Testing branch-and-bound methods for global optimization

Kaj Madsen
Julius Žilinskas*

May 26, 2000

Abstract

This report presents results of some experiments with two codes implementing attraction based branch-and-bound global optimization methods: A non-deterministic method using real arithmetic, and a deterministic interval arithmetic method. The domain of the optimization is a compact right parallelepiped in $\Re^n$, parallel to the coordinate axes.
Traditional mathematical test functions as well as two practical problems were used for the testing.

*Department of Informatics, Kaunas University of Technology, Studentu 50-214b, Kaunas, Lithuania. *(jzil@dsplab.ktu.lt)*

1 Introduction

The idea of using interval arithmetic for global optimization has been investigated by many authors. A recent review is presented, e.g. in [3]. Besides of many advantages interval methods have also some disadvantages as discussed for instance in [8]. The last paper proposes to use a subdivision strategy of the feasible region which is similar to that used in interval methods. However the subdivision strategy is based on information from real arithmetic calculations only, so the need of applying interval arithmetic is avoided. Thus the method has a much broader field of applications than the corresponding interval methods, including problems with a "black box" calculation of the objective function. On the other hand the reliability of the interval method is lost.

In this report we compare two methods for solving the following optimization problem: Find the global minimum of a smooth function f,

$$f^* = \min \{f(x) \mid x \in D\} \tag{1}$$

as well as the set of points for which it is attained,

$$X^* = \{x \in D \mid f(x) = f^*\} \tag{2}$$

where D is a compact right parallelepiped in $\Re^n$, parallel to the coordinate axes. Two codes implementing the methods have been tested using a set of traditional mathematical test functions and two practical problems. The first code (which is denoted by StoGO) is a C++ implementation of the method proposed in [9] (inspired by the implementations of [11] and [2]), whereas the interval code (denoted by IntGO) is the C++ implementation [7] of the method of Jansson and Knüppel described in [5]. The latter method involves combination of local (real arithmetic) searches, branch-and-bound techniques and interval arithmetic.

2 Test functions

Test functions with different dimensions and different numbers of local and global minimizers were used in the experiments. Apart from the last problem, denoted by BoneGrowth, all test problems are know from the global optimization literature. The dimensions and the numbers of minimizers of

Function	Dimension	Number of local min.	Number of global min.
Rosenbrock	2	1	1
McCormic	2	1	1
Box&Betts	3	1	1
Paviani	10	1	1
Gen Rosenbrock	30	1	1
Gold&Price	2	4	1
Shekel 5	4	5	1
Shekel 7	4	7	1
Shekel 10	4	10	1
Levy 4	4	71000	1
Levy 5	5	10^5	1
Levy 6	6	10^6	1
Levy 7	7	10^8	1
Griewank	10	10^3	1
Cola	17	?	1
Growth	12	?	?
Six Hump Camel	2	6	2
Branin	2	23	5
Shubert	2	400	9
Hansen	2	760	9

Table 1: The dimensions and the numbers of local and global minimizers of test functions.

test functions are shown in Table 1 where the problems are divided into the categories: 1 local minimizer; 1 global minimizer and a few local; 1 global minimizer and many local; a few global minimizers; practical problems.

The first practical problem, denoted by Cola, is the MDS problem which is discussed in [10]; the data used in this test correspond to the classical "Cola testing" problem. The number of variables is 17. There are many local minimizers with function values close to the global minimum, see [10]. The second practical problem is given in [1]. It is related to a linear growth model of the human mandible (the lower jar). The dimension of this problem is 12 and there are many local minimizers.

Detailed descriptions of the test problems are given in Appendix A. C++ definitions of the test problems (and also of the StoGO program) can be found in [4].

3 Criteria of efficiency

The main criteria of efficiency of global optimization algorithms are the numbers of calls of the objective function and perhaps its gradient, and the calculation time of the optimization. The number of calls is used when the objective function is expected to be "expensive", i.e its calculation requires more time than the auxiliary calculations by the optimization algorithm. In case of the contrary relation, the calculation time is an important criterion. In this report we use the numbers of function calls as efficiency criterion.

Comparison of two algorithms with respect to function calls of the overall optimization is well grounded when the stopping conditions are the same. For StoGO the stopping condition is the time limit since it can normally not be detected if all global minimizers have been found. For IntGO, however, the calculations are normally stopped when all global minimizers have been found, and it has been justified that there are no more. Because of this difference in stopping conditions we chose to use the numbers of calls needed to find the first and the last global minimizer, respectively, as the criterion of efficiency for both algorithms.

For StoGO the criteria of efficiency are the number of objective function calls (Nrf) and the number of gradient calls (Nrg). If the gradient is expressed analytically, then the gradient call may cost the same as the objective function in some cases, in other cases up to n times a call of calls of the objective function. If the gradient is evaluated using automatic differentiation the gradient call usually costs about 3 times the objective function call. If the gradient is evaluated using finite difference approximations, the call of gradient function costs approximately $n + 1$ calls of the objective function. In our experiments most gradients expressed analytically, otherwise (for instance in the two practical problems) they are found using finite difference approximations.

IntGO does not use gradients, however interval function values must be found. Therefore the criteria of efficiency for IntGO are the number of real function calls (Nrf) and the number of interval function calls (Nif). The authors of IntGO state that the average cost of an interval function call is twice the cost of a corresponding real function call [5].

4 Experiments with StoGO

In the experiments with StoGO the default settings of parameters (see [2]) were used. The number of regular sample points $n_{reg} = 2n+1$, the number of random points $n_{rnd} = 0$. The parameter of the generation of regular points inside a box $r_s = 0.3$.
In the stopping condition of the local searches we use $\epsilon = 10^{-4}$, and two values of the cluster radius were used: $\epsilon_{cluster} = 0.1$ and $\epsilon_{cluster} = 0.01$. Improved initialization using the alternative variables method was not used.

The numbers of objective and gradient function calls and outer iterations needed to find the first and all minimizers are given in Tables 2 and 3. All global minimizers were found for all test problems. (For the Growth problem this means that the smallest known function value, $f^* = 205.104$ and the corresponding minimizer were found.) Notice that in almost all cases the solutions are found during the first outer loop of the algorithm.

For many test functions the two choices of $\epsilon_{cluster}$ give the same performance of StoGO. For the 8 problems with differences, $\epsilon_{cluster}$=0.1 is always the best, however the differences are not very large (except for the Levy 4 function). In general we conclude that the experiments indicate some independence of the choice of $\epsilon_{cluster}$.

The algorithm has difficulties with functions having very many oscillations and local minima, like the Levy, Shubert and Hansen functions. For such functions the performance seems to be a bit random: For Levy 4 which is the "easiest" of the four Levy functions, the solution requires many function evaluations, whereas the most difficult one, Levy 7, is solved rather quickly because the algorithm accidentally finds the global minimizer at the beginning of the optimization. The performance in such cases may depend on how close the starting points of the local searches are to the global optimum. Thus the performance may change if the boxes D of feasible regions are shifted.
For other difficult functions with many local minimizers, like the Griewank, Cola and Growth functions, the algorithm works rather well.

Function	n	all minimizers		OI
		Nrf	Nrg	
Rosenbrock	2	27	23	1
McCormic	2	11	10	1
Box&Betts	2	6	6	1
Paviani	2	14	11	1
Gen Rosenb	30	359	346	1
Gold&Price	2	75	51	1
Shekel 5	4	66	37	1
Shekel 7	4	24	13	1
Shekel 10	4	22	12	1
Levy 4	4	30502	23798	1
Levy 5	5	2786	2147	1
Levy 6	6	5292	4154	1
Levy 7	7	35	30	1
Griewank	10	124	81	1
Cola	17	8995	7226	1
Growth	12	529	414	1

Function	n	first minimizer		OI	all minimizers		OI
		Nrf	Nrg		Nrf	Nrg	
Six Hump C	2	37	33	1	49	45	1
Branin	2	21	10	1	666	527	2
Shubert	2	26	15	1	6825	4412	1
Hansen	2	218	129	1	7655	5037	1

Table 2: $\epsilon_{cluster}$=0.1. The numbers of objective function (*Nrf*) and gradient (*Nrg*) calls needed to find the first and all minimizers using improved StoGO with default parameters. *OI* denotes the number of outer iterations of the algorithm. The two blocks of the table indicate whether there is one or several global minimizers

Function	n	all minimizers		OI
		Nrf	Nrg	
Rosenbrock	2	27	23	1
McCormic	2	11	10	1
Box&Betts	2	6	6	1
Paviani	2	14	11	1
Gen Rosenb	30	359	346	1
Gold&Price	2	81	56	1
Shekel 5	4	66	37	1
Shekel 7	4	24	13	1
Shekel 10	4	22	12	1
Levy 4	4	44316	33438	1
Levy 5	5	2823	2184	1
Levy 6	6	5389	4241	1
Levy 7	7	35	30	1
Griewank	10	124	81	1
Cola	17	9229	7413	1
Growth	12	529	414	1

Function	n	first minimizer		OI	all minimizers		OI
		Nrf	Nrg		Nrf	Nrg	
Six Hump C	2	37	33	1	49	45	1
Branin	2	21	10	1	733	580	2
Shubert	2	26	15	1	8810	5380	1
Hansen	2	218	129	1	10296	6409	1

Table 3: $\epsilon_{cluster}$=0.01. The numbers of objective function (*Nrf*) and gradient (*Nrg*) calls needed to find the first and all minimizers using improved StoGO with default parameters. *OI* denotes the number of outer iterations of the algorithm. The two blocks of the table indicate whether there is one or several global minimizers

5 Comparison with IntGO

We compare the performance of StoGO with a method of Jansson and Knüppel, [5], which is based on a combination of local searches, branch-and-bound techniques and interval arithmetic. The implementation of Knüppel, [7], here denoted by IntGO, was tested using the same test functions as for StoGO. We have used three values of the parameter n_d of IntGO which determines the number of bisections made in each iteration: $n_d = 2$ (which is the default value used for the corresponding parameter of StoGO), $n_d = n+1$, and $n_d =$ *a value tuned for each individual test problem.* The numbers of real and interval function calculations needed to find the first and all minimizers are shown in Tables 4, 5 and 6.

The tables demonstrate some dependence of the choice of n_d. Table 6 shows that sometimes quite a lot may be gained by tuning this parameter. In the Generalized Rosenbrock, for instance, the number of real function calls used by IntGO varies from 2698 to 14260.

The IntGO failed to minimize the two practical problems Cola and Growth. To be more specific the computation broke down because of overflow of memory. For the Cola problem the number of unexplored boxes generated by the algorithm was more than 131072 at the stopping time, and the number of interval function calls was greater than 393213. Similar results were seen for the Growth problem.

The numbers of calls when minimizing other problems are relatively small, i.e. the method performs well for these problems.

When comparing IntGO with StoGO we notice that normally IntGO is clearly best for the trigonometric functions Levy, Shubert and Hansen (which have very many oscillations and local minimizers). For the other functions StoGO seems to be the best. In Table 7 we provide comparisons between the optimal case of IntGO (i.e. n_d being tuned) and StoGO with $\epsilon_{cluster}$=0.1 and default parameter values otherwise, i.e. Table 7 is a combination of figures from the Tables 2 and 6. In this table we have divided the problems into three categories: Problems for which IntGO is clearly the best, problems for which StoGO is slightly best, and problems for which IntGO is clearly best (according to function calls, and assuming that a real gradient call costs approximately the same as an interval function call, which is true when automatic differentiation is used).

Function	n	all minimizers Nrf	Nif
Rosenbrock	2	170	13
McCormic	2	97	9
Box&Betts	2	97	21
Paviani	2	366	41
Gen Rosenbrock	30	6513	125
Gold&Price	2	129	9
Shekel 5	4	172	17
Shekel 7	4	166	17
Shekel 10	4	211	17
Levy 4	4	614	6839
Levy 5	5	555	202
Levy 6	6	527	330
Levy 7	7	543	506
Griewank	10	fails	
Cola	17	fails	
Growth	12	fails	

Function	n	first minimizer		all minimizers	
		Nrf	Nif	Nrf	Nif
Six Hump Camel	2	225	554	321	819
Branin	2	324	119	fails	
Shubert	2	524	1543	fails	
Hansen	2	89	17	862	1421

Table 4: Results of optimization using the implementation IntGO of Knüppel [7]. The number of subdivisions per iteration, n_d, is 2 for all problems. n = dimension, Nrf = the number of real function calls, Nif = the number of of interval function calls. The two blocks of the table indicate whether there is one or several global minimizers

Function	n	all minimizers Nrf	Nif
Rosenbrock	2	104	19
McCormic	2	96	13
Box&Betts	2	103	33
Paviani	2	524	221
Gen Rosenbrock	30	14260	1921
Gold&Price	2	243	226
Shekel 5	4	98	41
Shekel 7	4	101	41
Shekel 10	4	101	41
Levy 4	4	111	41
Levy 5	5	127	61
Levy 6	6	235	85
Levy 7	7	235	113
Griewank	10	186	221
Cola	17	fails	
Growth	12	fails	

Function		first minimizer		all minimizers	
	n	Nrf	Nif	Nrf	Nif
Six Hump Camel	2	268	4532	342	5411
Branin	2	506	453	803	687
Shubert	2	940	3802	1550	4776
Hansen	2	193	200	892	1686

Table 5: Results of optimization using the implementation IntGO of Knüppel [7]. The number of subdivisions per iteration, n_d, is n+1 for all problems. n = dimension, Nrf = the number of real function calls, Nif = the number of of interval function calls. The two blocks of the table indicate whether there is one or several global minimizers

Function	n	n_d	all minimizers	
			Nrf	Nif
Rosenbrock	2	3	104	19
McCormic	2	3	96	13
Box&Betts	2	1	93	11
Paviani	2	2	366	41
Gen Rosenbrock	30	7	2698	431
Gold&Price	2	1	120	5
Shekel 5	4	4	94	33
Shekel 7	4	4	85	33
Shekel 10	4	4	85	33
Levy 4	4	5	111	41
Levy 5	5	4	187	41
Levy 6	6	4	178	49
Levy 7	7	4	233	57
Griewank	10	8	419	161
Cola	17	any	fails	
Growth	12	any	fails	

Function	n	n_d	first minimizer		all minimizers	
			Nrf	Nif	Nrf	Nif
Six Hump Camel	2	1	273	352	365	443
Branin	2	3	506	453	803	687
Shubert	2	4	357	1008	1060	3472
Hansen	2	2	89	17	862	1421

Table 6: Results of optimization using the implementation IntGO of Knüppel [7]. The number of subdivisions per iteration, n_d, is tuned for each individual problem. n = dimension, *Nrf* = the number of real function calls, *Nif* = the number of of interval function calls. The two blocks of the table indicate whether there is one or several global minimizers

Function	n	n_{loc}	n_{glob}	StoGO		IntGO	
				Nrf	Nrg	Nrf	Nif
Levy 4	4	71000	1	30502	23798	111	41
Levy 5	5	10^5	1	2786	2147	187	41
Levy 6	6	10^6	1	5292	4154	178	49
Shubert	2	400	9	6825	4412	1060	3472
Hansen	2	760	9	7655	5037	862	1421
Rosenb.	2	1	1	27	23	104	19
McCormic	2	1	1	11	10	96	13
Gold&P.	2	4	1	75	51	120	5
Shekel 5	4	5	1	66	37	94	33
Branin	2	23	5	666	527	803	687
B.&B.	3	1	1	6	6	93	11
Paviani	10	1	1	14	11	366	41
Gen Ros.	30	1	1	359	346	2698	431
Shekel 7	4	7	1	24	13	85	33
Shekel 10	4	10	1	22	12	85	33
Levy 7	7	10^8	1	35	30	233	57
Griewank	10	10^3	1	124	81	419	161
Cola	17	?	1	8995	7226	fails	
Growth	12	?	?	529	414	fails	
Six H.C.	2	6	2	49	45	365	443

Table 7: Numbers of function calls of StoGO and IntGO needed to find all global minimizers (from Tables 2 and 6). n_{loc} and n_{glob} are the numbers of local and global minimizers, respectively, and the other names are identical to those of Tables 2 and 6. The upper block is the set of problems for which IntGO is clearly the best, in the middle block StoGO is slightly best, whereas StoGO is clearly best in the last block .

A Test Problems

In the following descriptions of test problems the notation will be used:
n - dimension,
X - feasible region,
$f(x)$ - objective function,
$G(x)$ - Gradient function,
f^* - global minimum,
X^* - set of global minimizers,
n_{loc} - number of local minimizers.

A.1 Rosenbrock function

$n = 2$, $X = [-2, 2]^n$

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

$$\begin{aligned} G_1(x) &= 200(x_2 - x_1^2)(-2x_1) - 2(1 - x_1) \\ G_2(x) &= 200(x_2 - x_1^2) \end{aligned}$$

$f^* = 0$, $X^* = (1, 1)$, $n_{loc} = 1$

A.2 McCormic function

$n = 2$, $X = ([-1.5, 4], [-3, 4])$

$$f(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1$$

$$\begin{aligned} G_1(x) &= \cos(x_1 + x_2) + 2(x_1 - x_2) - 1.5 \\ G_2(x) &= \cos(x_1 + x_2) - 2(x_1 - x_2) + 2.5 \end{aligned}$$

$f^* = -1.9133$, $X^* = (-0.54719, -1.54719)$, $n_{loc} = 1$

A.3 Box and Betts exponential quadratic sum

$n = 3$, $X = ([0.9, 1.2], [9, 11.2], [0.9, 1.2])$

$$f(x) = \sum_{i=1}^{10} g(x)^2$$

where:

$$g(x) = (\exp(-0.1ix_1) - \exp(-0.1ix_2) - (\exp(-0.1i) - \exp(-i))x_3)$$

$$\begin{aligned} G_1(x) &= \sum_{i=1}^{10} -0.2g(x)i\exp(-0.1ix_1) \\ G_2(x) &= \sum_{i=1}^{10} 0.2g(x)i\exp(-0.1ix_2) \\ G_3(x) &= \sum_{i=1}^{10} 2g(x)(-\exp(-0.1i) + \exp(-i)) \end{aligned}$$

$f^* = 0$, $X^* = (1, 10, 1)$, $n_{loc} = 1$

A.4 Paviani function

$n = 10$, $X = [2.001, 9.999]^n$

$$f(x) = \sum_{i=1}^{n} \left(\ln^2(x_i - 2) + \ln^2(10 - x_i)\right) - \left(\prod_{i=1}^{n} x_i\right)^{0.2}$$

$$G_j(x) = \frac{2\ln(x_j - 2)}{x_j - 2} - \frac{2\ln(10 - x_j)}{10 - x_j} - \frac{0.2\prod_{i=1}^{n} x_i}{x_j\left(\prod_{i=1}^{n} x_i\right)^{0.8}}$$

$f^* = -45.778470$, $X^* = (9.350266, 9.350266, \ldots, 9.350266)$, $n_{loc} = 1$

A.5 Generalized Rosenbrock function

$n = 30$, $X = [-n, n]^n$

$$f(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$$

$$
\begin{aligned}
G_1(x) &= -400(x_2 - x_1^2)x_1 + 2(x_1 - 1) \\
G_i(x) &= -400(x_{i+1} - x_i^2)x_i + 2(x_{i+1} - 1) + 200(x_i - x_{i-1}^2) \\
G_n(x) &= 200(x_n - x_{n-1}^2)
\end{aligned}
$$

$f^* = 0$, $X^* = (1, \ldots, 1)$, $n_{loc} = 1$

A.6 Goldstein and Price function

$n = 2$, $X = [-2, 2]^n$

$$
\begin{aligned}
f(x) = &\left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \\
&\times \left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]
\end{aligned}
$$

$f^* = 3$, $X^* = (0, -1)$, $n_{loc} = 4$

A.7 Shekel function

$n = 4$, $X = [0, 10]^n$

$$
f(x) = -\sum_{i=1}^{m} \frac{1}{(x - A_i)(x - A_i)^T + c_i}
$$

where :

$$
A = \begin{bmatrix}
4 & 4 & 4 & 4 \\
1 & 1 & 1 & 1 \\
8 & 8 & 8 & 8 \\
6 & 6 & 6 & 6 \\
3 & 7 & 3 & 7 \\
2 & 9 & 2 & 9 \\
5 & 5 & 3 & 3 \\
8 & 1 & 8 & 1 \\
6 & 2 & 6 & 2 \\
7 & 3.6 & 7 & 3.6
\end{bmatrix}
\qquad
c = \begin{bmatrix}
0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.5
\end{bmatrix}
$$

$$
G_j(x) = \sum_{i=1}^{m} \frac{2(x_j - a_{i,j})}{((x - A_i)(x - A_i)^T + c_i)^2}
$$

for $m = 5$: $f^* = -10.1532$, $X^* = (4.00004, 4.00013, 4.00004, 4.00013)$, $n_{loc} = 5$

for $m = 7$: $f^* = -10.4029$, $X^* = (4.00057, 4.00069, 3.99949, 3.99961)$, $n_{loc} = 7$

for $m = 10$: $f^* = -10.5364$, $X^* = (4.00075, 4.00059, 3.99966, 3.99951)$, $n_{loc} = 10$

A.8 Levy function

$n = 4, 5, 6, 7$
for $n = 4$: $X = [-10, 10]^n$
for $n = 5, 6, 7$: $X = [-5, 5]^n$

$$\begin{aligned} f(x) = \quad & \sin^2(3\pi x_1) + \textstyle\sum_{i=1}^{n-1}(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1})) + \\ & (x_n - 1)(1 + \sin^2(2\pi x_n)) \end{aligned}$$

$$\begin{aligned} G_1(x) = \quad & 6\sin(3\pi x_1)\cos(3\pi x_1)\pi + 2(x_1 - 1)(1 + \sin^2(3\pi x_2)) \\ G_i(x) = \quad & 6(x_{i-1} - 1)^2 \sin(3\pi x_i)\cos(3\pi x_i)\pi + \\ & 2(x_i - 1)(1 + \sin^2(3\pi x_{i+1})) \\ G_n(x) = \quad & 6(x_{n-1} - 1)^2 \sin(3\pi x_n)\cos(3\pi x_n)\pi + \\ & 1 + \sin^2(2\pi x_n) + 4(x_n - 1)\sin(2\pi x_n)\cos(2\pi x_n)\pi \end{aligned}$$

for $n = 4$: $f^* = -21.502356$, $X^* = (1, 1, 1, -9.752356)$, $n_{loc} = 71000$
for $n = 5, 6, 7$: $f^* = -11.504403$, $X^* = (1, \ldots, 1, -4.754402)$, $n_{loc} = 10^5, 10^6, 10^8$

A.9 Griewank function

$n = 10$, $X = [-500, 700]^n$

$$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$G_j(x) = \frac{x_j}{2000} + \frac{\prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) \sin\left(\frac{x_i}{\sqrt{i}}\right)}{\sqrt{i}\cos\left(\frac{x_i}{\sqrt{i}}\right)}$$

$f^* = 0$, $X^* = (0, 0, \ldots, 0)$, $n_{loc} = 10^3$

A.10 Six Hump Camel Back function

$n = 2$, $X = [-5, 5]^n$

$$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

$$\begin{aligned} G_1(x) = &\quad 8x_1 - 8.4x_1^3 + 2x_1^5 + x_2 \\ G_2(x) = &\quad x_1 - 8x_2 + 16x_2^3 \end{aligned}$$

$f^* = -1.03163$, $X^* = \{(0.08984, -0.71266), (-0.08984, 0.71266)\}$, $n_{loc} = 6$

A.11 Branin function

$n = 2$, $X = [-10, 10]^n$

$$f(x) = \left(1 - 2x_2 + \frac{1}{20}\sin 4\pi x_2 - x_1\right)^2 + \left(x_2 - \frac{1}{2}\sin 2\pi x_1\right)^2$$

$$\begin{aligned} G_1(x) = &\quad -2\left(1 - 2x_2 + \frac{1}{20}\sin 4\pi x_2 - x_1\right) - \\ &\quad 2\left(x_2 - \frac{1}{2}\sin 2\pi x_1\right)\pi\cos 2\pi x_1 \\ G_2(x) = &\quad 2\left(1 - 2x_2 + \frac{1}{20}\sin 4\pi x_2 - x_1\right)\left(-2 + \frac{\pi}{5}\cos 4\pi x_2\right) + \\ &\quad 2\left(x_2 - \frac{1}{2}\sin 2\pi x_1\right) \end{aligned}$$

$f^* = 0$

$$X^* = \left\{ \begin{array}{crrl} (& 1, & 0 &), \\ (& 0.148696, & 0.402086 &), \\ (& 0.402537, & 0.287408 &), \\ (& 1.59746, & -0.287408 &), \\ (& 1.85130, & -0.402086 &) \end{array} \right\}$$

$n_{loc} = 23$

A.12 Shubert function

$n = 2$, $X = [-10, 10]^n$

$$f(x) = -\sum_{i=1}^{n}\sum_{j=1}^{5} j \sin((j+1)x_i + j)$$

$$G_j(x) = -\sum_{i=1}^{5} i(i+1)\cos((i+1)x_j + i)$$

$f^* = -24.062499$

$$X^* = \left\{ \begin{array}{cccc} (& -6.774576, & -6.774576 &), \\ (& -6.774576, & -0.491391 &), \\ (& -6.774576, & 5.791794 &), \\ (& -0.491391, & -6.774576 &), \\ (& -0.491391, & -0.491391 &), \\ (& -0.491391, & 5.791794 &), \\ (& 5.791794, & -6.774576 &), \\ (& 5.791794, & -0.491391 &), \\ (& 5.791794, & 5.791794 &) \end{array} \right\}$$

$n_{loc} = 400$

A.13 Hansen function

$n = 2$, $X = [-10, 10]^n$

$$f(x) = \sum_{i=1}^{5} i\cos((i-1)x_1 + i)\sum_{j=1}^{5} j\cos((j+1)x_2 + j)$$

$$\begin{aligned} G_1(x) &= -\sum_{i=2}^{5} i(i-1)\sin((i-1)x_1 + i)\sum_{j=1}^{5} j\cos((j+1)x_2 + j) \\ G_2(x) &= -\sum_{i=1}^{5} i\cos((i-1)x_1 + i)\sum_{j=1}^{5} j(j+1)\cos((j+1)x_2 + j) \end{aligned}$$

$f^* = -176.541793$

$$X^* = \left\{ \begin{array}{lrrl} (& -7.589893, & -7.708314 &), \\ (& -7.589893, & -1.425128 &), \\ (& -7.589893, & 4.858057 &), \\ (& -1.306708, & -7.708314 &), \\ (& -1.306708, & -1.425128 &), \\ (& -1.306708, & 4.858057 &), \\ (& 4.976478, & -7.708314 &), \\ (& 4.976478, & -1.425128 &), \\ (& 4.976478, & 4.858057 &) \end{array} \right\}$$

$n_{loc} = 760$

A.14 Cola function

$n = 17$, $U = ([0,4], [-4,4]^{n-1})$

$$x_1 = y_1 = y_2 = 0, x_2 = u_1, x_i = u_{2(i-2)}, y_i = u_{2(i-2)+1}$$

$$f(x,y) = \sum_{j<i} (r_{i,j} - d_{i,j})^2$$

$$d = \left\{ \begin{array}{cccccccccc} \ldots \\ 1.27 & \ldots \\ 1.69 & 1.43 & \ldots \\ 2.04 & 2.35 & 2.43 & \ldots \\ 3.09 & 3.18 & 3.26 & 2.85 & \ldots \\ 3.20 & 3.22 & 3.27 & 2.88 & 1.55 & \ldots \\ 2.86 & 2.56 & 2.58 & 2.59 & 3.12 & 3.06 & \ldots \\ 3.17 & 3.18 & 3.18 & 3.12 & 1.31 & 1.64 & 3.00 & \ldots \\ 3.21 & 3.18 & 3.18 & 3.17 & 1.70 & 1.36 & 2.95 & 1.32 & \ldots \\ 2.38 & 2.31 & 2.42 & 1.94 & 2.85 & 2.81 & 2.56 & 2.91 & 2.97 & \ldots \end{array} \right\}$$

$$r_{i,j} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{\frac{1}{2}}$$

$f^* = 11.7464$

$$\begin{aligned} U^* = \; & (0.651906, 1.30194, 0.099242, -0.883791, -0.8796, \\ & 0.204651, -3.28414, 0.851188, -3.46245, 2.53245, -0.895246, \\ & 1.40992, -3.07367, 1.96257, -2.97872, -0.807849, -1.68978) \end{aligned}$$

A.15 Growth problem

This problem represents a growth model of the human mandible (the lower jar). The data of the problem are the coordinates of 271 points of equivalent morphology from 3 mandibles of the same patient at the age of 9 months, 21 month and 7 years [1]. The goal of the problem is to position the three mandibles in the space so that the squared sum of distances of points from the middle mandible to the lines connecting corresponding points from the first and third mandible is minimal. The middle mandible is fixed and each of the two others has 6 degrees of freedom: 3 angles of rotation and 3 directions of translation. Thus the dimension of the problem is $n = 12$. The domain is $X = [-\pi, \pi]^6 \times [-120, 120]^6$. The best known function value found previously using a multistart strategy is $f^* = 205.104$. The corresponding point is

$$\begin{aligned} X^* &= \{-0.125288, -0.048084, 0.0683822, -5.28448, 13.5913, 47.4381, \\ &\quad 0.100655, 0.00663149, 0.0861344, 15.1711, -19.2757, -44.7489\} \end{aligned}$$

The numbers of local and global minimizers are not known, but more than 15000 have been found. A precise description of the objective function can be found in [4].

References

[1] P.R. Andersen, M. Nielsen, and S. Kreiborg. *4D Shape - preserving modelling of bone growth.* Medical Image Computing and Computer-Assisted Intervention - MICCAI'98, Lecture Notes in Computer Science, Vol 1496, pp 710-719, Springer Verlag 1998.

[2] S. Gudmundsson. *Parallel Global Optimization.* M.Sc. Thesis, IMM, Technical University of Denmark, 1998.

[3] P. van Hentenryck. *Numerica: a modeling language for global optimization.* MIT Press, 1997.

[4] http://www.imm.dtu.dk/~km/GlobOpt/testex/

[5] C. Jansson and O. Knüppel. *A global minimization method: The multidimensional case.* Report 92.1, Technical University Hamburg-Harburg, 1992.

[6] C. Jansson and O. Knüppel. *A Branch and Bound Algorithm for Bound Constrained Optimization Problems without Derivatives.* Journal of Global Optimization **7**, 297-331, 1995.

[7] O. Knuppel. *A PROFIL/BIAS inplementation of a global minimization algorithm.* Report 95.4, Technical University Hamburg-Harburg, 1995.

[8] K. Madsen and S. Zertchaninov. *Branch-and-Bound for Global Optimization.* IMM-REP-1998-05, Department of Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark, 1998.

[9] K. Madsen, S. Zertchaninov, A. Žilinskas and J. Žilinskas. *A global optimization using branch-and-bound.* Submited to the Journal of Global Optimization, 1999.

[10] R. Mathar and A. Žilinskas. *A class of test functions for global optimization.* Journal of Global Optimization **5**, 195-200, 1994.

[11] S. Zertchaninov and K. Madsen. *A C++ Programme for Global Optimization.* IMM-REP-1998-04, Department of Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark, 1998.