Performance Measures for Dynamic Environments

Karsten Weicker

University of Stuttgart, Institute of Computer Science,
Breitwiesenstr. 20–22, 70565 Stuttgart, Germany,
email: Karsten.Weicker@informatik.uni-stuttgart.de

Abstract. This article investigates systematically the utility of performance measures in non-stationary environments. Three characteristics for describing the goals of a dynamic adaptation process are proposed: accuracy, stability, and recovery. This examination underpins the usage of the best fitness value as a basis for measuring the three characteristics in scenarios with moderate changes of the best fitness value. However, for dynamic problems without coordinate transformations all considered fitness based measures exhibit severe problems. In case of the recovery, a newly proposed window based performance measure is shown to be best as long as the accuracy level of the optimization is rather high.

1 Introduction

Research on evolutionary algorithms applied to non-stationary problems dates back to the work by Goldberg and Smith (1987) and is still a topic of increasing popularity. In the past 15 years a vast variety of problems was considered. However, most of the research is rather problem or technique oriented and, therefore, deals with a specific niche in the space of the dynamic problems. There are only few classifications of dynamic problems available trying to establish a foundation for systematic research of dynamic optimization. Moreover, the even more important question concerning the comparability of several evolutionary algorithms acting in dynamic environments is still unanswered. In static optimization, there exist persistent reference quantities like the global optima and their fitness values. Not only these quantities might change over time in dynamic optimization, there are also certain variants of the focus concerning the actual goals of such an optimization. Within a formal framework this article specifies different goals of dynamic optimization, summarizes and develops a number of possible performance measures, and examines empirically how well these measures reflect the different goals for problems from various problem classes.

2 A Classification of Dynamic Problems

There are only few very coarse grained classifications distinguishing alternating (or cyclic) problems, problems with changing morphology, drifting landscapes, and abrupt and discontinuous problems (cf. Collard, Escazut, & Gaspar, 1997; De Jong, 2000). Other classifications are based on the parameters of problem generators (e.g. Branke, 1999). A third type of classification by Weicker (2000) decomposes the dynamic problem, that is defined on the search space Ω, into several components and the change of each component i is determined by a sequence of coordinate transformations $\left(c_i^{(t)}\right)_{t\in\mathbb{N}}$ with $c_i^{(t)} : \Omega \to \Omega$ and $d(c_i^{(t)}(\omega_1), c_i^{(t)}(\omega_2)) = d(\omega_1, \omega_2)$ for all $\omega_1, \omega_2 \in \Omega$ and fitness rescalings $\left(r_i^{(t)}\right)_{t\in\mathbb{N}}$ with $r_i^{(t)} \in \mathbb{R}_0^+$.

An exemplary non-stationary fitness function is

$$f^{(t)}(A) = \max_{1\leq j\leq hills} \begin{cases} 0, & \text{if } d(A, opt_j) > 150 \\ maxfit_j \frac{150-d(A,opt_j)}{150}, & \text{otherwise} \end{cases}$$

with $A \in \Omega = [-500, 500] \times [-500, 500]$, euclidean distance d, and *hills* randomly chosen local optima $opt_j \in \Omega$ – each local optima corresponds to one component. The coordinate transformation for each component j is a linear translation of length *coordsev* into a direction dir_j which is randomly determined at the beginning and as soon as a point outside of Ω would be created. The fitness rescaling is a factor $fitchange_j$ which is added to $maxfit_j$. Again, $fitchange_j \in [-fitsev, fitsev]$ is chosen randomly at the beginning and when $maxfit_j$ would leave the range $[0, 1]$. In non-alternating problems the maximum hill with $j = 1$ must have maximal fitness in $[0.5, 1]$ and all other hills in $[0, maxfit_1]$.

Weicker (2000) defines several properties within the framework. In this work, it is relevant whether the coordinate transformation is the identity mapping, i.e. the coordinates are static. If this is not the case the coordinate dynamics are constant over a period of time. Moreover, the fitness rescalings may be an identity mapping too. Here it is of interest whether the overall best fitness value may alter between various components.

Based on these properties the following problem classes are considered:

Class 1: coordinate translation, no fitness rescaling, no alternation
Problem instance: $hills = 5$, $coordsev = 7.5$, $fitsev = 0$
Various hills are moving while their height remains constant and the best hill remains best.

Class 2: coordinate translation, fitness rescaling, no alternation
Problem instance: $hills = 5$, $coordsev = 7.5$, $fitsev = 0.01$
Various hills are moving while their height is changing, but the best hill remains best.

Class 3: no coordinate translation, fitness rescaling, alternation
Problem instance: $hills \in \{2, 5\}$, $coordsev = 0$, $fitsev = 0.01$
The hills are not moving but changing their height leading to alternating best hills.

Class 4: coordinate translation, fitness rescaling, alternation
Problem instance: $hills \in \{2, 5\}$, $coordsev = 7.5$, $fitsev = 0.01$
The hills are moving while changing their height and different hills take the role of the best hill at different generations.

The problem instances with 2 hills are chosen such that there is at least one alternation while both hills are changing their height into the same direction. This additional characteristic is supposed to be problematic when measuring the performance. Note, that the fitness severity is chosen moderately in all classes.

3 Goals of Dynamic Optimization

The goal of an evolutionary search process in a dynamic environment is not only to find an optimum within a given number of generations but rather a perpetual adjustment to changing environmental conditions. Besides the accuracy of an approximation at time t, the stability of an approximation is of interest as well as the recovery time to reach again a certain approximation level.

The *optimization accuracy* at time t for a fitness function F and optimization algorithm EA is defined as

$$accuracy_{F,EA}^{(t)} = \frac{F(best_{EA}^{(t)}) - Min_F^{(t)}}{Max_F^{(t)} - Min_F^{(t)}} \tag{1}$$

where $best_{EA}^{(t)}$ is the best candidate solution in the population at time t, $Max_F^{(t)} \in \mathbb{R}$ the best fitness value in the search space, and $Min_F^{(t)} \in \mathbb{R}$ the worst fitness value in the search space. Note, that the accuracy is only well defined if the fitness function is non-constant at each time step, i.e. the fitness landscape is not a plateau covering the complete search space at any generation. The optimization accuracy ranges between 0 and 1, where accuracy 1 is the best possible value. It is also independent of fitness rescalings. This formula was introduced by Feng et al. (1997) as a performance measure in stationary environments.

As a second goal, *stability* is an important issue in optimization. In the context of dynamic optimization, an adaptive algorithm is called stable if changes in the environment do not affect the optimization accuracy severely. Even in the case of drastic changes an algorithm should be able to limit the respective fitness drop. The stability at time t is defined as

$$stab_{F,EA}^{(t)} = \max\{0, accuracy_{F,EA}^{(t-1)} - accuracy_{F,EA}^{(t)}\} \tag{2}$$

and ranges between 0 and 1. A value close to 0 implies a high stability. In the case of mere drifting landscapes, e.g. classes 1 and 2, it is a means to gain insight into the ability to track the moving optimum by observing the stability over a period of time. However, the stability must not serve as the sole criteria since it makes no statement on the accuracy level. In classes 3 and 4, the stability at the generations where changes occur are of interest.

An additional aspect is the ability of an adaptive algorithm to react quickly to changes. An algorithm has ε-reactivity at time t

$$react_{F,A,\varepsilon}^{(t)} = \min\left\{t' - t \,\middle|\, t < t' \leq maxgen,\ t' \in \mathbb{N},\ \frac{accuracy_{F,A}^{(t')}}{accuracy_{F,A}^{(t)}} \geq (1-\varepsilon)\right\} \cup \{maxgen - t\}$$

where *maxgen* is the number of generations. A smaller value implies a higher reactivity. This aspect of adaptation is especially of interest if the problem has short phases of big severity alternating with extensive phases of no severity with regard to the coordinate transitions or if the problem is alternating concerning the fitness rescalings (with rather low severity for the coordinates).

4 Performance Measures

In the previous section, the goals of optimization in dynamic environments are carefully defined using the accuracy. The accuracy is always a useful value, even in cases where the fitness of best approximation by the algorithm is decreasing, a situation where the overall best fitness could decrease or increase. But if the overall best fitness is not known the accuracy cannot be computed. Since this value might change, there is no common basis for assessment of the quality of a solution. A good fitness value at one time can be a bad fitness value at another time – but this is not transparent. Then, other means to assess the performance of an algorithm are necessary.

Those performance measures can be classified by the knowledge they need.

- Knowledge on the position of the optimum is available (which is usually only the case in academic benchmarks).
- Knowledge on the best fitness value is available.
- No global knowledge is available.

In order to enable a fair comparison of different algorithm with respect to the question how they reach the goal of dynamic optimization, a single value for the algorithm's performance is gained by averaging over all generations (see De Jong, 1975). In the following sections only the measures for one generation are given.

4.1 Measures for optimization accuracy

The definition of the accuracy was used by Mori, Kita, and Nishikawa (1996) as a performance measure averaged over a number of generations T. An almost similar performance measure was used by Trojanowski and Michalewicz (1999) where the normalization using the worst fitness value was omitted. They considered also only those time steps before a change in the environment occurs. If more emphasis is to be put on the detection of the optimum, Mori, Kita, and Nishikawa (1998) proposed a different weighting of those successful generations. A more gradual approach seems to be the usage of the square error to the best fitness value proposed by Hadad and Eick (1997).

In the case that no global knowledge is available, the following performance measures are discussed and examined in the remainder of the paper.

$$\begin{aligned}
\mathit{currentBest}_{F,EA}^{(t)} &= \max\{F(\omega) \mid \omega \in P_{EA}^{(t)}\} \\
\mathit{currentBestOffl}_{F,EA}^{(t)} &= \max_{1 \leq t' \leq t} \mathit{currentBest}_{F,EA}^{(t')} \\
\mathit{currentAverage}_{F,EA}^{(t)} &= \frac{1}{\mid P_{EA}^{(t)} \mid} \sum_{\omega \in P_{EA}^{(t)}} F(\omega) \\
\mathit{windowAcc}_{F,EA,W}^{(t)} &= \max\left\{ \frac{F(\omega) - \mathit{windowWorst}}{\mathit{windowBest} - \mathit{windowWorst}} \;\middle|\; \omega \in P_{EA}^{(t)} \right\} \quad \text{with}
\end{aligned}$$

$$\begin{aligned}
\mathit{windowBest} &= \max\{F(\omega) \mid \omega \in P_{EA}^{(t')},\ t - W \leq t' \leq t\} \\
\mathit{windowWorst} &= \min\{F(\omega) \mid \omega \in P_{EA}^{(t')},\ t - W \leq t' \leq t\}
\end{aligned}$$

The majority of publications uses the best fitness value $\mathit{currentBest}_{F,EA}^{(t)}$ to assess the quality of the algorithm (e.g. Angeline, 1997; Cobb, 1990; Dasgupta & McGregor, 1992; Goldberg & Smith, 1987; Grefenstette, 1992; Hadad & Eick, 1997; Lewis, Hart, & Ritchie, 1998; Mori et al., 1996; Vavak, Fogarty, & Jukes, 1996). This measure is better suited to dynamic problems than $\mathit{currentBestOffl}_{F,EA}^{(t)}$, the usual basis for offline performance (De Jong, 1975), that compares incommensurable values from different generations (cf. Grefenstette, 1999). Branke (1999) uses a mixed approach where those values from generations without environmental change are compared which requires global knowledge on any possible change in the environment.

The average fitness value $\mathit{currentAverage}_{F,EA}^{(t)}$ is used as a performance measure e.g. by Dasgupta and McGregor (1992), Goldberg and Smith (1987), Mori et al. (1996). Averaged over generations this leads to the online performance measure of De Jong (1975), which was originally defined as the average over all function evaluations since the start of the algorithm. Presumed that the population size is constant and the algorithm is generational, the online performance may be defined as mean $\mathit{currentAverage}_{F,EA}^{(t)}$. Online performance reflects the focusing of the search on optimal regions (see Grefenstette, 1992, 1999). In the online performance actually each new created individual is supposed to contribute a high fitness value. However, Cobb (1990) argued that this conception might not be suited for many dynamic problems because focusing too much on good fitness values might have negative effects on the adaptability.

Another approach to measure the accuracy without knowing the actual best possible fitness is based on the assumption that the best fitness value will not change much within a small number of generations. As a consequence a local window of interest $W \in \mathbb{N}$ is introduced and the accuracy $\mathit{windowAcc}_{F,EA,W}^{(t)}$ is measured within this window. This window based measure has not been used in the experiments reported in the literature.

Alternatively to the fitness based measures, genotype or phenotype based measures may be used to approximate the optimization accuracy. Though inde-

Table 1. Average accuracy and standard deviation for the genetic algorithm with and without hypermutation.

	w/out hypermut.		w/ hypermut.	
problem	avg	sdv	avg	sdv
class 1	0.45	0.023	0.87	0.0049
class 2	0.45	0.018	0.87	0.0035
class 3	0.82	0.035	0.96	0.0054
class 3 (2 hills)	0.97	0.0029	0.99	0.00086
class 4	0.46	0.025	0.87	0.0031
class 4 (2 hills)	0.41	0.023	0.86	0.0019

pendent of fitness rescalings, they require full global knowledge of the position of the current optimum. There are two variants: Weicker and Weicker (1999) used the minimal distance of the individuals in the population to the current optimum $\omega^* \in \Omega$ and Salomon and Eggenberger (1997) used the distance of the mass center ω_{center} of the population to ω^*.

$$bestDist^{(t)}_{F,EA} = \max \left\{ \frac{maxdist - d(\omega^*, \omega)}{maxdist} \;\middle|\; \omega \in P^{(t)}_{EA} \right\}$$

$$centerDist^{(t)}_{F,EA} = \frac{maxdist - d(\omega^*, \omega_{center})}{maxdist}$$

Where the first approach seems to be straightforward to assess the approximation quality, the second performance measure is more difficult to interpret. It requires that the population as a whole describes very closely the region of the optimum.

4.2 Measures for stability and reactivity

In this paper, the measures for stability and reactivity are defined by replacing the accuracy in the definition of stability or reactivity by an accuracy measure from the previous section. However, the genotype/phenotype based measures are omitted in this examination since they also require global knowledge.

5 Empirical Results

5.1 Experimental Setup

To optimize the dynamic problems two genetic algorithms are used. Both algorithms are based on a standard genetic algorithm where each search space dimension is encoded using 16 bits, the crossover rate is 0.6, the bit flipping mutation is executed with probability $\frac{1}{32}$, a tournament selection with tournament size 2 is used, and the algorithm runs for 200 generations. In addition to this standard algorithm, a version using hypermutation with a fixed rate of 0.2 is used (cf. Grefenstette, 1999). Table 1 shows the accuracy averaged over 10 problem instances and 50 independent experiments for each instance as well as

Table 2. Ranking based on pairwise hypothesis tests concerning the MSE of the curves

	standard GA						GA with hypermutation					
	Class 1	Class 2	Class 3	Class 3 (2 hills)	Class 4	Class 4 (2 hills)	Class 1	Class 2	Class 3	Class 3 (2 hills)	Class 4	Class 4 (2 hills)
Accuracy:												
best fitness	1	1	1	4	1	1	1	2	1	4	1	1
average fitness	2	2	2	3	2	2	4	3	2	3	3	3
window based	3	3	5	5	3	5	2	1	5	5	2	2
shortest distance	4	4	3	1	4	3	3	3	4	1	4	4
distance of center	5	4	3	2	5	4	5	5	2	2	5	5
Stability:												
best fitness	1	1	1	1	1	1	1	1	1	1	1	1
average fitness	2	2	2	2	2	2	2	2	2	2	2	2
window based	3	3	3	3	3	3	3	3	3	3	3	3
0.05-Recovery:												
best fitness	1	1	1	1	1	1	1	1	1	1	2	1
average fitness	3	3	3	3	3	3	3	3	3	3	3	3
window based	2	2	2	2	2	2	2	1	1	1	1	1

the respective standard deviation. The GA with hypermutation performs superior – however the performance of both algorithms should be expressed by a performance measure equally well.

5.2 Statistical examination of the measures

The goal of this investigation is to find some empirical evidence for the question how good the various measures approximate the exact adaptation characteristics. A first approach is based on the assumption that the curves of the performance measures should match the curves of the respective exact values to guarantee a meaningful statement of the performance measure. The second approach considers the averaged performance values only and tests how well they correlate to the averaged exact values.

In the first approach, the measurements are normalized $(g(t) - E_g)/\sqrt{V_g}$ where E_g is the expectancy value and V_g the variance. This makes the values of different performance measures comparable since the values are independent of the range of the values. To assess the similarity of the curves of the exact values h' and the normalized performance measure g', the mean square error $MSE_{g',h'} = \sum_{t=1}^{maxgen}(g'(t)-h'(t))^2$ is computed. In order to get a statistical confidence of one measure over another, a hypothesis test is carried out using the 500 independent mean square errors of each performance measure. Those pairwise hypothesis tests are used to establish a ranking concerning the suitability of the performance

Table 3. Percentage of problem instances with a high correlation to the exact averaged value

	standard GA						GA with hypermutation					
	Class 1	Class 2	Class 3	Class 3 (2 hills)	Class 4	Class 4 (2 hills)	Class 1	Class 2	Class 3	Class 3 (2 hills)	Class 4	Class 4 (2 hills)
Accuracy:												
best fitness	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
average fitness	1.0	1.0	1.0	1.0	1.0	1.0	0.8	1.0	1.0	1,0	1.0	1.0
window based	0.9	0.9	0.6	0.0	0.8	0.4	0.9	0.9	0.5	0.0	1.0	1.0
shortest distance	0.7	0.7	0.9	1.0	0.7	0.8	0.9	0.7	0.9	1.0	0.8	0.6
distance of center	0.7	0.7	0.9	1.0	0.7	0.9	0.7	0.4	0.9	0.5	0.6	0.3
Stability:												
best fitness	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
average fitness	1.0	1.0	0.4	0.0	1.0	1.0	0.0	0.2	0.0	0.0	0.0	0.0
window based	0.4	0.4	0.2	0.2	1.0	0.5	0.1	0.5	0.4	0.2	0.5	0.3
0.05-Recovery:												
best fitness	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.6	1.0	1.0	1.0	0.8
average fitness	1.0	1.0	0.4	0.2	1.0	1.0	0.3	0.3	0.0	0.0	0.1	0.1
window based	0.9	0.8	0.8	1.0	0.6	0.4	1.0	1.0	1.0	1.0	1.0	1.0

measures. Student's t-test is used as a hypothesis test with a significant error probability of 0.05. Table 2 shows the results of this analysis.

In the second approach, the averaged measures at the end of a optimization run are used to determine how well the algorithm performed on the problem. Therefore, a statistical test is used to compute the correlation of the approximated measures to the exact measures. The input data for the correlation analysis are the averaged performance values of the 50 different runs of an algorithm on a problem instance. (Since the recovery measures depend highly on the successive generations, the values of generation 150 are used for those measures instead of the final performance values in generation 200). As statistical method Spearman's rank correlation is used. A series of data is considered to be highly correlated if the Spearman's rank correlation is positive and the two-sided significance level of its deviation from zero is less than 0.001. The correlation is computed for each of the ten instances of a problem class. Table 3 shows the percentage of instances where a high correlation between exact value and performance measure could be identified.

5.3 Discussion of the Results

The results concerning the accuracy show, that the averaged best fitness values are a good indicator for all classes and both high and low quality algorithms.

However, the examination of the MSE shows that all fitness based measures have severe problems with class 3 (2 hills) where only fitness rescalings are occuring, in a misleading way. Especially the windows based measure has severe problems with all class 3 instances. The MSE of the GA with hypermutation on class 2 indicates that the window based measures can be a better indicator than best fitness although this is not approved by the averaged performance values.

Also, the stability is measured best using best fitness values. The average fitness shows very poor results with the averaged performance values. The windows based measure is insufficient regarding the MSE.

Concerning the recovery, the window based measure proofs to be equivalent or superior to the best fitness in case of the high quality experiments (GA with hypermutation).

The good results of the averaged best fitness for the accuracy contradicts partly the examination of the MSE. This indicates that especially in dynamic environments averaged performance measures should be used and interpreted carefully to rule out statistical effects.

6 Conclusions

This paper presents the first systematic approach to examine the usefulness of performance measures in time-dependent non-stationary environments. The goals of an adaptation process are discussed in detail and accuracy, stability, and recovery are proposed as key characteristics. Existing performance measures from literature are reviewed and a new window based performance measure is proposed.

On a wide set of dynamic problems the measures are examined for an algorithm with high accuracy and an algorithm generating low accuracy. Altogether the best fitness value proves to be the best performance measure for problems with moderate fitness severity – deficiencies exist for problems without coordinate transitions and as a basis for recovery measures. In the latter case, the window based measure exhibits a superior performance.

Future work has to examine problems with more severe fitness rescalings or additional problem characteristics. Also an investigation concerning the validity and strength of averaging performance values in dynamic domains is necessary.

References

Angeline, P. J. (1997). Tracking extrema in dynamic environments. In P. J. Angeline, R. G. Reynolds, J. R. McDonnell, & R. Eberhart (Eds.), *Evolutionary Programming VI* (pp. 335–345). Berlin: Springer. (Lecture Notes in Computer Science 1213)

Branke, J. (1999). *Evolutionary algorithms for dynamic optimization problems: A survey* (Tech. Rep. No. 387). Karlsruhe, Germany: Institute AIFB, University of Karlsruhe.

Cobb, H. G. (1990). *An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments* (Tech. Rep. No. 6760 (NLR Memorandum)). Washington, D.C.: Navy Center for Applied Research in Artificial Intelligence.

Collard, P., Escazut, C., & Gaspar, A. (1997). An evolutionary approach for time dependant optimization. *International Journal on Artificial Intelligence Tools*, *6*(4), 665–695.

Dasgupta, D., & McGregor, D. R. (1992). Nonstationary function optimization using the structured genetic algorithm. In R. Männer & B. Manderick (Eds.), *Parallel Problem Solving from Nature 2 (Proc. 2nd Int. Conf. on Parallel Problem Solving from Nature, Brussels 1992)* (pp. 145–154). Amsterdam: Elsevier.

De Jong, K. (2000). Evolving in a changing world. In Z. Ras & A. Skowron (Eds.), *Foundation of intelligent systems* (pp. 513–519). Berlin: Springer.

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* Unpublished doctoral dissertation, University of Michigan.

Feng, W., Brune, T., Chan, L., Chowdhury, M., Kuek, C. K., & Li, Y. (1997). *Benchmarks for testing evolutionary algorithms* (Tech. Rep. No. CSC-97006). Glasgow, UK: Center for System and Control, University of Glasgow.

Goldberg, D. E., & Smith, R. E. (1987). Nonstationary function optimization using genetic algorithms with dominance and diploidy. In J. J. Grefenstette (Ed.), *Proc. of the Second Int. Conf. on Genetic Algorithms* (pp. 59–68). Hillsdale, NJ: Lawrence Erlbaum Associates.

Grefenstette, J. J. (1992). Genetic algorithms for changing environments. In R. Männer & B. Manderick (Eds.), *Parallel Problem Solving from Nature 2 (Proc. 2nd Int. Conf. on Parallel Problem Solving from Nature, Brussels 1992)* (pp. 137–144). Amsterdam: Elsevier.

Grefenstette, J. J. (1999). Evolvability in dynamic fitness landscapes: a genetic algorithm approach. In *1999 Congress on Evolutionary Computation* (pp. 2031–2038). Piscataway, NJ: IEEE Service Center.

Hadad, B. S., & Eick, C. F. (1997). Supporting polyploidy in genetic algorithms using dominance vectors. In P. J. Angeline, R. G. Reynolds, J. R. McDonnell, & R. Eberhart (Eds.), *Evolutionary Programming VI* (pp. 223–234). Berlin: Springer. (Lecture Notes in Computer Science 1213)

Lewis, J., Hart, E., & Ritchie, G. (1998). A comparison of dominance mechanisms and simple mutation on non-stationary problems. In A. E. Eiben, T. Bäck, M. Schoenauer, & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V* (pp. 139–148). Berlin: Springer. (Lecture Notes in Computer Science 1498)

Mori, N., Kita, H., & Nishikawa, Y. (1996). Adaptation to a changing environment by means of the thermodynamical genetic algorithm. In H. Voigt, W. Ebeling, & I. Rechenberg (Eds.), *Parallel Problem Solving from Nature – PPSN IV (Berlin, 1996) (Lecture Notes in Computer Science 1141)* (pp. 513–522). Berlin: Springer.

Mori, N., Kita, H., & Nishikawa, Y. (1998). Adaptation to a changing environment by means of the feedback thormodynamical geentic algorithm. In A. E. Eiben, T. Bäck, M. Schoenauer, & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V* (pp. 149–158). Berlin: Springer. (Lecture Notes in Computer Science 1498)

Salomon, R., & Eggenberger, P. (1997). Adaptation on the evolutionary time scale: A working hypothesis and basic experiments. In J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer, & D. Snyders (Eds.), *Artificial Evolution: Third European Conf., AE'97* (pp. 251–262). Berlin: Springer.

Trojanowski, K., & Michalewicz, Z. (1999). Searching for optima in non-stationary environments. In *1999 Congress on Evolutionary Computation* (pp. 1843–1850). Piscataway, NJ: IEEE Service Center.

Vavak, F., Fogarty, T. C., & Jukes, K. (1996). A genetic algorithm with variable range of local search for adaptive control of the dynamic systems. In *Proc. of the 2nd Int. Mendelian Conf. on Genetic Algorithms.* ?

Weicker, K. (2000). An analysis of dynamic severity and population size. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, & H.-P. Schwefel (Eds.), *Parallel problem solving from nature – PPSN VI* (pp. 159–168). Berlin: Springer.

Weicker, K., & Weicker, N. (1999). On evolution strategy optimization in dynamic environments. In *1999 Congress on Evolutionary Computation* (pp. 2039–2046). Piscataway, NJ: IEEE Service Center.