Journal of Global Optimization **13:** 445–454, 1998.
© 1998 *Kluwer Academic Publishers. Printed in the Netherlands.*

Random and Quasi-Random Linkage Methods in Global Optimization

FABIO SCHOEN
Dipartimento di Sistemi e Informatica, Firenze, Italy

(Accepted in final form 30 June 1998)

Abstract. In this paper a brief survey of recent developments in the field of stochastic global optimization methods will be presented. Most methods discussed fall in the category of two-phase algorithms, consisting in a global or exploration phase, obtained through sampling in the feasible domain, and a second or local phase, consisting of refinement of local knowledge, obtained through classical descent routines. A new class of methods is also introduced, characterized by the fact that sampling is performed through deterministic, well distributed, sample points. It is argued that for moderately sized problems this approach might prove more efficient than those based upon uniform random samples.

Key words: Two phase methods, Simple linkage, Quasi-random sequences

1. Introduction

In this paper an analysis of some stochastic algorithms for global optimization will be carried out with the aim of discovering some characteristic of their finite time behaviour. In particular methods belonging to the class of *two-phase* algorithms will be analyzed. Two-phase methods consist of a global, or exploration, phase aimed at sampling, as evenly as possible, the feasible region (Phase I), coupled with a strategy for the refinement, or approximation, of local optima (Phase II). Many, if not all, methods of global optimization are based upon such a general scheme. Usually the first phase is implemented through uniform random sampling or through deterministic sequences, while the second phase, depending on the structure of the objective function, is usually based upon local optimization routines. For objective functions whose structure is so poor (or whose derivatives are so difficult to evaluate) that classical, gradient based, local optimization is not possible, frequently the local phase is obtained through localized random sampling, i.e., sampling in small neighborhoods of the current point. An interesting method in this class is described in Locatelli (1996), where theoretical results are derived for a class of simulated-annealing algorithm based upon sampling in small spheres around the current iterate.

Two phase methods in the strict sense, however, are based upon starting local optimization routines from carefully selected points in a sample. For a brief survey of two-phase methods in the strict sense the reader might wish to consult Schoen (1998).

In this paper, after a survey of recent results concerning a class of two phase methods with strong theoretical properties and good practical behaviour, some considerations will be made on the finite time behaviour of such methods. In fact all known theoretical results are based upon asymptotic considerations, while little is known on the practical behaviour in the first iterations. Such an analysis will lead to the proposal of discarding random samples, at least for moderately sized global optimization problems, and to substitute them with deterministic sequences of points, built in such a way as to guarantee an extremely even coverage of the feasible region. While this idea is not new in the literature, published results usually consist in just substituting well-spaced quasi-random sequences in place of uniform samples; here it is argued that such a substitution implies a radical change in the definition itself of the algorithm.

2. Simple Linkage methods

In this section a survey will be presented of the main definitions and properties of *Simple Linkage*, a class of methods which was recently developed and analyzed in Locatelli and Schoen (1996, 1998). Let us consider the basic global optimization problem

$$f^{\star} = \min_{x \in [0,1]^d} f(x)$$

with f a continuous function, as smooth as it is required by the local optimization algorithm we plan to use. The scheme of Simple Linkage is the following:

1. set $k := 0$; choose $\sigma > 0$ and $\epsilon > 0$;
2. let $k := k + 1$;
3. generate a single random point X in $[0, 1]^d$;
4. let the threshold $r_{k;\sigma}$ be defined as follows:

$$r_{k;\sigma} := \pi^{-1/2} \left(\Gamma(1 + d/2)\, \sigma\, \frac{\log k}{k} \right)^{1/d}; \tag{1}$$

5. apply a local search algorithm from X except if $\exists\, X_j$ in the sample:

$$\|X - X_j\| \leq r_{k;\sigma} \text{ and } f(X_j) \leq f(X) + \epsilon \tag{2}$$

6. Stop? If not, add X to the sample and goto 2.

It is not prescribed in this method to store the local optima found during the iterations; even if the theoretical results remain unchanged, some experiments in this direction displayed that this inclusion had no significant impact on the overall performance.

This method was originally inspired by Multi-Level Single-Linkage (Rinnooy Kan and Timmer, 1987a,b) and was built in order to circumvent the most relevant defects of that algorithm, in particular the necessity of sampling in batches of points, the necessity of reconsidering, after each sample, the whole set of sampled points as possible candidates for starting local searches, and the impossibility of starting local searches from points near to the boundary. All these defects were removed, without sacrificing any of the theoretical properties of the method. Moreover, computational results were obtained which confirmed the superiority of this method as well as its applicability to problems with a high number of variables. Some details on such experiments are reported in the above cited papers and in Schoen (1997), where a variant of Simple Linkage is applied to the minimization of the Lennard-Jones potential energy for clusters of up to 20 atoms: these are extremely hard global optimization problems with a number of variables which is 3 times the number of atoms and an estimated number of local optima which is exponential in the number of atoms.

The main theoretical properties of Simple Linkage are the following:

1. the best observed function value converges, with probability 1, to the global optimum value $f^\star$;
2. the probability of starting a local search goes to 0 if $\sigma > 0$;
3. the total number of local searches performed even in the case the algorithm is never stopped remains finite if $\sigma > 2^d/d$.

Unfortunately, these assertions express asymptotic properties and fail to give any insight in what actually happens in the first iterations of these methods. In particular, all of the properties rely on the fact that a continuous function in a compact set is also uniformly continuous; thanks to this fact, it is possible to prove Locatelli and Schoen (1998) that, provided that k is large enough, a local search is started from a sample point X if and only if none of the points $X_1, X_2, \dots, X_k$ is closer to X than the threshold $r_{k;\sigma}$. This means that, for large enough k, the decision whether to start or not a local search no more depends on f, but only on the relative density of sample points. While this fact is very desirable from the point of view of theoretical analysis, its practical effects are largely unknown. It should be observed that the values of k which allow us to neglect function values are usually astronomically high; what happens in finite time may thus be completely different in general from what asymptotic theory predicts.

In Schoen (1997) some considerations on this finite time behaviour were carried out. In particular, it was observed that for high-dimensional problems the value of the threshold (1) is very high; its order of magnitude, for fixed k and increasing dimension d, is comparable to the diameter of the unit hypercube. This fact implies that the neighborhood inside which no better sample point than the current one should be found in order to start a local search is comparable with the whole feasible set. Thus the finite time behaviour of the method is similar to that of Best Start, a simple algorithm which prescribes to start a local search only when a record, i.e. the best observed function value so far, is hit.

A partial remedy has been given in Schoen (1997), where two devices were introduced in order to reduce the threshold (1). The first one is to change the norm used in the comparison from the Euclidean to the infinite (or maximum) norm. This has the effect of keeping the diameter of the feasible region fixed as the dimension d increases. A second device used was that of letting the threshold be different depending on the position of the current sample point with respect to the boundary. The rationale behind this modification is that the magnitude of the threshold is largely determined by the density of uniform points near the border and, in particular, near the vertices of the hypercube. The results obtained with this modifications are quite encouraging, in particular for large global optimization problems. The modified method becomes the following:

1. set $k := 0$; choose $\sigma > 0$ and $\epsilon > 0$;
2. let $k := k + 1$;
3. generate a single random point X in $[0, 1]^d$;
4. let

$$R_{i;k;\sigma} := \left(\sigma\, 2^{i-d} \frac{\log k}{k}\right)^{1/d}; \tag{3}$$

5. let $i :=$ number of components of X which are less than $R_{0;k;\sigma}$ or greater than $1 - R_{0;k;\sigma}$,
6. apply a local search algorithm from X except if $\exists\, X_j$ in the sample:

$$\|X - X_j\|_\infty \leq R_{i;k;\sigma} \quad \text{and} \quad f(X_j) \leq f(X) + \epsilon$$

7. Stop? If not, add X to the sample and goto 2.

All the theoretical properties of the basic Simple Linkage algorithm are retained in this modification; however the fact that the threshold is different for central points and border points makes the method more efficient for higher dimensional problems.

Even with this modifications however, the problem remains of relatively high thresholds in the first iterations; the reader should not be illuded by the word ‘few’: even in quite small dimensional problems this number may well be over several thousands. The net effect of this starting anomaly is that for a long initial period methods based upon uniform random sampling behave almost like Best Start; thus from one side a complex threshold mechanism is set up without being useful; on the other side, Best Start is very inefficient, and it can become very slow especially when a local optimum whose function value is quite near to the global one is discovered: in this case, even if the region of attraction of the global optimum might in principle be very large, a local search in its basin of attraction will not be started until a point better than the best so far is sampled.

3. Sampling with quasi random sequences

As it has been argued in the preceding section, methods based upon starting local searches from points which are locally optimal can become quite inefficient when too large a threshold is chosen in the definition of 'local optimality'. In particular all of the thresholds used in methods based upon random uniform sampling are $O\left((\log k/k)^{1/d}\right)$ and, depending on the constant of proportionality and/or on the dimension d, this value is usually so large in the first iterations as to prevent most local searches to be started. It should come as no surprise the fact that the asymptotic behaviour of the threshold is determined by classical properties of uniform random sampling. In particular it was shown in Deheuvels (1983) that, given a uniform sample $X_1, \ldots, X_k$ in $[0, 1]^d$, the *maximum dispersion*

$$d'(X_1, \ldots, X_k) := \sup_{x\in[0,1]^d} \min_{i=1,k} \|x - X_i\|_\infty$$

is almost surely

$$O\left((\log k/k)^{1/d}\right).$$

Thus it is natural that all the results related to the finiteness of the total number of local searches performed rely on thresholds with similar asymptotic behaviour. But, as we already pointed out, if this is a desirable asymptotic feature, its finite time behaviour may inhibit most local searches.

Another difficulty with methods based upon random sampling is that, even in finite time, there are two distinct possibilities for a local search to be started: first, it might be started from a local record, i.e., from the best point in a neighborhood whose diameter is regulated by the chosen threshold. As we already noticed, this event, in particular during the first iterations and when using slowly decreasing thresholds, happens usually only when the current point is a global (as opposed to local) record. The second possibility for starting a local search comes from the fact that the randomness of the sample might produce a 'large gap', i.e. a point might be sampled very far from all the other points in the sample; in this case a local search will be started because that point is obviously a local record (being the unique point in a sufficiently large region). It is thus likely that a local search is started erroneously, just as a consequence of the fact that a 'hole' in the sample has been produced.

In order to try to overcome these two difficulties, different sampling strategies have been recently investigated. In particular both defects can be, at least in part, avoided by means of sampling in a more even way. The literature on deterministic sequences with low dispersion is quite large; the interested reader might consult the monograph Niederreiter (1992) or the recent book Drmota and Tichy (1997). Some attempts of using quasi-random sequences with low dispersion can be found even in the global optimization literature; however in those attempts a quasi-random sequence was just used as a substitute for a uniform random one; here we claim that the substitution must be carried out with great care and possibly with a redesign of the algorithm.

One of the most well known sequences of point in $\mathbb{R}^d$ characterized by low dispersion is the so-called *Halton sequence*, first published in Halton (1960). These sequences possess very interesting properties which make them quite suitable for use in our setting. In particular, if $X_1, \dots, X_k$ have been generated as a Halton sequence, the following hold:

- $d'(X_1, \dots, X_k) \leq k^{-1/d} \max_{i=1,\dots,d} b_i$ where $b_1, \dots, b_d$ are different prime numbers;
- the generic k–th point in the sequence may be generated directly, without the necessity of knowing the previous $k-1$ points;
- being the sequence extremely regular, it may be possible to speed up computations when looking for the nearest neighbor of the current point.

The first characteristics shows that the Halton sequence is sensibly more dense and regular than the uniform one, being the dispersion $O(k^{1/d})$. So, for a given dimension d, using quasi-random points in place of uniformly distributed ones, guarantees a much more dense coverage of the feasible set. Unfortunately the constant of proportionality grows quite rapidly as the dimension d increases (but the same happens also for random samples). Results can be derived from the well known prime number theorem, a consequence of which is that the d–th prime number grows asymptotically as $d(\log d + \log\log d - 1)$ Ribenboim (1995). This fact makes Halton sequences adapt only for low-dimensional problems. The second property, i.e. the possibility of directly generating each point in the sequence, can be exploited in our setting in order to avoid one of the major problems in two-phase methods, i.e. the necessity of storing the whole sample in order to be able to compute nearest neighbor distances. In this case it is sufficient to store only function values at each sampled point, with a memory requirement of $O(k)$, as opposed to $O(kd)$, necessary when the whole sample has to be memorized. Finally, an implementation might exploit also the fact that the regularity of the sequence is predictable and it is possible to restrict the explicit distance computation only to a few candidate points. This possibility is actually under investigation and details are expected to appear in a forthcoming paper.

In our first experiments we decided not to use the Halton sequence, but to generate points according to a more complex mechanism described in the cited monograph Niederreiter (1992). The sequence we used is described in Chapter 4 of Niederreiter's monograph, under the name of (t, s)-sequence; in particular we used $(T_2(d), d)$-sequences in base 2 and based our computations on a public domain code published in Bratley et al. (1994). We refer the interested reader to the cited monograph for details on these sequences. Here, we just cite the result on dispersion which guarantees the following upper bound

$$d'(X_1, \dots, X_k) \leq \frac{2^{1+T_2(d)/d}}{k^{1/d}}$$

where $T_2(d)$ is a function (whose values are tabulated in Niederreiter, 1992) which does not grow too fast. In particular it can be proven that

$$T_2(d) < d(\log_2 d + \log_2 \log_2 d + 1)$$

Again, the constant in the asymptotic expression for the dispersion is asymptotic to $d \log d$. However, this is an upper bound and, in practice, the actual constant is quite lower; recent results also appeared which enable to build sequences in base 2 with a coefficient in the dispersion which is only $O(d)$ (which is the lowest possible theoretical bound), but we did not have the opportunity to test them.

Even for moderately high-dimensional problems, (t, s) sequences give very regularly spaced points. Even better results for what concerns the coefficient in the asymptotic bound on dispersion can be obtained using (t, d) sequences in a prime base different from 2, but we choose base 2 nets as they can be implemented in a much more efficient way.

From a theoretical point of view, using quasi-random sequences in place of uniform random samples does not change the main theoretical properties of the method; however, it is possible to fully exploit the regularity of quasi-random sequences by using a threshold which is only $O(k^{-1/d})$. We have in particular the following result.

THEOREM 1. *If the threshold (1) in Simple Linkage is modified in the following way:*

$$r_{k;s} := sk^{-1/d} \tag{4}$$

and sample points are generated according to a quasi-random sequence whose dispersion is bounded above by $\alpha(d)k^{-1/d}$, (where $\alpha(d)$ is constant with respect to the iteration counter k) then the total number of local searches started even if the algorithm is never stopped will remain finite provided that

$$s > \alpha(d)$$

Proof. Thanks to the continuity of f and the compactness of the feasible domain, f is also uniformly continuous; thus, given any $\epsilon > 0$ (and, in particular, the ϵ used in the criterion (2) used for deciding whether to start or not a local search), there exists $\delta = \delta(\epsilon)$ such that

$$\|x - y\| \leq \delta \Rightarrow |f(x) - f(y)| < \epsilon$$

Thus, when k is so large that $\alpha(d)k^{-1/d} \leq \delta(\epsilon)$, for every point X there will exist another point Y in the sample whose distance satisfies $\|X - Y\| \leq \delta(\epsilon)$; this implies that function values at X and Y will not differ by more than ϵ and, according to criterion (2), no local search will be started from X when using a threshold which is larger than $\alpha(d)k^{-1/d}$. □

The above proof, although extremely simple, again is based upon asymptotic considerations and does not give any insight on the finite time behaviour of the algorithm; in particular it should be observed that the theorem comes into effect only when the sample is so dense that an observation of f has been placed in every element of a covering of $[0, 1]^d$ with hypercubes of volume δ^d. In practice, in the first iterations, f will play an important rôle and thus it is advisable to choose a threshold which might also be strictly lower than the value prescribed in this theorem. In fact, choosing too large a threshold in the first iterations will inhibit any local search except those from the global records, thus leading again to Best Start.

4. A few numerical experiments

An extremely limited set of numerical experiments have been performed in order to see the effect of changing both the sampling strategy from uniform to quasi-random, and the threshold, from $O(\log k/k)^{1/d})$ to $O(k^{-1/d})$. Here only a single case will be presented, while more extensive numerical experiments are planned and will appear elsewhere.

The tests concern a quite difficult, even if low-dimensional, global optimization problem, known as ‘the penalized Shubert function’:

$$\begin{aligned} f(x_1, x_2) = &\prod_{i=1}^{2} \sum_{j=1}^{5} (j \cos((j+1)x_i + j)) \\ &- 0.5((x_1 + 1.42513)^2 + (x_2 + 0.80032)^2) \end{aligned}$$

This 2-dimensional test function has 760 local optima, a single global one and, provided that its special structure is not taken into account, is a quite challenging test for general purpose global optimization algorithm which consider f as a ‘black-box’. We performed 100 independent tests on this function both using Simple Linkage and the new method based upon quasi-random sampling; with the term ‘independent’, we mean that for algorithms based upon random sampling we used different seeds in initializing the random number generator; for quasi-random sequences, we used a similar device, i.e. the generation of quasi random vectors was started from a ‘random’ point in the sequence. The algorithms were stopped as soon as the global optimum (which for this test function is known) was observed for the first time with a relative error smaller than 10^{-6}. The following table gives the medians (over 100 runs) of the number of function evaluations, of gradient evaluations, of local searches, the median cardinality of the sample and the average CPU time in seconds (on a SUN Ultra workstation). In the first column of the table we give the value of σ and s used in the experiments.

Simple Linkage

Parameters	f.e.	g.e.	l.s.	Sample	CPU
$\sigma = 1$	5283	4104.5	165	957.5	1.71
$\sigma = 2$	3395.5	1908.5	84	1333	1.54
$\sigma = 4$	2612.5	880	43.5	1699.5	1.56

Quasi Random Linkage

Parameters	f.e.	g.e.	l.s.	Sample	CPU
$s = \sqrt{2}$	3095.5	2106	84	941	0.94
$s = 2$	1667.5	789.5	38	799	0.98
$s = 4$	1925	432	24	1425.5	0.64

It is quite evident that a sensible improvement, both in terms of function evaluations, and in CPU time, is obtained through a more regular sample than that obtained through quasi random sequences. The exact value of $T_2(2)$ is 0, so the maximum dispersion in dimension 2 of the sequence we used is bounded above by $2/\sqrt{k}$; we performed 3 groups of experiments, with a strictly lower, an exact, and a higher threshold with respect to that required by the hypothesis of the preceding theorem.

We performed similar experiments on other moderately low-dimensional problems and the results confirm the superiority of quasi-random methods over stochastic ones. When however tests were made with high-dimensional problems, like, e.g., the minimization of the Lennard-Jones potential energy function of clusters of atoms (see for a recent survey on computational biology Neumaier (1997), the results were of very difficult interpretation, with quasi-random methods being ‘randomly’ much better or much worse than simple linkage. An explanation of this unpredictable behaviour is quite easily found: in high dimensional spaces there is no point in distinguishing between evenly distributed families of points in terms of dispersion. In fact, even regular sequences like those we used, need to sample 2^d points just to place a single point in each of the hypercubes obtained by dividing into two equal segments each edge of the unit hypercube. For a relatively small problem like, e.g., the minimum energy configuration of a cluster of 13 atoms, d is equal to 36 (variables in this problem are the coordinates, in $\mathbb{R}^3$, of the centers of each atom, except one which we arbitrarily place at the origin). So we need 2^{36} iterations just to place an observation in each box whose edges are one half of the original one. It is clear that, both for space and for time limitations, we always stop our algorithms well before 2^{36} iterations. The impression of this author is that for high dimensional problems there is no hope of finding either stochastic or deterministic reliable global optimization methods, unless the structure of the problem is exploited as much as possible. Obviously these considerations are a natural consequence of the fact that global optimization problems, even when restricted to very special instances, remain NP-hard and, thus, no general purpose algorithm can be expected to perform well on any problem, unless the structure of the problem is taken into account as much as possible.

Investigations are currently planned to understand how the structure of the problem might be taken into account in designing simple-linkage-like methods for high dimensional problems.

Acknowledgment

This research has been partially supported by project M.O.S.T.

References

Bratley, P., Fox, B.L., and Niederreiter, H. (1994), Algorithm 738: Programs to generate Niederreiter's low-discrepancy sequences, *ACM Transactions on Mathematical Software* 20: 494–495.

Deheuvels, P. (1983), Strong bounds for multidimensional spacings, *Z. Wahrsch. Verw. Geb.* 64: 411–424.

Drmota, M. and Tichy, R. (1997), Sequences, discrepancies and applications, Vol. 1651 of *Lecture Notes in Mathematics*, eds., A. Dold, F. Takens, Springer Verlag, New York.

Halton, J.H. (1960), On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numer. Math.* 2: 84–90.

Locatelli, M. (1996), Convergence properties of simulated annealing for continuous global optimization, *Journal of Applied Probability* 33: 1127–1140.

Locatelli, M. and Schoen, F. (1996), Simple Linkage: Analysis of a threshold-accepting global optimization method, *Journal of Global Optimization* 9: 95–111.

Locatelli, M. and Schoen, F. (1998), Random Linkage: a family of acceptance/rejection algorithms for global optimisation, *Mathematical Programming*, to appear.

Neumaier, A. (1997), Molecular modeling of proteins and mathematical prediction of protein structure, *SIAM Review* 39: 407–460.

Niederreiter, H. (1992), Random number generation and quasi-Monte Carlo methods, SIAM

Ribenboim, P. (1995), The New book on Prime Number Records, 3rd edn, Springer Verlag. New York.

Rinnooy Kan, A.H.G. and Timmer, G. (1987a), Stochastic global optimization methods. Part I: Clustering methods, *Mathematical Programming* 39: 27–56.

Rinnooy Kan, A.H.G. and Timmer, G. (1987b), Stochastic global optimization methods. Part II: Multi level methods, *Mathematical Programming* 39: 57–78.

Schoen, F. (1997), Global optimization methods for high-dimensional problems, *European Journal of Operations Research*, submitted.

Schoen, F. (1998), Stochastic global optimization: Two phase methods, in C. Floudas and P. Pardalos (eds.), *Encyclopedia of Optimization*, to appear. Kluwer Academic Publishers, Dordrecht/Boston/London.