Earthline Journal of Mathematical Sciences
ISSN (Online): 2581-8147
Volume 5, Number 2, 2021, Pages 329-343
https://doi.org/10.34198/ejms.5221.329343

Water Distribution Network Design using Hybrid Self-adaptive Multi-population Elitist Pollination Intelligence (HSAMPEPI) Jaya Algorithm

Akintayo Emmanuel Akinsunmade[1,*] and Ibukun Isaac Aina[2]

[1] Department of Mathematics, Faculty of Physical Sciences, University of Ilorin, Ilorin, Nigeria
e-mail: akintayoakinsunmade@gmail.com

[2] Department of Mathematics, Faculty of Physical Sciences, University of Ilorin, Ilorin, Nigeria
e-mail: ibkisaac@gmail.com

Abstract

An optimization model to minimize the cost of designing water distribution network is presented in this study. The model was formulated to reduce the cost coefficient in a plumbing system. A new hybrid method of optimization was constructed by combining the search abilities of Jaya-based algorithm and pollination intelligence algorithm, and was used to solve the designed model. The model was implemented by obtaining geometrical information of a water distribution network layout stationed at Gaa Odota, Ilorin, Kwara State, Nigeria. Result obtained from the model showed a significant reduction in the cost coefficient compared to that of the study area.

1 Introduction

The cost associated with new water distribution networks and the extension of existing ones has led researchers to take great interest in employing mathematical methods to find optimal design strategy which will help to minimize the cost of materials. A properly distributed system allows flow of water through plumbing components such as pipes, reservoirs and valves, thus connecting consumers

Received: September 22, 2020; Revised & Accepted: October 9, 2020
2020 Mathematics Subject Classification: 91B76, 90C59, 90C90.
Keywords and phrases: mathematical model, metaheuristic methods, multi-population Jaya algorithm, network designs.

*Corresponding author

to source of water. Optimization processes involving design system for water supply can be defined as the best possible combination of reducing cost of design components, such that all water demands are met, as well as curbing the occurrence of system failures.

In practice, optimizing water distribution network takes numerous forms depending on the kind of components integrated into the network system. A large number of optimization models for water distribution and system network aiming at minimizing cost incurred, have been developed over the last three decades. A number of these models attempt to obtain optimal layout of network distribution to minimize cost, Pilar *et al.* [1], Idel *et al.* [2]. Researchers like Saldarriaga *et al.* [3] and Douglas *et al.* [4], have worked extensively on water distribution network (WDN) model putting into consideration sensitive factors, and has highlighted the importance of optimization methods in water distribution network.

Optimization methods most especially stochastic algorithms have been gaining significance in producing fast, low cost and robust solutions to complex problem problems, these methods of optimization have been applied to virtually all fields of life including engineering, science and finance to mention but a few. A significant number of these methods are algorithms that are derived from studying biological and environmental factors, human and animal activities, as well as chemical processes. Differential Evolution Method (DE), Storn and Price [5], Firefly Algorithm (FA), Yang [6], Wind Driven Optimization (WDO), Bayraktar *et al.* [7], Brain Storm Optimization (BSO), Shi [8], Flower Pollination Algorithm (FPA), Yang [9], Crow Search, Askarzadeh [10], Rainfall Optimization, Aghay *et al.* [11], to mention but a few, are different types of stochastic methods developed in recent years. Meanwhile, a significant number of these algorithms works well with parameter fitting, the implication of this is, the algorithm have to be provided with right parameter to work well in many instances. Other algorithm developed in recent years that tend to run on stochastic procedures with grouping of fitness functions are the Jaya-based-Algorithm, Rao [12], Rao and Patel [13]. Elitist-based self-adaptive multi-population (SAMPE) Jaya algorithm, (Rao and Saroj [14]) was designed to overcome parameter tuning known for most stochastic optimization algorithms. Basically, the advantage of SAMPE-Jaya algorithm over other stochastic algorithms is the adaptive tuning, massive generation of sub-populations and regrouping of solutions. For a broadened knowledge on the different variants of Jaya algorithm see (Rao [15]).

However, there exist numerous optimization algorithms, and it is difficult to test and determine which algorithm is most suitable for solving a particular optimization problem, because most of the algorithms work on generalized concept and do not have specific ability to every problem. In this situation, hybridization process gains importance as it combines the desirable properties of different approaches to mitigate their individual weaknesses. Improvement on solution accuracy, reduced computation time, enhancement of algorithm stability and the handling of searching convergence can be considered as targets of hybridization and improvement processes.

In this work, a hybrid version of SAMPE-Jaya Algorithm, (Rao and Saroj [14]) and Flower Pollination Algorithm (FPA), (Yang [9]) is constructed and used to solve water distribution network problem.

2 Materials and Methods

2.1 Model Formulation

A water distribution network model is being formulated as a least-cost optimization problem with a solution of pipe sizes. The pipe layout and its connectivity, nodal demand, and minimum requirement are known. The parameters involved in the water distribution network are presented in Table 1.

Minimize

$$C_T = \sum_{r=1}^{U} EL_r D_r^{1.997} \tag{1}$$

subject to

$$\sum_{r=1}^{U} RL_r (F_r)^{\beta} D^{-\gamma} = 0 \tag{2}$$

Table 1: Parameters of the model and definition.

Parameter	Definition
E	Cost coefficient
C_T	Cost of network
L	Pipe length
L_r	Length of pipe r
D	Pipe diameter
m	Exponent obtained by regression analysis
F_r	Flow in pipe r
M_j	Demand at node j
N	Total number of nodes in the network
U	Total number of pipes
R	Constant depending on pipe material
β	Exponent equal 2 for Darcy-Weibach formula = 1.85 in Hezen-William formula
γ	Exponent equation 5 in Darchy-Weisbach formula= 4.87 in Hezen-William formula
h^{min}	minimum residual pressure allowed
M^{min}	minimum demand
D^{min}	minimum diameter
F^{min}	minimum Flow in pipe k

where R is a constant depending on pipe material given as

$$R = \frac{\alpha}{C_{hw}^{1.85}} \tag{3}$$

from equation (3) above, $\alpha = 2.234 \times 10^{12}$ and $C_{hw} = 130$.

$$\sum_{r=1}^{U} F_r + \sum_{j=1}^{N} M_j = 0 \tag{4}$$

$$\sum_{r=1}^{U} RL_r(F_r)^{\beta} D^{-\gamma} h^{min_j} = 0 \tag{5}$$

$$L_r \geq L^{min} \tag{6}$$

$$D_r \geq D^{min} \tag{7}$$

$$F_r \geq F^{min} \tag{8}$$

to solve the model equations (1) to (8), two new class of optimization algorithm, Self-Adaptive Multi-population Elitist-Jaya Algorithm (SAMPE), (Rao and Saroj [14]) and Flower Pollination Algorithm (FPA), (Yang [9]), is presented. A newly constructed hybrid version of the two algorithms is also presented is also proposed and used to solving the model equation.

2.2 Self-Adaptive Multi-population Elitist-Jaya Algorithm

Let $G(u)$ be an objective function to be optimized. Assuming that at any iteration k, the design variable d is given as $x = 1, 2, \ldots, d$ with a generated population size P given as $r = 1, 2, \ldots, P$. If $U_{x,r,t}$ is the value of the xth variable for the rth candidate during the kth iteration, a new solution at iteration $k+1$ can be modified as

$$U_{x,r,k+1} = U_{x,r,k} + r_1(U_{x,best,k} - |U_{x,r,k}|) - r_2(U_{x,worst,k} - |U_{x,r,k}|), \tag{9}$$

where $U_{x,best,k}$ and $U_{x,worst,k}$ are the value of the xth variable for the best and worst solution in the generated population P. $U_{x,r,k+1}$ is the new value of $U_{x,r,k}$ at iteration $k+1$, r_1 and r_2, are randomly generated number in the range of $[0, 1]$. From equation (9), the new solution $U_{x,r,k+1}$ tries to approach the best solution given by the term $r_1(U_{x,best,k} - |U_{x,r,k}|)$ while at the same time tries to avoid worst solution given by the term $-r_2(U_{x,worst,k} - |U_{x,r,k}|)$. The new solution $U_{x,r,k+1}$ is accepted at every stage of iteration if there is improvement over the previous solution $U_{x,r,k}$.

This method uses number of sub-population by dividing it into numbers of groups based on the quality of the fitness value, inferior groups are characterized by the quality of the fitness value and are termed worst solution which are then replaced with the higher fitness value (elite solution) obtained from other groups. During search process, the number of sub-populations are modified by the algorithm based on the quality and strength of the problem. This determines either the sub-population will be increased or decreased at every stage of iteration. The basic operation of Jaya algorithms is random generation of populations at every stage of iteration.

Step 1: Start with setting the number of design variable (d), number of population to be generated (P), elite size (ES) and define a stopping criterion which can be set as the maximum number of function evaluation (F_{Emax}).

Step 2: Find the initial solution based on the defined fitness function for the defined problem using

$$U_{initial} = U_{min} + rand(0,1) \times (U_{max} - U_{min}). \tag{10}$$

rand(0,1) is a random number between 0 and 1 with a normal distribution whose mean is zero and the standard deviation is 1.

Step 3: Group the entire population (P) into m number of groups based on the quality of the solution and replaced the worst solution of the inferior group with solutions of the superior group.

Step 4: Using equation (9) in every sub-population, modify the solution in each group independently. Keep the modified solution if and only if they are better than the old solution generated in step 2.

Step 5: Combine the entire sub-population and check whether the previous solution of the entire population $U(best-before)$ is better than the current solution $U(best-after)$ of the entire solution. Increase the group number m by $1 (m = m + 1)$ if the value of $U(best - after)$ is better than $U(best - before)$, as this helps to increase the exploration feature of the search process. If otherwise, decrease the group number m by $1 (m = m-1)$ this shows the algorithm need to be more exploitative.

Step 6: If stopping condition is reached, terminate the loop and report best solution. Otherwise, replace the solution with randomly generated new solution and go to step 3.

2.3 The Flower Pollination Algorithm

Flower Pollination Algorithm (FPA) is a nature inspired algorithm that have the ability of solving different types of optimization problems. The principle behind the transfer of pollen grains in flowering plants which is ultimately reproduction process is mimicked to develop this algorithm. Pollen agents such as insects tends to visit flowering plants having been attracted by its nature(bright colour, scent), this process is categorized as biotic factor, meanwhile factors like rainfall, wind can also influence pollination process this is categorized as biotic factor. There are over 2000 varieties of pollinators, and they tend to behave by moving randomly. Based on this characteristic, the flower pollination algorithm was developed. For simplicity the following rules are used:

1. biotic and cross pollination are modeled as a process of global pollination, as pollen-carrying pollinators move in a random direction which obeys L*é*vy flight.
2. self-pollination and abiotic factors are used to generalized local pollination.
3. pollinators such as insects can develop flower constancy which is equivalent to reproduction probability that is proportional to the similarity of two flowers involved
4. the interaction of switching of local pollination and global pollination is controlled by a switch probability $p \in [0, 1]$, with a slight bias toward local pollination.

A set of updating formula are needed to convert the rules into updating equations. In global pollination, flower pollen gametes are carried by pollinators such as insects and pollen can travel over a long distance because insects can often fly and move in a much longer range. Rule 1 and flower constancy is represented as

$$u_i^{t+1} = u_i^t + L(\lambda)(g^* - u_i^t), \tag{11}$$

where u_i^t is the pollen or solution vector u_i at iteration t and g^* is the current best solution found among all solution at the current generation/iteration. $L(\lambda)$ is the parameter that correspond to the strength of the pollinators which is essentially the step size. Since insects may move over a long distance, Lévy flight is designed to mimic this characteristics efficiently. $L > 0$ is drawn from a Lévy distribution Rule 2 and Rule 3 can be represented as

$$u_i^{t+1} = u_i^t + e(u_j^t - u_k^t), \tag{12}$$

where u_j^t and u_k^t are pollen from different flowers of the same plant specie. This essentially mimics the flower constancy in a limited neighborhood. Mathematically, if u_j^t and u_k^t comes from the flower type or selected from the same population, this equivalently becomes a local random walk if we draw e from a uniform distribution in $[0, 1]$. Though flower pollination activities can occur at all scales, both local and global, adjacent flower patches or flowers in the not so far away neighborhood are more likely to be pollinated by locally. In order to mimic this, we can effectively use a switch probability(Rule 4) or proximity probability p to switch between common global pollination to intensive local pollination.

Flower Pollination Algorithm

```
Minimize $f(u)$, $u = (u_1, u_2, \dots, u_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution $g^*$ in the initial population
Define a switch probability $p \in [0, 1]$
Define a stopping criterion
while ($t < MaxGeneration$)

    for $i = 1 : n$ (all n flowers in the population)
      if $rand < p$,
        Draw a (d-dimensional) step vector L which obeys a Lévy distribution
        Global pollination via $u_i^{t+1} = u_i^t + L(\lambda)(g^* - u_i^t)$
      else
        Draw e from a uniform distribution in $[0, 1]$
        Do local pollination via $u_i^{t+1} = u_i^t + e(u_j^t - u_k^t)$
        end if
        Evaluate new solutions
        If new solutions are better, update them in the population
    end for
        Find the current best solution $g**$

end while
Output the best solution found
```

2.4 Hybrid Self-Adaptive Multi-Population Elitist Pollination Intelligence Jaya-Algorithm (HSAMPEPI)

The no free lunch theorem holds for all optimization algorithm, all algorithm have special abilities in solving a particular problem. In a bid to achieve a global optimum and have a more realistic and robust solution when using optimization algorithms, two or more methods are combined to sequentially perform search processes. Jaya algorithm are known for randomness, grouping and sub-grouping of random solutions, thus making the search process to go through the same process until an optimum solution is achieved. The grouping of generated fitness solution into superior and inferior groups makes Jaya-Algorithm a good method. Meanwhile, Flower Pollination Algorithm is a parameter based method, although it also randomly generates solution from an assigned population, this solution are not grouped, rather a best solution is only obtained at once while other solutions are discarded. Jaya-Algorithm can thus compliment Flower Pollination Algorithm through grouping the generated fitness value in a given population. To achieve this, the two algorithms are combined as one to have better algorithm.

The basic process involved in combining these algorithms is setting Flower Pollination Algorithm (FPA) to perform initial search process which by default is to be randomly generated by Jaya-Algorithm in equation (10). This process outsmart equation (10) as Flower Pollination Algorithm has solution generated from both exploitative and exploratory search process which will be categorized by Jaya-Algorithm into best and wost solutions. The hybrid process is well explained in steps below:

Step 1: Start with setting the number of design variable (d), number of population to be generated (P), elite size (ES) and define a stopping criterion which can be set as the maximum number of function evaluation (F_{Emax}) just as in original Jaya-Algorithm.

Step 2: Find the initial solution based on the defined fitness function for the defined problem using Flower Pollination Algorithm. This overwrites the randomly generated solution in the original Jaya-Algorithm and also have in store robust solution that have passed through local and global solution. The beauty of this is to ease the process at step 3, as Jaya-Algorithm will only be classifying best solutions and regrouping the least of the best solution as inferior in the generated population.

Step 3: Group the entire population(P) obtained from Flower Pollination Algorithm into m number of groups based on the quality of the solution and replaced the worst solution of the inferior group with solutions of the superior group.

Step 4: Using equation (9) in every sub-population, modify the solution in each group independently. Keep the modified solution if and only if they are better than the old solution generated in step 2.

Step 5: Combine the entire sub-population and check whether the previous solution of the entire population $U(best-before)$ is better than the current solution $U(best-after)$ of the entire solution. Increase the group number m by $1(m = m+1)$ if the value of $U(best-after)$ is better than $U(best-before)$, as this helps to increase the exploration feature of the search process. If otherwise, decrease the group number m by $1(m = m-1)$ this shows the algorithm need to be more exploitative.

Step 6: If stopping condition is reached, terminate the loop and report best solution. Otherwise, replace the solution with randomly generated new solution and go to step 3.

To implement the model, a case study of water network layout for Gaa Odota, Ilorin, Kwara State Nigeria is used. The network of the study area comprises of thirteen (13) pipes and nine (9) nodes, the head losses were calculated using Hazen-Williams equation and every diameter has a dimensionless roughness coefficient C equal to 130. The network design of the study area is presented in Figure 1. Using the data obtained from the study area as presented in Table 2.

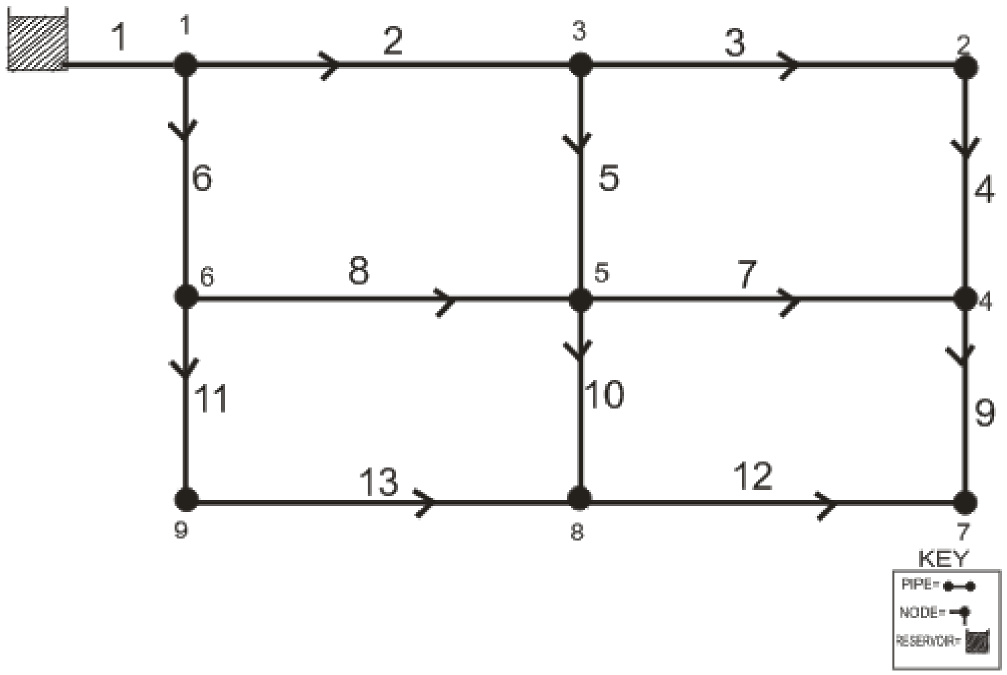

Figure 1: Pipe Network layout for Gaa Odota, Ilorin.

3 Result and Discussion

The model was solved using Self-Adaptive Multi-population Elitist-Jaya Algorithm (Rao and Saroj [14]), Flower Pollination Algorithm (Yang [9]) and the newly constructed Hybrid Self-Adaptive Multi-Population Elitist Pollination

Table 2: Geometric data of network layout for Gaa Odota, Ilorin.

Pipes	Length (m)	Diameter (mm)	Flow along the pipe (l/s)
1	150	200	0.472
2	120	100	0.366
3	100	200	0.305
4	100	190	0.305
5	130	200	0.397
6	125	100	0.381
7	120	200	0.365
8	155	115	0.473
9	110	100	0.336
10	150	200	0.458
11	150	200	0.458
12	130	200	0.397
13	100	200	0.305

Intelligence Jaya-Algorithm. Results obtained from the model are presented in tables below:

Table 3: Comparison of optimized pipe diameter obtained from the model using SAMPE, FPA and HSAMPEPI.

Pipe	Study Area (mm)	SAMPE (mm)	FPA (mm)	HSAMPEPI (mm)
1	200.00	182.00	200.00	199.99
2	100.00	102.95	99.30	99.29
3	200.00	133.11	107.21	101.73
4	190.00	117.67	190.67	136.08
5	200.00	128.69	128.05	114.13
6	100.00	96.92	96.00	95.18
7	200.00	176.07	199.45	190.49
8	115.00	200.00	196.71	127.99
9	100.00	198.28	151.12	190.25
10	200.00	177.83	198.98	199.04
11	200.00	121.06	200.00	95.04
12	200.00	160.83	145.28	174.81
13	200.00	127.73	95.60	95.00

Table 4: Comparison of optimized cost obtained from the model using SAMPE, FPA and HSAMPEPI.

Model Method	Cost of Network (Naira)
Study Area	12256449.44
SAMPE	9465714.67
FPA	10737968.51
HSAMPEPI	8756487.47

Result obtained from the model shows a significant reduction in the cost of the network distribution in the study area. Optimized diameter of the 13 pipes used in the network layout is presented in Table 3. A minimized network cost was obtained from the model using SAMPE, FPA and the newly constructed hybrid algorithm. Result obtained from Table 4 shows that the newly designed algorithm produced an improved cost of $8,756,487.47$ Naira, as against $10,737,968.51$ Naira and $9,465,714.67$ Naira, obtained using FPA and SAMPE respectively. It is worthwhile to note that all the three algorithms used in this study to solve the model equation, produced a reduced cost as against the geometric data of the study area presented in Table 2.

4 Conclusion

The cost of installation can be determined by the diameter(size) of a pipe. In a water distribution network, the diameter of pipe is an important factor that must be considered in layout designs. In this study, an optimized model was developed to minimize the cost that may be incurred in a water distribution network. The efficacy of Jaya-Algorithm and Stochastic Algorithm in solving water distribution network problem has also been achieved along side the construction of a hybrid version of Jaya-Algorithm and Stochastic Algorithm. The newly constructed Hybrid Self-Adaptive Multi-Population Elitist Pollination Intelligence Jaya-Algorithm (HSAMPEPI) minimized the cost of the network than other methods reported in this study. Optimization methods have a lot to do with helping structural engineers and plumbers by providing a cost effective framework to which water distribution network can be constructed.

References

[1] M. Pilar, G. Adela and L. A. Jose, Water distribution network optimization using a modified genetic algorithm, *Water Resources Research* 35(11) (1999), 3467-3473. https://doi.org/10.1029/1999WR900167

[2] M. Idel, I. Joaquin, P. Rafael and M.T. Michael, Particle swarm optimization applied to the design of water supply, *Computer & Mathematics with Application* 56 (2008), 769-776. https://doi.org/10.1016/j.camwa.2008.02.006

[3] J. Saldarriaga, D. Paez, P. Cuero and N. Leon, Optimal design of water distribution networks using mock open tree topology, *World Environmental and Water Resources Congress* (2013), 869-880. https://doi.org/10.1061/9780784412947.083

[4] Douglas F. Surco, Thelma P. B. Vecchi, Mauro A. S. S. Ravagnani, Optimization of water distribution networks using a modified particle swarm optimization algorithm, *Water Supply* 18(2) (2018), 660-678. https://doi.org/10.2166/ws.2017.148

[5] R. Storn and K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11 (1997), 341-359. https://doi.org/10.1023/A:1008202821328

[6] X. S. Yang, *Nature Inspired Metaheuristic Algorithm*, Luniver Press, Cambridge, United Kingdom, 2008.

[7] Z. Bayraktar, M. Komurcu and D. H. Werner, Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetic, 2010 IEEE Antennas and Propagation Society International Symposium, Toronto, Canada, 2010, pp. 1-4. https://doi.org/10.1109/APS.2010.5562213

[8] Y. Shi, An optimization algorithm based on brainstorming process, *International Journal of Swarm Intelligence Research* 2(4) (2011), 35-62. https://doi.org/10.4018/ijsir.2011100103

[9] X. S. Yang, Flower pollination algorithm for global optimization, *Unconventional Computation and Natural Computation*, Lecture Notes in Computer Science, vol. 7445, Springer, Berlin, Heidelberg, 2012, pp. 240-249. https://doi.org/10.1007/978-3-642-32894-7_27

[10] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, *Computers and Structures* 169 (2016), 1-12. https://doi.org/10.1016/j.compstruc.2016.03.001

[11] S. Hr. Aghay Kaboli, J. Selvaraj and N. A. Rahim, Rainfall optimization: A population based algorithm for solving constrained optimization problems, *Journal of Computational Science* 19 (2017), 31-42. https://doi.org/10.1016/j.jocs.2016.12.010

[12] R. V. Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, *Int. J. Ind. Eng. Comput.* 7(1) (2016), 19-34. https://doi.org/10.5267/j.ijiec.2015.8.004

[13] R. V. Rao and V. K. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, *Int. J. Ind. Eng. Comput.* 3(4) (2012), 535-560. https://doi.org/10.5267/j.ijiec.2012.03.007

[14] R. V. Rao and A. Saroj, An elitism-based self-adaptive multi-population Jaya algorithm and its applications, *Soft Comput.* 23 (2019), 4383-4406. https://doi.org/10.1007/s00500-018-3095-z

[15] R. V. Rao, *Jaya: An Advanced Optimization Algorithm and its Engineering Applications*, Springer International Publishing, Switzerland, 2019.