Theoretical and experimental analysis of random linkage algorithms for global optimization

Marco Locatelli

Università di Milano

Dip. di Scienze della Informazione, via Comelico 39/41, 20139 Milano, Italy. Tel: +39 - 2 - 55006 300.

e-mail: locatelli@hermes.mc.dsi.unimi.it

Fabio Schoen

Università di Firenze

Dip. di Sistemi e Informatica, via di Santa Marta 3, 50139 Firenze, Italy. Tel: +39 - 55 - 4796 358. Fax: +39 - 55 - 4796 363.

e-mail: schoen@ingfi1.ing.unifi.it

www: www-dsi.ing.unifi.it/~schoen/home.html

Abstract

In this paper a class of randomized algorithms for the optimization of multimodal functions is introduced and some theoretical results briefly recalled. A comparison, both theoretical and experimental, is also performed between some members of this class of algorithms and a well known method of global optimization - Multi Level Single Linkage. It is shown that it is possible to obtain similar, and even better, results with the proposed algorithms, with significantly lower computational effort.

Keywords

Global Optimization, Stochastic Algorithms, Clustering, Multi Level Single Linkage

1 PROBLEM STATEMENT

We consider the following problem: find

$$f^* := \max_{x \in \Omega} f(x) \tag{1}$$

where $\Omega \subseteq \mathbb{R}^d$ is a compact set, $d \geq 1$, $f \in \mathcal{C}(\Omega)$; in this paper it will be assumed that $\Omega = [0, 1]^d$. The function f is usually non concave, so that classical “local” methods cannot

be employed to solve the problem. However, if f is sufficiently regular, good and reliable local optimization algorithms usually do exist, so that it is quite natural to design global optimization methods which mix in an appropriate way local and global procedures. As problem (1) is NP-hard (even very special sub-classes of this problem have been proven to be NP-hard – see, for example, (Pardalos and Vavasis, 1991)), randomized algorithm often are considered as a good choice; one quite attractive such an approach, and one which has been successfully implemented in recent years, consists of repeatedly sampling points from a uniform distribution over the feasible set, coupled with local searches started from selected points in the sample. At two opposite extrema of this class of algorithms we find Multistart, in which a local search is started from each sampled point, and Single Start, where a single local search is started from the best (record) point at the end of the sampling process.

In this paper we introduce and analyze a class of algorithms which in some sense is intermediate between Multistart and Single Start, where the decision whether to start or not a local search from a sampled point is made on-line, depending both on the relative merit (function value) of the current point and on the distance between the current point and other points in the sample. This way it is hoped that unnecessary local searches, possibly leading to already discovered or to non interesting local optima, can be avoided, while retaining some good convergence property.

2 MULTI LEVEL SINGLE LINKAGE

Multi Level Single Linkage (MLSL for short) is a popular stochastic global optimization algorithm introduced in (Rinnooy Kan and Timmer, 1987 a-b). Let $N > 0$, $\sigma > 0$, $\gamma \in (0, 1]$ be parameters chosen by the user and let $k := 0$; MLSL can be defined as follows:

1. let $k := k + 1$;
2. let

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

3. generate a uniform random sample of size N in Ω ;
4. sort the whole sample of Nk points in order of decreasing function value, and select the γNk best of them; the resulting sample is called *reduced sample*;
5. apply a local search algorithm from each point in the reduced sample except if there exists another point with higher function value at a distance which is less than or equal to the threshold $r_{k;N;\sigma}$;
6. check if some stopping condition is satisfied, otherwise repeat from 1.

The rationale behind MLSL is that if a sampled point is “near” to a better one, either a local search was started from the latter, or that point was in turn “near” to another one, with even better function value, and so on. Eventually, a non decreasing chain of sampled points will be detected, the last point being a point from which a local search has been

started. The true meaning of “near” obviously depend on the sample size, and is reflected in (2).

Among the main drawbacks, in the authors’ opinion, of MLSL, there are both the necessity of sampling in batches of N points (with a related difficult decision about which value to assign to N) and the necessity of reconsidering, after each sample, each point in the reduced sample (except those points from which a local search was already started) in order to judge whether, with the updated threshold (2), a local search should be started. Moreover the original definition of MLSL does not allow starting a local search from a point which is too near to the boundary of Ω (this fact is crucial in the proofs of convergence). On the positive side, MLSL enjoys strong theoretical properties, among which most notably are:

1. convergence of the record value to f^* with probability 1;
2. the probability of starting a local search decreases to 0 if $\sigma > 2$;
3. the total expected number of local searches performed, even if the algorithm is run forever, is finite if $\sigma > 4$.

3 RANDOM LINKAGE

Starting from MLSL, the authors reconsidered the basic idea of the algorithm, and generalized it, while introducing significant simplifications. The theoretical analysis of Random Linkage is reported in full details in (Locatelli and Schoen, 1995). Here we simply recall that, differently from MLSL, a Random Linkage algorithm

1. samples sequentially (*i.e.* $N = 1$);
2. does not reconsider previous decisions about whether to start or not a local search; in other words, at iteration k a local search may possibly be started only from X_k , the most recently sampled point;
3. a local search may be started also from points near or over the boundary of the feasible region;
4. the decision whether to start or not a local search may be randomized, with the probability of starting a local search being a non increasing function of the distance between the current point and the nearest better point in the sample (the name “Random Linkage” came from this particular acceptance/rejection mechanism).

In this paper we will report only results relative to a sub-class of Random Linkage, named Threshold Random Linkage (TRL) which is characterized by the fact that the decision whether to start or not a local search is not randomized, but instead it is given by the following deterministic rule: start a local search from X_k , having observed f at $X_1, \dots, X_{k-1}$ if and only if

$$\min\{\|X_k - X_j\|, j = 1, \dots, k-1 : f(X_j) \geq f(X_k)\} \geq \alpha_k \quad (3)$$

where $\{\alpha_k\}$ is a non-increasing positive sequence. For this special class of algorithms, and the more general one with randomized decisions, a complete characterization was given

in (Locatelli and Schoen, 1995). In particular, if the criterion for deciding whether to start a local search is slightly relaxed as follows:

$$\min\{\|X_k - X_j\|, j : f(X_j) \geq f(X_k) - \epsilon\} \geq \alpha_k \quad (4)$$

where ϵ is a small positive constant, then it was proven that

Theorem 1 *The probability of starting a local search tends to 0 with $k \rightarrow \infty$ if and only if*

$$\alpha_k k^{1/d} \rightarrow \infty \quad (5)$$

If α_k is chosen in a way similar to (2), that is letting

$$\alpha_k = r_{k;1;\sigma}, \quad (6)$$

then the probability of starting a local search will decrease to 0 for any choice of $\sigma > 0$; moreover the expected number of local searches started even if the algorithm is run forever is finite if $\sigma > 2^d/(2d)$. If no local search is allowed to start from within a prefixed distance from the boundary, then finiteness of the total number of local searches is achieved for all $\sigma > 1$. In any case, the global optimum will be found in finite time with probability 1.

4 COMPARISON BETWEEN MLSL AND TRL

A first numerical comparison between different Random Linkage implementations and MLSL with different values for σ is reported in (Locatelli and Schoen, 1995). There the results of running these algorithms over a set of test functions with dimension d ranging from 2 to 30 and with a number of local optima between 2 and 10^{30} were compared. The difference in standard statistics, like, *e.g.*, number of function evaluations, number of local searches, total sample size at stopping, were not very high. One of the difficulties in making a fair comparison was the very high computed variance in all of these statistics. Numerical experiments have thus been repeated using, for TRL and MLSL, the same sequence of randomly generated sample points: this way the only visible difference between the algorithms is caused by the different criterion for the choice of starting points for local searches. In table 1 we report the median statistics computed over 100 independent runs of TRL with threshold defined as in (6) and $\sigma = 1$ and of MLSL with $N = 50$ and $\sigma = 4$. Moreover in MLSL it was assumed that $\gamma = 1$ (*i.e.*, no reduction in the sample).

It is quite evident even from these numerical results that the performance of the two algorithm is very similar; it seems that, for difficult test functions, TRL is quite superior to MLSL. CPU times were not recorded, mainly because they strongly depend upon the implementation details and, in particular, the data structure used to store sampled points and to compute nearest neighbor distances. It should however be quite evident that, being the number of local searches quite comparable in the two algorithms, the main difference in performance should be caused by the number of nearest neighbor distance computed; a very naive implementation of both MLSL and TRL, using the simplest data structure, confirmed this analysis: for example, 100 runs of TRL on function “pen. shubert

function	method	funct.ev.	grad.ev.	loc. srch.	sample size
quartic $d = 2$	MLSL	73	22	1	50
	TRL	34	31.5	1	1
6 hump $d = 2$	MLSL	68	17	1	50
	TRL	33	31	1	1
shubert $d = 2$	MLSL	102	34	2	50
	TRL	175	110	6	48
pen.shubert 0.5 $d = 2$	MLSL	2765	623.5	35	1975
	TRL	2174	505	25.5	1589.5
pen.shubert 1. $d = 2$	MLSL	2017.5	433	24	1625
	TRL	1725.5	404	20.5	1209
picc. 1 $d = 2$	MLSL	69.5	18.5	1	50
	TRL	63	49.5	3	4
picc. 1 $d = 3$	MLSL	87.5	36.5	1	50
	TRL	93.5	84.5	3	4
picc. 1 $d = 4$	MLSL	107.5	56	1	50
	TRL	117.5	109	3	4
picc. 2 $d = 5$	MLSL	361.5	308	4	50
	TRL	435	409	6	11.5
picc. 2 $d = 8$	MLSL	908.5	855	6	50
	TRL	1126	1095.5	8	18.5
picc. 2 $d = 10$	MLSL	1241.5	1189.5	7	50
	TRL	1420	1395	7	15
picc. 2 $d = 20$	MLSL	4149	4096	6	50
	TRL	3924.5	3904.5	6	13
picc. 2 $d = 30$	MLSL	7005	6952.5	6	50
	TRL	8844	8810.5	8	18

Table 1 Comparison between MLSL and TRL

0.5" (the most difficult of the test functions) required roughly 24' on a SUN Sparc Station, while MLSL on the same test required 7 hours and 20'.

5 THEORETICAL ANALYSIS OF MLSL AND TRL

One of the strongest features of MLSL is that, thanks to the fact that the starting points for local searches may be found also in previously sampled points, the overall process consists in building chains of sampled points, with better and better function value, up to a point from which a local search is started; this way the algorithm can adapt very well the chain of sampled points to the shape of the region of attraction of local (and global) optima. This mechanism is absent in TRL, so it is quite natural to ask whether TRL makes sensible choices; the theoretical analysis in this section is devoted to prove that, with high probability, MLSL and TRL make the same choices, that is, they both decide to start local searches from the same points.

For the comparison we assume that in MLSL the choice $N = 1$ has been made. The analysis of the general case is actually being investigated and will be published elsewhere.

Let us define $M_k(h)$ (resp. T_k) to be the events that MLSL (TRL) started a local search from the k -th sampled point X_k ; the index h in $M_k(h)$ indicates that the local search starting from X_k was decided at iteration $h \geq k$ by MLSL.

We start with a straightforward observation: if the same threshold is used in MLSL and TRL, then

$$P(M_k(k) \mid T_k) = 1. \quad (7)$$

For what concerns the reverse, *i.e.* the probability that TRL starts a local search from X_k given that *at some iteration* MLSL will decide the same, the situation is a little bit more complex. However the following result is valid, which states that, when the threshold parameter used in MLSL is strictly greater than that in TRL (like in our experiments, where $\sigma = 4$ was chosen for MLSL and $\sigma = 1$ for TRL), then TRL will surely start a local search if MLSL will start a local search from the same point *not too late*. It should be recalled that one of the main distinguishing features of MLSL versus TRL is the fact that MLSL might decide to start a local search from X_k at a later iteration $h \geq k$, when the threshold for acceptance/rejection has been decreased enough. The next theorem says that, if the decision of starting a local search from X_k with MLSL does not occur "too late", then surely also TRL will start a local search as soon as it will sample X_k , *i.e.* at iteration k . This is, in the authors' opinion, a strong property, which enables to use TRL as a much simpler and computationally efficient implementation of MLSL.

Theorem 2 *Let the threshold parameter for TRL be $\sigma > 0$ and let the analogous parameter for MLSL be $\beta\sigma$ with $\beta > 1$. Then, there exists $\bar{k}$ such that, for all $k \geq \bar{k}$,*

$$P(T_k \mid \bigcup_{h=k}^{\beta k} M_k(h)) = 1 \quad (8)$$

Proof. As f is continuous and Ω is compact, f is uniformly continuous; then it is easy to see that, thanks to the relaxation (4), there exists $\bar{k}$ such that, for all $k \geq \bar{k}$, a local search is started in TRL if and only if

$$\min\{\|X_k - X_j\|, j = 1, \dots, k-1\} \geq \alpha_k \quad (9)$$

and similarly for MLSL (for a proof of this statement, see the cited paper (Locatelli and Schoen, 1995), lemma 3.2). This means that, at least for sufficiently large k , the dependance of the acceptance criterion on f can be neglected and all of the analysis can be carried out with explicit reference only to the relative positions of the sampled points.

If we denote by $\alpha_k = r_{k;1;\sigma}$ the threshold used in TRL, then the threshold for MLSL is $\beta^{1/d}\alpha_k$. The event that a local search is started from X_k in MLSL at iteration h with $k \leq h \leq \beta k$ is equivalent to the event

$$\min\{\|X_j - X_k\|, j = 1, \dots, h, j \neq k\} \geq \beta^{1/d}\alpha_h \quad (10)$$

We have also

$$\min\{\|X_j - X_k\|, j = 1, \dots, k, j \neq k\} \geq \min\{\|X_j - X_k\|, j = 1, \dots, h, j \neq k\}$$

so that a local search in TRL will surely be started from X_k if

$$\beta^{1/d}\alpha_h \geq \alpha_k$$

(notice that $k \leq h$). But, recalling the definition of the threshold, we have that $\beta^{1/d}\alpha_h \geq \alpha_k$ if and only if

$$(\beta \frac{\log h}{h})^{1/d} \geq (\frac{\log k}{k})^{1/d}$$

and, using the fact that $k \leq h \leq \beta k$, we obtain

$$\beta \frac{\log h}{h} \geq \beta \frac{\log k}{\beta k} = \frac{\log k}{k}$$

which completes the proof. $\square$

6 CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

After the preceding theorem, we observe that, while in both MLSL and TRL the probability of starting a local search decreases to zero, as the iterations proceed it is more and more likely that both algorithms will perform very similar decisions regarding the points from which to start a local search. Obviously, being the analysis based on some sort of asymptotic result (being in particular strongly dependent upon the consequence of the uniform continuity of f), MLSL and TRL might significantly differ in the first iterations. It seems that an accurate analysis of this “transient” phase of both algorithms is very difficult and no deep result is expected, unless very strong assumptions on f are assumed. On the other end, it seems plausible that a more complete characterization of the relative behaviour of MLSL and TRL can be effectively obtained, allowing to state results more general than that of the last theorem and comprising the event that MLSL starts a local search at any iteration later than k (and also later than βk). In this case it seems plausible that the probability of starting a local search from TRL will be strictly positive (although not necessarily equal to 1), thus confirming our impression on the similarity of the two

algorithms. The details of this more elaborate analysis are a subject of current research and will appear elsewhere.

REFERENCES

- Locatelli, M. and Schoen, F. (1995) Random Linkage: a family of acceptance/rejection algorithms for global optimisation. Submitted. (Available for anonymous ftp at <ftp-dsi.ing.unifi.it>, directory <pub/schoen/papers/randlink>).
- Pardalos, P.M. and Vavasis, S.A. (1991) Quadratic programming with one negative eigenvalue is NP-Hard. *Journal of Global Optimization*, **1**, 15–22.
- Rinnooy Kan, A.H.G. and Timmer, G.T. (1987a) Stochastic global optimization methods. Part I: Clustering methods. *Mathematical Programming*, **39**, 27–56.
- Rinnooy Kan, A.H.G. and Timmer, G.T. (1987b) Stochastic global optimization methods. Part II: Multi Level methods. *Mathematical Programming*, **39**, 57–78.

ACKNOWLEDGEMENT

This research has been partially supported by MURST research project “Metodi di Ottimizzazione nella Ricerca Operativa”.