

Optimization of Dejong Function using GA under Different Selection Algorithms

Kapil Juneja

Department of Computer Science & Applications
Maharshi Dayanand University,
Rohtak 124001, Haryana, India

Nasib Singh Gill

Department of Computer Science & Applications
Maharshi Dayanand University,
Rohtak 124001, Haryana, India

ABSTRACT

There are numbered of Numerical and Mathematical Problem that takes the exponential time to process called NP Problems. These kinds of problems always need some optimization algorithm to complete the process in effective time. Genetic is one of the adaptive evolutionary algorithmic approaches to solve such NP problems in optimize time frame. In this work, A Genetic approach is defined to optimize the solution of Rosenbrock's valley problem. The work also includes the Optimization analysis of different parametric changes in selection and crossover phase. In this paper, the result of the diverse models is presented in the form of the graph.

Keywords

DeJong Function, Genetics, Optimization, Selection, Crossover

1. INTRODUCTION

All the available algorithmic problems are divided into two major categories called the Polynomial and the Non Polynomial Problems. The Non-Polynomial is the special class of the problem that provides the best solution in exponential time and gives the complexity in form of $O(2^n)$, $O(3^n)$ or $O(m^n)$. There are numbered of algorithmic, numerical and mathematical problems that come within this category. In these kinds of problems, as the size of the dataset increase, the time complexity within the system also raised exponentially. In such a case, some optimization process is required to obtain the best results in the effective time frame. There are numbered of problems that require the optimization such as Travelling Salesman Problem[1], Knapsack problem[2], shortest path problem[3], etc. There also exist a number of optimization techniques such as Approximate Algorithm[4], Linear Programming[5], Exact Algorithms[4], Genetics[6], Neural Network[7], Partical Swarm Optimization[8], Ant Colony Optimization[9], etc. Each algorithm is defined under some constraints and parameters to solve the NP problems up to some extent. These optimization algorithms are shown in Fig. 1.

Exact Algorithm is one of the conventional optimization approaches that provides the optimization solution for finite size instance problem. Another significant traditional and accurate optimization approach is the Approximate Algorithm. This algorithmic is further classified in two sub approaches called integer programming and local search approach. Integer Programming approach provides the effective decision support variables that process only for integers. Local Search is another useful algorithmic approach under an approximate algorithmic program. It is an iterative method with small localized moves to derive the high-quality solutions for the complex problems like Tabu Search, Memetic Algorithm, Simulated Annealing, etc[4][10].

The Linear Programming uses an additive function defined with real variables to optimize the problem. In many mathematical models where the linear equation can be directly implemented, this method provides the best solution.

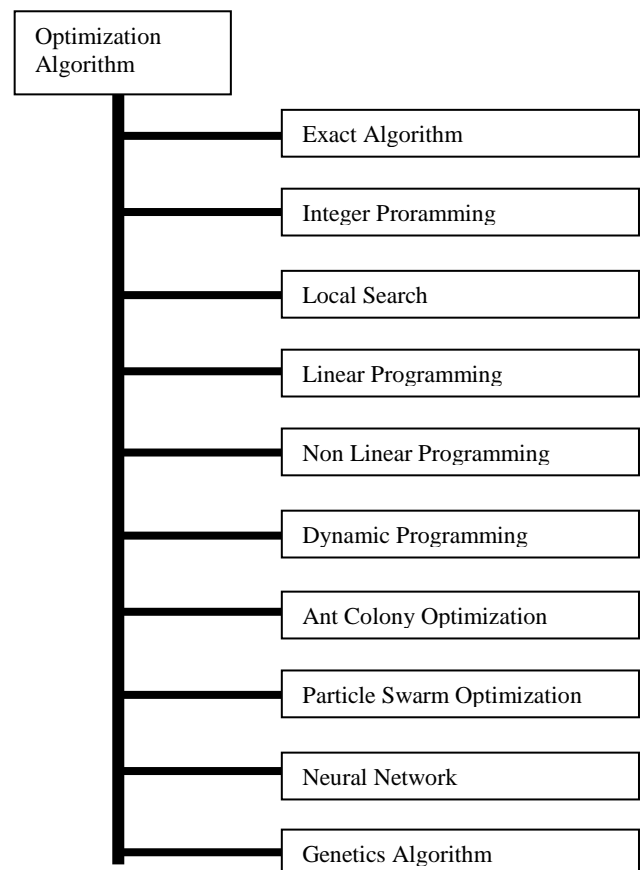


Fig. 1: Optimization Algorithms

Non-Linear Programming is the optimization problem with non linear objectives and related constraints. It provides a general-purpose model that will work for both the linear and non-linear constraint bounded problems. Dynamic programming is another optimization approach that split the main problem in smaller sub problems. The approach dynamically processes these sub problems and builds the final solution collectively.

Ant Colony Optimization is the bio inspired optimization approach that works on the phenomenon based on the foraging behavior of the ants. This approach defines a trail pheromones trail to find the solution direction. In communication and other networks based application it one of the most appreciated optimization approaches. Another delightful bio inspired approach is Partical swarm optimization approach. This approach

selects some random particles into the search space and performs a periodic update along with the velocity and position estimation over the population. It uses the concept of local best and global best to derive the optimized solution to the problem.

The neural network is the learning based approach that performs the decision marking based on the pattern analysis. The system provides the activation throughput on the network to perform the mapping based on three-step processes. The optimized solution depends on weigh rules, learning rate and pattern order representation. Genetics is one of the evolutionary optimization algorithms that provides a solution by processing the population of the problem set. It implements a series of operation like selection, crossover and mutation and derives the best evolved solution over the time. In this present work, the main focus is on genetic optimization to solve the mathematic model. In section 2, the DeJong Function is defined, and section 3 explores the basic genetic function. In section 3, the complete algorithmic approach adapted to the work is shown, and section 4 represents the results derived from the system.

2. DEJONG FUNCTION

The present work is about the optimization of some mathematical or numerical model, which is given by DeJong Function. DeJong provides a series of operations out of which DeJong Function 2, called Rosenbrock's Vally, is taken. It is also called Banana Function because of the graphical shape driven by the implementation of the model[11]. The Banana function is represented by a long, narrow and parabolic shaped flat valley as shown in Fig. 2.

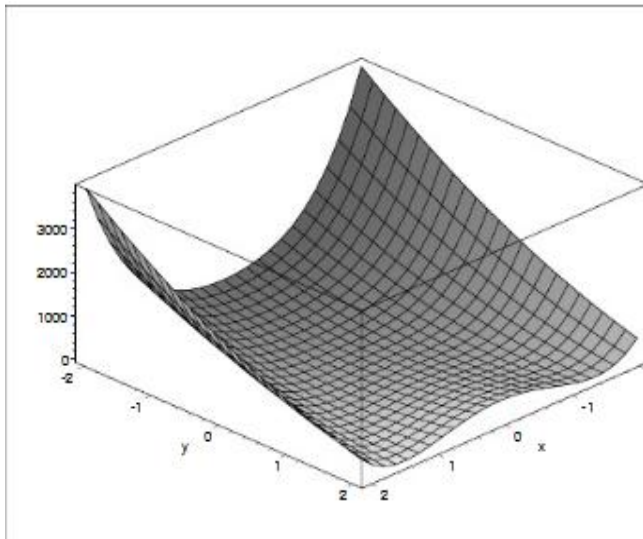


Fig. 2: DeJong Function 2[11]

The main problem in this mathematical model is to identify the valley is trivial and for which convergence of the problem is required. Which a difficult task and it requires the optimization to boost up the process. The equational representation of this mathematical model is shown in given as.

$$f2(x) = \sum_{i=1}^{n-1} 100 (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \quad (1)$$

Where $-2.048 \leq x_i \leq 2.048$.

In this model, the alteration in x is linearly and y will be changed according to the convergence function. To find the superlative solution of the problem the Genetic based optimization model is implemented in this paper. The GA model is defined here with some algorithmic step substitution for the selection procedure. In the later sections, the GA model for the DeJong problem is defined with result analysis.

3. GA MODELING

GA[12][13] is the evolutionary algorithmic approach in which the solution of the problem is driven by maintaining the population set. The population set is defined under some encoded rule set. Once the population generated a rule is defined to perform the candidate selection from this population set. On these selected candidates; t crossover operation is defined to generate a new child candidate. The mutation operation will be implemented on this fresh candidate in terms of partial value change or the replacement. Finally, the mutated child will become the part of main population. This operation is called reconstruction of the population set. The basic genetics steps are shown in Fig. 3.

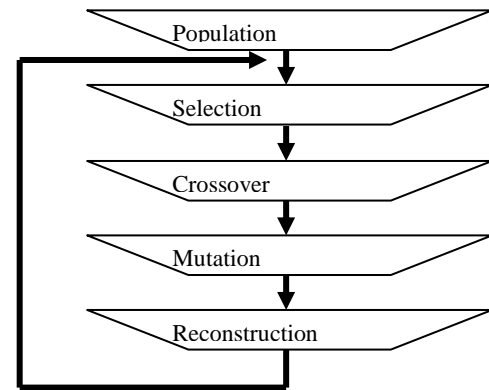


Fig. 3: Genetics Operations

In this present work, the main focus is on the selection algorithm. Here the work is about the function optimization using genetics with the variation on selection function. The analysis is being performed on different selection approaches and based on these results are being obtained. Based on selection procedure, three main models are discussed and analyzed in this section and given as under.

3.1 Rank Selection Approach

In this module, the rank selection is been included as the selection algorithm in the basic genetic model. In case of rank selection approach, the population individuals are arranged in a sorted order. On this ordered dataset, the rank based selection is performed by using the following formula.

$$P_i = 1 / N * (P_{\text{worst}} + (P_{\text{best}} - P_{\text{worst}}) * ((I-1) / (N-1))) \quad (2)$$

Here, $P_{\text{worst}} \Rightarrow$ Worst Fit Individual

$P_{\text{best}} \Rightarrow$ Best Fit Individual

$N \Rightarrow$ Size of Population

$I \Rightarrow$ Population Individual

The rank selection based genetic model is shown in Fig. 4.

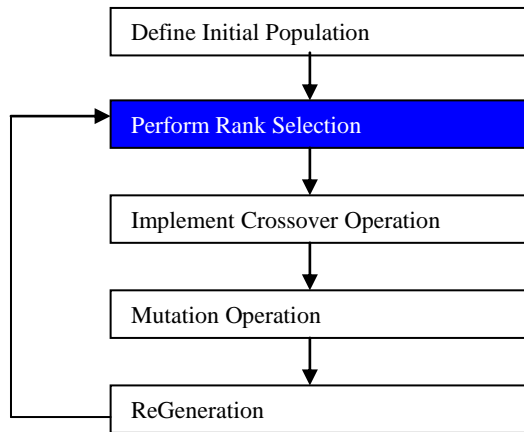


Fig. 4: Rank Selection based Genetic Model.

The rank selection algorithm based genetic algorithm to optimize the Convergence of DeJong function is given as under.

1. Create an initial generation of population using Dejong function 2.
2. Define the Minimization based Fitness Rule over the population.
3. Select the top-ranked rule and calculate its fitness over the selection period. Save it as the initial best rule.
4. Randomly select two rules, using weights attaching higher probability to more highly ranked rules.
5. Measure the fitness of each rule in the new generation over the training period. Take the best rule in the training period and measure its fitness over the selection period. If it outperforms the previous foremost rules, save it as the new optimal rule.
6. Stop after 100 generations.

3.2 ROULETTE Wheel Selection Approach

ROULETTE Wheel is the classical and effective selection operator described by Goldberg. According to this approach, the member pool is defined based on a roulette wheel proportional model to determine the fitness value. Higher probability values will have the highest chances of selection. The algorithmic model is shown in Fig. 5.

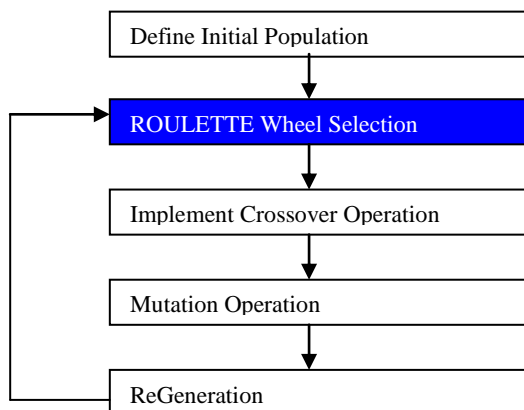


Fig. 5: ROULETTE Wheel Selection based GA

According to this model, the chance of the individual selection is proportional to the fitness value of individual. It means the individual with higher fitness will be selected more often. The basic work of this model is summarized her as.

1. Some of the fitnesses of all the population members; call the result total fitness.
2. Generate n , a random number between 0 and total fitness
3. Return the first population member whose fitness, added to the fitnesses of the preceding population members, is greater than or equal to n .

3.3 Elitism Selection Approach

Elitism is the selection procedure based on preemptive seeding of the newly initialized population. This selection procedure finds the best individual by performing the analysis on all the previous run of genetic. It will provide the evolution for new run. This method gives the advantage to find the immediate high fitness for the population member. The Elitism based genetic model is shown in Fig. 6.

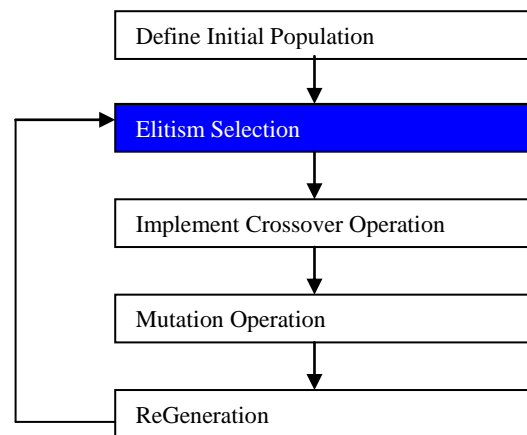


Fig. 6: Elitism Selection based GA

3.4 Crossover

The crossover operation is basically used to generate a new child instance based on the selected candidates. There are number of crossover approaches to construct a new child instance. In this work, the One-point crossover is been implemented. The single point will split the first genome in two halves at the crossover point. Both halves are then matched with the second genome to find the longest common subsequence. Matching is done as follows: The first half is matched with the first character from the second genome, then with the first two characters and so on. The second half is matched with the last character from the second genome, then with the last two characters and so on.

3.5 Mutation

The mutation is the slightly change in the new child instance. The change here is performed by the inclusion of 1 in the obtained new child value.

4. RESULTS

The complete work has been drawn on different scenarios of genetics. The work is implemented with some constant stages and implemented the variations at one stage called selection algorithm. The implementation is here done for three main selection algorithms: Rank Selection Algorithm, ROULETTE

Wheel Selection, ELITISM Selection. The work is implemented in matlab 7.8. The results obtained based on this all analysis is shown as under. The work is defined for different algorithmic models as given in previous section. The respective results are given here as under.

4.1 Rank Selection based GA

The basic parameters of the model is given in table 1.

Table 1 : Model 1 Parameters

Parameter	Value
Encoding	Real Encoding
Fitness Function	De Jong Function 2
Selection	Rank Selection
Number of Iterations	100
Mutation Rate	1%
Population Size	20
Mutation	Interchanging

In this scenario, the Rank Selection Algorithm is implemented at selection stage of genetics to perform the optimization of Dejong function 2. All other factors are same in all scenarios. The results obtained from this scenario is shown in Fig. 7.

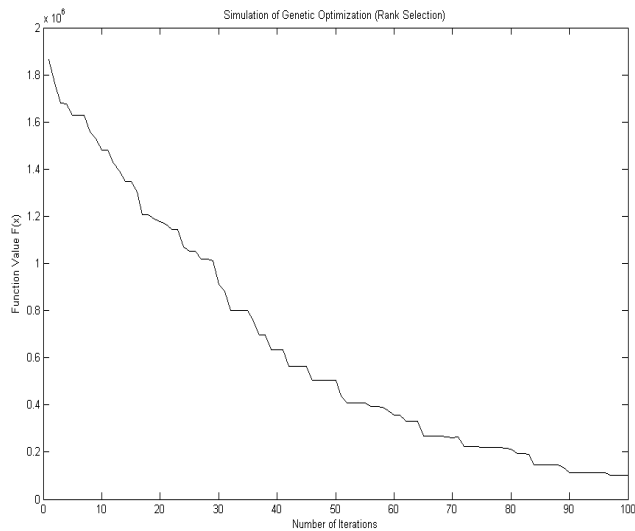


Fig. 7 : Simulation of Rank Selection based GA (100 Iter)

In this work, the minimization function is been implemented to obtain the best value from the function. The different results obtained after different iterations are shown in table 2.

Table 2: Iteration Based Analysis of Rank Selection

Sr. No.	Number of Iterations	Rank Selection F(x)
1	30	750030
2	60	279058
3	90	99313.7
4	120	48589.2
5	150	26478.2
6	180	21911.8
7	210	17110.8
8	240	12168
9	270	9749.14
10	300	6952.99
11	330	6238.02
12	360	3369.44
13	390	3294.09
14	420	3258.17
15	450	413.557
16	480	295.652
17	510	200.276
18	540	161.111
19	570	138.57
20	600	127.298

4.2 ROULETTE Wheel Selection approach

The basic parameters of the model is given in table 3.

Table 3 : Model 2 Parameters

Parameter	Value
Encoding	Real Encoding
Fitness Function	De Jong Function 2
Selection	ROULETTE WHEEL Selection
Number of Iterations	100
Mutation Rate	1%
Population Size	20
Mutation	Interchanging

In this scenario, the ROULETTE Wheel Algorithm of Dejong function 2 is been implemented to perform the optimization. All other factors are same in all scenarios. The results obtained from this scenario is shown in Fig. 8 for 100 iterations.

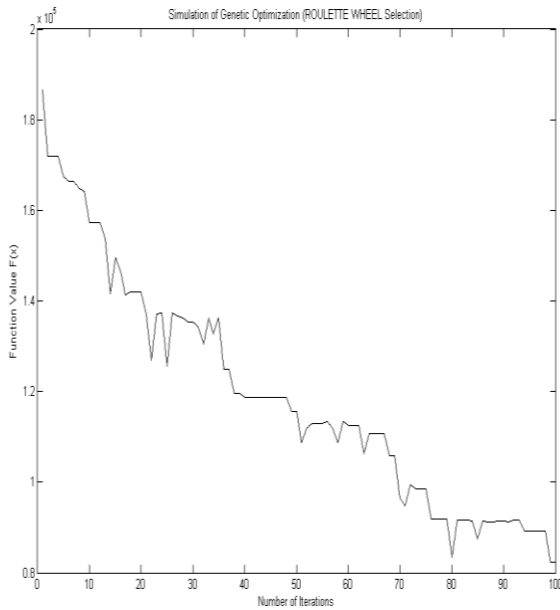


Fig. 8 : Simulation of Roulette Wheel based GA (100 Iter)

Same work is implemented for different number of iterations and the obtained results are shown in table 4.

Table 4 : Iteration Analysis of Roulette Wheel

Sr. No.	Number of Iterations	ROULETTE WHEEL Selection F(x)
1	30	63125.63
2	60	48623.25
3	90	76235.37
4	120	54362.19
5	150	34653.54
6	180	23156.66
7	210	16325.78
8	240	13963.76
9	270	7632.68
10	300	5563.76
11	330	4369.78
12	360	2936.89
13	390	1963.19
14	420	1546
15	450	1137
16	480	316
17	510	126.69
18	540	68.28
19	570	35.56
20	600	23.59

4.3 ELITISM Selection Algorithm

The basic properties of this Scenario is

Table 5 : Model 3 Parameters

Parameter	Value
Encoding	Real Encoding
Fitness Function	De Jong Function 2
Selection	ELITISM Selection
Number of Iterations	100
Mutation Rate	1%
Population Size	20
Mutation	Interchanging

In this scenario, the ELITISM Selection Algorithm is implemented of Dejong function 2 to perform the optimization. All other factors are same in all scenarios. The result obtained from this scenario is shown in Fig. 9.

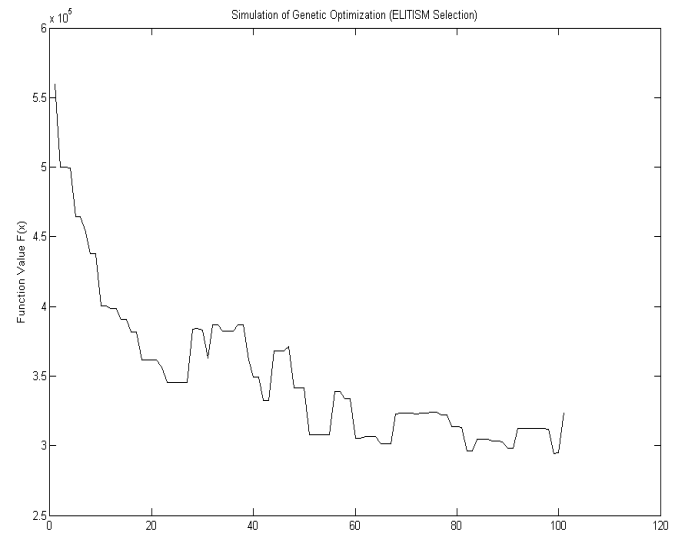


Fig. 9: Simulation of ELITISM based GA (100 Iter)

In this work, the minimization function is implemented to obtain the best value from the function. The different results obtained after various iterations are shown in table 6.

Table 6 : Iterative Analysis of Elitism Based GA

Sr. No.	Number of Iterations	ELITISM Selection F(x)
1	30	89326
2	60	75623.36
3	90	62569.26
4	120	59.236.66
5	150	86236.32
6	180	46236.69
7	210	34269.23
8	240	15693.28

9	270	26236.99
10	300	14236.85
11	330	9326.19
12	360	4236.29
13	390	3125.19
14	420	1028.96
15	450	316.56
16	480	496.69
17	510	239.26
18	540	129.36
19	570	78.59
20	600	59.63

The effective result analysis of different GA models on the mathematical model DeJong 2. In this model, the result analysis is performed for three selection algorithms and perform the analytical work on it. To identify the actual impact of selection function, all other genetic tools and parameters are set to constant. As the results obtained shows that the ROULETTE WHEEL and Elitism Selection approach give best results then rank selection algorithm. The better results mean more optimized value is achieved with the minimum number of iterations.

4. CONCLUSION

The work is about to implement and analyze the capability of Genetic Algorithm for Algorithm Optimization. Here the optimization of Dejong Function 2 is performed. At the initial stage, the genetic algorithm is implemented as the basic architecture on this algorithm. Further, the analysis is performed in form of the different selection algorithm. Here, three main selection algorithms Rank Selection, Roulette Wheel Selection and Elitism Selection Algorithm are discussed comparatively to identify the suitability of selection function.

5. REFERENCES

- [1] Victor M. Kureichick, "Genetic Algorithm for Solution of the Traveling Salesman Problem with New Features against Premature Convergence", Working Paper, 1996.
- [2] Rajeev Kumar, "Evolution of Hyperheuristics for the Biobjective 0/1 Knapsack Problem by Multiobjective Genetic Programming", GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA, pp 1227-1234
- [3] Christian Horoba, "Ant Colony Optimization for Stochastic Shortest Path Problems", GECCO'10, July 7–11, 2010, Portland, Oregon, USA, pp 1465-1475
- [4] Dervis KARABOGA, "A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm", Turk J ElecEngin, VOL.12, NO.1 2004, cTUB_ ITAK
- [5] JORGE J. MORE, "Benchmarking derivative-free optimization algorithms". SIAM J. Optimization, Vol. 20 (1), pp.172-191, 2009
- [6] A. Rangel-Merino, "Optimization Method based on Genetic Algorithms". Apeiron, Vol. 12, No. 4, October 2005
- [7] Martin Pelikan, "BOA: The Bayesian Optimization Algorithm", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), I, pp 525-532
- [8] Gerhard Venter, "A Parallel Particle Swarm Optimization Algorithm Accelerated by Asynchronous Evaluations", 6th World Congresses of Structural and Multidisciplinary Optimization, 2005
- [9] Zhouchen lin, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix", In Intl. Workshop on Comp. Adv. in Multi-Sensor Adapt. Processing Aruba, 2008
- [10] Gulshan Singh, "Comparison of Multi-Modal Optimization Algorithms Based on Evolutionary Algorithms". GECCO '06 Proceedings of the 8th annual conference on Genetic and evolutionary computation, Pages 1305-1312
- [11] Marcin Molga, "Test functions for optimization needs", 3 kwietnia 2005
- [12] Jeffrey W. Heath, "New Global Optimization Algorithms for Model-Based Clustering". Journal of Computational and Graphical Statistics. 2009
- [13] M. A. Panduro, "A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays", Progress In Electromagnetics Research 2008