Cmsc 191 Long Exam 1 Part 1: Analysis of Convergence of 10 Standard Test Functions Using the Hybrid Simulated Annealing Algorithm

Gimel David F. Velasco

October 4, 2016

Abstract

This paper demonstrates an implementation of the Hybrid Simulated Annealing in solving the global minimum for 10 continuous 2-variabled standard test functions.

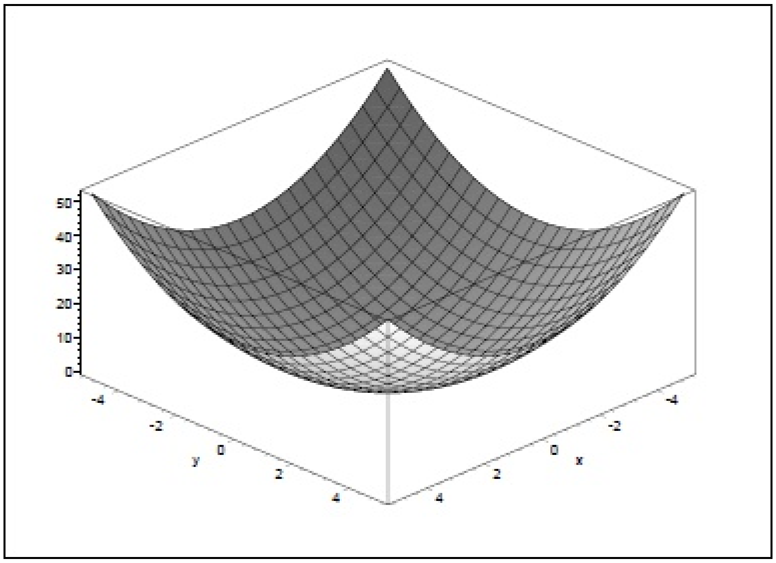
1 Introduction

The Simulated Annealing is a type of Heuristics which is used for solving complex problems. This algorithm is of the analogy that if a liquid material cools slowly, the system will then get into a state of minimum energy optimally. On the onther hand, if the liquid material cools too quickly, it will have a sub-optimal configuration.In this paper, the concept of the Hybrid Simulated Annealing is presented wherein instead of looking at a single X_i as the possible solution for a problem, a whole number of population of X_is are used. Which was adapted in the concept of the Genetic Algorithm. Now, for these standard test functions, the Hybrid Simulated Annealing Algorithm will solve or get close to their respective global minimum.

2 10 Test Functions

The algorithm will seek the global minimum of the following standard test functions [1]:
Problem 1: De Jong's Function ($|x| \leq 5.12; n = 5, 10$)

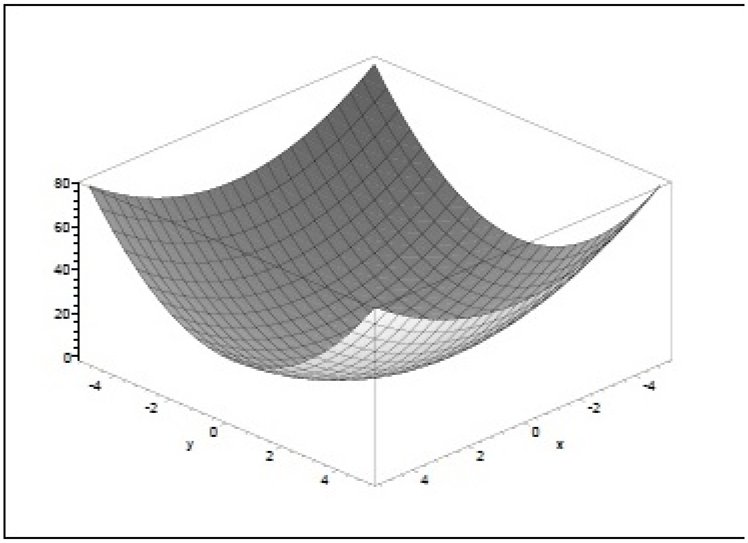
$$f(x) = \sum_{i=1}^{n} x_i^2 \tag{1}$$



Global Minimum: $f(x) = 0$

Problem 2: Axis Parallel Hyper-Ellipsoid Function ($|x| \leq 5.12; n = 5, 10$)

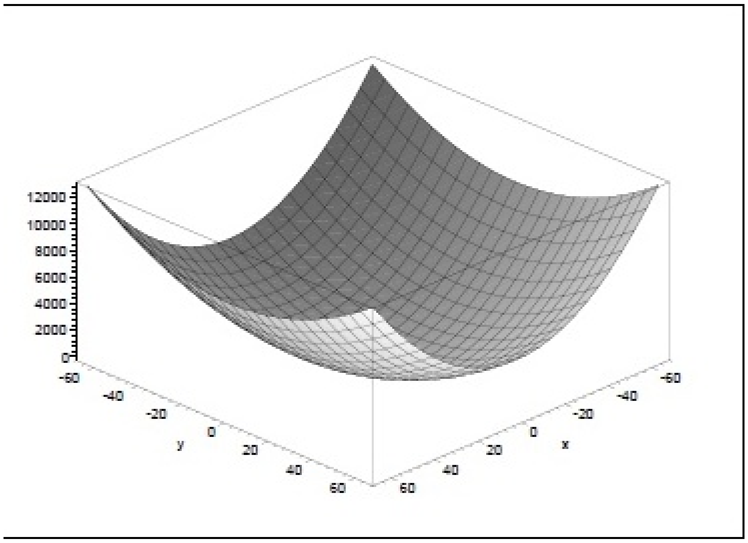
$$f(x) = \sum_{i=1}^{n} (i * x_i^2) \tag{2}$$



Global Minimum: $f(x) = 0$

Problem 3: Rotated Hyper-Ellipsoid Function ($|x| \leq 65.536; n = 5, 10$)

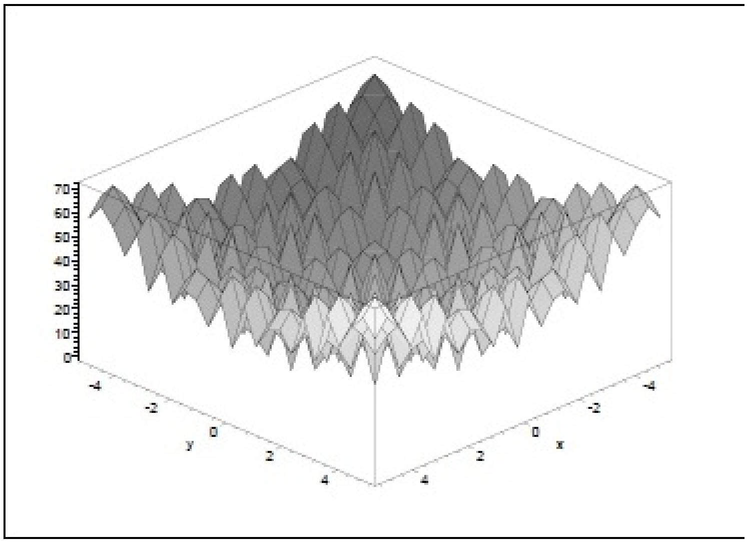
$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{i} x_j^2 \tag{3}$$



Global Minimum: $f(x) = 0$

Problem 4: Rastrigin's Function ($|x| \leq 5.12; n = 5, 10$)

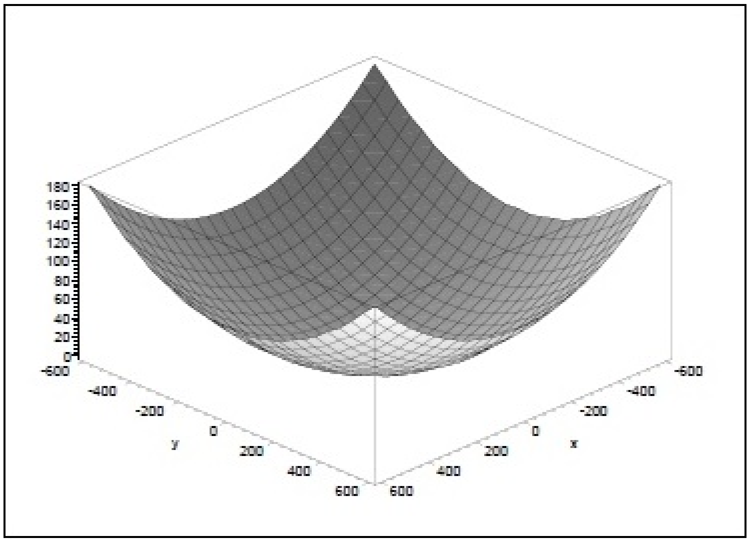
$$f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i)] \tag{4}$$



Global Minimum: $f(x) = 0$

Problem 5: Griewangk's Function ($|x| \leq 600; n = 5, 10$)

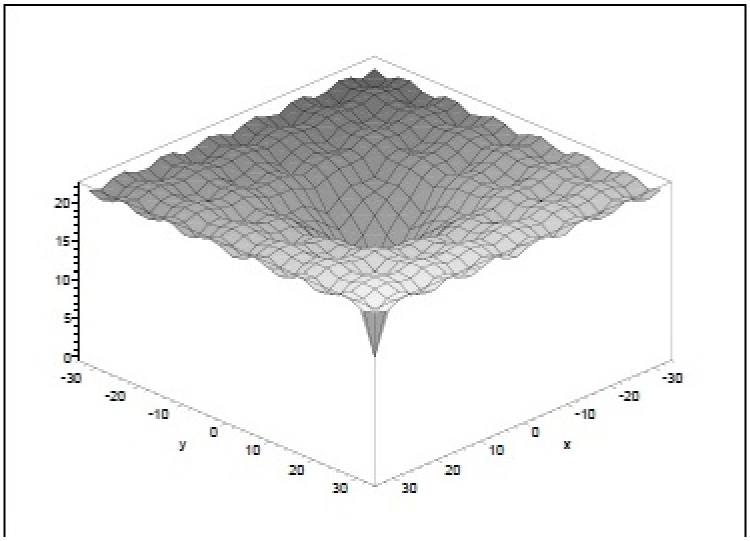
$$f(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{5}$$



Global Minimum: $f(x) = 0$

Problem 6: Ackley's Function ($|x| \leq 32.768; n = 5, 10$)

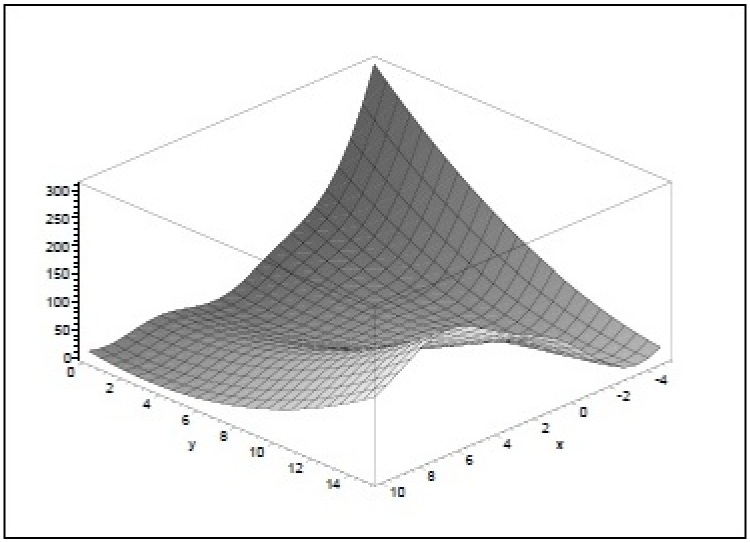
$$f(x) = -a * exp(-b * \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(cx_i)) + a + exp(1) \tag{6}$$



Global Minimum: $f(x) = 0$
where $a = 20, b = 0.2, c = 2\pi$

Problem 7: Branin's Function ($|x| \leq 12.5; n = 2$)

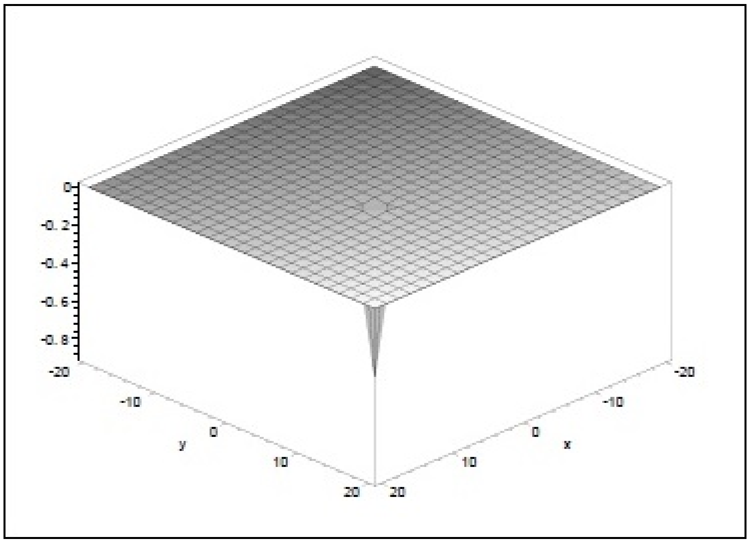
$$f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)cos(x_1) + e \tag{7}$$



Global Minimum: $f(x_1, x_2) = 0.397887$
where $a = 1, b = \frac{5.1}{4\pi^2}, c = \frac{5}{\pi}, d = 6, e = 10, f = \frac{1}{8\pi}$

Problem 8: Easom's Function ($|x| \leq 100; n = 2$)

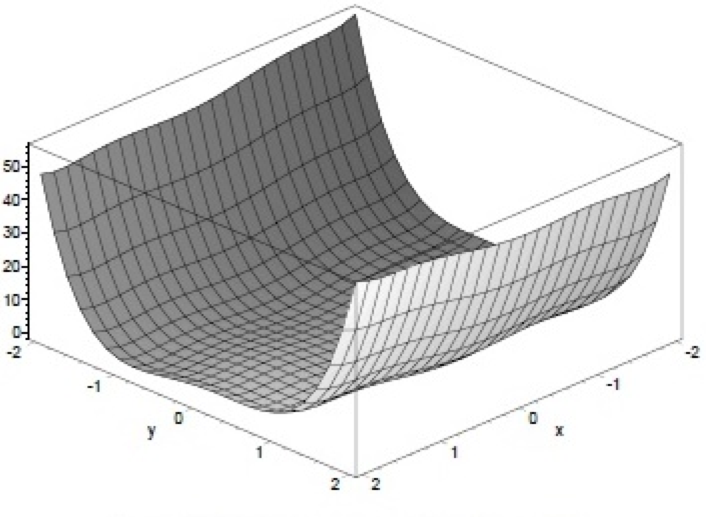
$$f(x_1, x_2) = -cos(x_1)cos(x_2)exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2) \qquad (8)$$



Global Minimum: $f(x_1, x_2) = -1$

Problem 9: Six-Hump Camel Back Function ($|x| \leq 3; n = 2$)

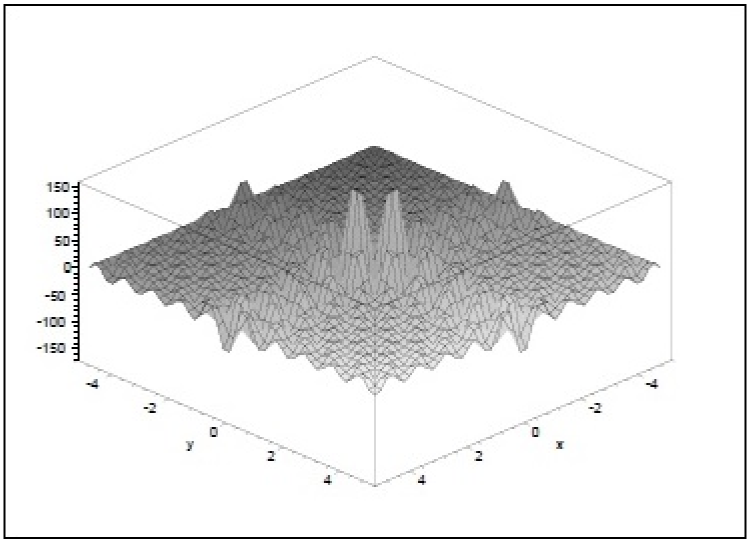
$$f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \qquad (9)$$



Global Minimum: $f(x_1, x_2) = -1.0316$

Problem 10: Shubert's Function ($|x| \leq 5.12; n = 2$)

$$f(x_1, x_2) = -\sum_{i=1}^{5} icos((i+1)x_1 + 1) \sum_{i=1}^{5} icos((i+1)x_2 + 1) \tag{10}$$



Global Minimum: $f(x) = -186.7309$

3 Methodology

The fitness function or objective function of which is defined as:

$$OBJ(x) = |f(x) - globalminimum| \tag{11}$$

A Minimization Problem is implemented in the algorithm wherein the objective of the algorithm is to get very close if not equal to zero.

3.1 The Hybrid Simulated Annealing Algorithm Pseudocode

The Hybrid Simulated Annealing Algorithm implemented in this program is based on how the algorithm is explained in the Test Paper. The pseudocode of the Hybrid Simulated Annealing Algorithm is shown below:

```
1. Set $T$ to a sufficiently high value
2. Initialize the population randomly
3. Repeatedly generate each new population from current population until $T$ has reached a desireable minimum value:
  3.a. Do $N/2$ times:
    3.a.i. Select two parents at random from the $N$ population
    3.a.ii. Generate two offspring using recombination operator (crossover), followed by the neighborhood operator (mutation)
    3.a.iii. Hold on one or two Boltzman trials between offspring and parents
    3.a.iv. Overwrite the parents with the trial winners
  3.b. Periodically lower $T$
4. Take Solution
```

where N is the number of population and T is the System Temperature.

3.2 The Boltzman Trial Algorithm Pseudocode

The Boltzman Trial Pseudocode implied in this algorithm is as follows:

```
1. Pick the Single or Double Acceptance/ Rejection by flipping a coin
2. If the Double Acceptance/Rejection is picked, take $E_i = E_{parent1} + E_{parent2}; E_j = E_{child1} + E_{child2}$
3. If the Single Acceptance/Rejection is picked, take $E_i = E_{parent}; E_j = E_{child}$. Do this so that each parent would be trialed with each child.
4. The logistic probability of $E_i$ to win is $1/(1 + exp(E_i - E_j)/T)$
```

where $E_{chromosome}$ is the cost of solution to a respective chromosome

3.3 The Recombination Operator Pseudocode

The Recombination Operator Pseudocode implied in this algorithm is as follows:

```
1. If the selected parents are to undergo crossover (based on $p_c$), perform a single point crossover at a random point in their chromosome producing two children. Then proceed to the Neighborhood Operator.
2. If the parents are not to undergo crossover, the parents will have their tournament. The better parent overwrites the other parent. Do not proceed to the Neighborhood Operator.
```

3.4 The Neighborhood Operator Pseudocode

The Neighborhood Operator Pseudocode implied in this algorithm is as follows [2]:

```
1. Each of the generated children will do steps 2 to 4.
2. Perturb $n$ neighbors of a child where $n = N * p_n$
3. If a neighbor is better than the child, replace the child
4. Else, replace the child with a probability $e^{-D/T}$ where $D = E_{neighbor} - E_{child}$
```

3.5 General Test Procedure

The whole program will test the values of $N, p_c, p_m, ratio_{cooling}$ that best suits each function. For $N = 100and1000$; $p_c = 1.0, 0.9and0.8$; $p_m = 0.05and0.01$; $ratio_{cooling} = 0.7and0.9$. Each function will run each pairing of the parameters for 3 trials and then the average energy, E_{aveg}, and the average runtime, $runtime_{aveg}$, will be computed for each 3 trials. The program has the initial and final T set to $T_{initial} = 1000$ and $T_{final} < 0.0000000000000001$.

4 Results and Discussion

4.1 Best Parameters per Function

After 3 trials on each $N, p_c, p_m, ratio_{cooling}$, the best results yielded per function are shown below:

Problem 1: De Jong's Function (Best)
Best Solution found at

Function 1	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	0.8	0.05	0.9

with $E_{average} = 0.0015764710498753$ and $Runtime_{average} = 41.3449611858984620$s

Problem 2: Axis Parallel Hyper-Ellipsoid Function
Best Solution found at

Function 2	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	0.9	0.01	0.7

with $E_{average} = 0.0009798755988856$ and $Runtime_{average} = 5.9209057866499180$s

Problem 3: Rotated Hyper-Ellipsoid Function
Best Solution found at

Function 3	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	0.8	0.01	0.7

with $E_{average} = 0.2953668378590396$ and $Runtime_{average} = 5.3379754476486250$s

Problem 4: Rastrigin's Function
Best Solution found at

Function 4	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	0.9	0.01	0.7

with $E_{average} = 0.1381402257857746$ and $Runtime_{average} = 5.8467638864548475$s

Problem 5: Griewangk's Function
Best Solution found at

Function 5	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	0.8	0.05	0.7

with $E_{average} = 0.0047081618645275$ and $Runtime_{average} = 27.2035168548162290$s

Problem 6: Ackley's Function
Best Solution found at

Function 6	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	0.9	0.05	0.7

with $E_{average} = 0.9123456551835348$ and $Runtime_{average} = 54.2163086700349270$s

Problem 7: Branin's Function
Best Solution found at

Function 7	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	0.9	0.05	0.7

with $E_{average} = 0.0174309157901743$ and $Runtime_{average} = 25.5826198382648900$s

Problem 8: Easom's Function
Best Solution found at

Function 8	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	1.0	0.05	0.9

with $E_{average} = 0.5086092506616432$ and $Runtime_{average} = 48.5967730565665260$s

Problem 9: Six-Hump Camel Back Function
Best Solution found at

Function 9	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	0.8	0.01	0.9

with $E_{average} = 0.0030237695257636$ and $Runtime_{average} = 8.7973803821196856$s

Problem 10: Shubert's Function
Best Solution found at

Function 9	N	p_c	p_m	$ratio_{cooling}$
n=5	1000	1.0	0.01	0.7

with $E_{average} = 0.6965685603289558$ and $Runtime_{average} = 6.5674327747791894$s

4.2 Observations

Based on the observations on all the results stated above, each parameter $N, p_c, p_m, ratio_{cooling}$ is discussed below:

4.2.1 Population Size

It is easily observed that the runs that had a larger number of population, N, yielded the best results. So it can be implied that the bigger the population size, the better are the results that will be yielded. The side effect of that though is that the runtime would significantly be affected as it also has a relationship with the mutation rate p_m.

4.2.2 Crossover Rate

From what was observed, a bigger crossover rate, p_c, is very effective in searching for the global minimum only on the functions that has their global minimum very steep. Such functions were Rastrigin's Function, Ackley's Function, Easom's Function and Shubert's Function. The Algorithm was able to find the best where the crossover rate is larger.

4.2.3 Mutation Rate

The Mutation Rate, p_m, has a very significant effect on the runtime of the algorithm. As observed from the results, the runs that had a larger p_m yielded a very large runtime.

4.2.4 Cooling Ratio

The Cooling Ratio, $ratio_{cooling}$, also gives an effect in the runtime but not as significant as that of the p_m.

5 Conclusion

The Hybrid Simulated Annealing Algorithm was able to solve or get near the coordinates of where the global minimum of each function is. With the different observations stated above, the proper parameters to use for the Hybrid Simulated Annealing Algorithm to solve a function is now made known. For a user to be more accurate for solving the solution or global minimum/maximum of a function, the stated parameters above can be used as guidelines.

6 References

[1] Molga, M. and Smutnicki, C., "Test Functions for Optimization Needs", 2005

[2] de Weck O., "Simulated Annealing: A Basic Introduction", Massachusetts Institute of Technology, 2010