Noname manuscript No.
(will be inserted by the editor)

The GLOBAL Optimization Method Revisited

Tibor Csendes[1], László Pál[2], J. Oscar H. Sendín[3], Julio R. Banga[3] *

[1] University of Szeged, Institute of Informatics, Szeged, Hungary
[2] Sapientia University, Faculty of Business and Humanities, Miercurea-Ciuc, Romania
[3] Instituto de Investigaciones Marinas, IIM-CSIC Vigo, Spain

Received: 01.05.2007 / Revised version: date

Abstract The multistart clustering global optimization method called GLOBAL has been introduced in the 1980s for bound constrained global optimization problems with black-box type objective function. Since then the technological environment has been changed much. The present paper describes shortly the revisions and updates made on the involved algorithms to utilize the novel technologies, and to improve its reliability. We discuss in detail the results of the numerical comparison with the old version and with C-GRASP, a continuous version of the GRASP method. According to these findings, the new version of GLOBAL is both more reliable and more efficient than the old one, and it compares favorably with C-GRASP too.

Key words Global optimization, direct methods, clustering, numerical tests.

1 Introduction

Global optimization has a wide range of applications. It is used more and more both in science and for the solution of real life problems. This phenomenon is related to the stronger and stronger computational capacities and to the complex, highly nonlinear nature of the investigated models. For the first set of problems let us mention as examples some of our recent publications in which global optimization was used to cope with hard mathematical problems arising in the field of qualitative analysis of dynamical

Send offprint requests to: Tibor Csendes
* *Email address:* csendes@inf.u-szeged.hu

systems [4,5,8,9] and discrete geometry, for optimal packing of circles in the square [16,23]. The method to be discussed in the present paper has also been applied for theoretical chemical problems [1], for the evaluation of bounding methods [25], and for optimization in abstract spaces, on Stiefel manifolds [2].

The real life or industrial applications are again too many to be covered here. We just mention some from the fields of bioprocess analysis [3], climate control [17], and integrated process design for wastewater treatment plants [18].

The global optimization problem for the solution of which we analyze a stochastic algorithm is

$$\min f(x) \tag{1}$$

$$x \in X, X = \{a_i \leq x_i \leq b_i,\ i = 1, 2, \ldots, n\},$$

where $f : R^n \rightarrow R$ is a real valued function, X is the set of feasibility, an n-dimensional interval with vectors of lower and upper bounds of a and b, respectively. In general we assume that the objective function is twice continuously differentiable, although it is not necessary for the global optimization framework procedure, and with a proper local search algorithm also nondifferentiable problems can be solved.

On the other hand one of the local search algorithms applies numerical derivatives calculated inside of it, so the user must not include subroutines for the calculation of derivatives, only that for the objective function itself. In this sense GLOBAL is a direct search method. According to our experience, it is a quite important feature, that allows a wide range of applications. With advanced quasi-Newton type local search methods a similar good convergence speed can be achieved as with local search methods applying first and second order derivatives. Since our algorithm uses only objective function evaluations on places determined by the algorithm, the black-box type objective function can also be defined by only a procedure, without having an explicit expression for it. On the other hand – if necessary – sophisticated automatic differentiation tools enable us to calculate the necessary derivatives without much human interaction (for a successful application see e.g. [24]).

We also have a random walk type robust local search method built in: UNIRANDI [15] that can be used when the problem structure does not allow us to utilize the locally quadratic behavior as it is the case for the quasi-Newton technique.

Although the bound constrained problem definition (1) does not allow explicit nonlinear constraints, the use of penalty functions enables our method to cope also with constrained problems. According to our experiences, this approach can be successful in most of the cases encountered (see e.g. the difficult one addressed in [9]). Still we are also working on the extension of the algorithm to constrained problems [22].

Algorithm 1 A concise description of the GLOBAL algorithm

Step 1: Draw N points with uniform distribution in X, and add them to the current cumulative sample C. Construct the transformed sample T by taking the γ percent of the points in C with the lowest function value.
Step 2: Apply the clustering procedure to T one by one. If all points of T can be assigned to an existing cluster, go to Step 4
Step 3: Apply the local search procedure to the points in T not yet clustered. Repeat Step 3 until every point has been assigned to a cluster.
Step 4: If a new local minimizer has been found, go to Step 1.
Step 5: Determine the smallest local minimum value found, and stop.

2 The algorithm and the improvements made

The algorithm to be studied originates from the paper of Boender et al. [6]. The method was later investigated theoretically in [20, 21]. It is a multistart type stochastic method that collects iteratively information about the regions of attractions [14] of the best local minimizer points. The main algorithm steps are summarized in Algorithm 1.

The most important changes made on the original algorithm are:

- the single linkage clustering was selected after a respective testing,
- the clustering distance is not based on the Hessian (thus the latter should not be computed),
- the gradient criterion for forming clusters has been found to be less effective and thus left out,
- no steepest descent step is used to transform the original sample, and finally
- the less informative confidence intervals are not calculated for the global minimum value,
- a scaling of the original problem is applied to ensure better numerical stability.

The interested reader can consult beyond the original articles also the earlier description of GLOBAL in [7], where the numerical performance of that algorithm variant was discussed as well. Among other computational tests made on global optimization software, [19] evaluated GLOBAL as a favorable one for black-box type problems.

The program is freely available for academic and nonprofit purposes at `www.inf.u-szeged.hu/~csendes/regist.php` (after registration and limited for low to moderate dimensional problems). After further testing and refinement, the present, Matlab based version will also be available there soon.

The most important changes made on GLOBAL are:

- it is now coded in Matlab, utilizing the vectorization for better efficiency,

- we use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) local search algorithm instead of the earlier Davidon-Fletcher-Powell (DFP) method [11],
- better uniform and normal distribution random number generators are applied,
- some improvements were made in the uniform distribution direction selection procedure of the UNIRANDI local search method. The new code provide better statistical characteristic while it needs less computation. The present implementation of UNIRANDI works now without dimension related problems.

We have utilized the advantages of Matlab to obtain an efficient code. The vectorization of Matlab, a special syntax, makes it easy to obtain such a machine code that can feed the pipeline of the CPU in such a way that long vector calculations can achieve a closely full use of the pipeline processors. This is otherwise hard to be achieved by high level algorithmic languages. The last mentioned improvement in the list above had enhanced the efficiency of GLOBAL substantially in terms of CPU time used. Now we have the capability to solve larger problems than before with similar reliability.

3 Numerical test results

Our numerical experiments were made on a PC with 3.0 GHz P4 processor and 1 Gb memory. We used the standard time unit (1 000 evaluations of the Shekel-5 function at $x^T = (4.0, 4.0, 4.0, 4.0)^T$) to measure the computation time comparably. GLOBAL has three parameters to set: the number of sample points (abbreviated in the legends of the tables as sample size, values between 20 and 100, 000), the number of best points selected (between 1 and 20), and the stopping criterion parameter for the local search (precision, between 4 and 8 digits).

We have completed two numerical tests: the first aimed to show the efficiency and reliability changes compared to the old version, based on the published results in [7], and one to compare the new method to C-GRASP, a greedy adaptive search technique [10] modified to solve continuous global optimization problems published recently [12].

3.1 Comparison with the old GLOBAL version

We used the standard test functions applied for the old version. For each problem we made 100 independent runs (earlier it was just 10), and we recorded the average number of function evaluations and the average CPU time necessary, measured in the standard time unit. The parameters of the procedures were set so that the algorithm was able to find the global optimum each time. In this sense the present algorithm was set to achieve about one order of magnitude better reliability. Here we understand reliability as

Table 1 The number of function evaluations for the old and new versions of GLOBAL with the quasi-Newton local search routines on the standard test problems.

Problem		old	new				
	dim.	aver.	aver.	min.	max.	median	st. dev.
Shekel-5	4	990	1 090	540	2 235	1 067.5	352.52
Shekel-7	4	1 767	1 718	895	3 160	1 655.0	449.16
Shekel-10	4	2 396	2 378	1 145	4 630	2 317.5	697.19
Hartman	3	216	196	86	385	171.5	61.55
Hartman	6	1 446	703	132	1 647	702.5	263.32
Goldstein-Price	2	436	277	130	569	270.5	88.59
Branin	2	330	77	61	115	73.0	11.84
SHCB	2	233	107	58	224	103.5	35.67
Rosenbrock	2	410	125	52	334	109.0	62.27

Table 2 The number of function evaluations for the old and new versions of GLOBAL with UNIRANDI as a local search method on the standard test problems.

Problem		old	new				
	dim.	aver.	aver.	min.	max.	median	st. dev.
Shekel-5	4	1 083	1 450	751	2 718	1 374.0	413.88
Shekel-7	4	1 974	2 527	1 254	4 834	2 530.0	695.24
Shekel-10	4	2 689	3 429	1 636	5 795	3 277.5	881.97
Hartman	3	697	1 449	363	3 933	1 330.0	671.15
Hartman	6	2 610	2 614	274	9 004	2 132.5	1 676.16
Goldstein-Price	2	386	446	126	1 118	413.0	211.94
Branin	2	464	172	92	353	149.0	64.46
SHCB	2	267	176	95	328	175.0	66.85
Rosenbrock	2	1 524	1 081	277	2 687	902.0	516.38

the ability of an algorithm to find a global minimizer point a given number of times out of a preset number of independent runs. While in the case of the old GLOBAL only 10 successful runs were required out of 10 independent runs, now we tuned our algorithm to meet this criterion for 100 successful runs out of 100, which means the new one is certainly much more reliable. The good algorithm parameter values found (given in the tables) are telling also on how the procedure can achieve reliability. The criterion to accept an approximate solution as a successful estimate of the global minimizer was the same as in [7]: if the relative distance between them was less than the threshold value 10^{-2}.

Table 1 provides the average number of function evaluations for the old and the new versions using the quasi-Newton type local search routines on the standard test problems. In case of the new implementation also the min-

Table 3 The average CPU times measured in standard time units for the old and new versions of GLOBAL with the quasi-Newton and UNIRANDI local search methods on the standard test problems. The algorithm parameters are the sample size, the number of selected points, and the precision required from the local search, respectively.

Problem	quasi-Newton					UNIRANDI				
	old	new				old	new			
	av.	av.	parameters			av.	av.	parameters		
Shekel-5	3.0	5.14	100	10	6	3.5	3.87	100	12	6
Shekel-7	4.9	7.89	200	15	6	6.0	6.64	300	15	6
Shekel-10	7.0	10.46	250	15	6	8.8	9.14	400	15	6
Hartman-3	1.2	1.19	15	2	7	1.9	3.86	15	3	7
Hartman-6	4.2	3.39	10	3	6	14.2	7.68	20	3	6
Goldstein-Price	1.3	1.32	50	4	6	1.5	0.81	30	4	7
Branin	1.4	0.40	20	1	6	1.6	0.31	20	1	6
SHCB	1.2	0.61	20	2	6	1.3	0.33	20	2	6
Rosenbrock	1.0	0.83	2	1	7	1.5	1.93	2	1	7

imal, the maximal number of function evaluations are given together with the median and the standard deviation. According to the results, for most of the test functions the new algorithm was substantially better than the old one. It is so mostly just due to the more sophisticated local search technique used (BFGS instead of an old implementation of DFP). The reason for it can be seen on Table 2: with basically the same local search procedure, the improvement in the average number of function evaluations appeared for only slightly less than half of the test functions. Yet even the figures for the new method with UNIRANDI are nice since the reliability improved much.

The algorithm parameters (shown in Table 3) cannot be compared in detail, since they were not collected and reported for the old method, still they are close to each other (as far as it can be judged). The fitting parameter setting is straightforward for the sample size and the number of selected points: the larger the sample size the more reliable the algorithm. The smaller the selected / sample size ratio, the smaller local minima we can find (i.e. the larger local minima will not be identified – the more efficient the search). It was one of the important findings of the papers by Boender et al. [6,20,21] that good reliability can be achieved efficiently while not locating all the local minimizer points. The finding of a local minimizer point with much worse function value than the global optimum does not contribute much for the identification of a global minimizer. This is what makes it possible to save computational efforts while keeping reliability in the determination of global minimizer points.

The set precision of the local search method has a different role: in general it should be close to the required relative precision of the estimated

global minimum value. However, according to our experience, for both local search methods it is worth to set the precision value higher to avoid cases when a local minimum is not recognized since the results of local searches could not be identified as the same – although the same region of attraction was found. This phenomenon is reflected in Table 3 in the larger precision values for those test problems for which the objective function values do not change much close to the global minimizer point.

The average CPU times (measured in standard time units) necessary for the solution of the test problems are comprised in Table 3. The figures follow more or less the differences seen for the number of function evaluations in the previous tables. The differences to the anticipated ratios are due to overhead of startup and output in part independent of the problems, the different programming environments, hardware and even because of the way the standard time units were measured (now actually the time for the 10^6 evaluation of the Shekel-5 function was divided by 1000).

The conclusion of the first set of tests completed is that on standard test problems the new implementation is at least as good or even better in terms of efficiency as the old one was, while the reliability of the solution has been increased substantially. Due to the better quasi-Newton local search method, the new version is much better for smooth problems even in terms of the necessary number of objective function evaluations.

3.2 Comparison with the C-GRASP method

As it was already mentioned, the C-GRASP method extends the greedy randomized adaptive search procedure of Feo and Resende [10] from the domain of discrete optimization to that of continuous global optimization [12]. It does not make use of derivative information, thus it is a direct search method. Regarding its control structure, it is a multistart local search technique too, such as GLOBAL. Since its local search part does not utilize the possible smoothness of the objective function, it is fair only to compare it to GLOBAL with UNIRANDI.

The two computers used for the testing were of similar performance, but our one was slightly slower and had much less memory (however the latter fact has not affected the comparison results). In this way the direct CPU times can only be used to compare with some care.

We applied our new implementation of GLOBAL to the same set of 14 global optimization test problems on which C-GRASP was run. The global minimum value f^* was known for all problems in the test set. Both methods were run until the objective function value f was significantly close to global optimum (i.e. till $|f^* - f| \leq 10^{-4}|f^*| + 10^{-6}$ became true). GLOBAL could also be stopped when no new local minimizer point was found in the last iteration cycle. The use of the known optimum value within an optimization algorithm is not typical, still it is realistic in several cases, as explained in [12], but also according to our experiments e.g. with circle packing problems [16, 23]. It is by no way necessary for GLOBAL.

Table 4 The number of function evaluations for C-GRASP and GLOBAL with the UNIRANDI local search routine. The algorithm parameters were for all test problems the same: the number of sample points was 400, the number of selected points was 15, and the required number of precise digits for the local search was 8.

Problem	C-GRASP	GLOBAL with UNIRANDI				
	average	average	min.	max.	median	st. dev.
Shekel-5	5 545 982	1 489	897	2 259	1 470.0	298.56
Shekel-7	4 052 800	1 684	860	2 914	1 625.0	338.05
Shekel-10	4 701 358	1 815	1 034	3 623	1 713.5	532.08
Hartman-3	20 743	3 608	1 767	5 844	3 585.0	828.73
Hartman-6	79 685	16 933	6 240	29 665	16 739.5	5 040.56
Goldstein-Price	29	923	559	1 558	874.0	229.22
Branin	59 857	1 023	621	1 678	1 032.5	211.07
Rosenbrock-2	1 158 350	6 274	2 894	10 378	6 091.0	1 402.33
Rosenbrock-5	6 205 503	374 685	168 601	628 507	360 241.0	89 941.28
Rosenbrock-10	20 282 529	1 908 469	806 288	2 418 556	2 043 155	477 478.51
Easom	89 630	1 604	532	2 664	1 610.5	416.63
Shubert	82 363	1 399	936	1 859	1 407.5	171.34
Zakharov-5	959	8 227	4 629	11 420	8 367.5	1702.57
Zakharov-10	3 607 653	47 288	34 549	53 995	47 850.0	3 767.57

For each problem, 100 independent runs of GLOBAL were completed with the same, fixed algorithm parameter set: the number of sample points was 400, the number of selected points was 15, and the required number of precise digits for the local search was 8. We recorded the percentage of runs that found a significantly close solution, the time necessary for such solutions and the number of function evaluations. The algorithm parameters of GLOBAL were set again in such a way that it was able to find a global minimizer point in each run. We listed the published results for C-GRASP [12] in the tables.

Table 4 contains the numbers of objective function evaluations necessary to solve the test problems in the above described sense. According to it GLOBAL was (with the exception of the Goldstein-Price and Zaharov-5 problems) always much more efficient than C-GRASP, sometimes even by orders of magnitude. The reason for that may be due to the different model in the background: while C-GRASP is prepared for any ugly behavior, GLOBAL assumes inherently that at least one global minimizer points has a non-negligible sized region of attraction. Still, both algorithms are capable to solve black box problems, i.e. without additional information on the problems beyond the objective function value (when GLOBAL is used with UNIRANDI).

Note that for GLOBAL the algorithm parameters should be such that all problems could be solved all times it was run. In contrast to that, it can

Table 5 CPU time required by C-GRASP and GLOBAL with the UNIRANDI local search routine in seconds.

Problem	dim.	C-GRASP	GLOBAL
Shekel-5	4	2.3316	0.1313
Shekel-7	4	2.3768	0.1461
Shekel-10	4	3.5172	0.1614
Hartman	3	0.0026	0.3208
Hartman	6	0.0140	1.7880
Goldstein-Price	2	0.0000	0.0516
Branin	2	0.0016	0.0580
Rosenbrock	2	0.0132	0.4117
Rosenbrock	5	1.7520	24.7559
Rosenbrock	10	11.4388	130.6813
Easom	2	0.0042	0.0916
Shubert	2	0.0078	0.0930
Zakharov	5	0.0000	0.5369
Zakharov	10	1.0346	3.1428

be surprising that some problems (e.g. Shekel-10) could be solved by the new GLOBAL with less function evaluations than what is stated in Table 1. This phenomenon is caused by the new stopping rule that can stop iteration earlier. It can also be seen that the higher dimensional versions of some test function could be solved by both techniques with correspondingly larger computational efforts.

The less comparable CPU time values are summarized in Table 5. These reflect on one hand the anticipated differences in the problem difficulty according to the number of function evaluations. On the other hand the CPU times for the Rosenbrock test functions are hard to explain: although they are proportional to the number of function evaluations for GLOBAL, the efficiency relation between the two methods found earlier cannot be recognized here.

4 Conclusion and future work

Summarizing the results of the paper we can conclude that the new version of GLOBAL utilizes the advantages offered by Matlab, and the algorithmic improvements increased the size of the problems that can be solved reliably with it. The reliability of the algorithm is now better while the efficiency is improved too. The careful comparison both with the old version and with C-GRASP is favorable for the new version of GLOBAL.

A careful testing work is still ahead to clear the role of those fine details of the algorithm that can be still improved to achieve a higher level of reliability and even more efficiency. In this direction the clustering dissimilarity measure, the gradient condition and the proper algorithm parameter selection should be strengthened further. Note that better test results are available for an improved C-GRASP version with other algorithm parameters for a larger test problem set at [13]. A comparison with it on the wider set of test problems belongs also to the future work to be done.

Acknowledgements The present work was supported by the grants OMFB E-25/2004, Aktion Österreich-Ungarn 60öu6, OTKA T 048377 and T 046822, Spanish Government MEC AGL2004-05206-C02-01/ALI, and Xunta de Galicia grant PGIDIT05PXIC40201PN.

References

1. Balogh, J., T. Csendes, and R.P. Stateva: Application of a stochastic method to the solution of the phase stability problem: cubic equations of state. Fluid Phase Equilibria **212**, 257-267 (2003)
2. Balogh, J., T. Csendes, and T. Rapcsák: Some Global Optimization Problems on Stiefel Manifolds. J. of Global Optimization **30**, 91-101 (2004)
3. Banga, J.R., C.G. Moles, and A.A. Alonso, *Global Optimization of Bioprocesses using Stochastic and Hybrid Methods.* In: C.A. Floudas and P.M. Pardalos (eds.): *Frontiers in Global Optimization*, pp. 45-70. Springer Berlin (2003)
4. Bánhelyi, B., T. Csendes, and B.M. Garay: Optimization and the Miranda approach in detecting horseshoe-type chaos by computer. Int. J. Bifurcation and Chaos, accepted for publication
5. Bánhelyi, B., T. Csendes, and B.M. Garay: A verified optimization technique to bound topological entropy rigorously. Proceedings of the SCAN-2006 Conference, IEEE, accepted for publication
6. Boender, C.G.E., A.H.G. Rinnooy Kan, G.T. Timmer, and L. Stougie, A stochastic method for global optimization. Mathematical Programming **22**, 125-140 (1982)
7. Csendes, T., Nonlinear parameter estimation by global optimization – efficiency and reliability. Acta Cybernetica **8**, 361-370 (1988)
8. Csendes, T., B. Bánhelyi, and L. Hatvani: Towards a computer-assisted proof for chaos in a forced damped pendulum equation. J. Computational and Applied Mathematics **199**, 378-383 (2007)
9. Csendes, T., B.M. Garay, and B. Bánhelyi: A verified optimization technique to locate chaotic regions of Hénon systems. J. of Global Optimization **35**, 145-160 (2006)
10. Feo, T.A. and M.G.C. Resende: Greedy randomized adaptive search procedures. J. Global Optimization **6**, 109-133 (1995)
11. Gill, P.E., W. Murray, and M.H. Wright: Practical Optimization. Academic Press London (1984)
12. Hirsch, M.J., C.N. Meneses, P.M. Pardalos, and M.G.C. Resende, Global optimization by continuous GRASP. Optimization Letters **1**, 201-212 (2007)
13. Hirsch, M.J., P.M. Pardalos, and M.G.C. Resende, Speeding up Continuous GRASP. Submitted to European Journal of Operational Research

14. Horst, R. and P.M. Pardalos (Eds.): Handbook of Global Optimization, Kluwer Dordrecht (1995)
15. Järvi, T., A random search optimizer with an application to a max-min problem. Publications of the Institute for Applied Mathematics, University of Turku, No. 3, 1973
16. Markót, M.Cs. and T. Csendes: A new verified optimization technique for the "packing circles in a unit square" problems. SIAM J. on Optimization **16**, 193-219 (2005)
17. Moles, C.G., J.R. Banga, and K. Keller, Solving nonconvex climate control problems: pitfalls and algorithm performances. Applied Soft Computing **5**, 35-44 (2004)
18. Moles, C.G., G. Gutierrez, A.A. Alonso, and J.R. Banga, Integrated process design and control via global optimization – A wastewater treatment plant case study. Chemical Engineering Research & Design **81**, 507-517 (2003)
19. Mongeau, M., H. Karsenty, V. Rouzé, J.-B. Hiriart-Urruty: Comparison of public-domain software for black-box global optimization. Optimization Methods & Software **13**, 203-226 (2000)
20. Rinnooy Kan, A.H.G. and G.T. Timmer, Stochastic Global Optimization Methods, Part I: Clustering Methods. Mathematical Programming, **39**, 27-56 (1987)
21. Rinnooy Kan, A.H.G. and G.T. Timmer, Stochastic Global Optimization Methods, Part II: Multi Level Methods. Mathematical Programming, **39**, 57-78 (1987)
22. Sendín, J.O.H., J.R. Banga, and T. Csendes: Extensions of a Multistart Clustering Algorithm for Constrained Global Optimization Problems. Manuscript submitted for pubication. Available at `www.inf.u-szeged.hu/~csendes/ReportGOBALm.doc`
23. Szabó, P.G., M.Cs. Markót, T. Csendes, E. Specht, L.G. Casado, and I. García: New Approaches to Circle Packing in a Square – With Program Codes. Springer New York (2007)
24. Tóth, B. and T. Csendes: Empirical investigation of the convergence speed of inclusion functions. Reliable Computing **11**, 253-273 (2005)
25. Tóth, B., J. Fernández, and T. Csendes: Empirical convergence speed of inclusion functions for facility location problems. J. of Computational and Applied Mathematics **199**, 384-389 (2007)