Dynamic rotation and partial visibility

Karsten Weicker and Nicole Weicker
Institute of Computer Science, Dept. Formal Concepts
University of Stuttgart, Breitwiesenstr. 20–22
70565 Stuttgart, Germany.
email: first_name.last_name@informatik.uni-stuttgart.de

Abstract- This article generalizes a previously presented dynamic fitness function with two different concepts, namely a coordinate rotation and the concept of partial visibility. Those concepts define different classes of test problems. A set of standard evolution strategies and genetic algorithms with and without hypermutation are tested on two of the dynamic problem classes. They give insight in certain properties of the presented concepts and dynamic optimization in general.

1 Introduction

During the past years, dynamic environments have become challenging new applications for evolutionary computation. In literature we find various different approaches to deal with a manifold spectrum of dynamic problem, e.g. oscillating fitness functions or moving peaks. The work by Weicker and Weicker (1999), referred to as precursory work in the remainder of the article, has added a different problem, namely a rotating spiral, which was analyzed only for a few flavors of evolution strategies. Where most other work focuses on the tracking of optima in dynamic environments, the emphasis of the rotating spiral was on the ability of an algorithm to find the optimum although the available information leading to the optimum is changing with time. The question is whether an optimization algorithm can gather the necessary information and abstract from temporary changes.

This work generalizes the proposed problem by distinguishing between two different underlying principles:

- the rotation of a fitness landscape around the optimum
- the partial visibility of the problem

Moreover the dynamic problem in the precursory work was restricted to dimensionality 2. The test problems presented in this work repeals this restriction.

This test problem framework is used to examine genetic algorithms and evolution strategies on their performance to tackle those dynamic problems. In this examination, the tracking behavior as well as the algorithm's ability to abstract are important issues.

The article is organized as follows. After a quick review of related work (Section 2), the characteristics of the regarded dynamic problems and the test problem generator are described in Section 3. Then, the behavior of different algorithms on this class of problems is examined. The algorithms are presented in Section 4, the experiments in Section 5. Finally, a discussion concludes in Section 6.

2 Related work

Many examined dynamic problems reveal an underlying rule for the progression of the dynamics. Usually, linear translations of single peaks or random perturbances of the landscape are used. Linear Translations along an axis date back to the very early dynamic research articles, e.g. by Cobb (1990), Cobb and Grefenstette (1993), Vavak, Fogarty, and Jukes (1996). Angeline (1997) made one of the first approaches to examine circular translations besides linear and random movements. Bäck (1998) carried on the examination of those translations. Nevertheless, in all these approaches the position of the optimum is shifted where in the article by Weicker and Weicker (1999) the optimum is kept at a constant position and the complete fitness landscape is rotated around the optimum. In fact, this kind of dynamic rotation is strongly related to the static coordinate rotation used by Salomon (1996b, 1996a), Salomon and Eggenberger (1997) where the fitness landscape is randomly rotated once at the beginning of the search process but kept fixed during the whole optimization. Salomon and Eggenberger (1997) examined in addition how those functions are affected by a linear transition over time.

Also strongly related to the concept of partial visibility are the constrained problems, cf. Michalewicz and Schoenauer (1996), where parts of the search space are infeasible. There are many approaches to deal with those fitness functions (see also Yu & Bentley, 1998). One general approach is to assign a constant bad fitness value to the infeasible points. The same mechanism takes place in the concept of invisibility in this work. In addition, this article's proposed dynamic fitness functions vary the visible regions with time.

Lately, inspired by the "No free lunch" theorems, a lot of research in the field was driven by test generators in order to create whole classes of similar problems. In the area of dynamic optimization various test problem generators have been proposed recently, e.g. by Morrison and De Jong (1999), Grefenstette (1999), Branke (1999) for moving peaks problems. Instead of a new test problem generator, this article offers a possible framework for new classes of dynamic fitness functions.

3 Test problem framework

This section presents the underlying principles of the proposed dynamic problems as well as the resulting framework for dynamic test problems.

The problem in the precursory work consists of a two-dimensional search space where each point in the search

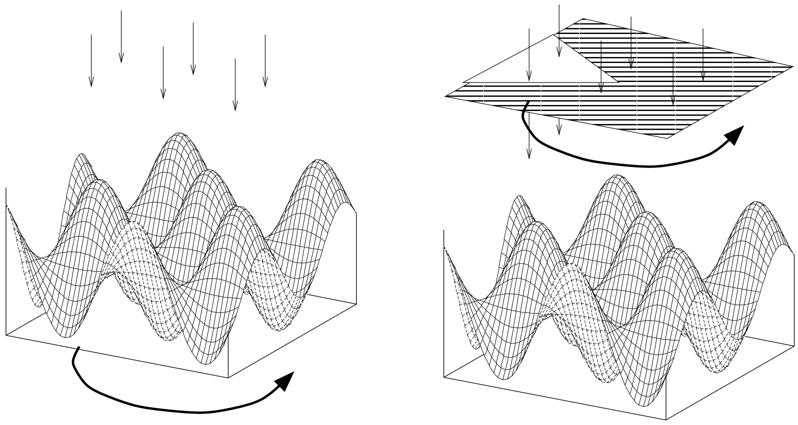


Figure 1: Schematic description of the considered fitness functions: rotating function F_1 (left) and static function with rotating visibility mask F_2 (right). The arrows indicate the view of the optimizer. In case of F_1 the underlying fitness function is changing with time. In case of F_2 the underlying fitness function is kept static but the rotating visibility mask uncovers only a part of the fitness function while the rest is kept a worse fitness value.

space gets the distance to the optimum assigned. Moreover, a mask of visibility is superposed which lets pass only the fitness value for certain regions of the search space. The other points get a constant bad fitness value assigned. In the original work a mask consisting of four spirals leading to the optimum was chosen. The dynamics have been created by rotating the visibility mask around the optimum.

To achieve a more general framework for similar dynamic problems, the approach is decomposed into two different concepts.

1. The concept of *rotation* is a transformation of the n-dimensional search space $\mathcal{S}$

$$R^{(t)}_{C,s} \quad : \quad \mathcal{S} \to \mathcal{S}$$

where $\vec{0} \mapsto \vec{0}$.

Such a rotation is determined by $\frac{1}{2}n(n-1)$ angles C between n axes for the direction of the rotation and a rotation speed s.

2. The concept of *partial visibility* changes the fitness values for certain points in the search space

$$M_{V,f,c} \quad : \quad \mathcal{S} \to \mathbb{R}$$

Such a visibility mapping is determined by a set of visible points $V \subset \mathcal{S}$, the actual underlying fitness function f, and a constant fitness value c for the invisible points in the search space. Then, for a point $x \in \mathcal{S}$ the function $M_{V,f,c}$ is defined as

$$M_{V,f,c}(x) \quad = \quad \begin{cases} f(x), & \text{if } x \in V \\ c, & \text{otherwise} \end{cases}$$

In order to combine both concepts, we require that the optimum of f is positioned at $\vec{0}$ and $\vec{0} \in V$.

Now, there are various possibilities how we can combine these ingredients for creating a dynamic test problem. We will give a quick overview on those different connections and focus on two variants in the remainder of the paper. Let $f : \mathcal{S} \to \mathbb{R}$ be the static fitness functions which should be turned dynamic.

Rotation only. By omitting the concept of visibility the dynamic function $F^{(t)}_1(x)$ is defined as

$$F^{(t)}_1(x) \quad = \quad f(R^{(t)}_{C,s}(x)).$$

The difficulty is to optimize although the fitness landscape is rotating (cf. left part of Figure 1). As underlying fitness function rather complex functions should be used. Symmetric functions like the sphere model suspend the rotation.

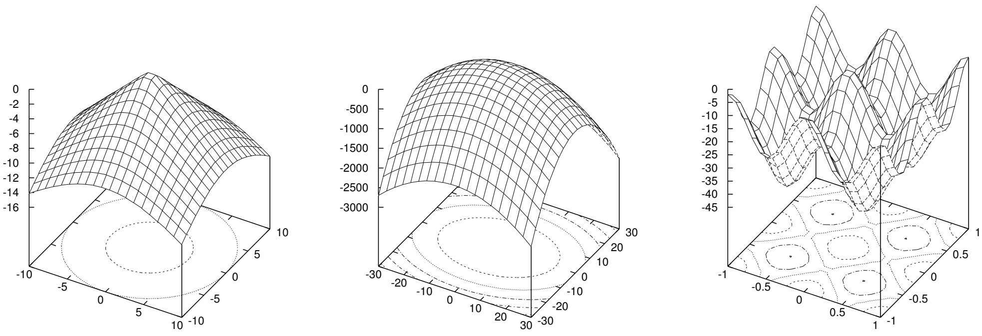
Rotating partial visibility. The function f is kept static and dynamics of $F^{(t)}_2(x)$ are only introduced by rotating the mask of visibility (cf. Weicker & Weicker, 1999).

$$F^{(t)}_2(x) \quad = \quad M_{R^{(t)}_{C,s}(V),f,c}(x),$$

where R is extended to sets of points. This work combines the tracking of the visibility range with an optimization task (cf. right part of Figure 1). It is necessary either to abstract from the visible parts or to follow the visible part.

Rotating fitness function with static partial visibility. The function f is rotated, but only a static segment of the search space is visible.

$$F^{(t)}_3(x) \quad = \quad M_{V,R^{(t)}_{C,s}\circ f,c}(x).$$



Cone Weighted Sphere Rastrigin

Figure 2: Inverse diagrams of the used static fitness functions

This involves the problem that tracking is not possible because of a fixed visibility. It is necessary either to abstract from the information passing by or to adapt quickly to the changing landscape.

Rotating fitness function with identically rotating visibility mask. The function f is rotated, and a visibility mask is rotating with the same rotation parameters.

$$F_4^{(t)}(x) \quad = \quad M_{R_{C,s}^{(t)}(V), R_{C,s}^{(t)} \circ f, c}(x).$$

This problem instance equals the rotation F_1 of a highly constrained fitness function. The visible part represents always the same segment of the landscape.

Rotating fitness function with differently rotating visibility mask. The function f is rotated, and a visibility mask is rotating with a different choice of rotation parameters.

$$F_5^{(t)}(x) \quad = \quad M_{R_{C_1,s_1}^{(t)}(V), R_{C_2,s_2}^{(t)} \circ f, c}(x).$$

This is the most difficult instance of the problem framework. A careful parameter setting should be chosen to avoid chaotic problem behavior.

This work considers only the first two cases F_1 and F_2. The remainder of this section discusses shortly the design of the rotation and the visibility mask.

The rotation must be supplied with a predetermined rotation time τ, which equals the number of generations until previous landscapes are repeated again. The rotation is managed by the multiplication of single rotations around two coordinate axes. The matrix for a rotation around axis 1 and 3 for a four-dimensional search space $\mathbb{R}^4$ is e.g.

$$R \quad = \quad \begin{pmatrix} cos(\frac{2\pi}{\tau}) & 0 & -sin(\frac{2\pi}{\tau}) & 0 \\ 0 & 1 & 0 & 0 \\ sin(\frac{2\pi}{\tau}) & 0 & cos(\frac{2\pi}{\tau}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Each coordinate axis may participate in only one pairwise rotation. The number of axis pairs γ to rotate pairwise is a second parameter for the rotation. The resulting rotation matrix describes the transition of the fitness landscape from one generation to the next. The complete rotation results in R^g for generation $g > 0$. In our experiments, all coordinate axes have been involved in the rotation. Since the dimensionality of the problem equals 30 it follows $\gamma = 15$. The values for τ are 5, 25, 50, 100, and 200.

The segment of visibility in function F_2 is defined by a cone with the optimum in the peak. The orientation of the cone is along axis 1 and the opening angle is determined by parameter α. Then, the visible points are defined as follows.

$$\mathcal{V} \quad = \left\{ \quad (x_1, \ldots, x_n) \,\middle|\, \arccos\left(\frac{x_1}{\sqrt{x_1^2 + \ldots + x_n^2}} \right) \leq \alpha \right\}$$

Because of the high dimensionality, we have chosen a rather large opening angle $\alpha = \frac{\pi}{2}$. The fitness value for the invisible points depends on the specific fitness function and the worst possible value.

The following fitness functions are used as static basis functions for the framework of rotation and partial visibility.
Cone: $f(\vec{x}) = \sqrt{\sum_{i=1}^{n} x_i^2}$ with $x_i \in [-10, 10]$
Weighted Sphere: $f(\vec{x}) = \sum_{i=1}^{n} i x_i^2$ with $x_i \in [-30, 30]$
Rastrigin: $f(\vec{x}) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right)$ with $x_i \in [-1, 1]$
Note especially the restricted definition range of the Rastrigin function. Cone is only used with the rotating partial visibility. Weighted Sphere and Rastrigin are used as a basis function for both F_1 and F_2.

4 Compared algorithms

On this problem two paradigms of evolutionary algorithms are considered, genetic algorithms (GA, Holland, 1992; Goldberg, 1989) and evolution strategies (ES, Rechenberg, 1973; Schwefel, 1981).

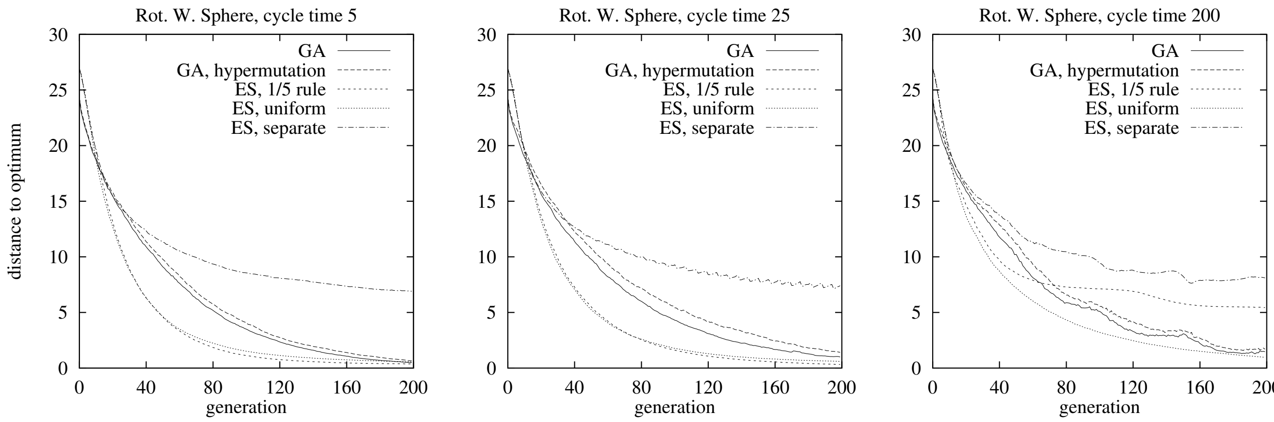


Figure 3: Rotating weighted sphere with cycle time 25 (left) and cycle time 200 (right)

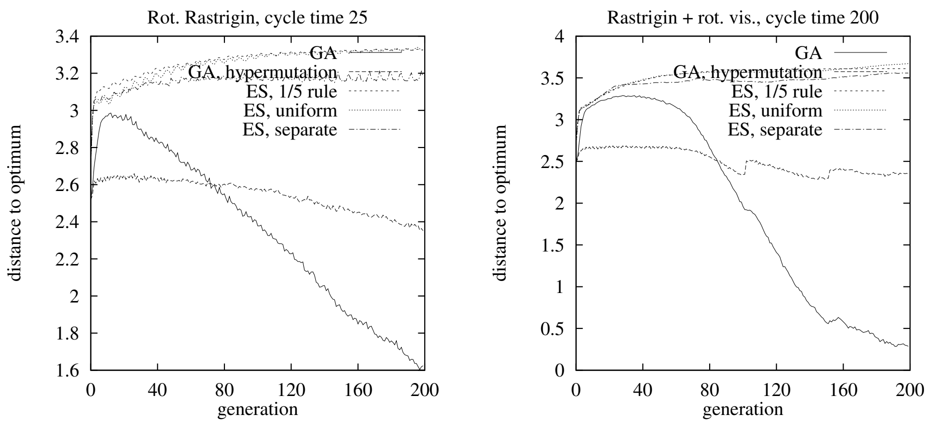


Figure 4: Rotating Rastrigin function with cycle time 25 (left) and Rastrigin function with rotating visibility mask and cycle time 200 (right)

The genetic algorithm is used with standard parameter settings: population size 100, crossover rate 0.6, mutation rate $\frac{1}{Number\ of\ Bits}$, and fitness proportional selection with linear scaling. In addition to the canonical genetic algorithm, often a successful extension for dynamic problems, the hypermutation operator (Cobb, 1990; Cobb & Grefenstette, 1993; Grefenstette, 1992, 1999), is used. In our study the fixed hypermutation model by Grefenstette (1992) was used where a fraction of 0.2 of the population is randomly reset.

In the case of the evolution strategies various adaptation mechanisms for the step-size are used in the context of this work. First of all, the 1/5-rule by Rechenberg (1973) is used as a global adaptation mechanism. Second, the uniform self-adaptation with one strategy parameter and the separate step-size self-adaptation with one strategy parameter for each dimension are used on the proposed problem (cf. Schwefel, 1981).

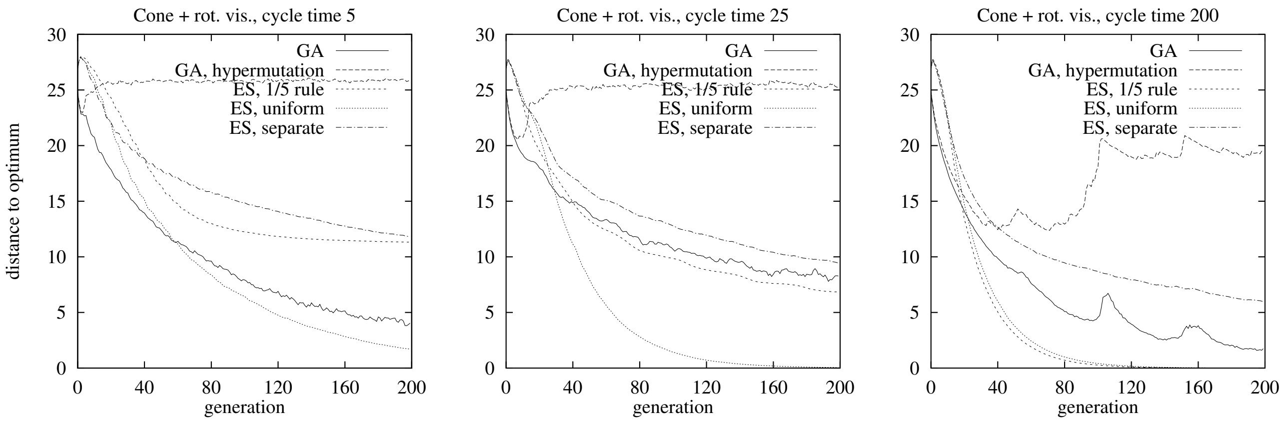
All evolution strategies use a $(15, 100)$-strategy and incorporate no recombination operator.

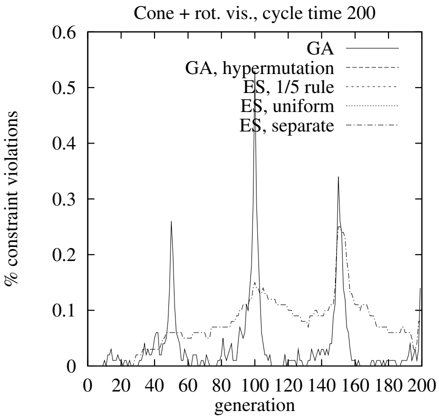
5 Experiments

All experiments are averaged over 100 independent runs. There was no special parameter tuning for both algorithms. As a basis for comparison the best fitness value of each generation is used for the rotating fitness function F_1. For the rotating visibility, the distance to the optimum is used as well as the percentage of generations where the visible region was completely lost.

In case of the rotating weighted sphere (cf. Figure 3) the 1/5-rule exhibits best performance for fast dynamics but worsens essentially with slow dynamics. Uniform step-adaptation proves to be almost unaffected by the rotation. With the exception of the disappointing separate step-adaptation, all other algorithms are only slightly worse than uniform adaptation. For this fitness function the hypermutation operator cannot improve the genetic algorithm.

In case of the rotating multimodal Rastrigin function (cf. left part of Figure 4) all variants of the evolution strategy are not able to locate the optimum. Note, that all used evolution strategies are mutation-only algorithms in this work. In-





Average lost generations

Cycle time	5	25	50	100	200
GA	0.281	0.275	0.173	0.090	0.036
GA w/ hypermut.	0.0	0.0	0.0	0.0	0.0
ES w/ 1/5-rule	0.226	0.275	0.218	0.032	0.0
ES w/ uniform	0.093	0.027	0.012	0.0	0.0
ES w/ separate	0.180	0.076	0.082	0.070	0.074

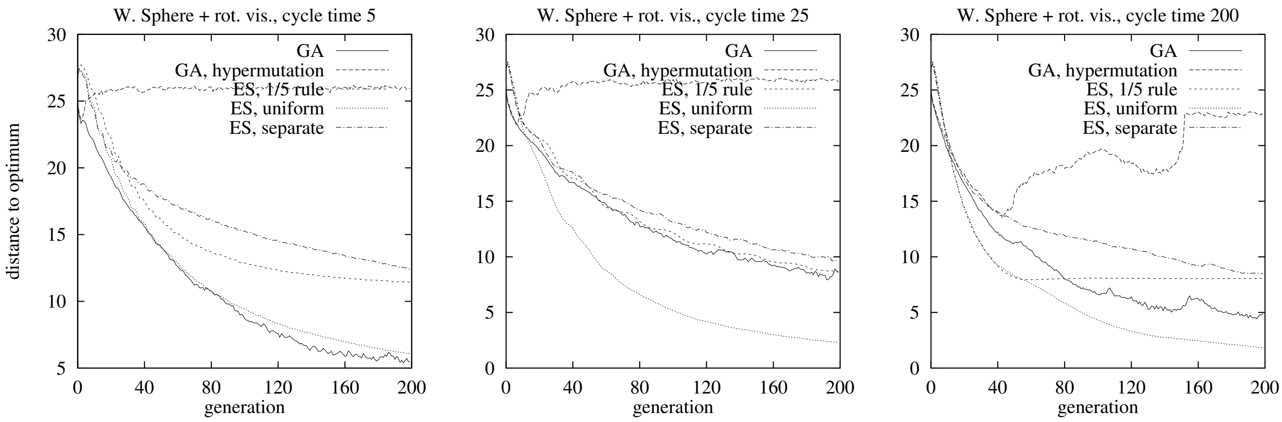
Figure 5: Cone with rotating visibility mask and cycle time 5, 25, 200 (top row from left to right), constraint violations for cycle time 200 (bottom row, left figure), and the average percentage of generations where the visibility range was completely lost for the cone fitness function in the tabular.

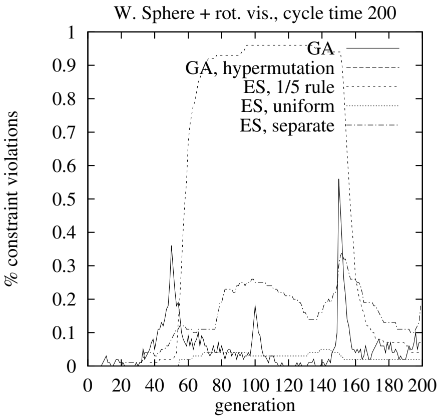
troducing intermediary recombination improves the ES performance significantly. The only successful optimization algorithm in the experiments is the genetic algorithm without hypermutation. Again the hypermutation operator rather hinders optimization than helps to improve the performance. The experiments show for all cycle times the same tendency. Almost similar results are obtained if the rotation of the fitness function is substituted by a rotating visibility mask (cf. right part of Figure 4). Not shown in the figures is the interesting fact that the genetic algorithm achieves better results for the rotating visibility mask than for the mere rotation (in case of very fast rotation ($\tau = 5$) and slow rotation ($\tau \geq 100$).

Now, the rotating visibility mask is closer examined for the cone and the weighted sphere. In almost all experiments with both functions the hypermutation extension was not able to get close to the optimum. This topic is discussed in more detail later.

In case of the cone with rotating visibility mask (cf. Figure 5), the uniform step-adaptation is outstanding where the other ES variants cannot convince with fast dynamics. There, the GA shows a fair performance which worsens considerably with cycle time 25, and improves again gradually. Interestingly the 1/5-th rule improves and outperforms the uniform step-adaptation with slow dynamics. The tabular in Figure 5 shows the percentage of generations where each algorithm was unable to track the visible region. There are several remarkable points. First, although performing fair on $\tau = 5$ the GA shows a high percent of missed tracks. This indicates that the genetic algorithm is able to solve the problem by abstracting from the changing landscapes rather than following the visible region. The lower left part of Figure 5 shows the average visibility violations for each generation with $\tau = 200$. Interestingly, the GA exhibits huge tracking problems after each quarter rotation – when the sign of half of the dimensions changes for the visibility range. Second, the GA with hypermutation has an optimal tracking rate by the inserted random immigrants. Nevertheless, it seems that this disturbs the GA so much that it is not able to optimize in addition to the tracking. Third, the tracking ability of the separate step-size adaptation does not improve with slower rotation. This indicates that this adaptation rule has severe problems with the rotation transformation (cf. lower left part of Figure 5).

In case of the weighted sphere with rotating partial visibility (cf. Figure 6), uniform mutation shows the most stable results on all cycle times, followed by the standard genetic algorithm. The 1/5-rule is not able to track the visible region





Average lost generations

Cycle time	5	25	50	100	200
GA	0.302	0.295	0.208	0.110	0.055
GA w/ hypermut.	0.0	0.0	0.0	0.0	0.0
ES w/ 1/5-rule	0.269	0.320	0.392	0.422	0.483
ES w/ uniform	0.240	0.034	0.0	0.020	0.022
ES w/ separate	0.204	0.089	0.124	0.134	0.142

Figure 6: Weighted sphere with rotating visibility mask with cycle time 5, 25, and 200 (top row from left to right), constraint violations for cycle time 200 (bottom row, left side), and the average percentage of generations where the visibility range was completely lost for the weighted sphere in the tabular.

as it is shown by the table in Figure 6. As the lower left part of Figure 6 exhibits, the algorithm seems to converge early and is not able to follow the visibility range. As in the previously described problem, the GA has tracking problems after each quarter of the rotation. Nevertheless, it outperforms all other algorithms with $\tau = 5$ (upper right figure). Concerning the GA with hypermutation, we observe that it behaves at the beginning as requested, but when 1/4-th of the first rotation is over, the algorithm looses track of the visibility region and worsens extremely. Again this effect is the focus of future research since the reason is not loosing track of the visibility region (cf. table in Figure 6).

6 Discussion

The generalization of the problem proposed in the precursory work results in five new classes of dynamic problems with different requirements. Two of these classes have been the focus of this work. There are a few conclusions we can draw from the investigations in this paper.

The rotation exhibits new kinds of difficulty. More complex step-size adaptations find it difficult to adapt to the rotation. But also genetic algorithms have severe problems when the visibility range changes the sign in a few dimensions. Usual means to improve genetic algorithm performance like the hypermutation show rather poor performance. Future research should investigate whether triggered hypermutation or a genetically controlled hypermutation rate shows a better performance. Presumably, hypermutation should be active after each quarter rotation.

We could also observe that under certain circumstances the global adaptation strategy of the 1/5-rule outperforms the other methods. This goes along with observations by Angeline (1997). Nevertheless, this is usually only the case with rather simple fitness functions. Since those fitness functions have a completely different behavior than other problems, the future research should move away from the simplifying models like moving spheres.

All in all, experiments with the proposed fitness functions should be interpreted carefully since there are two different effects taking place. With slow rotation a tracking effect takes place and with fast rotation there is only a sequence of few fitness landscapes with a common optimum. There might be different strategies an algorithm reaches its goal for the two extreme rotation speeds. Further experiments and analyses are necessary to clarify the working principles.

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)

Bäck, T. (1998). On the behavior of evolutionary algorithms in dynamic environments. In *IEEE Int. Conf. on Evolutionary Computation* (pp. 446–451). IEEE Press.

Branke, J. (1999). Memory enhanced evolutionary algorithms for changing optimization problems. In *1999 Congress on Evolutionary Computation* (pp. 1875–1882). Piscataway, NJ: IEEE Service Center.

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.

Cobb, H. G., & Grefenstette, J. J. (1993). Genetic algorithms for tracking changing environments. In S. Forrest (Ed.), *Proc. of the Fifth Int. Conf. on Genetic Algorithms* (pp. 523–530). San Mateo, CA: Morgan Kaufmann.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

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.

Holland, J. H. (1992). *Adaptation in natural and artifical systems.* Cambridge, MA: MIT Press.

Michalewicz, Z., & Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, *4*(1), 1–32.

Morrison, R. W., & De Jong, K. A. (1999). A test problem generator for non-stationary environments. In *1999 Congress on Evolutionary Computation* (pp. 2047–2053). Piscataway, NJ: IEEE Service Center.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Stuttgart: frommann-holzbog. (German)

Salomon, R. (1996a). Performance degradation of genetic algorithms under coordinate rotation. In L. J. Fogel, P. J. Angeline, & T. Bäck (Eds.), *Evolutionary Programming V: Proc. of the Fifth Annual Conf. on Evolutionary Programming* (pp. 155–161). Cambridge, MA: MIT Press.

Salomon, R. (1996b). Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions. *BioSystems*, *39*(3), 263–278.

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.

Schwefel, H.-P. (1981). *Numerical optimization of computer models.* Chichester: John Wiley & Sons, Ltd.

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

Yu, T., & Bentley, P. (1998). Methods to evolve legal phenotypes. In A. E. Eiben, T. Bäck, M. Schoenauer, & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V* (pp. 280–291). Berlin: Springer. (Lecture Notes in Computer Science 1498)