

Preliminary experiments with the Andean Condor Algorithm to Solve Problems of Continuous Domains

Boris Almonacid¹

¹Global Change Science

Corresponding author:
Boris Almonacid¹

Email address: boris.almonacid@globalchange.science

ABSTRACT

In this article a preliminary experiment is carried out in which a set of elements and procedures are described to be able to solve problems of continuous domains integrated in the Andean Condor Algorithm. The Andean Condor Algorithm is a metaheuristic algorithm of swarm intelligence inspired by the movement pattern of the Andean condor when searching for its food. An experiment focused on solving the problem of the function 1st De Jong's $f(x_1 \dots x_n) = \sum_{i=1}^n x_i^2$, $-100 \leq x_i \leq 100$. According to the results obtained, solutions have been obtained close to the overall optimum value of the problem.

1 INTRODUCTION

The Andean Condor Algorithm is a metaheuristic algorithm inspired by the movement pattern of the Andean condor when searching for its food [1]. It is a swarm intelligence algorithm [5, 3, 4], in which a balance of its population is made by means of a performance indicator. This balance will indicate which Andean condors will carry out an exploration or intensification procedure. The feature of Andean Condor Algorithm is that it has a flexible design, that is, it can be adapted to be used in problems with variables from different domains, such as binary, discrete or continuous.

The objective of this research is to be able to solve an optimisation problem described by a continuous domain function. The benefit of this research is to be able to integrate into the Andean Condor Optimization a set of elements and methods for continuous domains. In this way, the scope is extended to solve various types of problems with the Andean Condor Optimization.

The rest of the scientific article is organised as follows: Section 2 describes the Andean Condor Algorithm with the necessary modifications to be able to solve problems of continuous domains. Section 3 shows the experiment carried out and the results obtained. Finally, section 4 indicates the conclusions and indicates the guidelines for future work.

2 ANDEAN CONDOR ALGORITHM APPLIED TO CONTINUOUS DOMAINS

In this section the elements that have been used in the Andean Condor Algorithm to integrate continuous domains problems will be described. These elements are the essential elements for the integration of a continuous domain problem in the Andean Condor Algorithm. The elements that make up the structure of the Andean Condor Algorithm are the process of generation of random solutions, the exploration process and the intensification process. A general detailed explanation of Andean Condor Algorithm can be found in [1].

2.1 Andean Condor

The structure of an Andean Condor is composed of three main elements.

- Variables: A set the variables $x = \{x_1, x_2, \dots, x_n\}$, with a domain between $[LB, UB]$, where LB meaning lower bound, and UB meaning upper bound.
- Status: A binary value {True, false} that indicates the status procedures of the Andean condor.
- Fitness: A float value.

2.2 Initial Solutions

In order to be able to carry out initial solutions for the population of Andean condors, one technique consists of making random solutions for each set of Andean condor variables. To do this, for each variable a random value between LB and UP is determined.

2.3 Exploration

To perform the exploration movement in the ACA algorithm, half of the variables of an Andean condor are randomly chosen. Subsequently, for each variable chosen, the equation 1 is applied. Where δ corresponds to a random number within a distribution between $U \sim [LB - x_i^t, UB - x_i^t]$, and t corresponds to the iteration.

$$x_i^{(t+1)} = x_i^t + \delta \quad (1)$$

2.4 Intensification

To perform the intensification movement, equation 2 is used, where the movement of intensification to a variable is applied. The type of intensification to apply is conditioned by the option selector θ . The value of θ is obtained randomly between the numbers 1, 2, 3, 4.

$$x_i^{(t+1)} = \begin{cases} x_i^t + \alpha & \text{if } \theta = 1 \\ x_i^t + \beta & \text{if } \theta = 2 \\ x_i^t + \gamma & \text{if } \theta = 3 \\ x_i^t + \varepsilon & \text{if } \theta = 4 \end{cases} \quad (2)$$

Where, α is a random number within a distribution between $U \sim [LB * 0.3, UB * 0.3]$, β is a random number within a distribution between $U \sim [LB * 0.1, UB * 0.1]$, γ is a random number within a distribution between $U \sim [-1, 1]$, and finally ε is a random number within a distribution between $U \sim [-0.05, 0.05]$.

3 EXPERIMENT

3.1 Experiment configuration

For the experiment the 1st De Jong's function was chosen [2], expressed in the equation 3. The function has a *Lower Bound* (LB) of -100, a *Upper Bound* (UP) of 100, a minimum value of $f(0, \dots, 0) = 0$, and a element n that represents the number of dimensions. The figure 1 illustrates a 3-dimensional display of the 1st De Jong's continuous function.

$$f(x_1 \cdots x_n) = \sum_{i=1}^n x_i^2 \quad (3)$$

$$-100 \leq x_i \leq 100$$

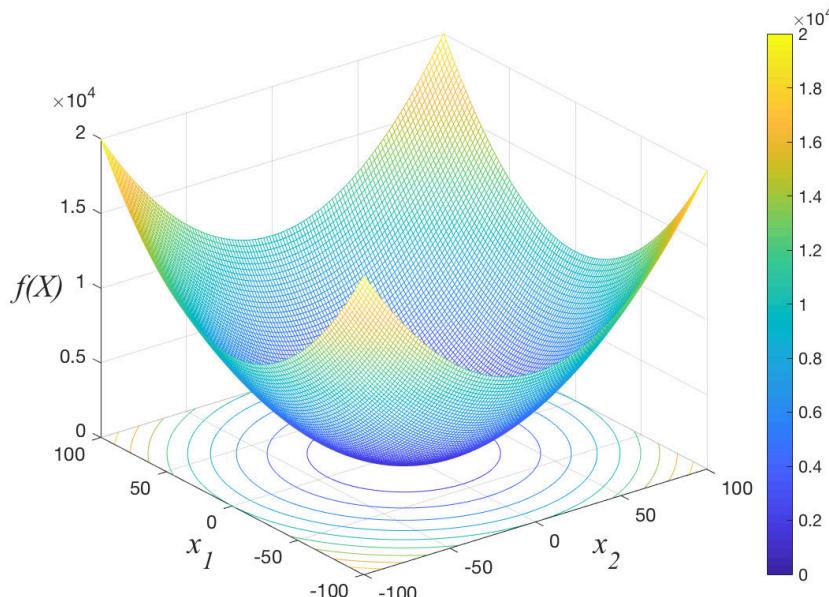


Figure 1. Sphere model. Technical name: Sphere Model (Spherical Contours, Square Sum, Harmonic, 1st De Jong's or Schumer-Steiglitz's Function No.01).

The configuration of the Andean Condor Algorithm used in the experiment corresponds to:

- Population of 25 Andean Condors.
- DP value in 0.5.
- PC value in 0.2.

In addition, the 1st De Jong's problem has been configured with 10 dimensions, 51 executions have been made for the problem, and a stop criterion of the Andean condor algorithm of 10,000 iterations has been configured.

3.2 Results

The best value found and the value of the variables is described below:

- Best value found: 7.55761E-05 (0.0000755761384167687).

- Variables: $x = [-3.63870\text{E-}03, 3.71562\text{E-}03, -5.93326\text{E-}04, 4.63425\text{E-}03, -1.73232\text{E-}03, -9.32016\text{E-}04, 1.31207\text{E-}03, -3.32202\text{E-}03, -3.15431\text{E-}03, 3.53939\text{E-}04]$.

The figure 2 describes the convergence of the algorithm for the best execution. That is, the execution that achieved the best solution. According to what is observed in the figure, the algorithm describes a convergence with a tendency to 0 between iterations 1 and 200. While from iteration 200 onwards, the algorithm concentrates on finding a local value close to 0.

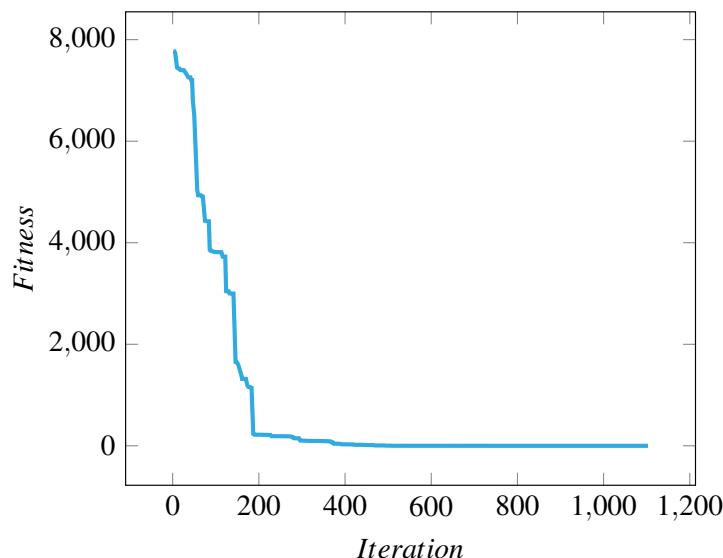


Figure 2. Convergence chart for the best solution found.

According to all the executions carried out, the following general results can be extracted:

- Worst value found: $1.39177\text{E-}02$ (0.0139176576452304), see execution 14 in table 1 and 2.
- Average: $1.43982\text{E-}03$
- Standard deviation: $2.41170\text{E-}03$
- Median: $4.33096\text{E-}04$

These results have been obtained by means of table 1 and 2. These tables describe the best value found in each iteration, in addition they describe the solution variables.

Table 1. Results obtained for the 51 executions part 1. The best value found for each iteration is described, and the solution variables from x_1 to x_4 .

Runs	Fitness	x_1	x_2	x_3	x_4
1	3.51915E-04	-2.75684E-03	-5.61110E-03	-3.91691E-03	-4.14229E-03
2	2.01866E-04	1.65102E-03	-3.02571E-03	-1.10184E-02	1.60338E-03
3	2.56437E-04	1.04766E-03	-9.89208E-04	9.95847E-03	-6.49856E-03
4	1.85462E-03	1.64811E-03	2.70429E-03	5.69402E-04	1.07365E-03
5	9.03293E-05	5.30856E-04	2.57298E-03	2.79539E-03	1.05917E-03
6	2.80646E-03	9.46281E-04	-3.10956E-03	-2.05757E-03	2.80132E-03
7	2.66713E-04	2.18173E-03	5.70659E-04	-8.51859E-03	-2.81482E-03
8	1.45970E-03	4.57272E-03	-3.31764E-02	7.68626E-03	-8.97136E-03
9	2.41536E-04	8.96258E-03	-3.44296E-04	5.95328E-04	-2.92045E-03
10	3.62368E-04	1.16019E-02	-8.54221E-04	3.04226E-03	1.35446E-02
11	1.20746E-03	-3.27931E-03	5.87958E-03	8.26694E-03	-2.29950E-02
12	4.72141E-03	4.60635E-04	-1.02437E-02	7.14664E-04	1.07081E-02
13	2.07874E-04	3.10879E-03	7.26111E-03	-3.47289E-03	2.59002E-04
14	1.39177E-02	-1.94019E-03	1.17300E-01	-6.07590E-04	-9.11073E-04
15	1.36415E-03	-1.68783E-03	-2.61033E-02	4.11816E-03	1.02096E-02
16	3.31498E-03	-6.30660E-03	-4.72272E-03	-6.19568E-03	1.66377E-03
17	1.24781E-04	-8.19605E-03	3.97838E-03	2.57659E-04	-1.25699E-03
18	1.69619E-04	5.94903E-04	-2.67444E-03	-8.53281E-04	-1.09748E-03
19	7.55761E-05	-3.63870E-03	3.71562E-03	-5.93326E-04	4.63425E-03
20	4.66217E-04	5.14852E-03	6.06774E-03	5.74498E-04	8.75551E-03
21	3.05860E-03	-1.31201E-02	1.28474E-02	-5.87972E-04	4.81562E-02
22	2.13416E-03	-2.02962E-03	-1.58508E-03	6.08860E-04	-3.08392E-03
23	4.58075E-04	-4.06143E-04	1.80308E-02	8.77044E-04	-4.23830E-03
24	6.41870E-03	1.19320E-02	2.90502E-03	-1.18107E-02	2.76212E-02
25	1.56878E-04	1.93634E-03	-5.04496E-03	-1.80577E-04	-1.18642E-03
26	1.23578E-03	-2.96796E-04	-1.95154E-02	4.74586E-03	1.41675E-03
27	3.62230E-04	2.88765E-03	1.84867E-03	-1.58525E-03	-6.04912E-03
28	2.01618E-03	-4.17557E-03	8.05914E-03	2.83616E-02	-5.70229E-04
29	9.39848E-05	1.08437E-03	-1.42961E-03	-4.06143E-03	-4.99166E-04
30	2.98243E-04	6.06102E-03	-6.90263E-03	1.23075E-02	4.34421E-03
31	3.31800E-04	-7.58205E-03	-2.08201E-04	-6.76189E-03	3.63188E-03
32	7.40846E-04	2.89644E-03	-1.79768E-03	4.16556E-03	-8.45386E-03
33	1.65309E-04	-9.37704E-03	3.93445E-03	-1.53178E-03	-5.22051E-04
34	4.34575E-03	5.12024E-03	8.97103E-05	-3.93810E-02	-8.81622E-03
35	4.80953E-04	9.56026E-03	3.08381E-03	-8.44327E-03	2.38210E-03
36	1.30450E-03	2.78951E-03	8.02345E-03	-1.58599E-04	2.59613E-03
37	5.49695E-04	-3.73045E-03	2.63508E-04	8.26195E-04	-1.77835E-02
38	2.05636E-04	6.89223E-03	-9.31673E-04	-4.21581E-03	-6.37329E-03
39	3.74484E-04	-5.95252E-03	-6.42049E-04	-1.08622E-02	-5.60421E-04
40	3.72051E-04	3.22479E-04	-1.54983E-03	-1.25657E-03	4.12088E-03
41	8.99300E-04	1.11899E-03	-2.34576E-03	-8.58673E-04	-1.59095E-03
42	2.63604E-04	-1.62404E-04	2.33881E-03	9.40476E-03	5.98086E-04
43	3.14265E-03	4.54396E-04	3.43496E-03	-5.43155E-02	-7.88314E-03
44	4.33096E-04	-1.23212E-02	1.78204E-03	7.88463E-03	5.62922E-03
45	8.85122E-04	1.92988E-04	2.60547E-02	4.07374E-03	-2.77745E-03
46	1.30134E-04	2.46884E-03	-3.35829E-03	1.52859E-04	-3.96423E-04
47	2.27097E-04	6.49925E-04	-2.58192E-03	2.11979E-04	-1.28660E-02
48	7.49492E-03	-1.72304E-03	-5.75874E-02	5.16715E-03	-4.77641E-02
49	3.71787E-04	1.52212E-03	1.36899E-03	7.02026E-03	-6.05863E-04
50	6.92663E-04	8.94140E-03	6.78584E-04	-1.44146E-02	1.01625E-03
51	3.24954E-04	4.00738E-03	-9.05209E-03	-6.06604E-03	9.76055E-03

Table 2. Results obtained for the 51 executions part 2. The solution variables are described from x_5 to x_{10} .

Runs	x_5	x_6	x_7	x_8	x_9	x_{10}
1	-1.39126E-02	5.93623E-04	-8.20481E-03	-7.20787E-04	1.95935E-03	3.83916E-03
2	4.06082E-03	-1.63934E-03	6.71758E-03	8.11671E-04	-9.89200E-04	2.58852E-04
3	9.70045E-04	-8.38475E-03	2.47623E-04	6.40565E-03	-7.01326E-04	-3.57588E-04
4	2.05332E-03	-1.36650E-04	1.01428E-02	4.11424E-02	5.24145E-03	-3.97997E-03
5	-2.50969E-03	-4.48013E-03	4.71819E-03	-5.04792E-03	5.68364E-04	-2.34956E-04
6	-5.62736E-03	5.19069E-02	-5.837733E-03	-4.69472E-03	1.30357E-03	-3.22407E-06
7	2.82267E-04	5.40235E-03	1.79326E-03	-1.02352E-04	1.02398E-02	6.61760E-03
8	6.53463E-03	3.12894E-03	-5.18044E-03	-1.96510E-03	-7.59784E-03	-7.59177E-03
9	-5.44516E-03	6.36252E-03	-2.18547E-03	-1.64447E-03	-1.82001E-03	-8.44285E-03
10	2.86473E-03	-3.25479E-04	-9.14660E-05	1.45791E-04	4.93513E-03	1.27529E-03
11	-1.39174E-02	8.04345E-03	-8.71083E-03	-9.53469E-04	1.50204E-02	-2.05641E-03
12	3.15834E-04	1.21326E-02	-2.46832E-03	-7.61820E-03	6.54763E-02	-1.58715E-03
13	2.71357E-03	-9.71038E-03	3.75532E-03	-1.02666E-03	3.74265E-03	-1.59331E-03
14	-3.04735E-03	-1.31431E-03	-5.64423E-03	-9.98006E-03	-2.43845E-03	2.22366E-03
15	-1.33176E-02	-5.23066E-03	2.84721E-05	3.18626E-04	-1.64998E-02	9.03625E-03
16	1.11352E-02	-2.62831E-03	-1.72373E-02	-1.73202E-02	4.98249E-02	-1.10449E-03
17	9.32103E-05	1.24838E-03	-1.21205E-03	-2.53910E-03	5.49468E-03	6.76293E-04
18	7.45962E-03	1.55286E-03	3.56546E-03	8.50735E-04	-2.88969E-04	-9.41291E-03
19	-1.73232E-03	-9.32016E-04	1.31207E-03	-3.32202E-03	-3.15431E-03	3.53939E-04
20	-1.24671E-02	3.65364E-03	1.08646E-02	2.79367E-04	4.01509E-03	-4.78390E-03
21	2.43608E-03	1.59076E-02	-1.62219E-03	-1.14811E-02	2.80446E-03	-8.66939E-04
22	1.79580E-02	1.33116E-02	6.57530E-03	-1.52693E-02	1.36184E-03	-3.66022E-02
23	-2.16626E-03	-4.22571E-03	5.94273E-04	-5.62566E-04	-1.30710E-03	9.44146E-03
24	6.14590E-03	4.35755E-03	-6.05438E-03	1.45551E-02	-7.11288E-02	9.45289E-04
25	2.62747E-03	8.74788E-03	3.53608E-03	-5.28826E-03	-5.00827E-04	1.44473E-03
26	1.52606E-02	1.04697E-02	-2.01744E-02	6.60985E-03	6.02311E-03	9.13009E-04
27	-4.22431E-04	1.32036E-02	-2.72066E-03	-3.86215E-03	1.38910E-04	1.07013E-02
28	5.35245E-03	4.37261E-04	-8.44895E-04	3.31930E-03	3.25089E-02	5.62952E-03
29	3.64803E-03	2.01451E-03	-3.54454E-03	2.70466E-04	-3.31328E-03	5.74802E-03
30	-2.48996E-03	1.99654E-03	3.27167E-03	2.58935E-04	5.67755E-04	-4.71522E-03
31	4.46571E-03	-5.16416E-03	4.80238E-03	1.61368E-03	1.15305E-02	-3.18222E-03
32	7.80257E-04	1.57806E-02	1.81007E-02	5.42831E-03	4.30992E-03	3.88491E-03
33	-1.68764E-03	5.67566E-04	2.79428E-05	2.29590E-03	1.36308E-03	-6.99867E-03
34	-5.15520E-02	-1.62859E-03	2.04702E-03	-3.60294E-03	3.50339E-03	-1.11143E-03
35	4.51254E-03	8.19866E-03	-1.25349E-03	-8.80369E-03	1.56399E-03	1.15749E-02
36	1.04526E-02	-6.57197E-03	-1.60412E-02	-4.29251E-03	-8.23044E-03	-2.70120E-02
37	5.73500E-03	-8.10106E-03	9.83353E-03	5.03120E-05	-4.82508E-03	-5.21908E-04
38	-8.60191E-04	-2.32423E-03	1.27847E-03	3.19261E-03	-3.91713E-04	-8.98614E-03
39	-7.45553E-03	-3.10428E-03	1.23711E-02	-8.85988E-04	6.14359E-04	-9.54568E-04
40	-8.25931E-03	-2.32325E-03	-4.42173E-03	1.60220E-02	-9.10136E-04	-5.35570E-04
41	-2.31365E-03	-1.88031E-03	-1.56628E-03	2.16752E-03	-1.79737E-03	2.94959E-02
42	-1.37378E-03	-4.45166E-04	5.88166E-03	1.83270E-03	-1.12011E-02	-1.94899E-03
43	-4.04997E-03	-7.00124E-03	4.16786E-03	2.22631E-03	5.05859E-03	-2.23434E-03
44	1.05325E-02	-8.03155E-04	-1.14325E-03	-1.89652E-03	-1.45745E-03	8.10218E-03
45	-2.43699E-03	2.31361E-03	8.62081E-03	9.46664E-03	6.70346E-04	2.50055E-03
46	6.40138E-03	4.15142E-04	-2.10122E-03	4.24741E-03	-6.75579E-03	1.82583E-03
47	-1.37555E-03	-3.08180E-03	4.47516E-03	-1.81831E-05	4.77822E-03	-4.26478E-04
48	1.32208E-02	-1.10047E-02	-1.66990E-03	-7.25176E-04	1.36244E-04	3.96019E-02
49	1.16560E-02	4.19905E-03	8.16236E-03	1.08198E-03	5.22750E-04	-9.81745E-03
50	-1.52990E-02	-7.16693E-03	-3.02310E-03	-1.03299E-02	-1.37255E-03	-5.35399E-04
51	-1.45088E-03	2.52216E-03	-9.27392E-03	-1.41419E-05	6.01077E-04	2.36817E-04

4 CONCLUSIONS

This research focused on presenting preliminary results of the use of Andean Condor Algorithm to solve problems of continuous domains. It has been possible to solve a problem through an experiment, giving results close to the optimum value described in the problem of the 1st De Jong's function. According to the flexibility shown in the implementation of this preliminary experiment it is possible to extend the research along this line of problems of continuous functions. A starting point to consider could be to focus on guidelines present each year at the CEP conference¹ [2], which presents a set of problems to solve. Finally, there is always the open window to solve various discrete problems using ACA. Including making an extended version of the ACA algorithm in order to solve multi-objective problems.

ACKNOWLEDGMENTS

Boris Almonacid is supported by Global Change Science, and by Ph.D (h.c) Sonia Alvarez, Chile. The founders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

REFERENCES

- [1] Boris Almonacid and Ricardo Soto. Andean condor algorithm for cell formation problems. *Natural Computing*, pages 1–31, 2018.
- [2] NH Awad, MZ Ali, JJ Liang, BY Qu, and PN Suganthan. Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization. In *Technical Report*. Nanyang Technological University Singapore, 2016.
- [3] Iztok Fister Jr, Xin-She Yang, Iztok Fister, Janez Brest, and Dušan Fister. A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*, 2013.
- [4] Adam Slowik and Halina Kwasnicka. Nature inspired methods and their industry applications—swarm intelligence algorithms. *IEEE Transactions on Industrial Informatics*, 14(3):1004–1015, 2018.
- [5] Vassilios Vassiliadis and Georgios Dounias. Nature–inspired intelligence: a review of selected methods and applications. *International Journal on Artificial Intelligence Tools*, 18(04):487–516, 2009.

¹http://www.ntu.edu.sg/home/epnsugan/index_files/cec2018/cec2018.htm