

# A Closer Look Down the Basins of Attraction

Erik Pitzer, Michael Affenzeller, Andreas Beham

**Abstract**—Formal fitness landscape analysis enables us to study basins of attraction with great detail. This seemingly simple concept shows more variability and influence on heuristic algorithms than might be expected. We have taken a new perspective and a closer look at the properties of basins of attraction using two-dimensional visualizations to arrive at a conceptually simple but heuristically challenging test function that exhibits “misleading” basins of attraction.

## I. INTRODUCTION

The concept of a fitness landscape since coined by Sewall Wright in [1] has found widespread adoption. However, most often the notion is used informally and the fitness distribution over the solution space is imagined as a three dimensional physical landscape with mountains and valleys. If the solution space has higher dimension, however, this can become a relatively misleading metaphor [2].

Using the notion of a fitness landscape in the light of heuristic optimization we soon arrive at the concept of local optima and their corresponding basins of attraction. These are the areas which *lead* to a certain local optimum. Hence, if we find the basin of attraction of an optimum we have practically found the optimum itself. The question that remains is which optimum have we found, is it merely local or is it the global optimum. In the following pages we will first formalize the notion of a fitness landscape, its embedded optima and their surrounding basins of attraction that we will further subject to a series of analyses.

## II. FORMALIZATION

### A. Fitness Landscapes

A fitness landscape can be defined over an arbitrary solution space  $\mathcal{S}$ . In general this solution space is further encoded into a solution representation space  $\mathcal{R}$  using an arbitrary encoding function  $e : \mathcal{S} \rightarrow \mathcal{R}$ . This space is the basis of the fitness landscape. Furthermore, we need an objective function  $f : \mathcal{R} \rightarrow \mathbb{R}$  that is closely related to the “fitness”. Depending on the nature of the problem, being either a maximization or a minimization problem, the objective function or respectively the negation of the objective function describes the fitness or desirability of a certain representation as solution to the given problem. These two items  $\mathcal{R}$  and  $f$  are often thought to already define a fitness landscape. So far, however, this only describes the underlying fitness function and we have no notion of connectedness yet.

There are several possibilities to define connectedness. While in [3] it is capitalized that every operator yields its own fitness landscape, it has been shown in [4] that for whole

classes of operators a distance measure or metric can be used instead. This has the further advantage of making different operators comparable within the same fitness landscape.

Let the distance function  $d : \mathcal{R} \times \mathcal{R} \rightarrow \mathbb{R}$  be a metric (symmetric, non-negative, identifying indiscernible values and satisfying the triangle inequality) then we can use it to define the triple in Equation 1 as fully describing a *fitness landscape*  $\mathcal{F}$ .

$$\mathcal{F} := \{\mathcal{R}, f, d\} \quad (1)$$

### B. Local Optima

When we are talking about optima in this paper we are always referring to minima to be congruent with the term of a *basin* of attraction and the fact that we are examining minimization problems in later sections.

First we define the  $\varepsilon$ -neighborhood of a representation  $r$  as the set of representations that have maximum distance  $\varepsilon$  to  $r$  in Equation 2.

$$N_\varepsilon(r) := \{n \mid n \in \mathcal{R}, d(r, n) \leq \varepsilon\} \quad (2)$$

Building upon the definition of a fitness landscape described in Equation 1 we can formalize a local optimum (a minimum in our case). To do so, we need to define a “downward path” from one point  $r_1$  in the landscape to another point  $r_2$  using  $\varepsilon$ -steps in Equation 3.

$$\begin{aligned} \Downarrow p_\varepsilon(r_1, r_2) \Leftrightarrow & \\ & f(r_2) < f(r_1) \wedge \\ & [d(r_1, r_2) \leq \varepsilon \vee \\ & \exists(n \in N_\varepsilon(r_1)) f(n) < f(r_1) \wedge \Downarrow p_\varepsilon(n, r_2)] \end{aligned} \quad (3)$$

We can then use this definition to say that a local optimum has no downward path to any other representation as shown in Equation 4.

$$\text{local optimum}_\varepsilon(o) \Leftrightarrow \nexists(r \in \mathcal{R}, r \neq o) \Downarrow p_\varepsilon(o, r) \quad (4)$$

Merely requiring all neighbors of an optimum (minimum) to be of higher value is not enough since the optimum could be either a plateau or a saddle point. The definition using downward paths nicely takes care of this while at the same time requiring us to choose an appropriate  $\varepsilon$  to allow our definition to cover both discrete and continuous spaces.

Interestingly the definition of a global optimum (minimum) is much simpler as it does not need the notion of a neighborhood:

$$\text{global optimum}(o) \Leftrightarrow \forall(r \neq o) f(r) \geq f(o)$$

Erik Pitzer is with the Upper Austria University of Applied Sciences, Softwarepark 11, 4232 Hagenberg, Austria (phone: +43-7236-3888-7129; fax: +43-7236-3888-7199; email: erik.pitzer@fh-hagenberg.at).

### C. Basins of Attraction

Further building upon our definition of a local optimum we can define the basin of attraction or the sub space that “leads” to the local optimum. Here, we would like to discern between two types of attraction basins. There is the *unconditional* or *strong basin*, which will ultimately lead to a certain optimum when going downward regardless of the search heuristic that was used and a broader *conditional* or *weak basin of attraction* that might be shared between several optima.

First, we define the weak basin of attraction  $b(o)$  simply as a set of solution representations that have a downward path towards a certain optimum  $o$  as shown in Equation 5. The definition of a downward path can be found in Equation 3.

$$b(o) := \{r \mid r \in \mathcal{R}, \downarrow p_\varepsilon(r, o)\} \quad (5)$$

This defines the total area from which a certain optimum can be reached when going downward. However, these areas might overlap with each other which creates ambiguity as to which optimum will be reached depending on the choice of algorithm and its optimization operators. Therefore, we define the strong basin of attraction as the area which leads exclusively to one optimum when only allowing downward convergence. Given the set of all local optima  $\mathcal{O}$  we can define the strong basin of attraction  $\hat{b}(o)$  of an optimum  $o$  in Equation 6.

$$\hat{b}(o) := \{r \mid r \in b(o), \nexists (o_2 \in \mathcal{O}, o_2 \neq o) r \in b(o_2)\} \quad (6)$$

This defines the exclusive basin of attraction for a certain optimum. Hence, if we reach this area, we are sure to converge to this optimum if we only go downward. Outside of this area, while still in the weak basin of attraction it is uncertain which of several optima we will reach. This fact can also be formalized as the probability of reaching a certain optimum once inside its weak attraction basin.

$$p(r, o) := \sum_{n \in \downarrow N_\varepsilon(r)} p(n|r) \cdot p(n, o) \quad (7)$$

In Equation 7 the probability  $p(r, o)$  that the optimum  $o$  is reached from  $r$  is given as the weighted sum of the probabilities  $p(n, o)$  that an average algorithm converges from a downward neighbor  $n$  to  $o$ . The probability to converge from an optimal representation to “itself” is set to 1 (i.e.  $p(o, o) = 1$ ). The weights  $p(n|r)$  which are the probabilities of choosing neighbor  $n$  coming from  $r$  depend on the algorithm and its operators. One possibility is to set  $p(n|r) = 1/|\downarrow N_\varepsilon(r)|$ , i.e. to give equal probability of choosing each downward neighbor. Another obvious choice would be to choose closer neighbors more often or to choose better neighbors more often, or a combination of the two. However, during our experiments a change in this selection probability has not influenced the shapes of the basins significantly.

With this definition we can chart and overlay the probability distributions for several local optima and obtain a better understanding of the interplay between overlapping basins of attraction.

## III. BACKGROUND METHODOLOGY

### A. Basin Fill Algorithm

To exhaustively and quickly analyze arbitrary functions the following algorithm was devised. As the definitions of basins of attractions and especially convergence probabilities are highly recursive the dynamic programming paradigm was employed to greatly reduce computational cost. This idea is inspired by the reverse hill climber developed in [3] to explore basins of attraction.

Paths in continuous spaces can only be sampled and cannot be computed symbolically in the general case. Therefore, we start by sampling a grid of fitness values over the entire area of interest with resolution  $\varepsilon$ . We sort the fitness values in increasing order. If we now conduct our recursive calculation in exactly this order, where we only ever consider downward paths, the values of all downward neighbors will already have been calculated and the previous—possibly nested—recursion becomes a simple iteration with constant complexity. For simplicity we are assuming all fitness values to be unique, otherwise we would have to take care of equal fitness values of neighboring representations. Listing I shows some pseudocode for this procedure: The container class `Value` can be thought of as an associative container that contains its coordinates, fitness value and probabilities to converge to each optimum. Choosing an appropriate list of neighbors is done through an arbitrary function `neighbors()`.

In our experiments we used several ranges of neighbors that satisfy a criterion such as  $d(r, n) \leq d_{\max}$  where  $d_{\max} \in \{1, \dots, 9\}$ . These particular choices showed hardly any influence on the pictures besides changing the transitions’ ruggednesses.

#### LISTING I

##### PSEUDOCODE FOR BASIN FILL ALGORITHM

```

// Calculate fitness values for all values
foreach index in data.Indices
  data[index] = new Value(index, f(transform(index)));
// Find local optima
localOptima =
  from v in data
    where neighbors(v).Max(n => n.v) > v.v
    select v;
// Traverse “landscape” by fitness value
foreach v in data.SortedByFitness()
  var downwardNeighbors =
    from n in neighbors(v)
      where n.v < v.v
      select new n;
// Update probabilities
foreach n in downwardNeighbors
  foreach o in localOptima
    p(v, o) += p(n, o)/localOptima.Count();
// No downward neighbors? We are at a local optimum!
if (downwardNeighbors.Count() == 0)
  v[nearest(localOptima, v)] = 1;

```

We can additionally consider the slope towards our neighbors and their distance when choosing a particular neighbor as already described in the previous section. However, in

our tests neither of the two refinements showed a significant impact on the shape or value of the convergence probabilities.

This procedure provides us with an interesting “perspective” onto the fitness landscape. For every representation and every observed optimum we obtain a generalized probability of converging to that optimum.

We have used this program to generate colorful pictures and to visualize the weak and strong basins of attraction and especially their overlaps on various two-dimensional fitness landscapes. This visualization uses a distinct color for every local optimum. The saturation of this color represents the probability of converging to one particular optimum, while the luminance of a point represents the fitness value. These colors are mixed by averaging their Lab color values [5] for any given point to provide an insight into the probability distribution surrounding different optima.

Additionally, we highlight the “borders” between basins, where the probability of converging to one optimum is equal to the probability of converging to another optimum, or more precisely where the most probable optimum to converge to differs between two neighbors (again using the same but arbitrary neighbors() function from above).

### B. Test Functions

In Figure 1 we can see the analysis method described in Section III-A applied to the Rastrigin [6], Griewank [7] and Ackley [8], [9] test functions. While they are all very colorful and visually pleasing, we can immediately observe that there is little variation in the shapes and sizes of their basins of attraction. Another interesting observation are the relatively large almost white areas that indicate either only weak attraction to a specific optimum or a strong overlay of several overlapping convergence probabilities to different optima.

One noteworthy aspect of basins of attraction is the relation between their height and their breadth as already mentioned in [3] using the reverse hill climber. With this respect the normal distribution becomes an intriguing candidate for investigation as it has an inverse relation between peak width to peak height depending on its variance parameter  $\sigma$ . If we take a look at Figure 2a there seem to be far reaching consequences. Here, we can see, that the large but shallow local optimum (blue, top right) has an immensely big basin of attraction that almost completely preempts the attraction basin of the global optimum (green, center), leaving only a narrow corridor towards the lower left.

We can see similar effects in the Figures 2b through 2d. Figure 2b shows an example of a deliberately placed shallow normal distribution (center) to confine the basins of attraction of the better optima (black dots), while Figures 2c and 2d show some random examples with up to five normal distributions in superposition. Here again, the large but shallow local optima dominate attraction in almost the entire solution space. This leads us to the assumption that a superposition of multiple multivariate normal distributions with different variances will make a good test function for heuristic algorithms as it emphasises exploratory properties

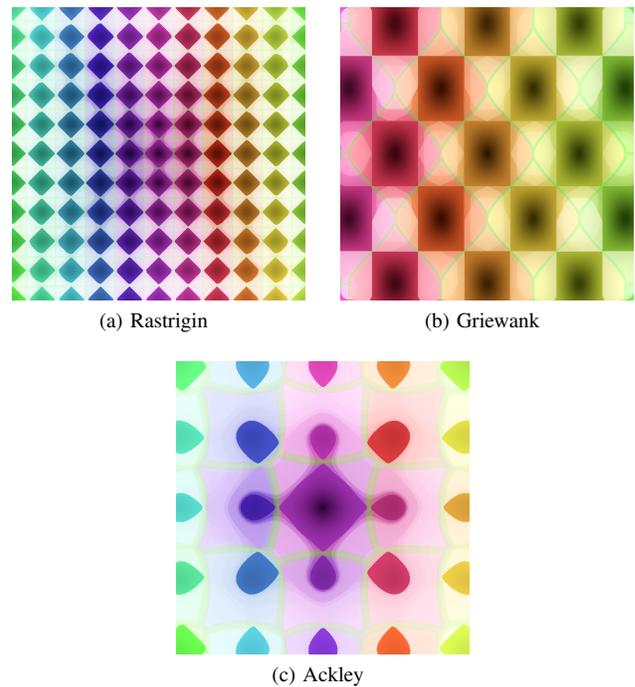


Fig. 1. Basin Analysis of Common Test Functions

and heavily penalizes early convergence to an “apparent” optimum. Moreover, as the normal distribution describes many phenomena that frequently occur, we think that these characteristics will be found in many optimization problems that deal with fitting data with measurement errors.

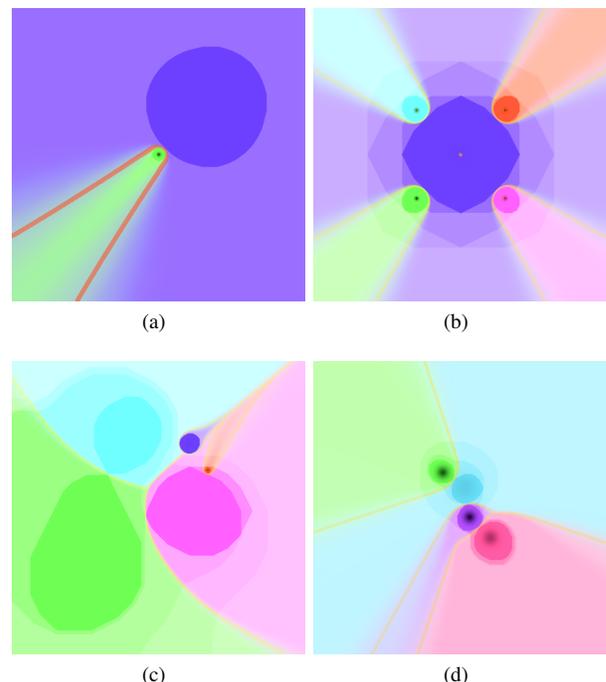


Fig. 2. Large Basins Overshadow Global Optimum

### C. Multi-Normal Test Function

There is a whole array of test functions that have been used to study the characteristics of various heuristic optimization algorithms [6], [7], [8], [9]. Interestingly, none have so far been used to study the effects of basin size vs. basin depth. Therefore, we have used an additive superposition of several multidimensional normal probability density functions as our new objective function. The general case of a multivariate normal distribution is shown in Equation 8. This simplifies to Equation 9 in case the variances are uniform in all dimensions. Our new test function  $f$  as shown in Equation 10 is then defined as a sum of  $n$  different normal distributions. We have taken the negative sum here to make it a minimization problem and not to break with our general metaphor of a “basin” analysis.

$$N(\vec{x}, \vec{\mu}, \Sigma) := \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})'\Sigma^{-1}(\vec{x}-\vec{\mu})} \quad (8)$$

$$N(\vec{x}, \vec{\mu}, \sigma) := \frac{1}{2\pi\sigma^2} e^{-\frac{(\vec{x}-\vec{\mu})^2}{2\sigma^2}} \quad (9)$$

$$f(\vec{x}) := -\sum_{i=1}^n N(\vec{x}, \vec{\mu}_i, \sigma_i) \quad (10)$$

We have used a simple scheme to scale the number of local optima (i.e. the number of normal distributions) with the number of dimensions exponentially. For two dimensions we have used four optima with the following values for  $\vec{\mu}$ : (5, 5), (5, 15), (15, 5), (15, 15) and  $\sigma^2$ : 0.2, 1.0, 1.0, and 2.0.

For higher dimensions we have uniformly perturbed the locations and variances of the non global optima by  $\pm 0.5$ . The locations of the optima in higher dimension are equivalent to the values of bits in binary number representations with as many digits as we have dimensions. Let  $l$  be the number of dimensions and, hence, the length of a binary string. Furthermore, let  $\text{bin}(x)$  be the binary encoding of  $x$  and subsequently  $\text{bin}(x)_i$  the  $i$ -th bit of the binary encoding. First we note that for dimension  $l$  we get  $2^l$  optima of length  $l$  of which one is the global optimum. Furthermore, the  $i$ -th coordinate in the  $j$ -th local optimum is defined as  $\text{bin}(j)_i \cdot 10 + 5 \pm 0.5$ . The variance for the first (and global) optimum is 0.2, for the last optimum it is 2 and all intermediate optima have variances between 0.5 and 1.5. The reason for this additional perturbation is to make recombination more difficult as the best and worst local optima are diametrically opposing each other across all dimensions with intermediate optima serving as “way points”.

Figure 3 shows a visualization of the basins of attraction of the two-dimensional multi-normal test function. We can see how the basin of attraction of the global optimum is preempted by the basins of the shallower local optima.

## IV. RESULTS

### A. Test Algorithms

As test platform, the new HeuristicLab 3.3 [10], [11] was used (freely available at <http://dev.heuristiclab.com>). It facilitates testing of a wide array of different

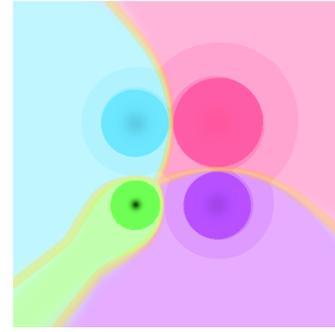


Fig. 3. Basins of Attraction of the Multi-Normal Test Function

heuristics on any given problem. While it allows definition of completely arbitrary algorithms it comes with a great selection of preconfigured heuristic algorithms.

Several different heuristic algorithms were used to estimate the difficulty of an example test problem using the multi-normal test function described in Section III-C.

The first test algorithm was an evolution strategy [12] with self adaptation [13] and recombination using the discrete crossover operator with two parents per child. The population size  $\mu$  and number of children  $\lambda$  where set to 100 and 500 respectively in an elitist population giving a (100+500)-ES. Moreover, we use a uniform one position mutation operator.

We have also applied a tabu search [14] algorithm, with 10 samples, 10,000 iterations and a tabu list length of 100 as well as a local search algorithm [15] with 10 samples and 10,000 iterations. For both algorithms the step size  $\sigma$  was set to 0.2.

We also tested a genetic algorithm [16] again with discrete crossover, a population size of 500, one elite individual, a maximum of 200 generations, linear rank selection, mutation probability 10% randomly choosing between either normally distributed all position mutation or uniformly distributed one position mutation.

An offspring selection genetic algorithm [17] with a population size of 700, one elite individual, discrete crossover, linear rank selection, comparison factor and success ratio 1.0, a mutation probability of 10% and again a mix of normally distributed all positions and uniformly distributed one position mutation was also tested. This time limiting the number of evaluated solutions to 100,000 instead of a maximum number of generations.

Finally, we have used a self-adaptive segregative genetic algorithm with simulated annealing aspects (SASEGASA) [17] with a population size of 100 on every of 8 initial villages, one elite individual, proportional selection, mutation probability of only 5% and the same mix of normal all positions and uniform one position mutation. The maximum selection pressure before merging was 15 while the maximum number of evaluated solutions was again capped at 100,000.

We ran all of these algorithms 10 times on a five-dimensional and a ten-dimensional multi-normal test function as well as on ten-dimensional, 50-dimensional and 100-

dimensional Ackley, Griewank and Rastrigin test functions.

The standard test problems, while quite simple to solve for all of these algorithms, have a much larger number of local optima. However, as we can see from Figure 1 the basins of attraction of these optima are very well behaved and all have the same size which is in stark contrast to the sizes and shapes of the basins of attraction in the multi-normal test function.

### B. Algorithm Performance

As we can see in Figure 3 the basin for the global optimum is greatly reduced in size by the nearby but much shallower local optima in the two-dimensional multi-normal test function. Therefore, we assume it to be an interesting test case for heuristic algorithms. We have applied the selection of algorithms described in Section IV-A to two versions of the multi-normal test function.

The first test was a five-dimensional multi-normal test function as defined in Section III-C. Figure 4a shows a boxplot of the relative closeness to the global optimum in terms of achieved objective function value. The genetic algorithms, especially advanced versions (OSGA and SASEGASA) appear to have little trouble with the five-dimensional case, while evolution strategy and especially local search and tabu search are easily trapped in one of the 31 non-global optima.

In the boxplots shown in Figure 4b we can see that no single algorithm is able to consistently find the optimum for a ten-dimensional multi-normal test function, while again GA and OSGA come a little closer we can see, that already a landscape with only 1024, but misleading, optima becomes quite difficult.

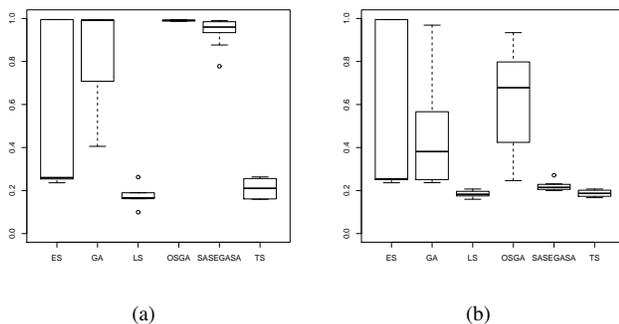


Fig. 4. Performance on 5D and 10D Multi-Normal Test Function

We contrast this with the performance of the exact same algorithms with the exact same settings on a set of standard test functions. Figure 5 shows that these algorithm have little to no trouble on these standard test functions with many more local optima than 1024. (The peculiarities of tabu search on the Ackley test function and local search on the Griewank test function performing worse on a lower number of dimensions can probably be explained by the inappropriately small step size  $\sigma$  of 0.2 for a function with much larger bounds.)

The apparent difference in performance between standard test functions with many optima compared to the multi-

normal test function with relatively few optima leads us to the conclusion that it is not only the number of local optima that influences problem difficulty but also the shape and relative size of their attraction basins that have to be considered.

## V. CONCLUSION

We conducted an in-depth analysis of the widely known concept of a basin of attraction. We have introduced a more fine-grained distinction between strong and weak basins of attraction as well as an analysis of generalized local convergence probabilities together with a dynamic programming algorithm to efficiently calculate these probabilities for model landscapes. We have further shown a comparison of standard test functions with a specially crafted multi-normal function with misleading properties of its optima and their basins of attraction and concluded that not only the number of local optima but also the shape and size of their basins of attraction plays an important role in fitness landscape analysis and the judgement of problem difficulty

### A. Future Work

While we are currently only visualizing two-dimensional landscapes, a similar approach can also be applied to higher dimensions. For these higher-dimensional spaces, we will derive a *weighted convergence volume* for each optimum which is simply the (estimated) number of representations weighted with their probability to converge to that optimum. The distribution of these volumes' extents will hopefully provide us with further insight into the convergence behaviour of heuristic algorithms.

## ACKNOWLEDGMENTS

The work in this paper has been made possible through a sabbatical grant from the Upper Austria University of Applied Sciences to Erik Pitzer and the collaboration with the Josef Ressel-Centre "Heureka!" in Hagenberg.

## REFERENCES

- [1] S. Wright, "The roles of mutation, inbreeding, crossbreeding and selection in evolution," in *Proceedings of the Sixth International Congress of genetics*, vol. 1, 1932, pp. 356–366.
- [2] L. Barnett, "Ruggedness and neutrality—the NKp family of fitness landscapes," in *ALIFE: Proceedings of the sixth international conference on Artificial life*. Cambridge, MA, USA: MIT Press, 1998, pp. 18–27.
- [3] T. Jones, "Evolutionary algorithms, fitness landscapes and search," Ph.D. dissertation, University of New Mexico, Albuquerque, New Mexico, 1995.
- [4] A. Moraglio and R. Poli, "Topological interpretation of crossover," in *In Proceedings of the Genetic and Evolutionary Computation Conference*. Springer, 2004, pp. 1377–1388.
- [5] R. S. Hunter, "Photoelectric color-difference meter," in *Proceedings of the Thirty-Third Annual Meeting of the Optical Society of America*, vol. 38, no. 12. OSA, 1948, pp. 1092–1092.
- [6] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computation*, ser. Natural Computing Series, G. Rozenberg, T. Baeck, A. E. Eiben, J. N. Kok, and H. P. Spaink, Eds. Springer-Verlag Berlin Heidelberg, 2003.
- [7] M. Locatelli, "A note on the griewank test function," *Journal of Global Optimization*, vol. 25, no. 2, pp. 169–174, 2003.
- [8] D. H. Ackley, *A connectionist machine for genetic hillclimbing*. Boston: Kluwer, 1987.

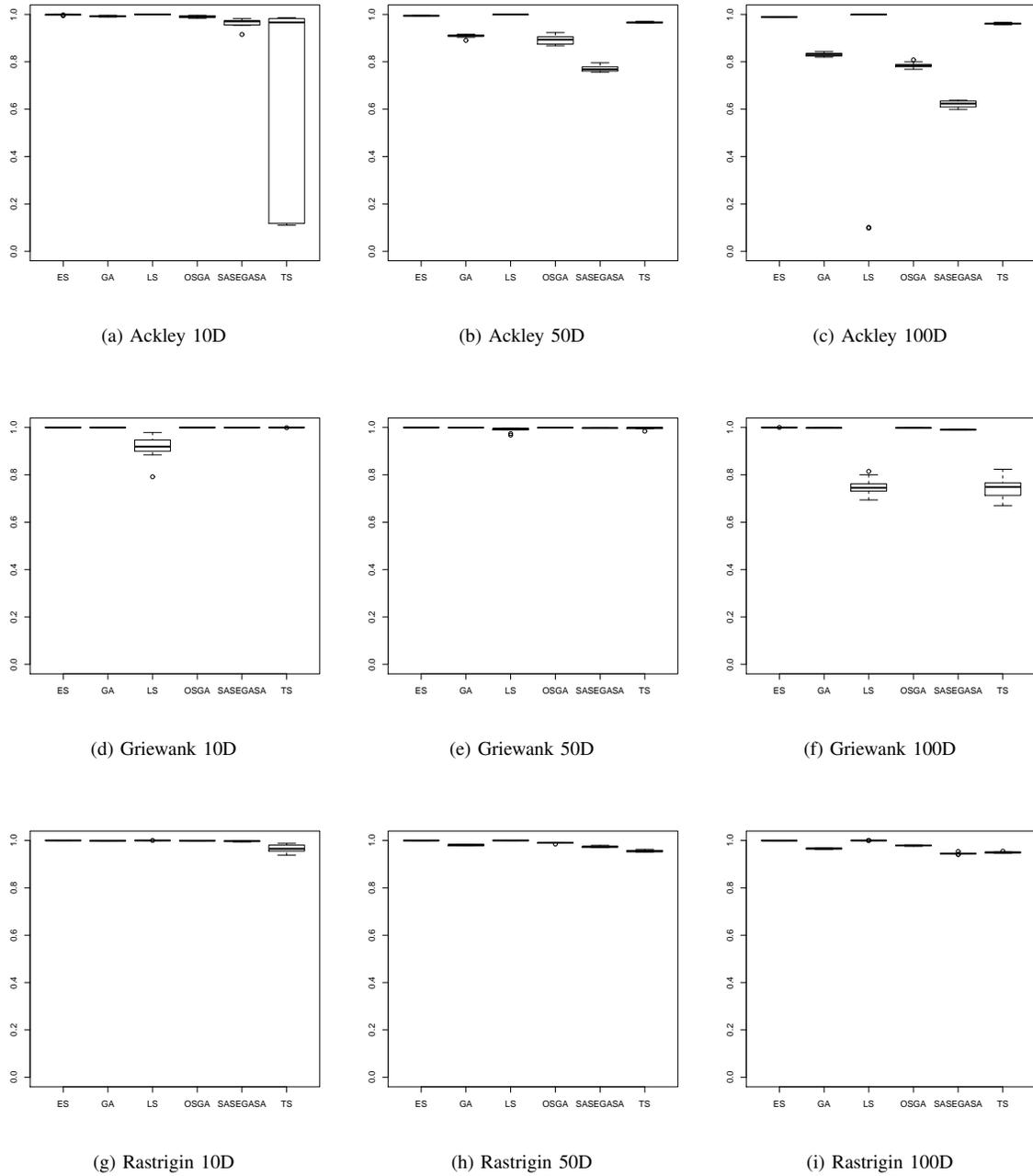


Fig. 5. Performances on Various Test Functions with 10, 50 and 100 Dimensions

- [9] T. Bäck, *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.
- [10] S. Wagner, "Heuristic optimization software systems - Modeling of heuristic optimization algorithms in the HeuristicLab software environment," Ph.D. dissertation, Johannes Kepler University, Linz, Austria, 2009.
- [11] S. Wagner, G. Kronberger, A. Beham, S. Winkler, and M. Affenzeller, "Modeling of heuristic optimization algorithms," in *Proceedings of the 20th European Modeling and Simulation Symposium*, A. Bruzzone, F. Longo, M. A. Piera, R. M. Aguilar, and C. Frydman, Eds. DIPTeM University of Genova, 2008, pp. 106–111.
- [12] I. Rechenberg, *Evolutionstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, 1973.
- [13] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies - A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, March 2002.
- [14] F. Glover and M. Laguna, *Tabu Search*. Kluwer, 1997.
- [15] E. Aarts and J. K. Lenstra, Eds., *Local Search in Combinatorial Optimization*. Princeton University Press, 2003.
- [16] J. H. Holland, *Adaption in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [17] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham, *Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications*, ser. Numerical Insights. CRC Press, 2009.