Global optimization algorithms for bound constrained problems

Theses of PhD Dissertation

László Pál

Supervisor:
Prof. Dr. Tibor Csendes

University of Szeged
PhD School of Computer Science
Szeged, 2010

1 Introduction

Global optimization is a multidisciplinary research field and a branch of applied mathematics and numerical analysis that deals with the task of finding the absolutely best set to satisfy certain criteria, formulated in mathematical terms. In other words, it refers to the characterization and computation of the global extrema of a given nonconvex function in a certain feasible region which may have several local minimizers. Global optimization problems are typically quite difficult to solve exactly since many of them belong to the class of NP-complete problems. On the other hand, many real-word application can be formulated as a global optimization problem. Hence, finding the optimum solution of such a problem is an important challenge.

In this thesis, two important fields of continuous global optimization have been considered: the stochastic and the interval arithmetic based global optimization. Our aim was to implement and investigate efficient algorithms in order to solve the general bound constrained global optimization problem.

During our investigations, we deals with bound constrained global optimization problems that can be formulated in the following way:

$$\min_{x \in X} f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a real valued function, $X = \{a_i \leq x_i \leq b_i,\ i = 1, 2, \ldots, n\}$ is the set of feasibility. In general, we assume that the objective function is twice continuously differentiable, although it is not necessary since the considered methods can solve also nondifferentiable problems, too.

2 Stochastic global optimization

As no algorithm can solve a general, smooth global optimization problem with certainty in finite time, stochastic methods are of eminent importance in global optimization. These methods incorporate probabilistic elements, which means that the result of the method is a random variable. Although stochastic methods do not offer an absolute guarantee of success, under mild conditions on f, the probability that the global optimum will be found by them approaches 1 as the sample size grows.

2.1 The GLOBAL optimization method

GLOBAL (see Csendes [3]) is a stochastic method based on Boender's algorithm (see Boender et al. [1]). In several recent comparative studies (e.g. Mongeau et al. [15]; Moles et al. [14]), this method performed quite well in terms of both efficiency and robustness, obtaining the best results in many cases.

Its goal is to find all local minimizer points that are potentially global. These local minimizers will be found by means of a local search procedure, starting from appropriately chosen points from the sample drawn uniformly within the set of feasibility. In an effort to identify the region of attraction of a local minimum, the procedure invokes a clustering procedure. The method steps are described in Algorithm 1.

Algorithm 1. The GLOBAL method

function GLOBAL(f, X)
  $k := 0; X^* := \emptyset; X^{(1)} := \emptyset$
  repeat
    $k := k + 1$
    Generate N points $x_{(k-1)N+1}, \ldots, x_{kN}$ according to the uniform distribution on X
    Determine the reduced sample consisting of the γkN best points from the sample $x_1, \ldots, x_{kN}$
    Clustering to X^* and $X^{(1)}$
    while *Not all points from the reduced sample have been assigned to a cluster* **do**
      $x^{(1)}$ – unclustered point with the smallest objective value
      $x^* := Loc(x^{(1)})$
      if $x^* \notin X^*$ **then**
        $X^* := X^* \cup \{x^*\}$
        choose $x_s := x^*$ as the next seed point
      else
        $X^{(1)} := X^{(1)} \cup \{x^{(1)}\}$
        choose $x_s := x^{(1)}$ as the next seed point
      end
      Add all unclustered reduced sample points which are within distance r_k of a point already in the cluster initiated by the seed point x_s
    end
  until *Some global stopping rule is satisfied*
  return *The smallest local minimum value found*

Although GLOBAL is based on Boender's method, it contains several modifications and improvements. The most important changes 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 it is left out.
- No steepest descent step is used to transform the original sample.
- 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.

Apart from the above changes, the original GLOBAL algorithm includes two different local search procedures: a quasi-Newton procedure with the Davidon-Fletcher-Powell (DFP) update formula and a random walk type direct search method UNIRANDI (see Järvi [11]), which 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.

The GLOBAL method 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. Therefore, our aim was to make some revisions and updates on the involved algorithms to utilize the novel technologies, and to improve its reliability. The main part of the work consists of experiments done with the GLOBAL algorithm. As a result of these tests we obtained a more efficient and robust method.

The most important changes made on GLOBAL are:

- It is now coded in MATLAB, utilizing the vectorization technique for better efficiency.
- We use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) (see Broyden [2]) local search algorithm instead of the earlier DFP method.
- 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 processor pipeline. This is otherwise hard to reach 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.

The BFGS local search method works similarly like the DFP algorithm. The main difference is that the first one uses a different update formula. The comparison results (see Powell [22]) show that the quasi-Newton method with the BFGS update formula performs better than the one with the DFP update formula.

In the UNIRANDI local search method, the random directions are uniformly generated in the interval $[-0.5, 0.5]^n$, but they are accepted only if the norm is less or equal than 0.5. This condition means that points outside the hypersphere of radius 0.5 are discarded in order to obtain a uniform distribution of random directions (i.e. to avoid having more directions pointing toward the corners of the hypercube). As the number of variables increases, it becomes more difficult to produce points satisfying this condition and for more than 15 variables it is never satisfied.

In order to fix this problem, we changed the UNIRANDI so that, the random directions will be generated by normal distribution $\mathcal{N}(0, 1)$, instead of uniform distribution and the respective vectors are normed.

3 Numerical results

We have completed two sets of numerical tests: the first aimed to show the efficiency and reliability changes compared to the old version, based on the published results in Csendes [3], and one to compare the new method to C-GRASP, a greedy adaptive search technique (see Feo and Resende [9]) modified to solve continuous global optimization problems published in Hirsch et al. [10].

GLOBAL has six parameters to set: the number of sample points set within an iteration step, the number of best points selected for the transformed sample, the stopping criterion parameter for the local search, the maximum number of function evaluations allowed for local search, the maximum number of local minima to be applied, and the type of used local method. All these parameters have a default value and usually it is enough to change only the first three of them.

In the first test, 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. We also 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.

The conclusion of the first set of tests completed is that on standard test problems the new implementation is closely as good in terms of efficiency as was the old one, 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.

As the C-GRASP method does not utilize the possible smoothness of the objective function, in the second test, we compared it to GLOBAL with UNIRANDI. 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.

For each problem, 100 independent runs of GLOBAL were completed. 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 such a way that it was able to find a global minimizer point in each run.

Summarizing the results, 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. The comparison results are published in Csendes et al. [6].

We also evaluated the performance of the GLOBAL algorithm on the BBOB (Black-Box Optimization Benchmarking) 2009 noiseless testbed, containing problems which reflect the typical difficulties arising in real-word applications. Results show that the GLOBAL algorithm performs well especially on functions with moderate number of local minima using a small budget of function evaluations. GLOBAL was ranked one of the best method for a function evaluation budget of up to $500n$ function values, but was no longer competitive when the budget was significantly larger. The results can be found in (Pál et al. [20]; Pošík et al. [21]).

3.1 Application: The flexible retirement problem

We analyzed the problem of designing a stable pension scheme described in Eső and Simonovits [7, 8]. There is a population of individuals who have private information regarding their life expectancies. The government's goal is to design an optimal pension system by maximizing an additive concave social welfare function.

The problem can be formulated in the following way:

$$\max_{(b_t,R_t)_t} \sum_{t=S}^{T} \psi(v_t) f_t, \tag{2}$$

subject to

$$v_t = [\bar{u} - w(b_t)]R_t + w(b_t)t, \quad t = S, \dots T, \tag{3}$$

$$\sum_{t=S}^{T} [(\tau + b_t)R_t - tb_t] f_t = 0, \tag{4}$$

$$v_{t+1} = v_t + w(b_t),\ t = S, \dots, T-1. \tag{5}$$

The life expectancies are denoted by t, f_t is the relative frequency of individuals with a life expectancy of t, τ is a yearly social security contribution rate, b_t is a yearly retirement benefit received after R_t active years. Denote further the lifetime utility of a worker with life expectancy t by v_t. The optimal benefit retirement schedule will have to satisfy all incentive compatibility constraints (equations (5)). The incentive compatibility of $(b_t, R_t)_{t=S}^{T}$ means that type t prefers to choose (b_t, R_t) from the schedule. It is also required that the population balance to be 0 (equation (4)).

The goal is to design an optimal pension system, described by $(b(R), \tau)$, maximizing an additive concave social welfare function (equation (2)).

Although the mathematical model of the problem contains constraints, we managed to solve it by GLOBAL using the penalty function method. The conclusion of the numerical test is that on the investigated problem the improved GLOBAL algorithm was able to find good approximations of the global minimizer points while the amount of computational efforts needed remained limited and in the acceptable region. These results are published in Pál and Csendes [16].

4 Interval global optimization methods

Global optimization methods that use interval techniques provide rigorous guarantees that a global minimizer is found. Interval techniques are used to compute global information about functions over large regions. Most global optimization methods using interval techniques employ a branch and bound strategy. These algorithms decompose the search domain into a collection of boxes for which the lower bound on the objective function is calculated by an interval technique.

Our aim was to implement an easy to use reliable global optimization method implemented in MATLAB by using the INTLAB package, in order to solve the general bound constrained global optimization problem.

4.1 Solution algorithms

We have implemented two algorithm variants which are based on the basic interval branch and bound framework described by Algorithm 2.

Both of the algorithms use only a subroutine calculating the objective function as information on the global optimization problem, i.e. the expression is not required.

The first method is a simple algorithm which does not apply the gradient and the Hessian of the objective function, although these can be computed by the automatic differentiation facility of INTLAB. In other words, we study now that algorithm variant that does not assume the differentiability of the objective function. Now just natural interval extension was applied to calculate the inclusion functions. We use only simple bisection along the widest interval component, and no multisection and advanced subdivision direction selection. The subdivision direction is determined according to the well-tested and simple A subdivision direction selection rule.

Algorithm 2. The basic interval branch and bound framework

```
function IntervalBranchAndBound(f, X)
  L_res := ∅;  L_work := {X}
  while L_work ≠ ∅ do
      Remove the first element (X) from the working list
      Calculate F(X)
      if X cannot be discarded then
          Subdivide X into X^i, i = 1, ..., p subintervals
          if X^i fulfils some stopping criteria then
              Enter X^i in the result list
          else
              Enter X^i in the working list
          end
      end
  end
  return L_res
```

The second algorithm is a more advanced one, which applies the most common accelerating devices: the cutoff test, the concavity test, the monotonicity test, and the interval Newton step. Beyond natural interval extension, a simple centered form inclusion function is also applied. Once the inclusion of the gradient is available, the intersection of these inclusion functions proved to be a good quality estimation of the range of the objective function.

We use also multisection and advanced subdivision direction selection (see Kearfott [12]), albeit without those based on the pf^* heuristic algorithm parameter (see Csendes [4]). Multisection means this time that each interval will be subdivided into three subintervals according to the most promising two coordinate directions. The subdivision directions are determined according to the well-tested and effective C subdivision direction selection rule (also used in Csendes [4] and Kearfott [12]).

During the implementation of both algorithms we have followed closely the original C-XSC code which was developed for bound constrained global optimization by Mihály Csaba Markót based on the algorithm documented in Markót et al. [13]. The control structures of the two algorithms are basically the same, while the vectorial array statements of MATLAB were applied wherever possible.

Summarizing our numerical results, we can state that the computational experiences confirm that the new implementation is in several indi-

cators (e.g. number of function, gradient and Hessian evaluations, number of iterations, and memory complexity) in essence equivalent to that of the old one. The CPU time needed is as a rule by at least two order of magnitude higher for the INTLAB version as it can be anticipated regarding the interpreter nature of MATLAB. In spite of the lower speed, the new interval global optimization methods can well be suggested as an early modeling and experimentation tool for the verified solution of bound constrained global optimization problems. These results can be found in (Csendes and Pál [5]; Pál and Csendes [17]).

4.2 Improvements on the Newton step

The Newton step is one of the most important accelerating tool which is used in our algorithm. We apply one step of the extended interval Newton Gauss-Seidel method to the gradient of the objective function in order to shrink the argument interval. The application of the Newton step on each single interval would be costly. Hence, it would be suitable to use a criterion in order to decide when to apply the Newton step. In our study Pál and Csendes [17], we used the criterion described in Markót et al. [13]. In this case we applied the Newton step if two of the trisected boxes were eliminated by other acceleration tools.

Our aim was to introduce a new condition for applying the Newton step in order to improve our algorithm. In this case, we apply an interval Newton step to each box which has a smaller width than a prescribed value. It is well known that the Newton step is more efficient when it is applied to a small box rather than to a large one. In the new condition we used 0.1 as the threshold value.

We have implemented an algorithm in MATLAB/INTLAB with the new condition in order to compare it with the old one. The algorithm is similar to one described in Pál and Csendes [17]. The most important differences are the following: the old interval selection rule is changed to the new one: we select the subinterval which has the maximal pf^* parameter value and the algorithm stops at the first box which satisfies the stopping criterion. The latter approach is acceptable in many practical situations when it is not necessary to find all global minimizer points.

We have also completed a computational test in order to compare the new condition with the old one. Based on these results (see Pál and Csendes [18]) it can be stated that we managed to reduce significantly the necessary computational time and the total number of Hessian evaluations with the help of the new condition.

4.3 Theoretical examination of the Newton step

As we have seen in the previous section, the interval based global optimization algorithm can be improved by using a well chosen threshold value in the new condition for applying the Newton step. However, there are cases when the Newton step with the new condition is not successful in the sense that the width of the interval does not decrease or the current interval will be divided into many subintervals. The latter case appears in the practice mostly and is not beneficial because we may achieve a similar result by using some subdivision, but with a much smaller expense.

We analyzed the theoretical aspects of the described problem on a predefined class of inclusion functions of the second derivative. During the investigation we have distinguished between symmetric and non-symmetric inclusion functions and between single-variable and multivariate objective functions. According to the results in Pál and Csendes [18], we managed to characterize the cases when the Newton step is not successful.

4.4 Application in sensor network localization

We applied interval techniques for two real-world applications (see Pál and Csendes [18, 19]). In the thesis, we focused mainly on the sensor network localization which refers to the process of estimating the locations of sensors, if the positions of some nodes and the noisy distance measurements between the nodes are known. The presented model can be formulated as a global optimization problem with the aim to minimize the sum of errors in sensor positions for fitting the distance measurements.

Finding the global optimum of such a problem with the INTLAB based global optimizer is a time consuming task especially using hundreds of sensors. Hence, our aim was to find an approximate solution using the multilateration technique, which tries to determine the position of an unknown node using at least three references located at known positions.

We obtained an initial solution of sensor locations (see Pál and Csendes [18]) by iterative multilateration with randomly selected four reference nodes. We also found that the localization error can be decreased by increasing the anchor nodes number or the radio range.

The published papers and the MATLAB implementation of the GLOBAL and the INTLAB based methods are available on the web page

`http://www.emte.siculorum.ro/~pallaszlo/Disszertacio`

References

[1] C.G.E. Boender, A.H.G. Rinnooy Kan, G.T. Timmer, and L. Stougie. A stochastic method for global optimization. *Mathematical Programming*, 22:125–140, 1982.

[2] C.G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.

[3] T. Csendes. Nonlinear parameter estimation by global optimization – efficiency and reliability. *Acta Cybernetica*, 8:361–370, 1988.

[4] T. Csendes. New subinterval selection criteria for interval global optimization. *Journal of Global Optimization*, 19:307–327, 2001.

[5] T. Csendes and L. Pál. A basic interval global optimization procedure for Matlab/INTLAB. In *Proceedings of International Symposium on Nonlinear Theory and its Applications (NOLTA2008)*, pages 592–595, Budapest, Hungary, 2008.

[6] T. Csendes, L. Pál, J.O.H. Sendín, and J.R. Banga. The GLOBAL optimization method revisited. *Optimization Letters*, 2:445–454, 2008.

[7] P. Eső and A. Simonovits. Designing Optimal Benefit Rules for Flexible Retirement. Technical Report CMS-EMS 1353, Northwestern University, Evanston, 2002.

[8] P. Eső and A. Simonovits. Designing Optimal Benefit Rules for Flexible Retirement (in Hungarian: Optimális járadékfüggvény tervezése rugalmas nyugdíjrendszerre). *Közgazdasági Szemle*, 50:99–111, 2003.

[9] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[10] M.J. Hirsch, C.N. Meneses, P.M. Pardalos, and M.G.C. Resende. Global optimization by continuous GRASP. *Optimization Letters*, 1: 201–212, 2007.

[11] T. Järvi. A random search optimizer with an application to a max-min problem. *Publications of the Institute for Applied Mathematics*, (3), University of Turku, 1973.

[12] R.B. Kearfott. *Rigorous global search: continuous problems*. Kluwer Academic Publishers, Dordrecht, 1996.

[13] M.C. Markót, J. Fernandez, L.G. Casado, and T. Csendes. New interval methods for constrained global optimization. *Mathematical Programming*, 106:287–318, 2006.

[14] C.G. Moles, 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 and Design*, 81:507–517, 2003.

[15] M. Mongeau, H. Karsenty, V. Rouzé, and J.B. Hiriart-Urruty. Comparison of public-domain software for black box global optimization. *Optimization Methods and Software*, 13:203–226, 2000.

[16] L. Pál and T. Csendes. Improvements on the GLOBAL Optimization Algorithm with Numerical Tests. In *Proceedings of the 8th International Conference on Applied Informatics (ICAI2007)*, pages 101–109, Eger, Hungary, 2007.

[17] L. Pál and T. Csendes. INTLAB implementation of an interval global optimization algorithm. *Optimization Methods and Software*, 24:749–759, 2009.

[18] L. Pál and T. Csendes. An INTLAB based global optimization method and it's application to a sensor network localization problem (in Hungarian: Egy intervallum alapú globális optimalizálási módszer és alkalmazása szenzor lokalizálási feladatra). Közlésre beküldve, 2010.

[19] L. Pál and T. Csendes. Efficient estimation of loads in service networks. Accepted for publication, 2010.

[20] L. Pál, T. Csendes, M.C. Markót, and A. Neumaier. Black-box optimization benchmarking of the global method. Submitted for publication, 2010.

[21] P. Pošík, W. Huyer, and L. Pál. Benchmarking Several Global-Optimization Algorithms. Submitted for publication, 2010.

[22] M.J.D. Powell. How bad are the BFGS and DFP methods when the objective function is quadratic? *Mathematical Programming*, 34: 34–47, 1986.