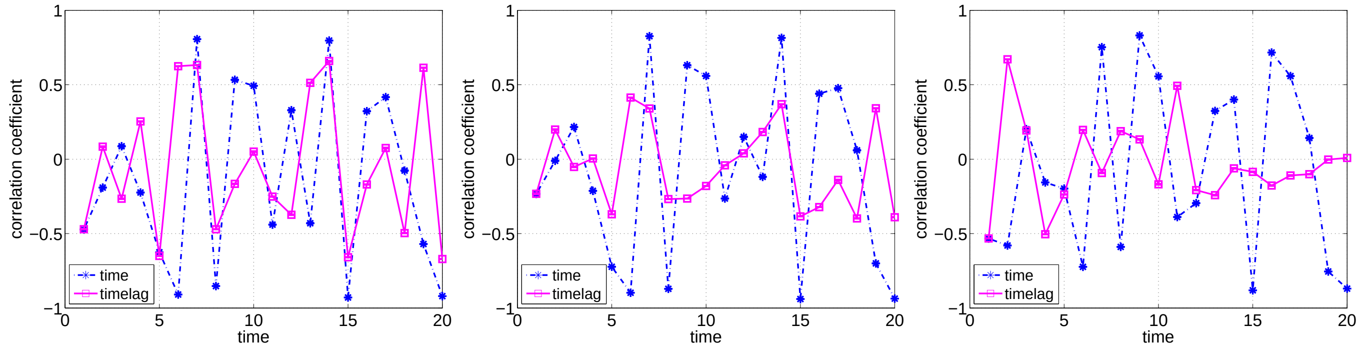
state.

6) Fitness Change Correlation of Similar Points: In practice, we may not be able to evaluate every solution's past fitness, based on which we could predict whether its fitness will increase or decrease in the future. Instead, we can evaluate a set of fixed solutions every time the environment changes and 'predict' how these solutions' fitnesses change due to future environmental changes. Based on the fitness change information of these fixed solutions, we could possibly 'predict' any solution's fitness change in the future according to how similar the solution is to the fixed solutions. In our study, we measure the similarity using Euclidean distance between two solutions. We assume that fitness change correlation decreases with distance increasing, which is experimentally demonstrated in Fig. 6. For a larger distance, the correlation turns negative, e.g., when distance is larger than 0.8 in Fig. 6a.

V. Conclusions and Future Work

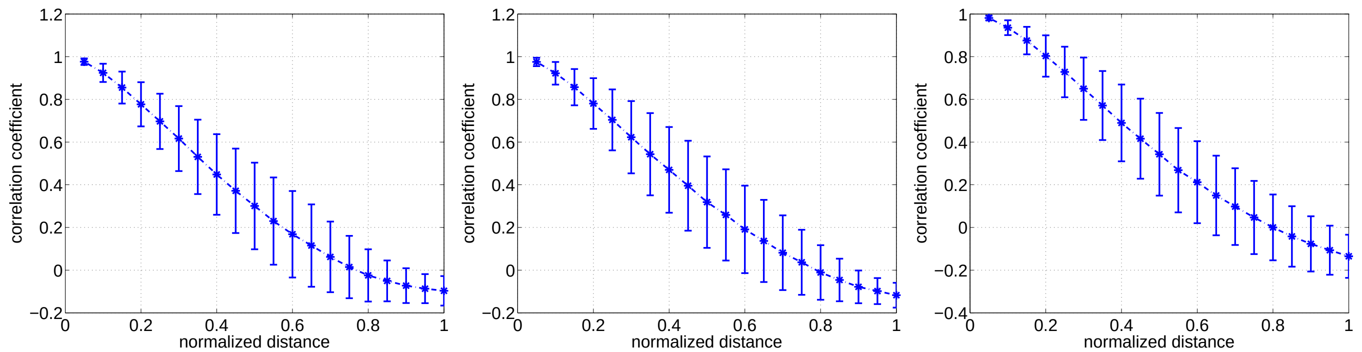
In this paper, we extend our previous work on ROOT [5] by first defining the single objective ROOT problem. Further, we develop and discuss a set of measures which can be used to characterize and analyse the underlying environmental change of dynamic optimization problems for the purpose of ROOT. We test these measures on several problem instances from the modified moving peaks benchmark, and show that these measures allow measurements of various aspects of environmental change, and most importantly give some inspirations of what kind of algorithms might or might not suit certain dynamic optimization problems with the aim being ROOT.

In order to encourage further research on ROOT, a benchmark problem specified for ROOT would be valuable. In the future, we will develop a proper ROOT benchmark problem which captures different aspects of environmental change based on the measures proposed in this paper, since existing dynamic optimization benchmarks are aiming at testing algorithm's TMO ability, and in some cases there won't exist any *robust solution over time* at all. Finally, it would be very helpful to have some of the measures incorporated into the design of algorithms for ROOT, for the obvious reason that if we can 'predict' how environment changes, the ROOT problem would be more tractable.



(a) 1st problem instance (b) 2nd problem instance (c) 3rd problem instance

Fig. 5: Fitness change correlation coefficient over time for each environmental change and fitness change correlation coefficient over time depending on time lag over 20 successive environmental changes with one standard deviation error bar (averaged over 20 independent runs). Time is the index of environmental change.



(a) 1st problem instance (b) 2nd problem instance (c) 3rd problem instance

Fig. 6: Fitness change correlation of similar points for different distances with one standard deviation error bar (averaged over 20 independent runs). Distance is normalized by dividing it by half the search range for each dimension.

ACKNOWLEDGMENT

The authors are grateful to Honda Research Institute Europe for funding this work. This work was also partially supported by two National Natural Science Foundation of China grants (No. 61028009 and No. 61175065).

REFERENCES

[1] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 3, pp. 303–317, 2005.

[2] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 4, pp. 440–458, 2006.

[3] X. Yu, K. Tang, T. Chen, and X. Yao, "Empirical analysis of evolutionary algorithms with immigrants schemes for dynamic optimization," *Memetic Computing*, vol. 1, no. 1, pp. 3–24, 2009.

[4] T. Nguyen and X. Yao, "Benchmarking and solving dynamic constrained problems," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 2009, pp. 690–697.

[5] X. Yu, Y. Jin, K. Tang, and X. Yao, "Robust optimization over time–A new perspective on dynamic optimization problems," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–6.

[6] K. Weicker, "An analysis of dynamic severity and population size," in *Parallel Problem Solving from Nature PPSN VI*. Springer, 2000, pp. 159–168.

[7] J. Branke, E. Salihoğlu, and Ş. Uyar, "Towards an analysis of dynamic environments," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, 2005, pp. 1433–1440.

[8] S. Wright, "The roles of mutation, inbreeding, crossbreeding and selection in evolution," in *Proc of the 6th International Congress of Genetics*, vol. 1, 1932, pp. 356–366.

[9] K. De Jong, "Evolving in a changing world," *Foundations of Intelligent Systems*, pp. 512–519, 1999.

[10] J. Branke, *Evolutionary optimization in dynamic environments*. Kluwer Academic Pub, 2002, vol. 3.

[11] H. Cobb and N. R. L. W. DC., *An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments*. Citeseer, 1990.

[12] S. Yang and X. Yao, "Population-based incremental learning with associative memory for dynamic environments," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 5, pp. 542–561, 2008.

[13] G. Lu, J. Li, and X. Yao, "Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms," *Evolutionary Computation in Combinatorial Optimization*, pp. 108–117, 2011.

[14] J. Rowe, "Finding attractors for periodic fitness functions," in *Genetic and Evolutionary Computation Conference*, 1999, pp. 557–563.

[15] ——, "Cyclic attractors and quasispecies adaptability," in *Theoretical aspects of evolutionary computing*. Springer-Verlag, 2001, pp. 251–259.

[16] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3. IEEE, 1999.