On the investigation of Stochastic Global Optimization algorithms

Bill Baritompa and Eligius M.T. Hendrix

June 1, 2001

Abstract

This discussion paper tries to consider the whole process of investigating algorithms. The idea is to make as explicit as possible all aspects and their interrelation. After an elaboration of viewpoints a list of discussion points follows.

1 Introduction

Research on Global Optimization (GO) mainly concerns the investigation of properties of algorithms and their interrelation with (mathematical structure of) Global optimization problems. The final aim of GO research is to get a better understanding of which methods are best suited to be used on which type of practical optimization problems.

The question posed here is how to study behaviour of algorithms in a systematic way. In figure 1 some relevant aspects are depicted. The main theme is that all aspects must be considered together:

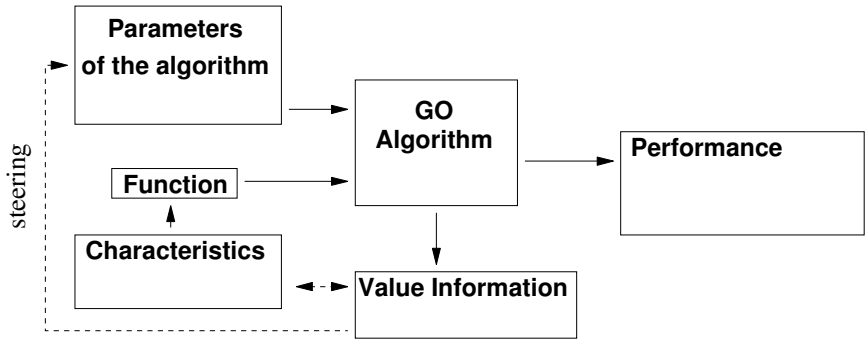

Figure 1: Aspects in investigating Stochastic Global Optimization Algorithms

1. first of all there is a description of the algorithm(s) under investigation.

2. measurable performance criteria are formulated to study the behaviour

3. the algorithm is run or analysed for certain test functions (instances, special cases). The instances correspond to a certain landscape with some **structure or characteristic**

4. often an algorithm contains certain parameters.

First an example is given and then it is described what we mean by the aspects.

Example
Pure Random Search on a compact robust set X, where N (parameter) trial points are generated uniformly over the feasible set gives an expression for the probability of "hitting" a global optimum as:

$$P = 1 - \left(1 - \frac{V(B_\epsilon(S^*))}{V(X)}\right)^N \tag{1}$$

Apparently the assumed target of a user is to reach one global minimum point. The search is assumed successful when the record value hits an epsilon environment of a global minimum point and the researcher has formulated as a performance criterion the probability of success. In this case the relevant (measurable) characteristic of the problem is $\frac{V(B_\epsilon(S^*))}{V(X)}$: the relative volume of the ϵ-neighbourhood of all global minimum points. Formula 1 is a basic result showing the infinity effort property that the optimum is reached when the effort N goes to infinity. Alternatively, the success could be defined as being in a δ-levelset from the global minimum value f^*, where the probability of success becomes:

$$P = 1 - (1 - \mu(f^* + \delta))^N \tag{2}$$

depending on the relative size of the $\delta + f^*$-levelset as a characteristic of the problem to be solved.

performance criteria
In this paper we look at several performance criteria that are used in Global optimization. We are not looking for *the* criterion, but interested in the interrelation between them. Moreover, we describe a tool which we call the *Performance Graph*. This will be described in section 2.

Characteristics
In experimental analysis in literature, the algorithm is often run over several test functions and the performance is compared with respect to other algorithms and/or for various parameter settings. Understanding the behaviour means we should go into the characteristics and landscape of test functions. The question is how to measure the characteristics. We will discuss some ideas which appear in literature in section 3. The main idea is that the relevant characteristics depend on type of algorithm as well as on the performance measure; in the example the regions of attraction, the form of the finite level sets, the barriers in the landscape do not matter for the result.

Algorithms
A profound discussion and classification of algorithms can be found in [12]. According to [5], one can roughly distinguish two major approaches:

1. Deterministic methods which aim at guarantee to approach the global optimum and therefore require a certain mathematical structure.

2. Stochastic methods which are based on the random generation of feasible trial points and nonlinear local optimization procedures.

In this paper, focus is on the investigation of stochastic optimization procedures, although many concepts are also applicable for general research questions on algorithms. Stochastic algorithms as such require no structural information on the problem to be solved. However, one can adapt algorithms to make use of additional information about the structure. Moreover, one should notice, that despite information on structure is assumed not necessarily to be available, other information, so-called **value information**,

becomes available when running algorithms, such as: number of local optima (better: results of a local optimization routine) found thus far, average number of function evaluations necessary for one local search, best function value found, etc. Such indicators can be measured empirically and on one hand can be used to get insight into what factors determine the behaviour of a particular algorithm and perhaps on the other hand can be used to improve the performance of an algorithm.

Main statement is that there does not exist **the** generic stochastic algorithm which covers all existing algorithms and perhaps we should not look for it. One of the most generic descriptions is that of Törn and Zilinskas (1989) [12]:

$$x_{k+1} = Alg(x_k, x_{k-1}, ..., x_0, \xi) \tag{3}$$

where ξ is a random variable.Can we come to a classification of stochastic global optimization algorithms? Let us try in section 6

2 Performance criteria

Globally there are two things to measure:

- Effectiveness: does the algorithm reach what we want.
- Efficiency: what are the computational costs.

First we will have a look at what can be found in literature. Then we have a look at which questions should be dealt with to define a good criterion. Finally, we suggest the performance graph as an option for studying the performance of an algorithm: We should have a picture of the trade-off between the two main criteria. The resulting question is, how criteria are related.

If we have a look at literature on investigation of the methods starting from the early works of Dixon and Szegö [3] and following publications in the Journal of Global Optimization which exist now for 10 years, the main property which is aimed at is that when effort increases to infinity, a point is generated in the ϵ-environment of a global minimum point with probability one. This defines the effectiveness of an algorithm. Analytically this main property can be derived whenever a designed algorithm is allowed to sample everywhere in the feasible area with a certain probability. Empirically, by means of test functions, one can measure how many times the global optimum has been reached by a certain algorithm (effectiveness). As in general with stochastic methods, we need many repetitions to measure performance criteria, as to average out the stochasticity. Words like *simulation*, *experiments* or *computational results* are used in this context.

Efficiency can also easily be measured in an empirical way. As a performance criterion, in general the expected number of function evaluations necessary to reach convergence, is taken. In literature experimental results are presented as tables where several algorithms or one algorithm with several parameter settings are run over some test functions and the required number of function evaluations and yes/no of reaching the global optimum.

This brings us to the question what are good ways to define the performance of an algorithm. Focusing on effectiveness there are several **targets a user** of the algorithm may have:

1. To discover all global minimum points. This of course can only be realised when the number of global minimum points is finite.
2. To detect at least one global optimal point.
3. A user wants a solution with a function value as low as possible.

4. Uniform covering: The idea as introduced by Klepper and Hendrix [10, 6], is that the final population of a population based algorithm should resemble a sample from a uniform distribution over a level set with a predefined level. The practical relevance is due to identification problems in parameter estimation. Performance indicators are difficult to construct. Usually one partitions the final region in subsets and sums the deviations from the expected number of points in each partition set.

The first and second target are typical satisfaction targets, was the search yes/no successful. This brings us to the question: what are good **measures of success**? In the older literature often convergence was measured; does the algorithm converge to "the" global optimum? If we want to make results comparable, we should be more explicit in what is assumed to be a **success**:

- is a user satisfied when the record of the search process hits a "low" level set, did it "see" the minimum or alternatively (according to the third target) should all compartments of the "low" level set be reached.
- what is considered "low", a predefined level or a percentile of the function range?
- should the process converge around a minimum point (or in population algorithms around several minimum points) to give the user a feeling this is really a minimum point?
- should alternatively the algorithm (a record value) hit (see) an ϵ-neighbourhood of one or all global minimum points?

Efficiency

Globally efficiency is defined as the effort the algorithm needs to reach the target. A usual indicator in stochastic algorithms is the (expected) number of function evaluations necessary to reach the optimum. This indicator depends on many factors such as the shape of the test function and the termination criteria used. By making more and more assumptions on the behaviour of the algorithm, one can come up with the complete distribution of number of function evaluations necessary to reach the optimum.

An alternative performance indicator for efficiency is the so-called Success Rate defined in [7] as the probability that the next iterate is better (a so called improvement) than the record value found thus far. The relevance for the concept of convergence speed are due to the analyses by Zabinsky and Smith [13] and Baritompa et al. [1], who show that a fixed success rate of an effective algorithm in the sense of uniform covering would lead to an algorithm which is polynomial in the dimension of the problem, i.e. the expected number of function evaluations grows polynomially with the dimension of the problem. However, the empirical measurement can only be established in the limit of the algorithm when it stabilises and only for specifically designed test cases.

Performance graph

In analytical studies we observe results which aim at showing the algorithm is successful in limit; when we put in infinite effort. In no realistic situation a user is going to wait up to infinity to get an answer. The question is now how to measure the performance for cases where the user has a finite amount of time (in [4] called a budget) to obtain a solution. Here we introduce the concept of the

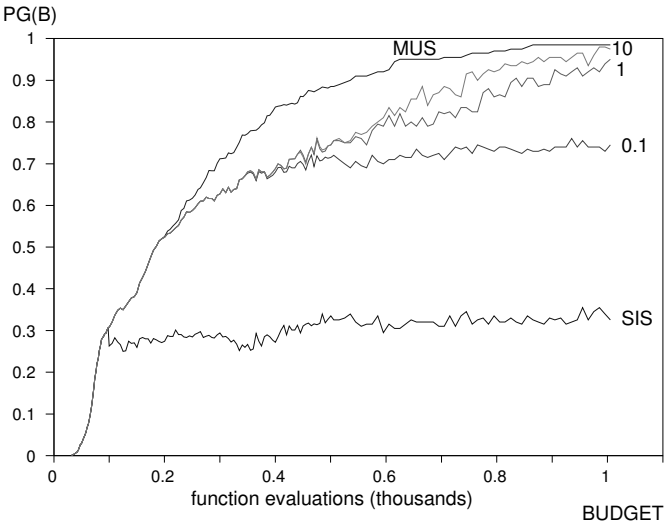

Figure 2: Probability of reaching the optimum for the Shekel-5 function and several algorithms given a number of function evaluations.

Performance Graph. The performance is defined as the probability the algorithm has been successful (depending on the successfulness indicator) for a certain amount of effort. An example [4] is given in figure 2; an estimate is given of the probability on success, based on running algorithms over many repetitions for a given amount of function evaluations. The success in this example was defined as the record being in an epsilon environment from the one global minimum point. All algorithms in the example use random points generated over the domain and (the same type of) local searches. The local searches as such were run to convergence and the same stopping criteria were used.

The probability concept should be left, when the success is not determined by a logical boolean indicator and the user just aims at low values according to target 3. In this case the performance graph intends to measure the expected value of the goal which is reached within the given effort. An example is given in figure 3.

relation between criteria

The performance graph suggests to consider the behaviour according to for instance $Psucces = Graph(Effort)$, where we can think of success of seeing a point in the ε-environment of a global minimizer. In many studies our focus is on the distrubution of Effort up to success (or convergence). How are those two concepts related?

Alternatively, let us imagine that success is defined as seeing a point in the level-set $S(y)$. If we move the level y upwards, the graph giving the probability of success is slowly moving upwards. What is the relation with the graph, which measures the expected (or distribution of the) record value? Can a performance graph be derived analytically from a given success rate (fixed rate of improvement)?

As empirical results depend on the test functions under consideration, it is good to have a better look at the structure of the functions and their corresponding landscape: which factors determine the performance. Like in a design of experiments one should construct extreme cases (best or worst cases) of the test functions as to investigate an expected relation as firmly as possible.

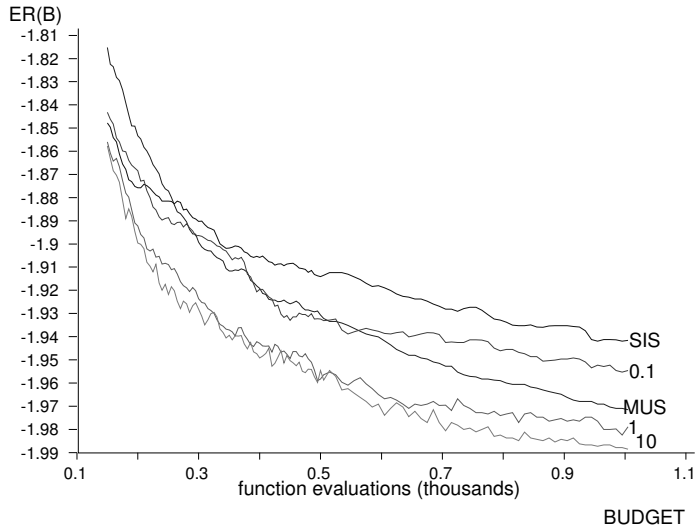

Figure 3: Average record value reached by 5 different algorithms for a given amount of function evaluations for the Rastrigin test function.

3 Characteristics of test cases

The most important concept is to vary the test cases systematically in order to get hold of how algorithms behave in various situations as is done for instance when using worst cases or extreme cases. In an experimental setting, depending on what one measures, one should construct a design of experiments which yields as much information as possible. To derive analytical results, it is not uncommon to make the weirdest assumptions about structure of the cases which have to be solved.

Examples of extreme cases to derive analytical results:

- to study the limit behaviour in the case the algorithm converges to one optimum, the spherical or conical problem.
- to illustrate the complexity, the concave quadratic problem over a hypercube, such that the number of local optima 2^n increases exponentially in the dimension n.
- in nonlinear optimization a usual characteristic is to look at the condition number of the Hessean in the minimum point, which influences efficiency of local nonlinear opimization algorithms.

When studying cases we should keep in mind that in literature on Global optimization the following types of problems are investigated.

- black box case: in this case it is assumed that nothing is known about the function to be optimized. Often the feasible set is defined as a box, but information about the objective function can only be obtained by evaluating the function at feasible points also called *the oracle case*.
- grey box case: something is known about the function, but the explicit form is not necessarily given. We can think of a lower bound on the function, the number of global and/or local optima. One step further towards deterministic methods is to assume structural information is known: the function is concave, a Lipschitz constant is known, a d.c-decomposition is known. Stochastic methods don't require this type of information, but information alike can be used to derive analytical or experimental results.

- white box case: explicit analytical expressions of the problem to be solved are assumed to be available. Specifically interval arithmetic algorithms require this point of view on the problem to be solved.

When looking at the structure of the instances for which we study the behaviour of the algorithm we should keep two things in mind.

- In experiments the researcher can try to influence the characteristics of the test cases such that the effect of that what is measured is as big as possible. This is called design of experiments. Notice that the experimentalist knows the structure in advance, but the algorithm doesn't.
- The algorithm can try to generate information which tells it about the landscape of the problems. We will enumerate some information which can be measured in the black box case.

When we have a look at the lists of test functions in literature (a.o. [12]), we observe as characteristics the number of Global minimum points, the number of local minimum points and the dimension of the problem.

Difficulty in the analysis is of a GO algorithm on a multiextremal case is, that everything seems to influence the behaviour: The orientation of compartments of lower level sets with respect to each other determine how good iterates can jump from one place to the other. The number of local optima up in the "hills" determine how algorithms may get stuck in local optima. The difference between the global minimum and one but lowest minimum determines the possibility to detect the global minimum point. The steepness around minimum points, valleys, creeks etc. which determine the landscape determine the success.

However, as shown in the first example, the **characteristics** which are important for the behaviour depend on the **type of algorithm** which is investigated and the **performance criteria** that describe the behaviour.

So in every analytical and experimental investigation of an algorithm we should be concerned with the type of characteristics which matter for the algorithm and performance criterion under consideration. Can we classify this?

- multistart type algorithms: size of regions of attractions of optima, number of local and global optima.
- clustering algorithms: idem + shape of regions of attraction.
- population based algorithm with focus on success rate: number, orientation (Hessean) of compartments of lower (limit) level sets.
- Hit and Run, MC type of algorithms with focus on expected number of function evaluations: barriers, creeks and rivers, location,orientation and number of saddlepoints.

If we want to look for weaknesses and strong points of algorithms we should consciously look for extreme cases with respect to this type of characteristics. It may be hard to measure characteristics from given test cases. The complete μ graph predicts the success of some random schemes and in [4] a characterising graph can be found that determines how successful it is to do more or less local searches. Clear drawback of this characterisation is that there is a complete graph describing the relevant structure of the landscape.

From the perspective of designing algorithms, running them empirically may **generate information** about the landscape of the problem to be solved. In [5] the following enumeration can be found of information one could measure during running a stochastic GO algorithm on a black box case:

- graphical information on the decision space

- current function value
- best function value found so far (record)
- number of evaluations in the current local phase
- number of optima found
- number of times every detected minimum point is found
- estimate of the time of one function evaluation
- estimate number of function evaluations for one local search
- indicator on how certain the optimum has been reached

For the latter figure a probability model is needed or simple considerations such as can be found in the early Karnopp result, see [9]. Measuring and using the information in the algorithm, usually leads to more extended algorithms with additional parameters complicating the research question on what are good parameter settings.

4 Euclidean landscape is misleading

One of the causes of diminishing research capability is that we look at the graph of the function with the Euclidean landscape eyes, because we are thinking in three dimensional landscape. Trying to measure characteristics from this perspective is just wrong. We should try to think more abstract and see the landscape from the perspective of the algorithm and its assumed neighbourhood structure.

For pure random search, the next generated point does not depend on the current iterate and therefore we can consider all points to be in the neighbourhood of the current iterate. In this perspective **local non-global optima do not exist**. We should redefine the neighbourhood structure by integrating over the probability distribution over all points which can be reached from a certain point given the algorithmic operation (3).

This defines a new topological structure. If we could do so, we'd get to the heart of what are relevant characteristics to measure. It is just a challenge to see algorithms and spaces from this perspective. Challeging question is, how to define points and neighbourhood structures for population algorithms?

5 Comparison of algorithms

In comparing algorithms, one could call algorithm[1] **dominated**, whenever there exists an algorithm which perfoms better, has a better performance graph for instance, than algorithm[1] for all possible cases under consideration. Usually however, one algorithm runs better on one case and another on another case as is illustrated comparing figures 2 and 3, where algorithm MUS does best on the Shekel-5 function, but there are better ones on the Rastrigin function.

So basically the performance of algorithms can be compared on the same test function, or preferably for many test functions with the same characteristic, where that characteristic is the only parameter that matters for the performance of the compared algorithm. As we have seen in section 3, it may be very hard to find out such determination characteristics. In any comparison we should be keen about some principles:

- Comparability: if we compare several algorithms they should make use of the same type of (structural) information, of the same stopping criteria, accuracies etc.
- Simple references: it is wise to include in the comparison simple benchmark algorithms such as Pure Random Search, Mulistart and pure Adaptive Search in order not to let analysis of the outcomes get lost in parameter tuning and complicated schemes.

Often in literature we see algorithms applied for solving "practical" problems. If we are comparing algorithms for a practical problem, we should keep in mind this is only one problem and up to now, nobody has defined what is a representative practical problem.

6 Classification of algorithms

Stochastic methods are understood to contain some stochastic elements. Either the outcome of the method is a random variable or the objective function itself is considered a realisation of a stochastic process. For an overview on stochastic methods we refer to Törn and Zilinskas (1989) [12] and Boender and Romeijn (1995) [2]. Apart from the continuity of f, the methods in general require no structure on the optimization problem. Therefore the methods are generally applicable. If we would like to classify algorithms, then it is useful to do this from the perspective of same information used and same characteristics which matter for the performance of the algorithm. Let us start with making a list:

- random function approaches (Kushner, Mockus, Zilinskas) where the outcome of the function is considered a stochastic variable up to the moment it is evaluated. Actually it is a deterministic approach in the sense that no stochastic variables are used to generate new points, but a stochastic model is used to describe the (random) function given all former iterates. Perhaps this teaches us some ways to tackle noisy objective functions.
- simple multistart type of approaches, where multistart is mixed with PRS (multi-singlestart) to generate random starting points.
- clustering algorithms.
- hit and run type of approaches: next iterate depends on current iterate and there is an acceptance/rejection rule.
- population based algorithms: generate offspring and select.

Is this enumeration exhaustive?

7 Summary and discussion points

- results of investigation of SGO algorithms consist of a description of the performance depending on algorithms (parameter settings) and characteristics of test functions or function classes.
- to obtain good performance criteria we should define what is assumed to be the target of an assumed user and what is considered as success.
- the performance graph is a useful instrument to compare performances of algorithms.
- What is the relation between the distribution of the number of necessary function evaluations to reach success and this performance graph?
- What is the difference when success is defined by seeing a point in the epsilon environment or seeing a point in the $S(y)$ level-set?
- the relevant characteristics of the investigated cases depends on the type of algorithm and performance criterion under consideration.
- it makes no sense to think in landscapes, when we don't consider it with the viewpoint of the algorithm.
- when algorithms are compared they should at least be comparable: make use of same information, accuracies, principles etc.

- it is wise to use pure simple benchmark algorithms like PRS, PAS and Multistart as a reference.
- a generic algorithmic description for all SGO algorithms does not exist.

References

[1] W. P. Baritompa, R.H. Mladineo, G. R. Wood, Z. B. Zabinsky and Zhang Baoping. Towards Pure Adaptive Search. *Journal of Global Optimization*, 7, 73-110. 1995.

[2] C. G. E. Boender and H. E. Romeijn. Stochastic methods, in *Handbook of Global Optimization* (Horst, R. and Pardalos, P.M. eds.), Dordrecht: Kluwer, 829-871. 1995.

[3] L. C. W. Dixon and G. P. Szegö, eds. *Towards Global Optimisation*, North Holland, 1975.

[4] E. M. T. Hendrix and J.Roosma. Global Optimization with a Limited Solution Time. *Journal of Global Optimization*,8,413-427, 1996.

[5] E. M. T. Hendrix. Global Optimization at Work. PhD thesis, Wageningen Agricultural University, 1998.

[6] E. M. T. Hendrix and O. Klepper. On uniform covering, adaptive random search and raspberries. *Journal of Global Optimization*, 18, 143-163. 2000.

[7] E. M. T. Hendrix, P. M. Ortigosa, and I. García. On Success Rates for Controlled Random Search. *Technical Note 00-01*, Department of Mathematics Wageningen, 2000. (to appear in JOGO).

[8] R. Horst and P. M. Pardalos, eds. *Handbook of Global Optimization*, Dordrecht: Kluwer, 1995.

[9] D. C. Karnopp Random search techniques for optimization problems. *Automatica*, no. 1, pp. 111–121, 1963.

[10] O. Klepper and E. M. T. Hendrix. A Method for Robust Calibration of Ecological Models under Different Types of Uncertainty *Ecological Modelling*, 74, 161-182. 1994.

[11] N. R. Patel, R. Smith and Z. B. Zabinsky. Pure adaptive search in Monte Carlo optimization. *Mathematical programming*, 43, 317-328. 1988.

[12] A. Törn and A. Zilinskas. *Global Optimization. Lecture Notes in Computer Science 350*. Springer-Verlag, Berlin, 1989.

[13] Z. B. Zabinsky and R. L. Smith. Pure Adaptive Search in Global Optimization. *Mathematical Programming*, 53, 323-338. 1992.