Finding all the Minimizers of Highly Multimodal Functions by Using a Monte-Carlo Method

ANDRE A. KELLER
Laboratoire d'Informatique Fondamentale de Lille / Equipe SMAC
UMR8022 CNRS/Université de Lille 1
59655 Villeneuve d'Ascq Cedex FRANCE
andre.keller@univ-lille1.fr - http://www.univ-lille1.fr – www.lifl.fr

Abstract: - In many test functions for solving continuous optimization problems, we may have a large number of local minimizers with at least one global minimizer. This situation increases the difficult challenge for finding an optimum solution and choosing one of them. Moreover, some minimizers with less performance may be preferred in the real-world problems. In fact, other qualitative criteria may be used in the decision process, such as cost or computational aspects. The purpose of this study is the location of the global minimizers and the exploration of their neighborhood. The stochastic search iterative procedure refers to the Monte-Carlo methods In this study, it consists in partitioning the rectangular search space in small spaces. In each space, a starting point is selected at random. Local minimizers are obtained by using these starting points. Thereafter, these minimizers are sorted with respect to their performances. This approach is focused on the two-dimensional Shubert's test function I. The computations are carried out by using the software *Wolfram Mathematica* ® 7.

Keywords: - Monte-Carlo method, Shubert's test function, minimizer, local search, global minimum, random starting point.

1. Introduction

This paper is a practical approach of optimization techniques [3]. The computations use the software package Wolfram *MATHEMATICA*® 7 [11]. The technique consists in partitioning the search domain of a function, for which optimum solutions are searched. A grid with a mesh is used. The problem is to find the best approximations of the exact global minimum solutions. The multimodal trigonometric polynomial Shubert test function is chosen [1] as the main application for this study.

2. Random Search Approach

The scatter search methods[2] refer to a class of evolutionary methods that make use of randomization, for searching global optimal solutions [4]. This approach belongs to the first type of scatter methods[3] in [6] that is a diversification generation method. A set of trial solutions is generated at random and then used as initial points in the optimization process.

2.1 Principle

A stochastic search procedure is proposed in this study for finding all the minimizers. It consists firstly in partitioning the rectangular search space. Thereafter, a starting point is selected at random in each sub-space. Then, a local minimizer is searched in this area and neighboring areas. In the following, all the minimizers are sorted with respect to their objective value. The doubles are eliminated while a number of local minimizers is obtained.

Figure 1 shows two techniques for selecting random points in the search domain. The first technique consists in drawing one point at random within a given arbitrary small sub-region. In the second technique, the same number of points is drawn in the whole feasible region of the optimization problem. A more homogenous distribution of the points is obtained with the first technique. In fact, several empty spaces (highlighted spaces in Figure 1) occur when using the second technique.

---

[1] Other multimodal two-dimensional functions for unconstrained global optimization with box constraints are in [1], [4], [7]. The 40 test problems by [8] have been coded in the Mathematica package:

'' Optimization`UnconstrainedProblems` package''.

[2] The origin of the scatter search is due to Glover [2].

[3] The five scatter methods by [6] are: a diversification generation method (1), an improvement method (2), a reference set updating method (3), a reference subset generation method (4) and a combination method (5).

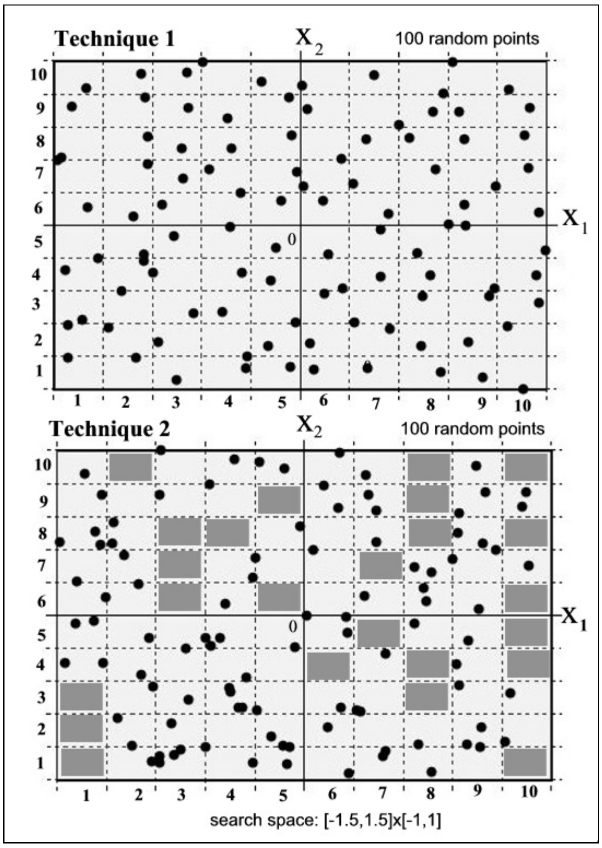

Figure 1: Two techniques to generate a random population of points in a given feasible region.

2.2 Bohachevsky Test Function

The Bohachevsky two-dimensional test-function [9], pp.79-80 yields a unique global minimum solution, and multiple, regular spaced, local optima. The function is

$$f(\mathbf{x}) = 0.7 + x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2),$$

where $\mathbf{x} \in [-1.5, 1.5] \times [-1, 1]$. The surface and the contours of this function are pictured in Figure 2. A hundred of points have been drawn at random, by using the first technique. Straight lines connect the initial random point that leads to the unique global minimum at $\mathbf{x} = (0,0)$. At this point, the function value is zero.

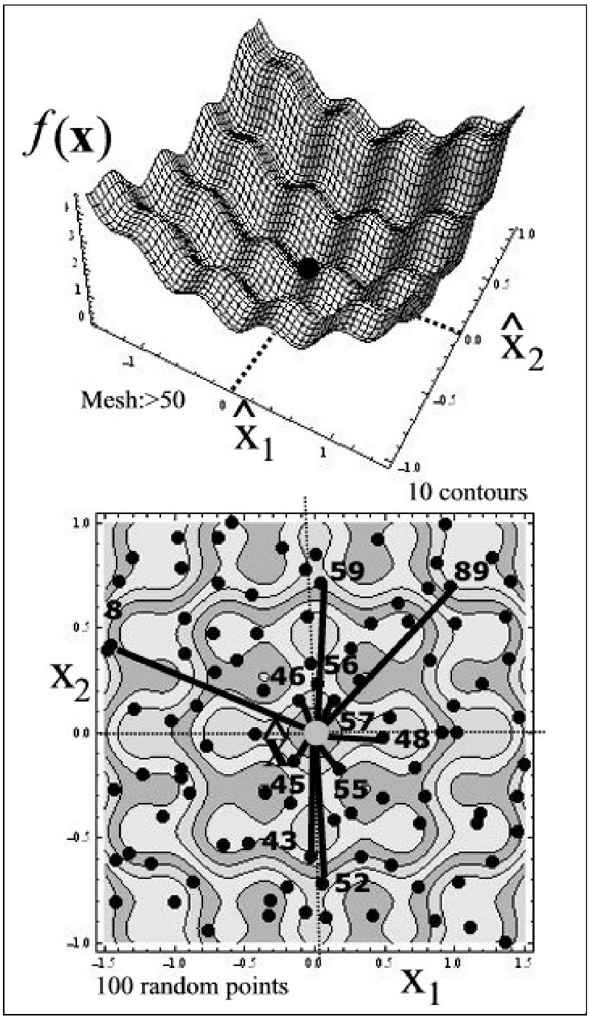

Figure 2: Surface and contours of the Bohachevsky test-function.

3. Multimodal Bivariate Test Function

Test functions are commonly used to evaluate the performance of search algorithms in [8],[9]. The Shubert test function I is chosen for this experiment.

3.1 Shubert Test Function I

The two-dimensional Shubert test function I for unconstrained global optimization is the trigonometric polynomial [5], [10]:

$$f(\mathbf{x}) = \prod_{i=1}^{2} \sum_{j=1}^{5} j\cos((j+1)x_i + j), \qquad (1)$$

For which the search domain is $\mathbf{x} \in [-10, 10]^2$. The surface of the Shubert test function I is pictured in Figure 3, and the contours are shown in Figure 5[4].

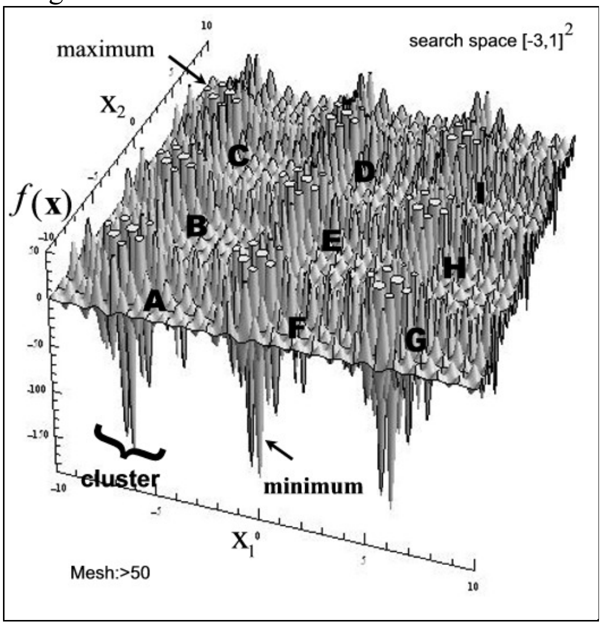

Figure 3: Clustered surface of the Shubert test function I.

3.2 Searching the Global Minimizers

The Figure 3 pictures nine clusters named ‘’A’’ to ‘’I’’, within a 3×3 square. In each cluster, we can find two close global minimum points, as in Figure 4. Therefore, there are 18 global minimizers in Table 1, for which the function value equals $f(\hat{\mathbf{x}}) = -186.731$ for all.

Let the global optima of clusters A and B be denoted by $\mathbf{x}^1_{\mathbf{A}}$, $\mathbf{x}^2_{\mathbf{A}}$ and $\mathbf{x}^1_{\mathbf{B}}$. The Euclidean distance between global minimizers are $\left\|\mathbf{x}^1_{\mathbf{A}} - \mathbf{x}^2_{\mathbf{A}}\right\| = 0.8836$ within one cluster, and $\left\|\mathbf{x}^1_{\mathbf{A}} - \mathbf{x}^1_{\mathbf{B}}\right\| = 6.2832$ between two neighboring clusters. Moreover, there are 742 exact local minimizers.

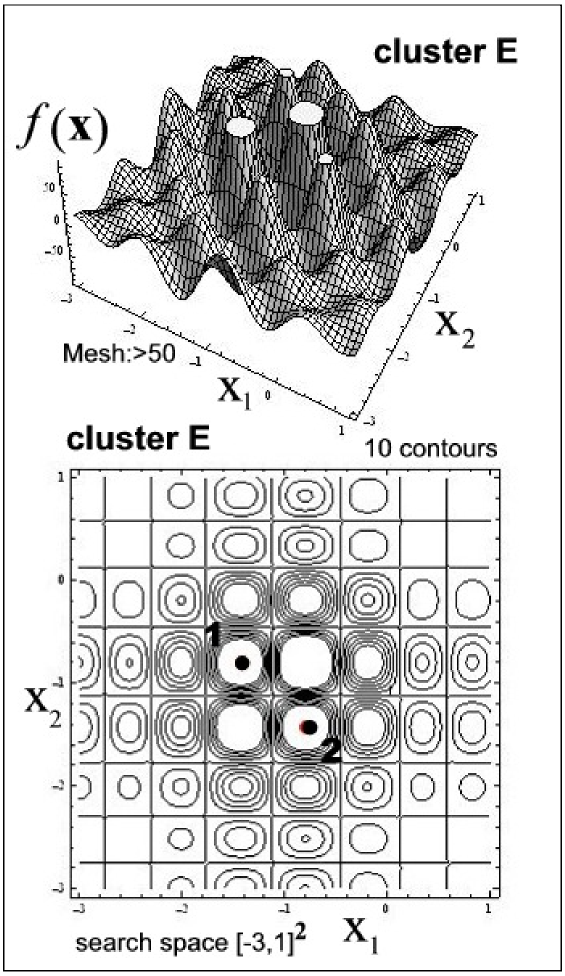

Figure 4: Surface and contours of cluster E.

Table 1: List of the 18 global minimizers.

Cluster	$\hat{\mathbf{x}}^1$		$\hat{\mathbf{x}}^2$		$f(\hat{\mathbf{x}})$
	x_1	x_2	x_1	x_2	
A	-7.7	-7.1	-7.1	-7.7	-186.7
B	-7.7	-0.8	-7.1	-1.4	-186.7
C	-7.7	5.5	-7.1	4.9	-186.7
D	-1.4	5.5	-0.8	4.9	-186.7
E	-1.4	-0.8	-0.8	-1.4	-186.7
F	-1.4	-7.1	-0.8	-7.7	-186.7
G	4.9	5.5	5.5	4.9	-186.7
H	4.9	-0.8	5.5	-1.4	-186.7
I	4.9	-7.1	5.5	-7.7	-186.7

[4] The surface of the Shubert function II with curvature control is pictured in the animated Figure 12

3.3 Search for All the Minimizers

At this stage, we are searching for all the global and local minimizers, which exact number is 760[5]. The search domain $\left[-10,10\right]^2$ is divided into 2500 squares, as in Figure 5. Inside each square (borders excluded), one point is drawn at random. Figure 5 pictures a population of random points.

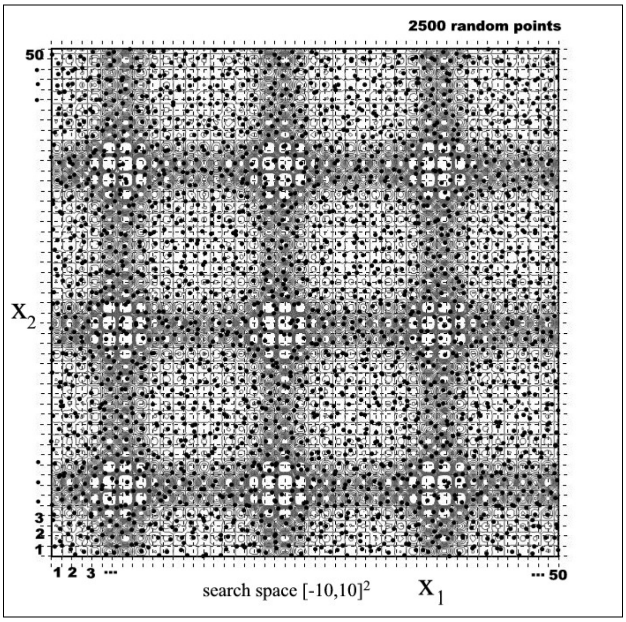

Figure 5: Generation of random initial points.

Each random point is used as an initial point of a search for a local minimum solution. Extending the calculation to all points of the population of the 2500 points yields 716 local and global minimizers (760 existing minimizers). All of the 18 global minimizers have been found. The distribution of the minimizers with respect to the function values is pictured in Figure 6.

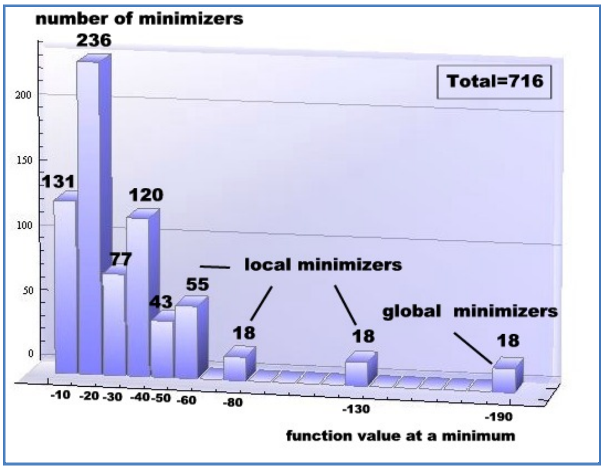

Figure 6: Estimated distribution of all the minimizers.

This process may be applied several times for different mesh accuracies, to get empirical distributions of all the results.

3.4 Local Search within a Neighborhood

Now, we are exploring the neighborhood of the global minimum $\hat{\mathbf{x}}^1$ of the cluster E. A population of numbered random points is generated, with the limitation to stay on or within a circle of unit radius centered at $\hat{\mathbf{x}}^1$. Thereafter, each of these points is used as the initial value of a local search for a minimum solution. The obtained local minima (named "A", "B", "C",…) are new points inside and outside the circular neighborhood. We may observe that the two global minimum solutions have been found. However other local minima are also found within the circular neighborhood and outside. In Figure 7, straight lines connect the initial points to one of the two global optimum points in cluster E.

The interest of this approach is to propose to the decision maker alternate local best solutions that could be preferred, due to the computational costs, or even for other environmental, economic, and social reasons.

[5] We may rewrite (1) as the product of two identical functions for which the argument differs. The graph of this function exhibits 19 maximum solutions for which the function value is positive, and 20 minimum solutions for which the function value is negative (see Figure 10). Then, we can determine the total number of minimizers as $\left(19\times 20\right)\times 2 = 760$.

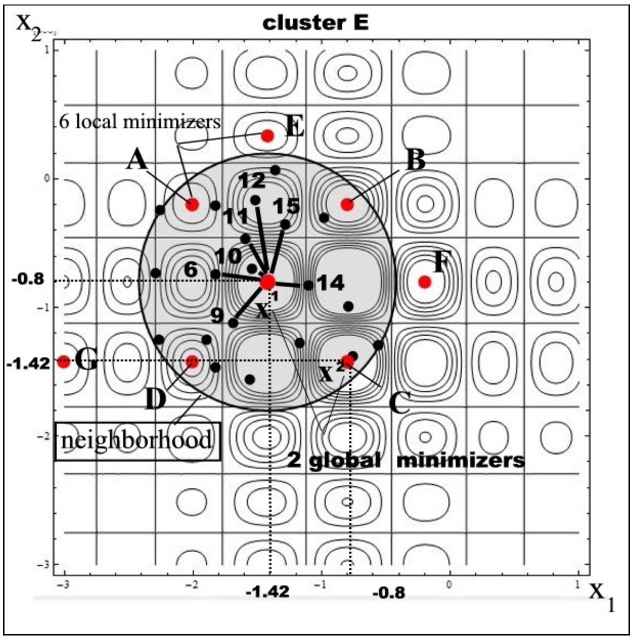

Figure 7: Neighborhood of a global minimum in cluster E.

4. Conclusion

This paper was devoted to the practical search of minimizers for highly multimodal functions. This study is centered on the trigonometric polynomial Shubert test functions I and II for experiments. A practical procedure is proposed to obtain the best technical solution and other comparable solutions (if any). Indeed, these alternative solutions may be of most interest for a decision maker, while considering other qualitative aspects of the decision making procedure.

The contributions of this study are three-fold. Firstly, a homogenous distribution of random points in the search region is proposed. Secondly, an empirical distribution of the minimizers is presented. Thirdly, an exploration of the neighborhood of a global minimum solution is effected to obtain other alternative solutions to the best technical solution.

Acknowledgment: I would like to thank two anonymous referees for their stimulating evaluations.

Appendix A. Random Search for Global Optimization in Mathematica ®[6]

In the Mathematica® software [11], global and local optimization problems have specific primitives. The global optimization problems can be solved either by using the primitive 'Minimize[…]' to get an exact solution or with the help of the primitive 'NMinimize[…] for solving this problem numerically. The local optimization problems can be solved by using the primitive 'FindMinimum[…]. This presentation is focused on global optimization methods and on the random search approach.

A.1. The 'NMinimize' Function

The NMinimize *Mathematica* function is: NMinimize[{function, constraints}, {variables}].

Example A.1. Let the unconstrained boxed variables minimization problem, which bivariate objective function (Figure 8) is

$$f(\mathbf{x}) := \sum_{i=1}^{2}\left(\left(x_i - 2\pi\right)^2 + 20\sin\left(\frac{\pi}{2}\left(x_i - 2\pi\right)\right)\right),$$

where $\mathbf{x} \equiv (x_1, x_2)$ and $\mathbf{x} \in [0,10]^2$.

The Mathematica primitive with the default global optimization method[7] is: NMinimize[f[x1,x2],{x1,0,10},{x2,0,10}]. The result is: {-22.703,{x1→6.14711, x2→6.14711}}.

[6] This appendix is inspired from the documentation provided by the Mathematica online help : ''Numerical Nonlinear Global Optimization'' available at http://reference.wolfram.com.

[7] The default method depends on the type of optimization problem. For numeric functions, the Nelder-Mead simplex algorithm is first used.

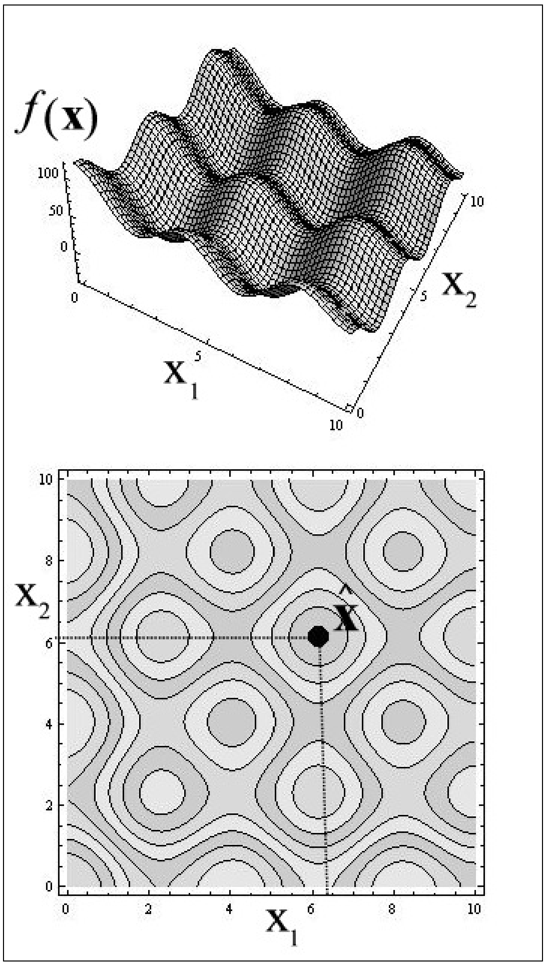

Figure 8: Surface and contours of **Example A.1**.

An initial rectangular region is needed for NMinimize to start. It is specified by using finite lower and upper bounds for each variable. This is done for each variable either by defining $\{x,a,b\}$ with $a<b$ or by inserting the constraint $a \le x \le b$. The default initial region is $-1 \le x \le 1$.

A.2. Global Optimization Methods

The alternative algorithms in *Mathematica* are “DifferentialEvolution” (DE), “NelderMead” (NM), “SimulatedAnnealing” (SA) and “RandomSearch” (RS). Then, a *Mathematica* primitive using the random search technique is NMinimize[f[x1,x2],{x1,0,10},{x2,0,10}, Method →’’RandomSearch’’]. The results on line RS of Table 2 yield a global optimum.

Table 2: Global optimization algorithms.

Method	x_1	x_2	$f(\mathbf{x})$	Optimum
DE	5.322	5.3222	**-38.078**	global
NM	9.166	9.1659	-22.703	local
SA	5.322	9.1657	-30.391	local
RS	5.322	5.3222	**-38.078**	global

A.3. The ‘RandomSearch’ Method

A population of initial points is generated randomly by using the random search algorithm. Thereafter, a local minimum is searched for each of the initial points[8] by using the Mathematica primitive ‘FindMinimum[…]’[9]. The solution is the best local minimum.

For this example, the random initial points have generated by using the primitive[10] Random[Real,{a,b}], where a<b. Twenty points have been generated for this application and placed into a file, named ‘data’. The local minimizations for all of the 20 initial points, use the primitive: Do[[FindMinimum[f[x1,x2],{x1,data[[i,1]]},{x2,data[[I,2]]}],{i,20}]. Figure 9 illustrates the position of the random initial points, that of the unique global, and of the eight local minimizers. A straight line relates the initial random points that reach the global optimum.

[8] The default number of initial points is $\min\{10n,100\}$, i.e. 20 points for a bivariate function for which $n=2$. A population of 100 points can be obtained by changing the Method option, as follows: Method → {’’RandomSearch’’,’’RandomPoints’’ → 100}.

[9] The possible settings for the Method option are : ‘’ ConjugateGradient’’, ‘’PrincipalAxis’’, ‘’LevenbergMarquardt’, ‘’Newton ‘’, ’QuasiNewton’’, ‘’InteriorPoint’’ and ‘’ LinearProgramming’’.

[10] The primitive Random[type,range] gives a pseudorandom number of the type integer, real or complex in the specified range.

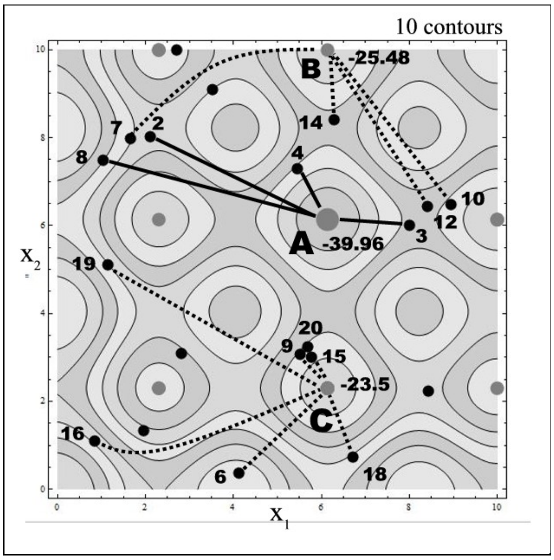

Figure 9: Random initial points and all minima.

Appendix B. All the Minimizers of Shubert Test Function I

We can rewrite $f(\mathbf{x})$ in (1) as:

$$f(\mathbf{x}) = u(x_1) \times u(x_2),$$

where

$$u(x_i) = \sum_{j=1}^{5} j \cos((j+1)x_i + j), \ i = 1, 2.$$

Figure 10 pictures the component function $u(x)$. The maximum solutions and the minimum solutions are in Table 3. Since the $20 - 1 = 19$ maximum solutions are characterized by positive function values and the 20 minimum solutions by negative function values of $u(x)$, we can easily deduce that the exact number of minimizers of $f(\mathbf{x})$ will be $((20-1) \times 20) \times 2 = 760$. The 760 minimizers are plotted in Figure 11: small size points are the 742 local minimizers and large size points are the 18 global minimizers.

Table 3: Optimum solutions of $u(x)$

	minimum		maximum	
	x	$u(x)$	x	$u(x)$
1	-9.7804	-2.6396	-10.0	-0.2583
2	-8.7941	-3.5877	-9.2863	3.6137
3	-7.7083	-12.8709	-8.2904	6.1698
4	-6.4786	-8.5178	-7.0835	14.508
5	-5.4614	-3.750	-5.9489	3.8473
6	-4.4775	-2.7288	-4.9632	2.9276
7	-3.4973	-2.6396	-3.9840	2.9257
8	-2.5109	-3.5877	-3.0032	3.6137
9	-1.4251	-12.8709	-2.0072	6.1698
10	-0.1954	-8.5178	-0.8003	14.5080
11	0.8218	-3.750	0.3342	3.8473
12	1.8057	-2.7288	1.3200	2.9276
13	2.7859	-2.6396	2.2992	2.9257
14	3.7723	-3.5877	3.2800	3.6137
15	4.8581	-12.8709	4.2760	6.1698
16	6.0878	-8.5178	5.4829	14.5080
17	7.1050	-3.7500	6.6174	3.8473
18	8.0888	-2.7288	7.6032	2.9276
19	9.0691	-2.6396	8.5824	2.9257
20	10.000	-3.3435	9.5632	3.6137

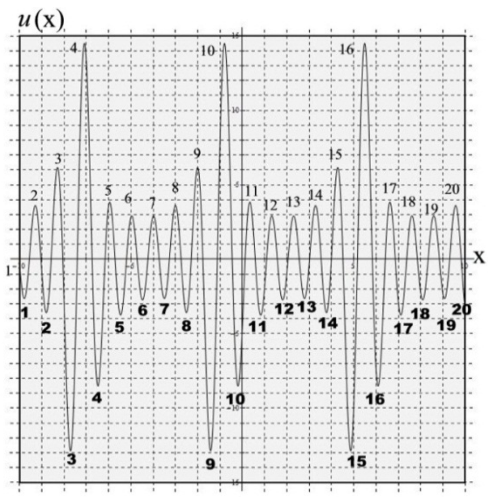

Figure 10: Component function $u(x)$

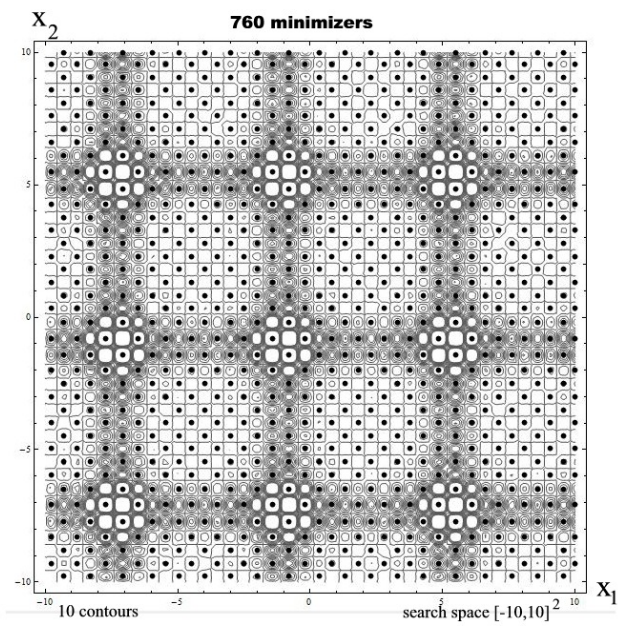

Figure 11: All minimizers of the Shubert's test function I.

Appendix C. Shubert Test Function II

The two-dimensional Shubert function II is

$$f(\mathbf{x}) = \prod_{i=1}^{2} \sum_{j=1}^{5} j \cos\left(\left(j+1\right) x_i + j\right) + \beta\left(\left(x_1 + 1.4251\right)^2 + \left(x_2 + 0.80032\right)^2\right)$$

where β denotes the curvature control. One animation[11] for this function is pictured in Figure 12.

References:

[1] Adorio, E. P. (2005). *MVF- Multivariate test function library in C for unconstrained global optimization.* Retrieved from http://extreme.adorio-research.org/download/mvf/html/mvf.html.

[2] Glover, F. (1998). A template for scatter search and path relinking. In J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer, & D. Smyers (Eds.), *Artificial Evolution, Lecture Notes in Computer Science* (Vol. **1363**, pp. 13-54). Berlin: Springer.

[3] Keller, A. A. (2013). *Nonconvex Optimization in Practice: Theory, Algorithms and Applications with Mathematica* (forthcoming). Athens, Greece: WSEAS Press.

[4] Laguna, M., & Marti, R. (2005). Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization, **33*** (2), pp. 235-255.

[5] Levy, A. V., & Montalvo, A. (1985). The tunneling algorithm for the global minimization of functions. *SIAM Journal on Scientific and Statistical Computing, **6*** (1), pp. 15-29.

[6] Marti, R., Laguna, M., & Glover, F. (2006). Principles of scatter search. *European Journal of Operational Research, **169***, pp. 359-372.

[7] Molga, M., & Smutnicki, C. (2005). *Test functions for optimization needs.* Retrieved from http//www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.

[8] More, J. J., Garbow, B. S., & Hillstrom, K. E. (1981). Testing unconstrained optimization software. *ACM Transactions on Mathematical Software, 7*(1), pp. 17-41.

[9] Rönkkönen, J. (2009). *Continuous Multimodal Global Optimization with Differential Evolution-Based Method.* Lappeenranta University of Technology, Lappeenranta, Finland.

[10] Shubert, B. 0. (1972). A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis, **9*** (8), pp. 379-388.

[11] Wolfram, S. (2003). *The Mathematica Book* (5th ed.). Champaign, IL: Wolfram Media.

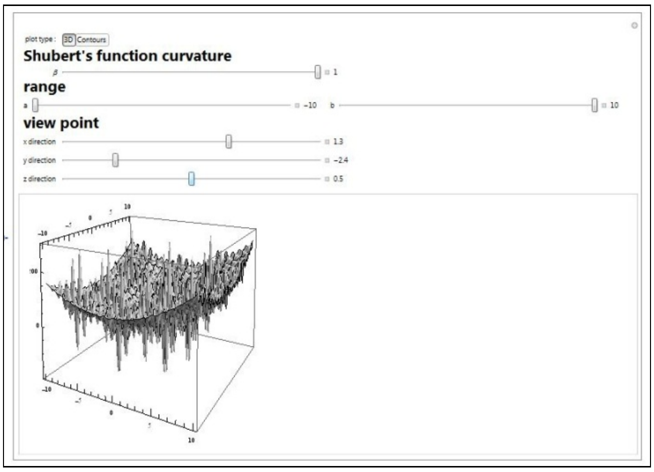

Figure 12: Shubert test function II.

[11] The Mathematica primitive Manipulate[…] creates an interactive object (see Figure 12) containing controls (slides) for different parameters, such as the curvature β, the type of plot (3D surface or 2D contours), the size of the search domain and the viewpoint coordinates.