Vol. 04, No. 1 (2022) 63-68, doi: 10.24874/PES04.01.009

Proceedings on Engineering Sciences

www.pesjournal.net

APPLYING ANT COLONY OPTIMIZATION ALGORITHM TO SOLVE GEAR BOX DESIGN

Branislav Milenkovic[1]
Mladen Krstic
Djordje Jovanovic

Received 20.11.2021.
Accepted 27.01.2022
UDC – 004.4:629.5.062.3-048.34

Keywords:

Optimization; Ant; Algorithm; Gear box

A B S T R A C T

In this paper, we present the ant colony optimization algorithm (ACO for short), which was used to solve gear box problem design. Biological fundamentals of ACO algorithm, as well as method explanation are presented. The pseudo code for this algorithm was written using Matlab R2018a software suite. At the end, the results obtained by ACO algorithm are compared to the results previously obtained by other algorithms.

1. INTRODUCTION

In practice, optimization is used to find extreme values, either maxima or minima, of functions of single or multiple variables. The problems that are considered by this area of expertise must be precisely defined using the form of mathematical programming.

To this day, several heuristic methods have been developed, and many new have been conceived. Each and every one of these methods has certain advantages and disadvantages. These methods draw inspiration heavily from natural processes and events. Examples of such methods include genetic algorithm, artificial immune system algorithm, ant colony optimization algorithm, bee colony optimization algorithm, and many others.

In this paper, the ant colony optimization algorithm Kurapati and Azarm (2007), which is used for solving engineering problems, is applied. The engineering problem considered in this paper is that of a gear box problem design. The code for this algorithm was written in Matlab 2018a software suite.

ACO algorithm was developed by Marco Dorigo (1992). This algorithm draws inspiration from the behavior of a colony of ants. Ants use pheromone paths or trails as a means of orientation, getting from point A to point B by traversing the shortest route. Whilst moving, ants mark the route through which they pass, leaving a pheromone trail behind. If many ants use the same route, the pheromone trail will become stronger, making the following ants to pick that very route. After each aunt traverses its path, the goal function checks the amount of pheromones at each point.

2. ANT SYSTEM

The way the ants find food in nature was not known for a long time, until it was discovered that when ants find food, they leave a trail of pheromones behind them. Other

[1] Corresponding author: Branislav Milenković
Email: bmilenkovic92@gmail.com

ants, having felt the scent of pheromone, follow the same path to food, gather it, and ultimately return to the anthill. If the ant senses that a trail of pheromone has a stronger scent, that trail will be chosen. This is for two reasons: the first being pheromone evaporations, whilst the other is that the strength of the scent is directly proportional to the number of ants that traversed that path. This means that the more ants choose a given path, the bigger the probability the other ants will follow.

If, perchance, there is an obstacle on the path that splits it into two inequal paths (Figure 1), the ants will not know, at first, which of the two paths is shorter. Therefore, there is an equal chance that either of the paths will be chosen. However, since one of the paths features a shorter path, that is where the pheromone trail will grow stronger, since the pheromone trail will evaporate faster on the longer path. Having sensed that the scent of pheromone is stronger on one side, more and more ants choose this path, making the rest of ants follow the same path as well.

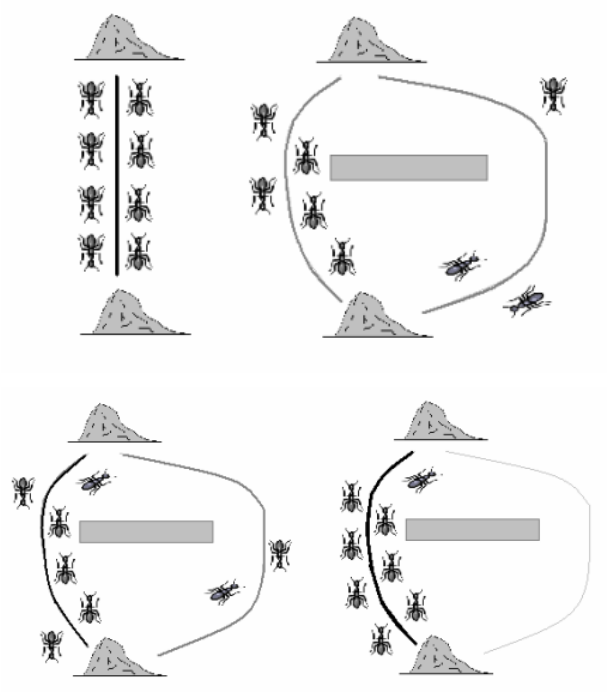

Figure 1. Basic ACO concepts (pheromone laying and shortest path searching)

An example of choosing between two different paths can be numerically represented in time in a different manner (Figure 2). In figure 2.1, the existing links between cities and their respective lengths are shown. In figure 2.2, half of the ants takes the longer path, while the other half takes the shorter. In figure 2.3, after the first few ants take the shorter road and leave a pheromone trail behind, the following ants will most probably take the very same trail. If the key elements of the path are substituted by points, what this image comes down to is a problem that is very similar to the travelling salesman problem.

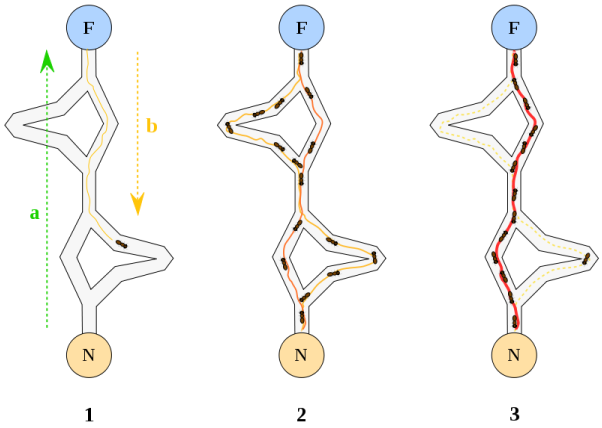

Figure 2. Experiment with Argentinian ants

Let us consider the travelling salesman problem. The number of cities to be traversed is n, while m is the total number of ants in the system, and $b_i(t)$ is the number of ants in city i at a given moment t.

In general, each and every ant decides at moment t about his whereabouts at moment t + 1, and is moved to that location in the next iteration of the algorithm. In such manner, all ants will pass through all the cities during the course of n iterations. In this algorithm, after each of the ants moved in the current iteration, the values for pheromones are updated by multiplying the old value with the evaporation rate ρ, and adding the product to the trail of each and that took that route (Equations 1 and 2)

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \quad (1)$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (2)$$

The number of pheromones that an ant adds to a certain route is equal to 0 if the ant did not pass through that route. If the ant, o the other hand, did take that route, he leaves a certain amount of trail that is inversely proportional to the length of the tour, as is shown in Equation 3.

$$\Delta\tau_{ij}^{k} = \frac{Q}{L_k} \quad (3)$$

In Eq. 3, Q is a constant, while $L_k$is the length of the tour the k-th ant traversed.

The probability of the next step is calculated based on Eq. 4:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}} \quad (4)$$

The previous equation is used only for cases where the ant has a path to take, meaning his taboo list does not contain all of the existing cities. α and β are parameters that govern the relationship between the trail and its visibility, whilst $\eta_{ij}$is the visibility between cities i and j.

3. ANT COLONY ALGORITHM

In the initial steps of the algorithm, the cities are to be created and placed, as well as initialization of pheromone trails is to be performed. After these steps, all of the ants are placed. During ant initialization taboo lists are created for each ant, and they are placed in their respective initial cities. What is to be noted is the fact that each ant has its own initial city. Therefore, this city is to be placed in the taboo list for that ant.

What followsis iterations during which the ants move. After n iterations, taboo lists are full for all the ants, so the pheromone trail is updated. The ants are then placed anew in a certain initial city, and their respective taboo lists are emptied. This cycle of n iterations is repeated until user abort, or reaching the upper limit of cycle number.

Flow diagram for ant colony optimization algorithm is shown in Figure 3. Standard ACO algorithm with its basic steps is shown in Figure 4.

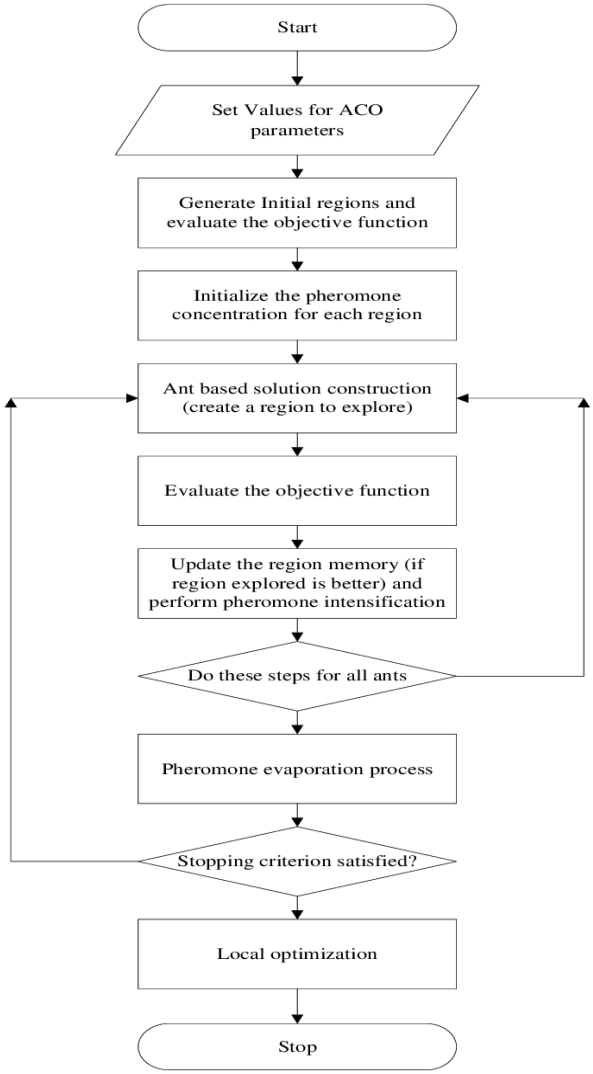

Figure 3. Flow diagram for ant colony optimization algorithm

ACO Algorithm
Input :pd, N
%%%% pd number of decision variables in ant, N iterations, Present position (ant) in the search universe X_{id} , ρ evaporation rate, %%%%%%%%
Output: Best_Solution
1: Initianlize_Node_Graph();
2: Initialize_Phermoni_Node();
3: While (num_of_Iterations>0) do
4: foreach Ant
5: η_j ***←objective function of the search space***
6: TRANSITION_RULE[j]= $p_j^m(t) = \frac{[\eta_j]\times[\tau_{ij}(t)]}{\sum_{i\in I_m}[\eta_i]\times[\tau_{ij}(t)]}$
7: Select node with the highest $p_j^m(t)$
8: Update Pheromone level $\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}(t)$
9: num_of_Iterations--;
10: end While
11:Best_sol←solution with best η_j
12: output(Best_sol)

Figure 4. The pseudo code for Ant Colony Optimization

4. GEAR BOX DESIGN

Gear box, shown in Figure 5, that is used in this paper, is taken from Kurapati and Azarm (2007), but with three goal functions (multicriteria optimization model), as described in (Huang et al, 2006). As it is to be seen in Figure 5, the problem consists of 7 project variables.

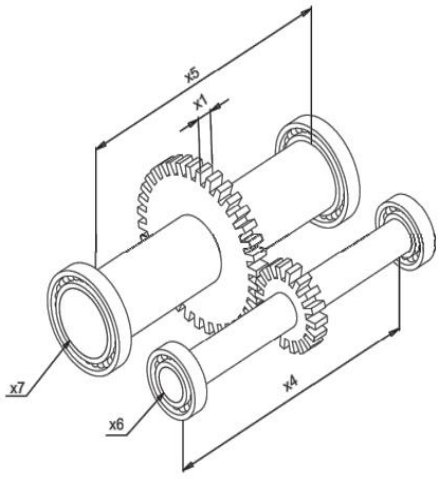

Figure 5. Gear box

The physical meaning of design variables and objective and constraint function are given in Table 1.

Table 1. The physical meaning of design variables and objective and constraint function

x_1	Gear face width (cm)	$g_2(X)$	Contact stress of teeth
x_2	Teeth module (cm)	$g_3(X)$	Transverse displacement of shaft 1
x_3	Number of teeth of pinion	$g_4(X)$	Transverse displacement of shaft 2
x_4	Distance between bearing 1 (cm)	$g_5(X)$	Generated torque constraint
x_5	Distance between bearing 2 (cm)	$g_6(X)$	Generated torque constraint

Table 1. The physical meaning of design variables and objective and constraint function (continued)

x_6	Diameter of shaft 1 (cm)	$g_7(X)$	Generated torque constraint
x_7	Diameter of shaft 2 (cm)	$g_8(X)$	Generated torque constraint
$f_1(X)$	Volume of the gear box (cm^3)	$g_9(X)$	Generated torque constraint
$f_2(X)$	Stress of shaft 1	$g_{10}(X)$	Stress of shaft 1
$f_3(X)$	Stress of shaft 2	$g_{11}(X)$	Stress of shaft 2
$g_1(X)$	Bending stress of teeth		

The mathematical formulation of the problem is given in Eq. (5) – (30).

$$F(X) = \{f_1(X), f_2(X), f_3(X)\}, \tag{5}$$

having:

$$f_1(X) = 0.7854x_1x_2^2\left(\frac{10x_3^2}{3} + 14.9334x_3 - 43.0934\right) - 1.508x_1(x_6^2 + x_7^2) + +7.4777(x_6^2 + x_7^2) + 0.7854(x_4x_6^2 + x_5x_7^2), \tag{6}$$

$$f_2(X) = \frac{A_1}{B_1,} \tag{7}$$

$$f_3(X) = \frac{A_2}{B_2,} \tag{8}$$

$$A_1 = \sqrt{(745x_4x_2^{-1}x_3^{-1}) + 1.69 \times 10^7}, \tag{9}$$

$$B_1 = 0.1x_6^3, \tag{10}$$

$$A_2 = \sqrt{(745x_5x_2^{-1}x_3^{-1}) + 1.575 \times 10^8}, \tag{11}$$

$$B_2 = 0.1x_7^3, \tag{12}$$

with constraint functions:

$$g_1(X) = 27x_1^{-1}x_2^{-2}x_3^{-3} - 1 \leq 0, \tag{13}$$

$$g_2(X) = 397.5x_1^{-1}x_2^{-2}x_3^{-2} - 1 \leq 0, \tag{14}$$

$$g_3(X) = 1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} - 1 \leq 0, \tag{15}$$

$$g_4(X) = 1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} - 1 \leq 0, \tag{16}$$

$$g_5(X) = x_2x_3 - 40 \leq 0, \tag{17}$$

$$g_6(X) = x_1x_2^{-1} - 12 \leq 0, \tag{18}$$

$$g_7(X) = 5 - x_1x_2^{-1} \leq 0, \tag{19}$$

$$g_8(X) = 1.9 - x_4 + 1.5x_6 \leq 0, \tag{20}$$

$$g_9(X) = 1.9 - x_5 + 1.5x_7 \leq 0, \tag{21}$$

$$g_{10} = \frac{A_1}{B_1} - 1300 \leq 0, \tag{22}$$

$$g_{11} = \frac{A_2}{B_2} - 850 \leq 0, \tag{23}$$

Whilst the variables are subjected to following constraints:

$$2.6 \leq x_1 \leq 3.6, \tag{24}$$

$$0.7 \leq x_2 \leq 0.8, \tag{25}$$

$$17 \leq x_3 \leq 28, \tag{26}$$

$$7.3 \leq x_4 \leq 8.3, \tag{27}$$

$$7.3 \leq x_5 \leq 8.3, \tag{28}$$

$$2.9 \leq x_6 \leq 3.9, \tag{29}$$

$$5.0 \leq x_7 \leq 5.5, \tag{30}$$

The results of ACO algorithm are obtained in code written for Matlab R2018a software suite and are given in Table 2 and Table 3.

Table 2. Comparative results for gear box design optimization

Design variables	WNNC (Huang et al, 2006)	FMO (Huang et al, 2006)	H-CS-FA (Martinez et al, 2009)	**ACO**
x_1	3.58	3.58	3.59	3.58
x_2	0.71	0.70	0.70	0.70
x_3	21.5	23	17	21
x_4	7.90	7.36	8.26	7.83
x_5	8.07	8.14	8.00	8.04
x_6	3.54	3.45	3.90	3.83
x_7	5.43	5.40	5.50	5.47

Table 3. Comparative results for goal functions for gear box design optmization problem

Goal funtion	WNNC (Huang et al, 2006)	FMO (Huang et al, 2006)	H-CS-FA (Martinez et al, 2009)	**ACO**
$f_1(X)$	3412.1	3425	3356.4753	**4124.9**
$f_2(X)$	829.4	879.8	698.4919	**735.03**
$f_3(X)$	754.7	797.6	754.9155	**766.79**

Based on obtained results, shown in Table 3, the conclusion can be drawn that ACO algorithm gives better results for $f_2(X)$ and $f_3(X)$ than FMO and WNNC algorithms. The best results for all the three goal functions are obtained by using the H-CS-FA algorithm. However, it is mentioned in the paper that the standard deviation which was obtained was pretty high, meaning that the algorithm could have, at some point, fallen into the trap of a local minimum.

5. CONCLUSION

In this paper, the results obtained using the ACO algorithm are presented and compared to existed optimal results present in literature.

The gear box design were described in detail using mathematical formulation and figures, while the results were shown in tables. By further development of this algorithm, this method can be modified and improved in order to obtain better results.

Acknowledgement: This work was supported by the Serbian Ministry of Education, Scienceand Technological Development through Mathematical Institute of the Serbian Academy of Sciences and Arts.

References:

Dorigo, M. (1992) *Optimization, Learning and Natural Algorithms*. Ph.D. Thesis, Politecnico di Milano, Italian.

Huang, H. Z., Gub, Y. K., & Duc, X. (2006). An interactive fuzzy multi-objective optimization method for engineering design. *Engineering Applications of Artificial Intelligence, 19*, 451-460.

Kurapati, A., & Azarm, S. (2007). Immune network simulation with multi-objective genetic algoritms for multidisciplinary design optimization. *Engineering Optimization, 33*(2), 245-260.

Martinez, M., Herrero, J. M., Sanchis, J., Blasco, X. & Garcia, S. (2009). Applied Pareto multi-objective optimization by stochastic solvers. *Engineering Applications of Artificial Intelligence,* 22, 455-465.

Branislav Milenkovic
Mathematical Institute of the Serbian Academy of Sciences and Arts,
Belgrade,
Serbia
bmilenkovic92@gmail.com

Mladen Krstic
Faculty of Mechanical and Civil Engineering Kraljevo,
Kraljevo,
Serbia
mladenkrstic994@gmail.com

Djordje Jovanovic
Mathematical Institute of the Serbian Academy of Sciences and Arts,
Belgrade,
Serbia
giorgaki.jovanovic@gmail.com