Journal of Global Optimization **5:** 349–358, 1994.

Topographical Multilevel Single Linkage

M. M. ALI and C. STOREY
Department of Mathematical Sciences, Loughborough University of Technology, Leics, LE11, 3TU, England

(Received: 31 March 1993; accepted: 10 June 1994)

Abstract. An iterative topographical Multilevel Single Linkage (TMSL) method has been introduced. The approach uses topographical information on the objective function, in particular the g-nearest-neighbour graph. The algorithm uses evenly distributed points from a Halton sequence of uniform limiting density. We discuss the implementation of the algorithm and compare its performance with other well-known algorithms. The new algorithm performs much better (in some cases several times) than the Multilevel Single Linkage method in terms of number of function evaluations but is not quite so competitive with respect to CPU time.

Key words: Global optimization, multilevel single linkage, topographs, graph minima.

1. Introduction

The global minimization problem for a function $f : \Omega \subset \mathrm{R}^n \to \mathrm{R}$ is to find x^* such that

$$f(x^*) \leq f(x), \quad \forall x \in \Omega. \tag{1}$$

It is also assumed that the problem is essentially unconstrained, that is, the global minimum f^* of f is attained in the interior of Ω.

In recent years, a number of deterministic and stochastic algorithms have been proposed (Törn and Žilinskas, 1989; Floudas and Pardalos, 1992; Horst and Tuy, 1990; Ratschek and Rokne, 1988) for solving (1). Deterministic algorithms are generally concerned with searching the whole region of feasible points of the objective function f and give a guarantee of successfully finding the global minimum only under highly restrictive conditions on f (for example, Lipschitz continuity with known Lipschitz constant). On the other hand, stochastic methods involve random sampling or a combination of random sampling and local search. They can be applied in less restrictive situations and with greater reliability. A probabilistic convergence guarantee is the most attractive and salient feature of stochastic methods. One type of stochastic approach is Simulated Annealing (SA). SA type algorithms for continuous global optimization have been proposed by a number of authors. For example, Vanderbilt and Louie (1984), Bohachevsky *et al.* (1986) and Aluffi-Pentini *et al.* (1985). More recently, Dekkers and Aarts (1991) derived a continuous SA algorithm which is theoretically similar to discrete SA. Among the stochastic methods various superior clustering variants of Multistart (MS) designed by Rinnooy Kan and Timmer (1987, 1987a) are well known. The aim of these clustering algorithms is to apply local search more efficiently, that is to apply local search

only once in every region of attraction. Among them the Multilevel Single Linkage (MSL) method is known to be the most successful adaptation (Timmer, 1984). Recently, however, Törn and Viitanen (1992) have developed a new Topographical Algorithm (TA). In their new algorithm they use topographical information on the objective function in identifying basins of local minima from the centre (graph minima) of each of which a local search is started. The TA algorithm is non-iterative and based on exploration of the search space. The graph minima are constructed by looking at the function values of the g-nearest neighbour points for each point of a sample of size N. The aim of TA is to construct a topographical graph and then start minimization from just one point in each identified basin. But for a general purpose algorithm a fixed value of g would be too restrictive to represent the appropriate number of local minima for any function. Moreover, it would be increasingly difficult to construct the exact graph minima of functions, especially when the number of dimensions of the functions increases. In principle, therefore, both MSL and TA cause errors of the following nature.

Type I Error, Local search will be repeated in some region of attraction.

Type II Error, Local search will not start in some region of attraction even if a sample point has been located in that region of attraction.

Rinnooy Kan and Timmer (1987) argued that in MSL the above two types of error would not occur after a sufficiently large number of iterations. But clearly continuing the search for too large a number of iterations is wasteful. Furthermore, in MSL extended samples are considered and the resulting overheads could also rise to a prohibitive level. The question therefore arises of whether these errors can be avoided in every iteration if we use topographical information on the underlying function in a sensible way.

We, therefore, propose a new algorithm that uses MSL together with topographical information on the objective function. This adaptation of MSL will guarantee that a local search will start at a point with a relatively low function value, thus ensuring that sample reduction is no longer necessary. The paper is organised in the following way. In Section 2 we introduce the proposed algorithm. In Section 3 we investigate the choice of user supplied parameters for the new algorithm and compare it numerically with MSL and also with some other leading algorithms. Finally in Section 4 we make some concluding remarks.

2. The New Algorithm

The MSL method is superior to other clustering variants of MS because of the fact that it uses function values in identifying clusters (Rinnooy Kan and Timmer 1987a). However, the mechanism of the algorithm is so simple that it turns out to be possible to avoid the clustering concept altogether (Rinnooy Kan and Timmer

1987a). In MSL the decision to start a local search in the k-th iteration depends only on the threshold

$$r_k = \pi^{-1/2} \left(\Gamma \left(1 + \frac{n}{2} \right) m(\Omega) \sigma \frac{\log\ kN}{kN} \right)^{1/n} \tag{2}$$

where σ is a constant, $m(\Omega)$ is the Lebesgue measure of Ω, N is the total number of sample points and Γ is the gamma function. This threshold is derived using asymptotic considerations. A point is taken as the starting point for local optimization if there is no other sample point, within the critical distance r_k, with lower function value. This check however, is carried out for all reduced sample points. The purpose of generating a comparatively large number of points and the sample reduction strategy in MSL is to make sure that:

A. The search region has been explored thoroughly so that points are drawn in every region of attraction.

B. A fraction of the points is discarded so that only points with relatively low function values are left for scrutiny.

We, however, propose using the selection of graph minima instead of sample reduction, together with evenly distributed sample points, to achieve conditions **A** and **B** above. Having found the reduced sample consisting of graph minima a local search is then carried out from each of a selected subset of the graph minima. No attempt is made to find the complete topograph of the function so the value of g is not so critical as it is in TA. Our experience shows however that a reasonably small value for g is generally to be recommended. We believe that this strategy is efficient because the size of the reduced sample depends on the choice of g and the particular function at hand as opposed to an empirical fraction γkN ($\gamma = 0.2$) as in MSL.

The basic principle of TA is to cover the search region with sample points as uniformly as possible. The greater the number of points, the greater will be the possibility of the graph representing all essential local minima. In our algorithm we, therefore, use a Halton sequence (Shaw, 1988) which is more evenly distributed than the pseudo-random numbers and has uniform limiting density. At the start of a new iteration we add all local minimizers found previously to the new set of sample points and then the construction of the graph minima takes place. In every successive iteration, therefore, local minima from previous iterations could become graph minima. In principle, we use previous sample points in an implicit way by representing them by the local minima they produced.

Because the Halton sequence is uniform asymptotically many of the theoretical results for MSL apply also to TMSL (Ali, 1994). We now give a stepwise description of TMSL on a typical iteration k with w different minima found previously.

TABLE I. Test functions

Symbol	Test Function	Number of local minima (m)	Dimension (n)
BR	Branin	3 (all same f^*)	2
GP	Goldstein and Price	4 (all different f^*)	2
S5	Shekel5	5 "	4
S7	Shekel7	7 "	4
S10	Shekel10	10 "	4
H3	Hartman3	4 "	3
H6	Hartman6	4 "	6

Step 1. Generate N sample points from the Halton sequence over the search region Ω. Compute f at each point. Let M be the set of sample points plus the w minimizers found previously.

Step 2. Construct a topographical graph and find graph minima for these M points.

Step 3. A graph minimum is a 'start' point for local search if it is greater than the critical distance r_k from any point with smaller f value and if it is not a previously obtained local minimizer.

Step 4. Carry out a local search from each such point. If new local minima are found then update w accordingly.

Step 5. Is stopping condition (Boender and Rinnooy Kan, 1987) $\frac{w(kN-1)}{kN-w-2} \leq w + \frac{1}{2}$ satisfied? Yes, stop. No, go to Step 1.

3. Numerical Results

In this section we compare our numerical results with MSL and other recent algorithms. The computing has been carried out on the HP9000/870 computer at Loughborough University of Technology using the programming language PASCAL. The test functions (see Table I) have been taken from Dixon and Szegö (1978), a set of commonly used functions in global optimization. We use limited memory BFGS, from the NAG Library routine (version E04DGF) for local search with a tolerance on the gradient of 10^{-10}.

3.1. Choice of Parameters in TMSL

The main user supplied parameters for TMSL are N, the sample size, σ in r_k and g the number of nearest neighbour points in the topographs. We carried out an extensive series of tests to see the effects of varying these parameters in the algorithm. We considered; $N = 10(n-2)$, $10(n-1)$, $10n$, $10(n+1)$, $15n$ and

TABLE II. Results for TMSL on S5

N	σ	CPU	LS	LM	FE	g
50	2	0.45	6	4	243	2
	4	0.40	2	2	117	2
	2	0.52	6	3	238	3
	4	0.52	2	2	117	3
	2	0.69	3	3	146	4
	4	0.65	2	2	117	4
	2	0.66	2	2	118	5
	4	0.70	1	1	90	5
	2	0.74	2	2	119	6
	4	0.74	1	1	90	6
	2	0.84	2	2	119	7
	4	0.84	1	1	90	7
	2	0.90	1	1	90	8
	4	0.90	1	1	90	8
	2	0.90	1	1	90	9
	4	0.90	1	1	90	9
	2	0.90	1	1	90	10
	4	0.90	1	1	90	10

$15(n+1)$ (where these were distinct), $\sigma = 2, 4$ (Rinnooy Kan and Timmer, 1987, 1987a), $g = 2, 3, ..., N-1$. These parameter values were used for each of the 7 test functions described in Table I. The general nature of the results was quite similar for each test function and so the results for only one representative function for a specific value of N are shown in Table II (the full set of results can be found in Ali, 1994). We use following notation: LS is the number of local searches performed, LM is the number of local minima found, FE is the number of function evaluations and CPU is the cpu time.

It is clear from Table II that $\sigma = 4$ is much better than $\sigma = 2$ for smaller values of g but this effect falls off as g increases. The global optimum was reached for almost all values of g (Ali, 1994) but as g increases towards $N-1$ the number of function evaluations decreased to a smallest value at which $LS = LM = 1$ (or possibly 2 for some test functions) and remained static subsequently; however, as FE decreased the cpu time increased.

Table III summarizes the interaction between g and N again for a typical function S5. The results given here are the values of g for which the smallest number of function evaluations mentioned above was achieved. It appears that $N = 10n = 40$ is the best sample size for this function a result which was generally true for all the 7 test functions.

It was interesting therefore to see how the best value for g varied amongst the 7 test functions. Table IV summarizes this effect for $N = 10n$.

TABLE III. Summary of results showing effect of g for different N on S5

N	σ	CPU	LS	LM	FE	g
75	2	3.00	2	1	127*	8
	4	3.01	2	1	98*	8
60	2	1.96	1	1	100	11
	4	1.98	1	1	100	11
50	2	0.90	1	1	90	8
	4	0.90	1	1	90	8
40	2	0.48	1	1	77	8
	4	0.48	1	1	77	8
30	2	0.22	2	2	100	7
	4	0.22	2	2	100	7
20	2	0.11	2	2	90	7
	4	0.11	2	2	90	7

* indicates only a local minimum was obtained

TABLE IV. Best g for $N = 10n$

	g	n	m
BR	17	2	3
GP	5	2	4
S5	8	4	5
S7	38*	4	7
S10	22	4	10
H3	12	3	4
H6	54	6	4

* This was the largest value for g to produce the global minimum

Clearly there is no obvious connection between the best value of g and the dimension, n, or the number of local minima, m, but the nature of the functions is important since for some functions to get the best value of g it has to be increased until it is close to $N - 1$. When $g = N - 1$, of course, the only graph minima are the point(s) with smallest function value(s).

In summary, the information on the use of user supplied parameters that was obtained from the numerical test function results was as follows. $N = 10n$ was a reasonable sample size. $\sigma = 4$ was the best value to use for small values of g but as g increased there was little difference between $\sigma = 2$ and $\sigma = 4$. All values of g eventually produced global optima (except for values of g very close to $N - 1$ for S7) and the number of function evaluations required decreased as g increased but at

TABLE V. Comparison of MSL and TMSL

	FE	CPU
MSL	3163	2.77
MSLH	2535	1.62
MSLG	799	6.18
TMSL	674	5.38

TABLE VI. Bifunctional catalytic reactor

	N	FE	CPU	f^*
MSL	150	11252	28676	10.094 (9.90, 9.86)
TMSL	150	6013	14361	10.094

the expense of extra cpu time. Choice of g would therefore seem to be dependent on the cost of computing the function values. Extrapolation of these suggestions must clearly be treated with caution since the test functions used are all well behaved mathematical functions with a relatively small number of local minima.

3.2. Comparison of MSL and TMSL

The major differences between MSL and TMSL lie in the use of the Halton sequence instead of a pseudo-random sequence and the use of the g-topograph rather than sample reduction. To judge the relative importance of these two changes we compared MSL with TMSL and also with MSLH, which is MSL with Halton instead of random sampling and with MSLG, which is MSL with the g-topograph replacing sample reduction. We used the same test functions as previously and have totalled the number of function evaluations and cpu time for all seven functions (the detailed results can be found in Ali, 1994) in Table V.

Clearly the dominating factor is the introduction of the g-topograph and the reduction in function evaluations by a factor of about 5 for TMSL is offset by it requiring about twice the cpu time. In the comparisons we tried to be as 'fair' as we possibly could to all four methods but fully realize the difficulties involved, particularly with respect to the stopping condition used (Ali, 1994).

In addition to the test functions we also used a number of 'real-life' problems with which to compare the methods. These included two chemical reactor problems, a problem involving the statistics of pig-liver behaviour, an atomic, many-body potential problem and a vehicle suspension problem. We show preliminary results for one of these problems in Table VI. This problem concerned a bifunctional catalytic reactor and has 25 local minima with a global minimum of value 10.09. (For full details see Ali, 1994.)

TABLE VII. Listing of different methods

Method	Name	Reference
A	Multistart (MS)	Rinnooy Kan and Timmer (1984)
B	Controlled Random Search (CRS1)	Price (1978)
C	Density Clustering	Törn (1978)
D	Clustering with distribution function	De Biase and Frontini (1978)
E	Multilevel Single Linkage (MSL)	Rinnooy Kan and Timmer (1987a)
F	Simulated Annealing (SA)	Dekkers and Aarts (1991)
G	Modified Controlled Random Search (CRS4)	Ali and Storey (1995)
H	Modified Controlled Random Search (CRS5)	Ali and Storey (1995)
I	Aspiration based Simulated Annealing (ASA)	Ali (1994)
J	TMSL	This paper

TABLE VIII. Comparison of TMSL with 9 currently available methods

Method	GP	BR	S5	S7	S10	H3	H6	Total
A	4400	1600	6500	9300	11000	2500	6000	41300
B	2500	1800	3800	4900	4400	2400	7600	27400
C	2499	1558	3649	3606	3874	2584	3447	21217
D	378	597	620	788	1160	732	807	5082
E	307	206	576	334	1388	166	324	3301
F	563	505	365*	558	797	1459	4648	8895
G	436	279	1423	1238	1213	545	1581	6715
H	402	346	1866	1719	1709	343	1321	7706
I	834	135	524	524	524	451	558	3550
J	53	46	98	116	100	60	127	600

* Local minima found

The results for MSL are the average of four runs, 2 of which produced minima of 9.90 and 9.86 and the other two the correct global minimum. For this problem TMSL exhibits superiority in both FE and cpu time. Detailed analysis of the results showed that the MSL method had to find more local minima than TMSL. Results from the other practical problems confirmed the competiveness of TMSL (Ali, 1994), especially in terms of number of function evaluations.

We also compared TMSL with the currently available algorithms given in Table VII using the number of function evaluations as a basis for comparison and the results are shown in Table VIII where the data other than that for MSL and TMSL has been taken from the references listed in Table VII. In Table VIII, the data for TMSL has been taken as $N = 10n$, $\sigma = 4$ and $g = 7$. The data for MSL has been taken as $N = 100$, $\gamma = 0.2$ and $\sigma = 2$.

4. Conclusion

We have developed a new global optimization algorithm based on MSL and topographical global optimization which seems to be robust and competitive with MSL. Research is continuing into the choice of user supplied parameters for the new method, on difficult practical problems, and into the effect of different stopping conditions.

Acknowledgement

The authors thankfully acknowledge useful suggestions from A. Törn of Abo Akademi, Finland.

References

Ali, M. M. (1994), *Some Modified Stochastic Global Optimization Algorithms with Applications*, Ph.D. Thesis, Loughborough University of Technology, Loughborough, Leicestershire, England.

Ali, M. M. and Storey, C. (1995), Modified Controlled Random Search Algorithms, to appear in *International Journal of Computer Mathematics*, 54, No. 3/4.

Aluffi-Pentini, F., Parisi, V., and Zirilli, F. (1985), Global Optimization and Stochastic Differential Equations, *Journal of Optimization Theory and Applications* **47**, 1–16.

Boender, C. G. E. and Rinnooy Kan, A. H. G. (1987), Bayesian Stopping Rules for Multistart Global Optimization Methods, *Mathematical Programming* **37**, 59–80.

Bohachevsky, M. E., Johson, M. E., and Stein, M. L. (1986), Generalized Simulated Annealing for Function Optimization, *Technometrics* **28**, 209–217.

Dekkers, A. and Aarts, E. (1991), Global Optimization and Simulated Annealing, *Mathematical Programming* **50**, 367–393.

De Biase, L. and Frontini, F. (1978), A Stochastic Method for Global Optimization: Its Structure and Numerical Performance, in *Towards Global Optimization 2*, Dixon, L. C. W. and Szegö, G. P. (eds.), North-Holland, Amsterdam, Holland, 85–102.

Dixon, L. C. W. and Szegö, G. P. (1978), *Towards Global Optimization 2*, North-Holland, Amsterdam, Holland.

Floudas, A. and Pardalos, M. (eds.) (1992), *Recent Advances in Global Optimization*; Princeton University Press, U.S.A.

Horst, R. and Tuy, H. (1990), *Global Optimization (Deterministic Approaches)*, Springer-Verlag, Berlin.

Price, W. L. (1978), A Controlled Random Search Procedure for Global Optimization, in *Towards Global Optimization 2*, Dixon, L. C. W. and Szegö, G. P. (eds.), North-Holland, Amsterdam, Holland, 71–84.

Ratschek, H. and Rokne, J. (1988), *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester.

Rinnooy Kan, A. H. G. and Timmer, G. T. (1984), Stochastic Methods for Global Optimization, *American Journal of Mathematical and Management Sciences* **4**, 7–40.

Rinnooy Kan, A. H. G. and Timmer, G. T. (1987), Stochastic Global Optimization Methods; Part I: Clustering Methods, *Mathematical Programming* **39**, 27–56.

Rinnooy Kan, A. H. G. and Timmer, G. T. (1987a), Stochastic Global Optimization Methods; Part II: Multilevel Methods, *Mathematical Programming* **39**, 57–78.

Shaw, J. E. H. (1988), A Quasirandom Approach to Integration in Bayesian Statistics, *The Annals of Statistics* **16** (2), 895–914.

Timmer, G. T. (1984), Ph.D. Dissertation, Econometric Institute, Erasmus University, Rotterdam, Holland.

Törn, A. and Žilinskas, A. (1989), *Global Optimization*, Springer-Verlag, Berlin.

Törn, A. and Viitanen, S. (1992), Topographical Global Optimization, in *Recent Advances in Global Optimization*, C. A. Floudas and P. M. Pardalos (eds.), Princeton University Press, Princeton U.S.A., 384–398.

Törn, A. (1978), A Search Clustering Approach to Global Optimization, in *Towards Global Optimization 2*, Dixon, L. C. W. and Szegö, G. P. (eds.), North-Holland, Amsterdam, Holland, 49–62.

Vanderbilt, D. and Louie, S. G. (1984), A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables, *Journal of Computational Physics* **56**, 259–271.