GLOBAL OPTIMIZATION METHOD FOR DESIGN PROBLEMS

Abdelkhalak El Hami[1] – Seifedine Kadry[2*]

[1]INSA of Rouen, Normandy University, France
[2]Engineering School, American University of the Middle East, Kuwait.

ARTICLE INFO

***Article history*:**
Received: 13.2.2015.
Received in revised form: 24.3.2015.
Accepted: 27.3.2015.

Keywords:
Global optimization
Simulated annealing (SA)
Triplate
Structural optimization
Simultaneous perturbation stochastic approximation (SPSA)

Abstract:

In structural design optimization method, numerical techniques are increasingly used. In typical structural optimization problems there may be many locally minimum configurations. For that reason, the application of a global method, which may escape from the locally minimum points, remains essential. In this paper, a new hybrid simulated annealing algorithm for global optimization with constraints is proposed. We have developed a new algorithm called Adaptive Simulated Annealing Penalty Simultaneous Perturbation Stochastic Approximation algorithm (ASAPSPSA) that uses Adaptive Simulated Annealing algorithm (ASA); ASA is a series of modifications done to the traditional simulated annealing algorithm that gives the global solution of an objective function. In addition, the stochastic method Simultaneous Perturbation Stochastic Approximation (SPSA) for solving unconstrained optimization problems is used to refine the solution. We also propose Penalty SPSA (PSPSA) for solving constrained optimization problems. The constraints are handled using exterior point penalty functions. The hybridization of both techniques ASA and PSPSA provides a powerful hybrid heuristic optimization method; the proposed method is applicable to any problem where the topology of the structure is not fixed; it is simple and capable of handling problems subject to any number of nonlinear constraints. Extensive tests on the ASAPSPSA as a global optimization method are presented; its performance as a viable optimization method is demonstrated by applying it first to a series of benchmark functions with 2 - 50 dimensions and then it is used in structural design to demonstrate its applicability and efficiency.

[*] Corresponding author. Tel.: 0096566610985;
E-mail address: skadry@gmail.com

1 Introduction

The domain of design engineering is in constant progress and the race toward an optimal solution continues in its good train. The increasing need for optimum structural designs with the most efficient use of material without violating constraints has given rise to several developments in the past three decades. A good deal of effort has been centered toward the search of optimal structural designs [1-4]. In many of the previous studies, the authors have used local search algorithms. Such algorithms can only be successful if they are used to improve the current design if only a small segment of the boundary is allowed to move or if the objective function is convex. In fact, such assumptions do not hold, a design engineer working in the field of research and development has often to design completely new structures. The loading and support conditions of a particular design problem are usually known in advance, but the designer is not sure about what the final or optimal structure should look like. The essential goal of a designer in using an optimization algorithm is just to state the boundary conditions and let the algorithm perform some iterations without human intervention until it has automatically produced the best design. In this respect, the previous studies had only a relative success. Many of them relied too much on designer's intuition including the choice of initial design, or tight restrictions imposed on the movements of boundary. For these reasons, global optimization should take part in structural problems. In some of the previous studies, global search algorithms were used, but the accuracy remained questionable. Structural engineering and mathematical programming techniques should, both of them, collaborate in some way to develop a powerful and sophisticated programming system for structural optimization. This includes a robust optimization method coupled with modern tools of computer-aided design.

This paper proposes a new hybrid simulated annealing algorithm for global optimization with constraints. Simulated Annealing (SA) has been developed rapidly since the past 20 years [5] as an effective and simple optimization technique, its superiority being in its good robustness and convenience to realize global optimization, whose conventional optimization techniques can seldom be attained. This algorithm, among few other heuristics, is suitable for complicated problems where global optimum is hidden among many local optima. The idea of the method is an analogy with the way the molten metal is cooled and annealed. For a slowly cooled process, this system is able to find the minimum energy state. Such slow cooling is essential for achieving a low energy state. Although simulated annealing is a global optimization method, it suffers from some disadvantages such as inaccuracy or slow convergence to the global minimum. In order to overcome these concerns, many modifications have been proposed, viz.:

(a) a new stopping rule and

(b) the application of a local search method.

The present article focuses on the enhancement of simulated annealing algorithms by proposing the new hybrid method ASAPSPSA. The method was designed in order to find the absolute minimum of an objective function without being sensitive to the starting point, itis capable of handling problems subject to any number of design variables or equality/inequality constraints. The present method can help researchers and practitioners devise optimal solutions to countless real-world problems. Numerical results demonstrate the efficiency, accuracy and applicability of the suggested method for structural optimization.

2 The basic simulated annealing

The basic simulated annealing algorithm used in this paper is a simulated annealing (Fig. 1) based on the work of Van Laarhoven and Aarts [6], the algorithm is also well described by Sitarz [7]. A brief summary of the algorithm is given below for completeness.
The following is a description of the periods of the algorithm in detail:

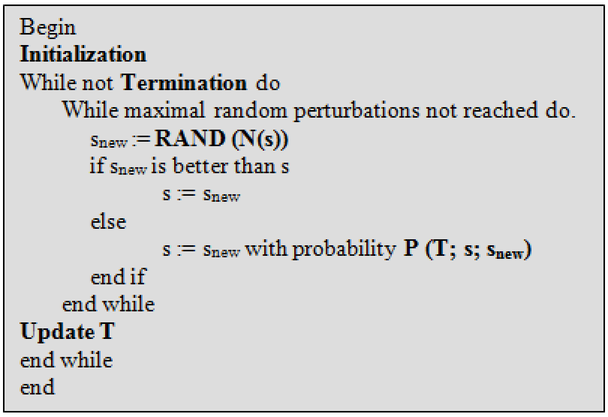
- **Initialization:** Generate initial solution s and initial parameters.

- **Termination:** The algorithm terminates when T reaches the value of the minimal temperature allowed.

- **RAND (N(s)):** Randomly choose a solution from N(s) where N(s) is the neighbourhood of s.

- **P (T; s; s_{new}):** P (T; s; s_{new}) is the acceptance probability defined as:
  P (T; s; s_{new}) $= \exp(-\Delta E / k_b . T)$
  where k_b is the Boltzmann constant. ΔE = E(s_{new}) - E(s), is the amount of increase in the objective value caused by the uphill move and T is a parameter referred to as "annealing temperature".

- **Update T:** Updating Temperature means cooling schedule. T_k temperature in the iteration k fulfils the following conditions:
  $T_k \geq 0 \; and \; \lim_{k \to \infty} T_k = 0$
  The cooling schedule may be classic [8, 11] $T_k = T_0 \; \alpha^k$, Logarithmic [9] $T_k = \mathrm{T}_0 / \log(\mathrm{k}+1)$ and so on. Where α is some constant such that $0 < \alpha < 1$, usually in the range of 0.90 - 0.99. T_0 = 0.01, 1, 100.



```
Begin
Initialization
While not Termination do
    While maximal random perturbations not reached do.
        $s_{new}$ := RAND (N(s))
        if $s_{new}$ is better than s
                s := $s_{new}$
        else
                s := $s_{new}$ with probability P (T; s; $s_{new}$)
        end if
    end while
Update T
end while
end
```

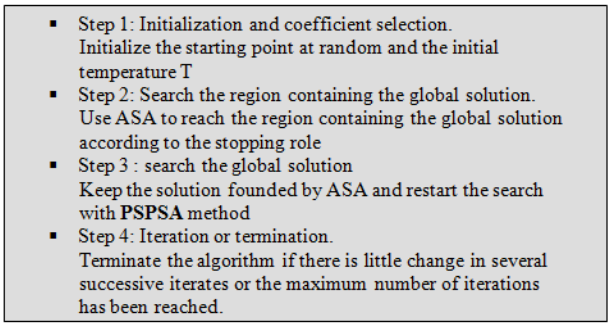
Figure 1. The basic steps of simulated annealing.

BSA Algorithm starts from an initial solution s at a high temperature T, and makes a series of moves according to RAND (N(s)). The change in the objective function values ΔE is computed at each move. If the new solution results in decreased objective function value, it is accepted with probability 1. If the new solution yields increased objective function value, it is accepted with probability P (T; s; s_{new}). To avoid accepting large uphill move in the later stage of the search, the parameter T will be decreased over time by a schedule which is called "the cooling schedule". By accepting worse solutions with a certain probability, SA can avoid being trapped in a local optimum. There are two drawbacks for applying BSA Algorithm directly: one is its slow convergence; the other is its accuracy. To obtain faster convergence and to ensure that the region containing the optimal solution was found, we first accelerate and follow the new stopping rule and in the final stage, we apply PSPSA in the solution found by ASA to improve the final result.

In this paper, a new ASA which is combined with PSPSA (local search) is proposed. From numerical experiments, ASAPSPSA shows its powerful ability of global optima searching and wide applicability.

3 The new proposed approach

Although the BSA method approaches the neighborhood of the global minimum, it has, however, a difficulty in obtaining some required accuracy; it seems judicious to modify the basic algorithm according to a new stopping rule, and then to finish the algorithm with a faster convergent method. According to this idea, we modify BSA method to obtain the ASAPSPSA method as follows:



- Step 1: Initialization and coefficient selection.
  Initialize the starting point at random and the initial temperature T
- Step 2: Search the region containing the global solution.
  Use ASA to reach the region containing the global solution according to the stopping role
- Step 3 : search the global solution
  Keep the solution founded by ASA and restart the search with **PSPSA** method
- Step 4: Iteration or termination.
  Terminate the algorithm if there is little change in several successive iterates or the maximum number of iterations has been reached.

Figure 2. The basic steps of ASAPSPSA.

Commonly, structural problems are constrained. For the reason that SPSA method involves penalizing constraints, a penalty method is used; the constrained problem is then converted into an unconstrained problem which allows us to design a new method for constrained optimization problems, called Penalty Simultaneous Perturbation Stochastic Approximation (PSPSA) method.

3.1 Penalty method

Penalty method is a procedure for approximating constrained optimization problems by the unconstrained ones. The approximation is accomplished in the case of penalty methods by adding to the objective function a term that prescribes a high cost for violation of the constraints.

Consider the problem

$$\begin{cases} Minimize \quad f(x), \quad x \in IR^n \\ \quad Subject\ to : \ x \in S \end{cases} \qquad (1)$$

Where f is a continuous function in IR^n and S is a constraint set in IR^n. The idea of a penalty method is to replace problem (1) by an unconstrained problem of the form

$$Minimize \quad f(x) + \mu\, P(x) \qquad (2)$$

where μ is a positive constant and P is a function in IR^n satisfying:

(i) P is continuous

(ii) $P(x) \geq 0$ for all $x \in IR^n$

(iii) $P(x) = 0$ if and only if $x \in S$

Supposing that $S = \{x : g(x) = 0\}$, the problem (1) can be replaced by the unconstrained one as follows:

$$\begin{cases} Minimize \quad f(x) + \mu\, g^2(x) \\ \mu \quad is\ large \quad enough \\ \qquad\qquad x \in IR^n \end{cases} \qquad (3)$$

Supposing now that S is defined by a number of inequality constraints

$$S = \{x : g_i(x) \leq 0, \quad i = 1,2,\ldots\ldots\ldots\ldots,p\}$$

A very useful penalty function in this case is

$$P(x) = \sum_{i=1}^{p} (\max\,[0, g_i(x)])^2 \qquad (4)$$

For large μ, it is evident that the minimum point of problem (2) will be in a region where P is small. Thus, by increasing μ, it is expected that the corresponding solution points will approach the feasible region S and, as subject to being close, will minimize f. Ideally then, as $\mu \to \infty$, the solution point of the penalty problem will converge to a solution of the unconstrained problem. More generally, providing that the subset S is defined as

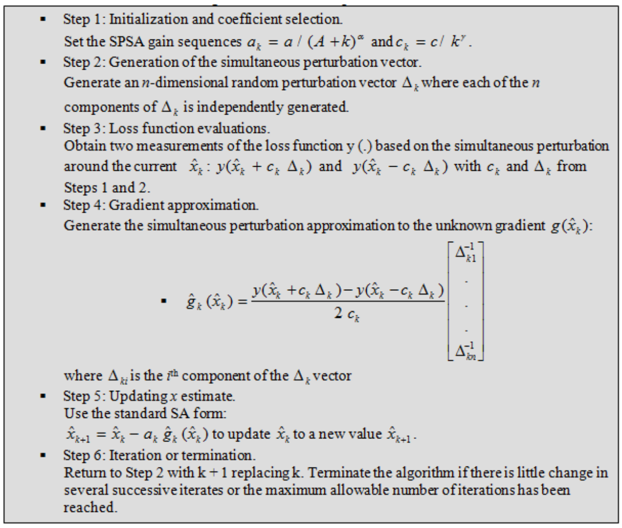
$$S = \begin{Bmatrix} x : g_i(x) \leq 0, & i = 1,2,\ldots\ldots\ldots,m \\ and \quad h_j(x) = 0, & j = 1,2,\ldots\ldots\ldots,p \end{Bmatrix} \qquad (5)$$

then the problem (1) is equivalent to that of (2) with a penalty function of the form:

$$P(x) = \sum_{i=1}^{m} (\max\,[0, g_i(x)])^2 + \sum_{j=1}^{p} \left|h_j(x)\right|^2 \qquad (6)$$

3.2 SPSA method

SPSA is based on a highly efficient and easily implemented simultaneous perturbation approximation to the gradient: this gradient approximation uses only two loss-function measurements, independent of the number of parameters being optimized. The following general algorithm of SPSA is based on the work of J. Spall [10-11].



- Step 1: Initialization and coefficient selection.
  Set the SPSA gain sequences $a_k = a / (A + k)^\alpha$ and $c_k = c / k^\gamma$.
- Step 2: Generation of the simultaneous perturbation vector.
  Generate an n-dimensional random perturbation vector Δ_k where each of the n components of Δ_k is independently generated.
- Step 3: Loss function evaluations.
  Obtain two measurements of the loss function y (.) based on the simultaneous perturbation around the current $\hat{x}_k$: $y(\hat{x}_k + c_k\,\Delta_k)$ and $y(\hat{x}_k - c_k\,\Delta_k)$ with c_k and Δ_k from Steps 1 and 2.
- Step 4: Gradient approximation.
  Generate the simultaneous perturbation approximation to the unknown gradient $g(\hat{x}_k)$:
  - $$\hat{g}_k(\hat{x}_k) = \frac{y(\hat{x}_k + c_k\,\Delta_k) - y(\hat{x}_k - c_k\,\Delta_k)}{2\,c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ . \\ . \\ . \\ \Delta_{kn}^{-1} \end{bmatrix}$$

  where Δ_{ki} is the i^{th} component of the Δ_k vector
- Step 5: Updating x estimate.
  Use the standard SA form:
  $\hat{x}_{k+1} = \hat{x}_k - a_k\,\hat{g}_k(\hat{x}_k)$ to update $\hat{x}_k$ to a new value $\hat{x}_{k+1}$.
- Step 6: Iteration or termination.
  Return to Step 2 with k + 1 replacing k. Terminate the algorithm if there is little change in several successive iterates or the maximum allowable number of iterations has been reached.

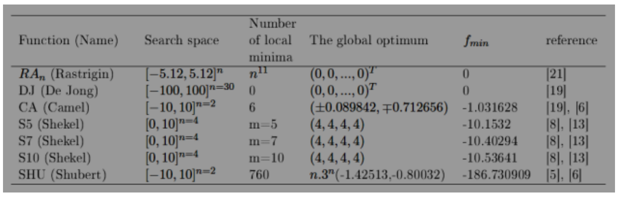
Figure 3. The basic steps of SPSA.

4 Numerical results

4.1 Benchmark test functions

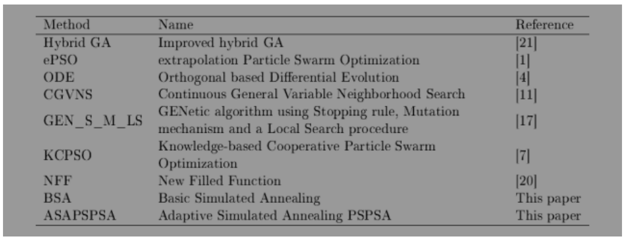
Several tests have been performed on some well-known benchmark test functions; see Appendix, with known global optima in order to demonstrate the efficiency as well as the accuracy of the proposed method. Basic information about the benchmark functions are reported in Table 1, global optimization methods used for performance analysis are provided by Table 2 and computational results are summarized in Table 3 for each problem.

Table 1. Basic information about 5 Benchmark functions



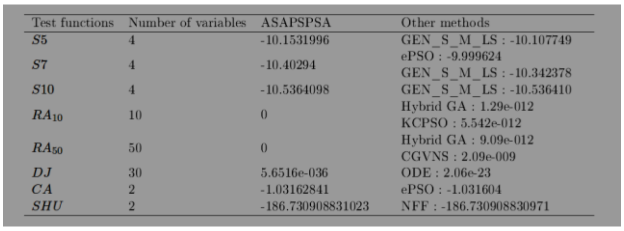
Function (Name)	Search space	Number of local minima	The global optimum	f_{min}	reference
RA_n (Rastrigin)	$[-5.12, 5.12]^n$	n^{11}	$(0, 0, ..., 0)^T$	0	[21]
DJ (De Jong)	$[-100, 100]^{n=30}$	0	$(0, 0, ..., 0)^T$	0	[19]
CA (Camel)	$[-10, 10]^{n=2}$	6	$(\pm 0.089842, \mp 0.712656)$	-1.031628	[19], [6]
S5 (Shekel)	$[0, 10]^{n=4}$	m=5	$(4, 4, 4, 4)$	-10.1532	[8], [13]
S7 (Shekel)	$[0, 10]^{n=4}$	m=7	$(4, 4, 4, 4)$	-10.40294	[8], [13]
S10 (Shekel)	$[0, 10]^{n=4}$	m=10	$(4, 4, 4, 4)$	-10.53641	[8], [13]
SHU (Shubert)	$[-10, 10]^{n=2}$	760	$n.3^n$(-1.42513,-0.80032)	-186.730909	[5], [6]

Table 2. Global optimization methods used for performance analysis



Method	Name	Reference
Hybrid GA	Improved hybrid GA	[21]
ePSO	extrapolation Particle Swarm Optimization	[1]
ODE	Orthogonal based Differential Evolution	[4]
CGVNS	Continuous General Variable Neighborhood Search	[11]
GEN_S_M_LS	GENetic algorithm using Stopping rule, Mutation mechanism and a Local Search procedure	[17]
KCPSO	Knowledge-based Cooperative Particle Swarm Optimization	[7]
NFF	New Filled Function	[20]
BSA	Basic Simulated Annealing	This paper
ASAPSPSA	Adaptive Simulated Annealing PSPSA	This paper

Table 3. Performance of ASASPSA on 5 test functions.



Test functions	Number of variables	ASAPSPSA	Other methods
$S5$	4	-10.1531996	GEN_S_M_LS : -10.107749
$S7$	4	-10.40294	ePSO : -9.999624 GEN_S_M_LS : -10.342378
$S10$	4	-10.5364098	GEN_S_M_LS : -10.536410
RA_{10}	10	0	Hybrid GA : 1.29e-012 KCPSO : 5.542e-012
RA_{50}	50	0	Hybrid GA : 9.09e-012 CGVNS : 2.09e-009
DJ	30	5.6516e-036	ODE : 2.06e-23
CA	2	-1.03162841	ePSO : -1.031604
SHU	2	-186.730908831023	NFF : -186.730908830971

As shown in Table 3, the ASAPSPSA can converge fairly close to the global optimum (verified by the analytical solutions). We also notice that, when compared with other methods, ASAPSPSA demonstrate its accuracy in achieving the global solution.

4.2 Optimal design of a triangular plate

The problem considered is a hexagonal steel plate, using thickness T_1 and fillet radius FIL as the optimization parameters (see Figure 4). This problem uses a 2-D model and takes advantage of symmetry. The loading is tensile pressure (traction) of 50 MPa at the three flat faces. The purpose of this optimization problem is to minimize the volume V of the triangular plate without exceeding the allowable stress σ_{von}. The maximum stress anywhere in the triangular plate should not exceed 150 MPa. The optimum set of design variables can be achieved by solving the following minimization problem:

$$\begin{cases} \quad Minimize\ Volume \\ Subject\ to: \qquad \sigma_{von} \leq 150 \\ \qquad\qquad 20 \leq \ T_1 \leq 40 \\ \qquad\qquad 5 \ \leq FIL \ \leq 15 \end{cases} \tag{7}$$

In order to solve problem 1, we have applied two schemes:

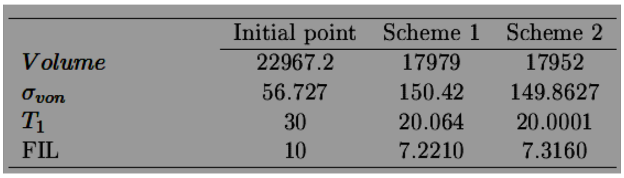
Scheme 1: both Structural Analysis and Optimization were done by using commercial software FEA code ANSYS (using First-Order method).

Scheme 2: Structural Analysis was performed in commercial software FEA code ANSYS but an external Optimization code is used (ASAPSPSA optimizer).

4.3 Numerical results

Calculations show that the results provided by ASAPSPSA are better than those provided by Scheme1 using First-Order method (see Table 4).

Table 4. Scheme 1 vs. Scheme 2



	Initial point	Scheme 1	Scheme 2
$Volume$	22967.2	17979	17952
σ_{von}	56.727	150.42	149.8627
T_1	30	20.064	20.0001
FIL	10	7.2210	7.3160

4.4 Graphical results

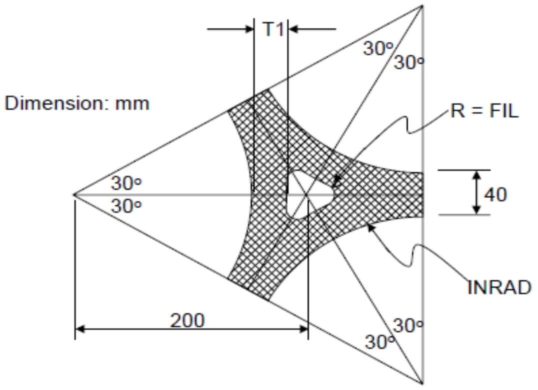


Figure 4. Triplate.

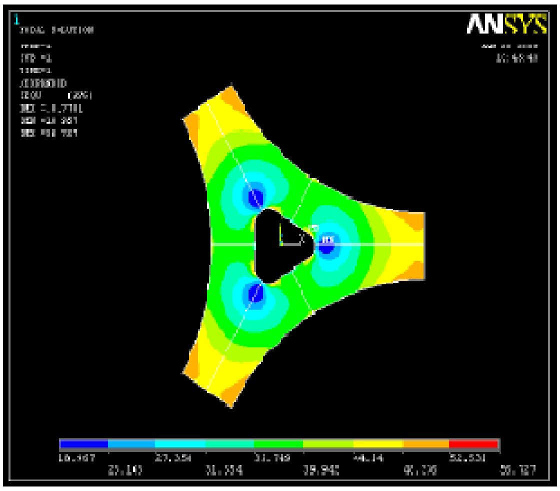


Figure 5. The Graph of a Hexagonal plate with initial solution.

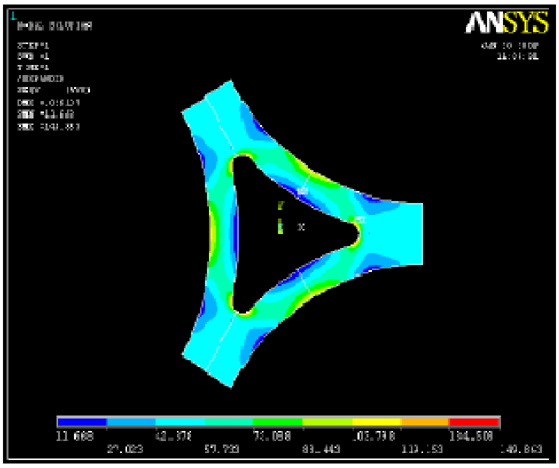


Figure 6. The Graph of the optimal hexagonal plate using ASAPSPSA.

5 Conclusion

In this paper a new global hybrid method ASAPSPSA has been proposed. The new algorithm can be widely applied to a class of global optimization problems for continuous functions with box constraints. The experimental results show that the present method has proved the robustness and high performance of its algorithm. We can see clearly from numerical and graphical results that the algorithm can yield the global optimum with high accuracy.

Acknowledgements

This work was supported by CMIFM, A.I. EGIDE, Number: MA/07/173.

Appendix

$$RA_n = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) \right] \qquad (8)$$

$SHU =$

$$\left\{ \sum_{i=1}^{5} i\cos\left[(i+1)x_1 + i\right] \right\} \left\{ \sum_{i=1}^{5} i\cos\left[(i+1)x_2 + i\right] \right\} \qquad (9)$$

$CA =$

$$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4 \qquad (10)$$

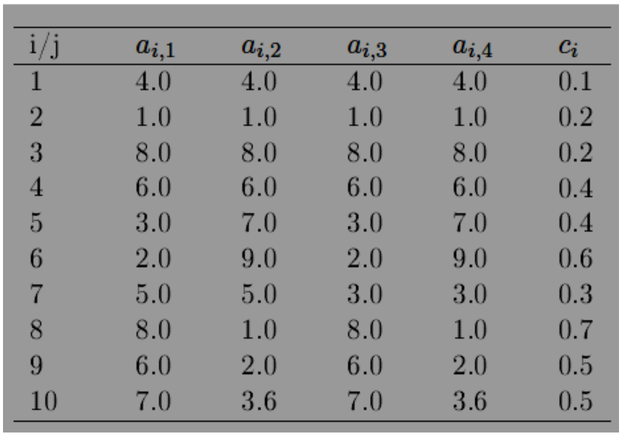
$$DJ = \sum_{i=1}^{n} x_i^2 \qquad (11)$$

Shekel =

$$-\sum_{i=1}^{m} \frac{1}{(x - a_i)^T (x - a_i) + c_i}, \quad x \in IR^n \qquad (12)$$

The Shekel function coefficients a_{ij}, c_i are (Table 5):

Table 5: Shekel function-four-dimensional



i/j	$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$a_{i,4}$	c_i
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.2
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.4
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3
8	8.0	1.0	8.0	1.0	0.7
9	6.0	2.0	6.0	2.0	0.5
10	7.0	3.6	7.0	3.6	0.5

References

[1] Bennett, J. A., Botkin, M. E.: *Structural shape optimization with geometric description and adaptive mesh refinement*, AIAA J., 238 (1984) 3, 458-464.

[2] Bruyneel, M., Duysinx, P., Fleury, C.: *A family of MMA approximations for structural optimization*, Structural Multidisciplinary Optimization, 24 (2002), 263-276.

[3] Meske, R., Sauter, J., Schnack, E.: *Nonparametric gradient-less shape optimization for real-world applications*, Structural Multidisciplinary Optimization, 30 (2005), 201-218.

[4] Queau, J. P., Trompette, P. H.: *Two-dimensional shape optimal design by the finite element method*, International Journal of Numerical Methods Engineering, 15 (1980), 1603-1612.

[5] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H.: *Equation of State Calculations by Fast Computing Machines*, Journal of Chemical Physics, 21 (1953), 1087-1092.

[6] Van Laarhoven, P. J. M., Aarts, E. H. L.: *Simulated annealing: theory and applications, Dordrecht*, D. Reidel Publishing Company, Kluwer, 1987.

[7] Sitarz, S: *Ant algorithms and simulated annealing for multicriteria dynamic programming*, Computers and Operations Research, 36 (2009), 433-441.

[8] Gong, W., Cai, Z., Jiang, L.: *Enhancing the performance of differential evolution using orthogonal design method*, Applied Mathematics and Computation, 206 (2008), 56-69.

[9] Shelokar, P.S., Siarry, P., Jayaraman, V. K., Kulkarni, B: D.: *Particle swarm and ant colony algorithms hybridized for improved continuous optimization*, Applied Mathematics and Computation, 188 (2007), 129-142.

[10] Spall, J. C.: *Multivariate Stochastic Approximation using a Simultaneous Perturbation Gradient Approximation*, IEEE transactions on automatic control, 37 (1992), 332-341.

[11] Gao, L., Zhou, Y., Li, C., Huo, L.: *Reliability assessment of distribution systems with distributed generation based on Bayesian networks*, Engineering Review, 34 (2014) 1, 55-62.