3rd International Conference on Experiments/Process/System Modeling/Simulation & Optimization
3rd IC-EpsMsO
Athens, 8-11 July, 2009

MULTISTART OPTIMIZATION WITH A TRAINABLE DECISION MAKER FOR AVOIDING HIGH-VALUED LOCAL MINIMA

N. Kyrgios[1], C. Voglis[1] and I.E. Lagaris[1]

[1]University of Ioannina, Dept. of Computer Science,
P.O. BOX 1186, 45110 Ioannina, Greece

Keywords: Global optimization, Multistart, Neural Networks, Molecular Conformation.

Abstract. *In this article we propose a time-saving technique to be used in conjunction with a multistart-based global optimization method, for determining low-valued local minima. The main idea is to avoid the local-search commencement from non-promising points. The decision for the start-point suitability turns out to be rather inexpensive when compared to the cost of a local-search. We employ a feedforward neural-network for the decision making that is fed with functional and gradient information obtained from a few selected points in the neighborhood of the candidate start-point. The network is trained from data collected during the optimization process. We report results for a number of computational experiments on a multitude of model test-functions, using multistart and a special local search that creates contiguous regions of attraction. This method can be particularly useful for the conformation problem in molecular mechanics.*

1. INTRODUCTION

Global optimization (GO) has received a lot of attention in recent years [1], due to the ever emerging scientific and industrial demand. For instance the collection of the stable conformations of a molecule, the management of mutual funds, engineering design and the design of drugs, to mention a few topics, are in need of efficient global optimization techniques.

There exist several categories of GO methods. We distinguish two main classes; the deterministic and the stochastic class and refer to [2] for a detailed account on classification. GO methods face various goals; some aim to find a single global minimum (Simulated Annealing, Genetic Algorithms, Controlled Random Search), others to find all the global minima (Modified Particle Swarm [3]), while others (Multistart with Clustering [4,5,6,7,8,9]) aim in finding all the local minima. Nowadays, with the availability of powerful computer systems, GO has become an affordable procedure. GO algorithms that can take advantage of parallel and/or distributed architectures, are particularly suitable for solving demanding problems. Among the plethora of such problems, we distinguish the determination of the stable conformations of a molecule, considered by "Molecular Mechanics" (MM), due to the far–reaching consequences of its solution. MM is employed to study molecular properties that are important in pharmacology (drug–design), bio–sciences, materials science, etc. Given a realistic interaction between the constituting atoms, MM aims to locate the minima of the molecular potential energy. When the molecule is small, all the local minima are rather easily determined. However, for extended molecules the number of minima may be notoriously high. In such cases the analysis of the molecular properties is quite involved, and the requirement is lowered to the determination of the global minimum and of a limited number of local minima with energy values below an appropriate threshold.

Mathematically the problem we are interested in may be expressed as:

$$\begin{gathered} \text{Find all } \underset{i}{x^*} \in S \subset R^n \text{ that satisfy :} \\ f(x_i^*) \le f_g + \Delta \\ x_i^* = \arg\min_{x \in S_i} f(x), \;\; S_i = S \cap \{x, | \, x - x_i^* \, | < \varepsilon \} \end{gathered} \tag{1}$$

where S is a bounded domain of finite measure, Δ a problem specific positive constant and f_g the value of the objective at the global minimum. Namely the problem is to determine all local minimizers in S with objective values not higher than $f_g + \Delta$.

The article is organized in the following way. In section (2), we lay-out the new ideas involved and we present the corresponding algorithm, while in section (3), we give a description of the numerical experiments that were performed along with the corresponding results. Finally in section (4), our conclusions are summarized and we give a recommendation for future research.

2. DESCRIPTION OF THE METHOD

In the following it will be assumed that the underlying GO method to be used is *"Multistart"*. Any of the better performing multistart–based clustering methods may be used with advantage. Here the emphasis will be given to the new idea of the timely start–point rejection, while keeping the GO procedure simple. We first outline the framework of the new procedure.

1. Pick at random a point $x_s^{(0)} \in S$. Apply only a few (say k) steps of a local search procedure, passing through points $x_s^{(i)}, i = 1,\ldots,k$. Let $f_s^{(i)} = f(x_s^{(i)})$ and $g_s^{(i)} = \nabla f(x_s^{(i)})$.
2. From this information, i.e. $\{f_s^{(i)}\}$, and $\{g_s^{(i)}\}, i = 0,1,\ldots,k$ predict f_s^*, the value of the objective function at the minimum that would be recovered if the local search was allowed to converge.
3. *If the prediction is higher than a preset threshold:* abandon the search and start over again from step *otherwise:* continue with the local search until a minimum is recovered.
4. Repeat from step 1.

Step 2 needs further description. The prediction of the objective value at the minimum is based on the following model

$$f(x_S^*) \approx f_M(p;Y_S) = f(x_S^{(0)}) - N(p;Y_S) \tag{2}$$

where $x_s^* = L(x_s^{(0)})$ is the minimum reached by starting local search L from point $x_s^{(0)}$. $N(p;Y_s)$ is a feedforward neural network with one hidden layer and p is the set of the network's weights and biases while Y_s is a set of input data collected during the run. More specifically

$$Y_s = \left\{ f_s^{(0)}, g_s^{(0)}, f_s^{(1)} - f_s^{(0)}, \| g_s^{(1)} \|, f_s^{(2)} - f_s^{(1)}, \| g_s^{(2)} \|, \ldots, f_s^{(k)} - f_s^{(k-1)}, \| g_s^{(k)} \| \right\}$$

Each node in the hidden layer requires $n + 2k + 3$ parameters (weights). Hyperbolic tangent was chosen for the activations in the hidden layer, while the output activation was taken to be linear. (Our implementation uses $k = 2$).

2.1 Network training

The weights are determined by training the model using collected data created during the global optimization procedure. Namely, we collect a number (M) of starting points $x_1^{(0)}, x_2^{(0)}, \ldots, x_M^{(0)}$, and the corresponding local minima $x_1^*, x_2^*, \ldots, x_M^*$ with $x_s^* = L(x_s^{(0)})$. The training set for the network is given by $\{(Y_1,t_1),(Y_2,t_2),\ldots,(Y_M,t_M)\}$ where $t_s = f(x_s^{(0)}) - f(x_s^*)$. The training is performed by minimizing the error function

$$E(p) = \frac{1}{M} \sum_{s=1}^{M} \left(N(p;Y_s) - t_s \right)^2 \tag{3}$$

2.2 Local search properties

For the prediction model $f_M(p;Y_S) = f(x_S^{(0)}) - N(p;Y_S)$ to be accurate the points $x_s^{(i)}, x_s^*$ should be connected via a monotonically decreasing path and even more x_s^* should be the closest minimum to $x_s^{(0)}$ that can be connected with such a path. This ensures the local character of the approximation. Note, that most common local search procedures do not share this property and hence are not suitable in this framework. A method that satisfies the above requirement is a steepest descent with an infinitesimal step. However, this is only a theoretical device and such a method in practice would be wasteful.

In figure 1 we present a univariate example of a multimodal function. Starting points in a "valley" should be associated with the surrounded minimum. In such a case the model has a local character and the approximation therefore is meaningful. To this end we have implemented quasi-Newton (BFGS) local search with a modified line search that maintains intact the Armijo condition. However the line-search uses an increasing step-size contrary to the common backtracking. We give a brief description of the line search in Algorithm 1.

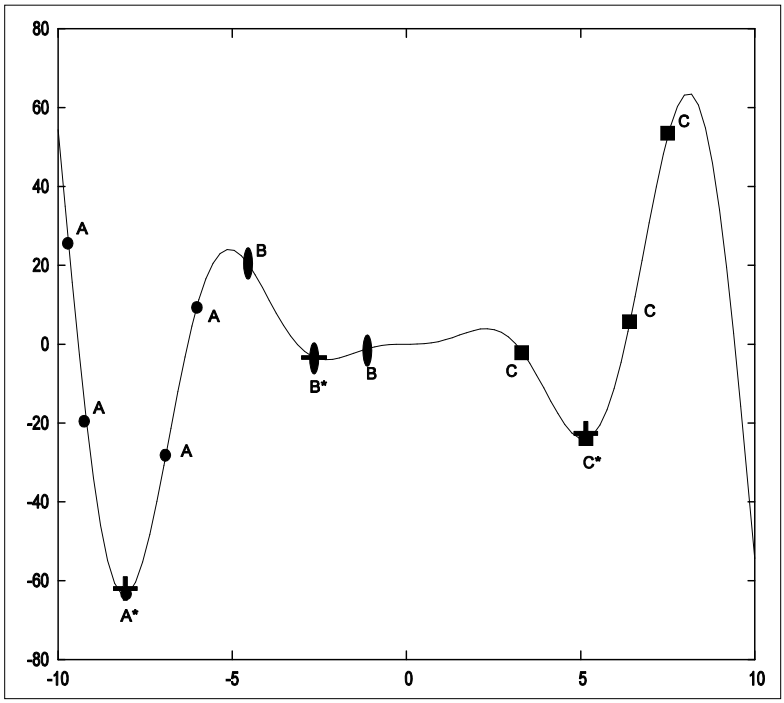


Figure 1. Starting points and associated minima

Algorithm 1 New line search

Input:

x: Current iterate
d: Descent direction from the outer quasi-Newton local search
ρ: Armijo rule parameter
$\mu, \nu > 0$: Method's parameters

Output:

x': Next iterate
α: Line search step
fc: Function calls

1. **Initialize:**
   $scale \leftarrow 1, fc \leftarrow 0, \text{term} \leftarrow$ **false**
2. **Main Step:**
   while term = **false do**
   for i=1, ν **do**
   $\lambda_i \leftarrow scale \cdot \frac{\mu^i - 1}{\mu^\nu - 1} \cdot \min\left(1, \frac{\max(1, \|x\|))}{\|d\|}\right)$
   if $f(x + \lambda_i d) < f(x) + \rho \lambda_i \cdot d^T \nabla f(x)$ **then** *{ Bellow ρ line}*
   if $f(x + \lambda_i d) > f(x + \lambda_{i-1} d)$ **then** *{No improvement}*
   $\alpha \leftarrow \lambda_{i-1}$
   $x' \leftarrow x + \alpha d$
   term $\leftarrow$ **true**, **break**
   end if
   else *{ Above ρ line }*
   $\alpha \leftarrow \lambda_{i-1}$
   $x' \leftarrow x + \alpha d$
   term $\leftarrow$ **true**, **break**
   end if
   $fc \leftarrow fc + 1$
   end
   $scale \leftarrow scale \frac{\mu^i - 1}{\mu^\nu - 1} \cdot \min\left(1, \frac{\max(1, \|x\|))}{\|d\|}\right)$
   end

We mention in passing that in Algorithm 1 the loop over the steps can be performed in parallel.

3. EXPERIMENTS AND COMPARISON

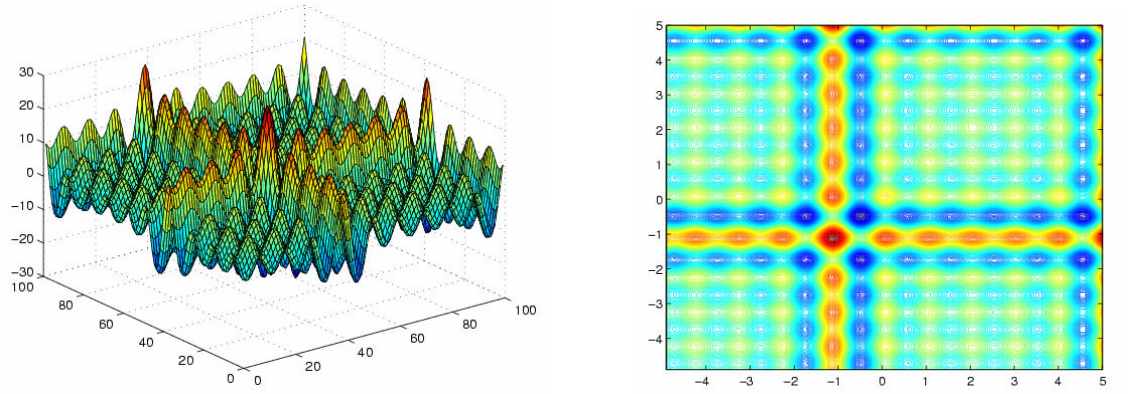
We used Matlab integrated environment to implement our methodology. Neural network's were created and trained using the Neural Network Toolbox, and the training was performed using a Levenberg-Marquard algorithm (+'trainlm'+ option).

3.1 Illustrative example

In this example we used the two-dimensional Shubert function inside $[0,5]^2$ given by:

$$f(x_1,x_2)=\left[\sum_{i=1}^{5} i\cos((i+1)x_1+i)\right]\left[\sum_{i=1}^{5} i\cos((i+1)x_2+i)\right] \tag{4}$$

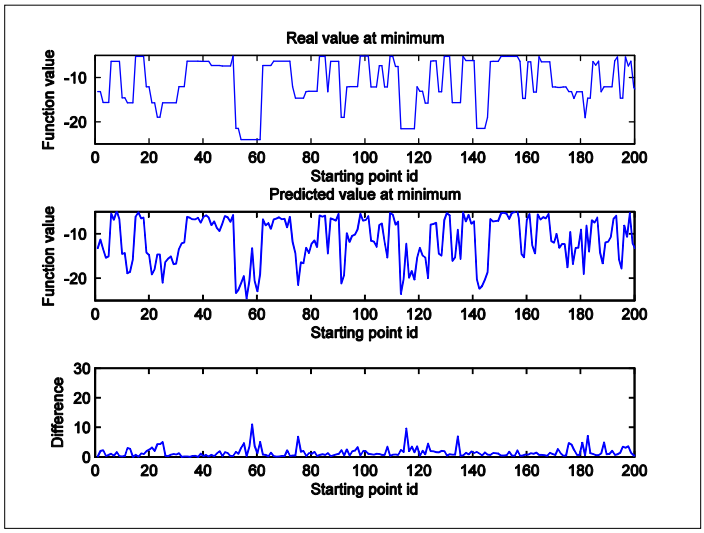
The training set was created by uniformly sampling 200 starting points, and by performing an equal number of local searches to obtain the associated minima, while similarly, the test set used 600 points.
In Figure 2 the surface and contour plot of the Shubert function is displayed.



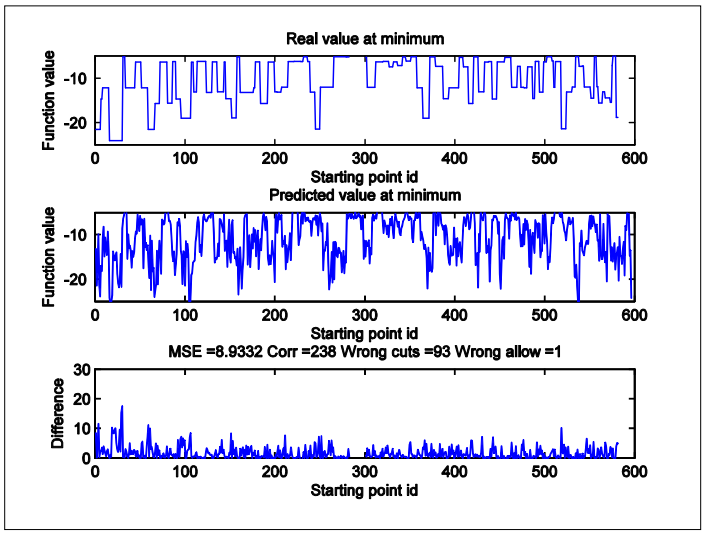
(a) Function's surface plot (b) Function's contour plot

Figure 2. Two dimensional Shubert function

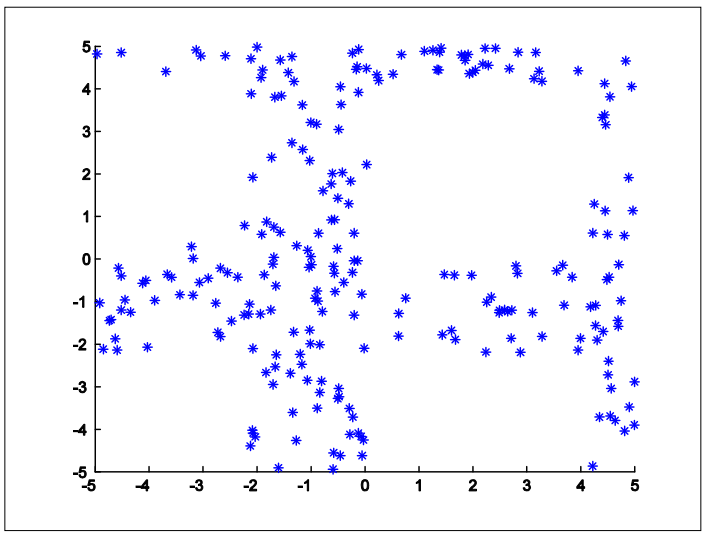
In figure 3(a) the horizontal axes register the starting point indices $i=1,\ldots,200$ used for the training. The vertical axes of the top, middle and bottom row hold the values of the objective at the associated minima x_i^*, the predicted value and their absolute difference correspondingly. Similarly in figure 3(b) the test set plots are given, while in figure 3(c) the accepted starting points are shown. We accepted a starting point $x_s^{(0)}$ when $f_M(p;Y_s)<-12$. There are two cases of misclassification. One, where a point is erroneously accepted, and the other when a point is erroneously ejected. The first case costs a local search, while the second costs only a few evaluations. In figure 3 the results refer to a neural network with 5 hidden nodes. Figure 4 illustrates the case of a 20-hidden nodes neural network. One may verify by inspection that the 20-node network obtains a lower MSE over the training set, and a higher MSE over the test set hindering that the 5-node network offers a better generalization. Namely the 5-node network attains a 84.33% success rate, and the 20-node network a corresponding 82.83%.



(a) Training set

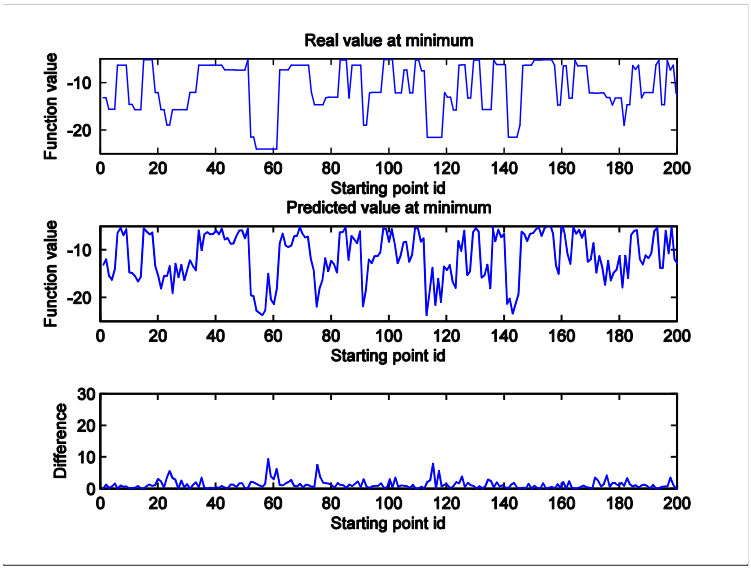


(b) Test set

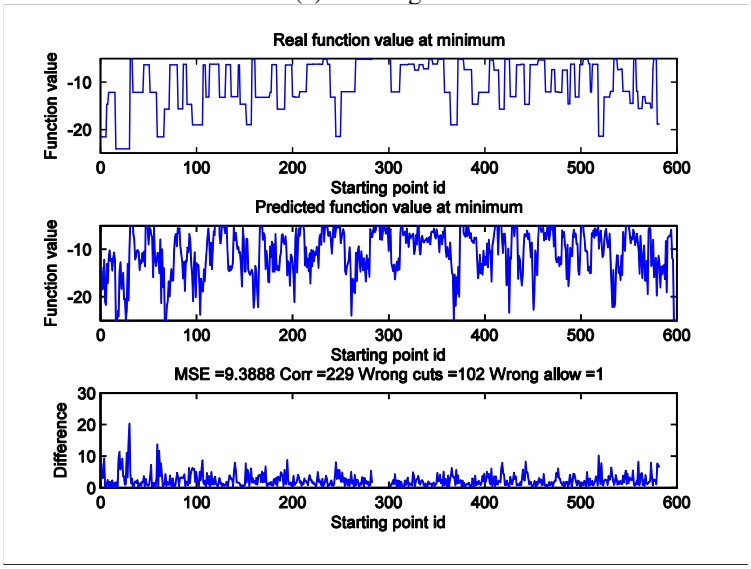


(c) Accepted starting points

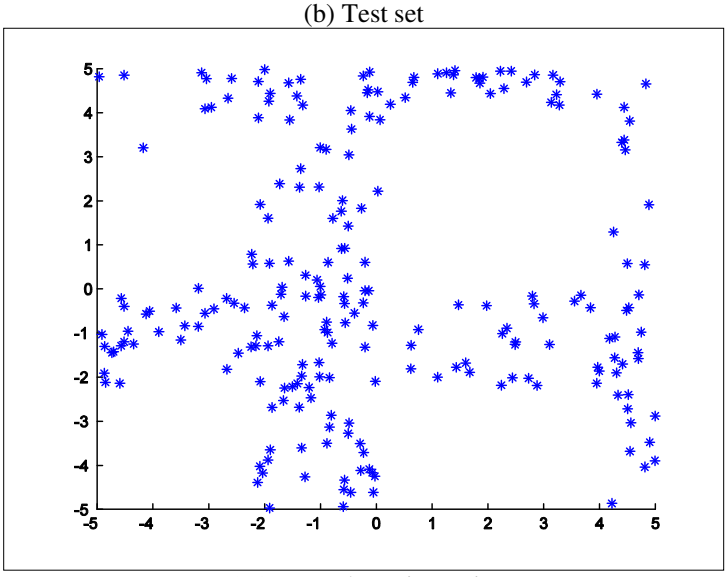
Figure 3. Results for a neural network with 5 hidden nodes



(a) Training set



(b) Test set



(c) Accepted starting points

Figure 4. Results for a neural network with 20 hidden nodes

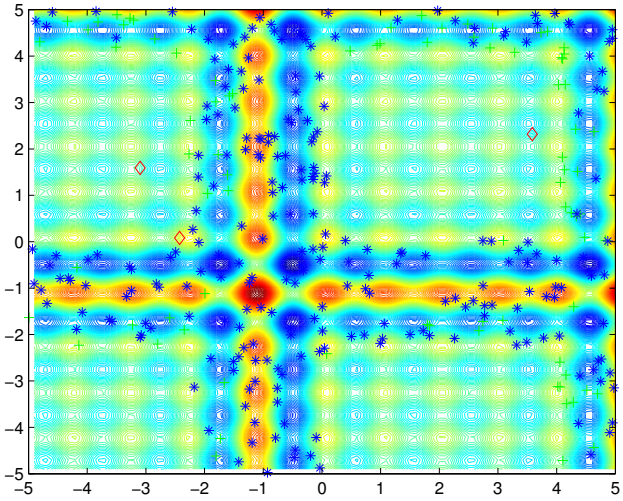


Figure 5. Accepted starting points printed on function's contour

We implemented our approach and tested it on a number of optimization problems. Namely we experimented with well known test-functions such as the Rastrigin, Giunta, Boha, Holder and Bird. Our results were in line with those of the Schubert test function discussed above and will be reported elsewhere.

4. CONCLUSIONS AND FURTHER WORK

In this paper we presented an early rejection criterion suitable for multistart based global optimization algorithms. The observed savings are substantial and hence the method may be suggested for application in time consuming global optimization problems like those appearing in molecular mechanics, where the objective function is the molecular potential energy while the atomic coordinates are the adjustable parameters.
Molecular mechanics problems are currently under intensive investigation by our research group.

REFERENCES

[1] Pardalos Panos M., Romeijn Edwin H., Tuy Hoang (2000), *Recent developments and trends in global optimization*, Journal of Computational and Applied Mathematics, pp. 209-228.
[2] Boender C.G.E. and Romeijn Edwin H. (1995), *Stochastic Methods*, in *Handbook of Global Optimization* (Horst, R. and Pardalos, P. M. eds.), Kluwer, Dordrecht, pp. 829-871.
[3] Parsopoulos, K. E., Vrahatis M. N. (2004), *On the computation of all Global minimizers through particle swarm optimization*, IEEE Trans. Evol. Comp., pp. 211-224.
[4] Boender, C.G.E. and Rinnooy Kan, A.H.G and Timmer, G.T. and Stougie, L., (1982), *A stochastic method for global optimization*, Mathematical Programming, pp. 125-140.
[5] Rinnooy Kan, A.H.G and Timmer, G.T. (1987), *Stochastic global optimization methods. Part I: Clustering methods*, Mathematical Programming, pp. 27-56.
[6] Rinnooy Kan, A.H.G and Timmer, G.T. (1987), *Stochastic global optimization methods. Part II: Multi level methods*, Mathematical Programming, pp. 57-78.
[7] Törn, A. A.(1978) *A search clustering approach to global optimization*, in Dixon, L.C.W and Szegö, G.P. (eds.), *Towards Global Optimization 2*, North-Holland, Amsterdam.
[8] Törn, A. and Viitanen, S. (1994) *Topographical Global Optimization Using Pre–Sampled Points*, Journal of Global Optimization, pp. 267-276.
[9] Ali, M.M. and Storey, C. (1994), *Topographical Multilevel Single Linkage*, Journal of Global Optimization, pp. 349-358.