Ananya Algorithm: A Simple and New Optimization Algorithm for Engineering Optimization

Neeraj Agarwal
Department of Mechanical Engineering
UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya,
Bhopal, India
E-mail: neeraj.bhopal@gmail.com

Nitin Shrivastava
Department of Mechanical Engineering
UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya,
Bhopal, India

M. K. Pradhan
Department of Mechanical Engineering
Maulana Azad National Institute of Technology
Bhopal, India

2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE) | 978-1-7281-9061-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICNTE51185.2021.9487788

***Abstract*— Optimization is the act of making the best or most effective use of a situation or resource. It involves maximizing or minimizing a mathematical function. A new and simple metaheuristic optimization algorithm is developed and proposed in this paper as Ananya Algorithm. Simplicity is the beauty of the proposed algorithm. Ananya algorithm is one of the simplest optimization algorithms to implement, among all optimization techniques. This algorithm has only two candidates hence it avoids large calculations. This algorithm moves towards a better solution with the difference between the mean of variables and the best variable. This algorithm works on simple calculations and does not involve any complicated calculations. This algorithm is tested for thirty unconstrained benchmark functions like sphere function, Beale function, Goldstein-Price function, Booth Function, Matyas Function, and convergence graph shown for the same. Every time this algorithm got successful to achieve an optimum solution. It takes a little CPU time to optimize. Ananya algorithm is compared to particle swarm optimization (PSO) and genetic algorithm (GA). It required lesser mean functional evaluation to achieve an optimal solution, hence the Ananya algorithm has better performance than the two algorithms.**

Keywords*— Optimization; New Algorithm; Ananya Algorithm; Metaheuristic; Advanced Optimization; Multi-Objective Optimization; Single Objective Optimization.

I. INTRODUCTION

Engineering optimization requires some optimization techniques to optimize a given objective function. The classical and analytical optimization technique involves some limitations [1]. This method sometimes fails for a complex problem. Optimized to local minima while searching for global minima. There is difficulty to solve the problem if the number of design variables increases. To overcome the limitation of classical optimization, advanced optimization techniques were developed by researchers. It is known as meta-heuristic optimization techniques [2]. Nowadays optimization based on meta-heuristic become more popular. An algorithm based on stochastic algorithms with randomization is known as a metaheuristic. Metaheuristics works on trial and error. Nature-inspired metaheuristics algorithms are mimicking on nature or biological systems [3]. Population-based metaheuristics require some common parameters like population size, generation size, etc. Ant Colony Optimization (ACO) is inspired by ant colony behavior [4]. Genetic Algorithm (GA) is a very popular optimization algorithm, it requires many algorithm-specific parameters like mutation, crossover, and operator selection [5]. Particle Swarm Optimization (PSO) is the most popular metaheuristic technique, it involves algorithm-specific parameter selection like inertia weight, maximum velocity [6]. The inappropriate parameter value may lead to the inefficient performance of the algorithm, hence this makes the algorithm more complicated [7]. Harmony Search algorithm (HS) is based on the musical process of searching for a solution with a perfect state of harmony [8]. Grey wolf optimizer (GWO) is a metaheuristic algorithm, which is inspired by grey wolves [9]. GWO mimics the hunting behavior of wolves. All of the above optimization discussed is a population-based optimization algorithm where multiple candidates participate in the solution simultaneously and they share information to improve the solution. Simulated Annealing (SA) is a single candidate based optimization method that used a single candidate. The solution moves towards optimum solution iteratively [10]. Butterfly optimization algorithm (BOA) is also a nature-inspired algorithm which mimics the behavior of butterfly for food search [11].

The proposed Ananya algorithm can be used to optimize engineering and other optimization problem. Ananya algorithm based on the behavior of togetherness as two-person may have any relation stay together, fight, interact, learn, and grow up as a better person. There are only two candidates in the proposed algorithm they interact with each other continuously moves towards an optimal solution.

II. ANANYA ALGORITHM

Ananya algorithm is one of the simplest algorithms among all optimization algorithm. It is easier to implement. This algorithm is used to attain a solution with less effort and high accuracy. Objective function f(x) to be maximized or minimized.

978-1-7281-9061-7/21/$31.00 ©2021 IEEE

Ananya algorithm is a multi-phase algorithm and each phase having a similar calculation. There are 'p' =total numbers of phase and 'q' = total numbers of iterations in every phase hence total 'p*q' iterations are required to solve the algorithm. The first phase of the proposed algorithm having two candidates (i=1, 2). There are n numbers of design variables (j=1, 2,....., n). Initial solutions are randomly generated within the upper bound and lower bound of each design variable and for both candidates. The objective function is calculated for both candidates. Compare the objective function for both candidates and select better candidates having better objective function value. Select all design variables corresponding to a better solution as better candidate variables like x(j)Better. Calculate the mean of each design variables (absolute value is considered), x(j)Mean= (|A(1,j)|+A|(2,j)|)/2. Now calculate the new value of all design variables as per equation (1) for both candidates. A(i, j) means ith candidate and jth design variable and r(j) is a random number generated in the range of [-1, 1] for jth design variable.

$$A'(i,j)=A(i,j)+r(j)*[(x(j)Better-x(j)Mean)] \quad (1)$$

Where

A'(i,j) = new value of the design variable for j^{th} variable for i^{th} iteration

A(i,j) = old value of the design variable for j^{th} variable for i^{th} iteration

r(j) = random number for design variable j in the range of (-1,1) for j^{th} variable

x(j)Better = selected batter value of the variable out of the population for j^{th} variable

x(j)Mean = mean of the j^{th} design variable

The flow diagram for Ananya Algorithm is shown in figure 1. New values of all design variables were calculated for the first and second candidates. If the new value of a variable exceeds the upper limit then assigns the upper limit for that variable, if the new value of the variable is less than the lower limit of the variable then assign a lower limit for that variable. The objective function value is calculated for the new candidate. Compare old and new candidates and select the candidate having a better objective function value f(x). Repeat this step for both first and second candidates. Thus the new table is prepared which has two candidates for the next iteration. Repeat this step for 'q' times where 'q' is the number of iterations. Suggested value of 'q' as 50 numbers. At last best candidate (out of two candidates) would be input for the next phase. In the next phase of the algorithm, there are again two candidates, the best solution produced by the first phase is considered as the first candidate while the second candidate is randomly generated within the lower bound and upper bound for each variable. Repeat the same procedure as per the first phase (say 'q' iterations). At the end of the second phase select the best candidate which would be input for the next phase. The next phase of the algorithm has two candidates the first candidate is a better candidate from the previous phase while the second candidate is randomly generated within the given range of each design variable. This process is repeated for 'p' times. There are total 'p' numbers of phases. The process has to be repeated total 'p' number of phases, having 'q' iterations in each phase, thus total 'p*q' iterations are required. Suggested value of numbers of phases 'p' = 20 to 50 numbers and iterations per phase 'q' = 50 numbers. Thus a total of 1000 to 2500 iteration required to solve the algorithm. In this research 'p' & 'q' is considered as 50, hence every function is evaluated 2500 times. Ananya means 'unique' in Hindi.

III. Demonstration of Ananya algorithm with sphere function

A. *The first phase - Sphere function*

Minimize

$$f(x) = \sum_{i=1}^{2} x_i^2 = \text{Minimize } (x_1^2 + x_2^2) \quad (2)$$

Range of variable considered for x_1 as [-100,100] and variable x_2 as [-100, 100]. There are only two candidates in the algorithm and as an initial population which is generated as a random number within the given range of design variable [-100, 100] for x_1 and x_2. The objective function is calculated for corresponding values as shown in table 1.

TABLE 1: INITIAL POPULATION FOR SPHERE FUNCTION

Candidate	X_1	X_2	f(x)	Status
1 (First)	4.9922	28.5975	842.7391	Better
2(Second)	59.8736	56.3847	6764.082	

From table 1 objective function f(x)= $(4.9922)^2$+ $(28.5675)^2$ = 842.7391 for first candidate and f(x) = $(59.8736)^2$ + $(56.3847)^2$ = 6764.0824 for second candidate. Hence it is obvious that better solution is located corresponding to first candidate (because minimum is better). Hence it is marked as 'better' solution. Corresponding to first candidate value of variable x_1= 4.9922 and variable x_2= 28.5975 hence $x_{1Better}$=4.9922 for x_1 variable; $x_{2Better}$=28.5975 for $x_2$variable; absolute mean x_{1mean}= ($|x_{(1,1)}|$ + $|x_{(1,2)}|$)/2 i.e. (|4.9922| + |59.8736|)/2 = 32.4329, x_{2mean}= ($|x_{(1,2)}|$ + $|x_{(2,2)}|$)/2 i.e. (|28.5975|+|56.3847|)/2 = 42.4911. Consider r_1= 0.0118 for x_1 is and r_2= 0.1223 for x_2 as random numbers within the range of [-1, 1]. Current solution having $x_{(1,1)}$=4.9922, $x_{(1,2)}$=28.5975, $x_{(2,1)}$=59.8736, $x_{(2,2)}$= 56.3847.

New values of variables are calculated for x_1 and x_2 using equation (1) and placed in Table 2 where 'i' represent the candidate and 'j' represents the variable. $A_{(i,j)}$ represent old value and $A'_{(i,j)}$ represents new value, $r_{(j)}$ represents the random number for j^{th} variable $x_{(j)}$. Better means the better value of j^{th} variable corresponding to a better candidate for the current solution. Rewrite equation (1) as follows.

$$A'(i,j) = A(i,j) + r(j) * [(x(j)Better - x(j)Mean)] \quad (3)$$

For the first candidate, the new value of variables x_1 calculated as $A'_{(1,1)}$= 4.9922 + 0.0118*(4.9922-32.4329) = 4.6684.

For the first candidate, a new value of the variable x_2 as $A'_{(1,2)}$= 28.5975+ 0.1223*(28.5975-42.4911)= 26.8983.

For the second candidate, the new value of the variable x_1 as $A'_{(2,1)}$= 59.8736+0.0118*(4.9922-32.4329)= 59.5498.

For the second candidate, the new value of the variable x_2 as $A'_{(2,2)}$= 56.3847+0.1223*(28.5975-42.4911)= 54.6855.

Now, the value of f(x) is calculated for both candidate with a new value of the variable x_1 and x_2, f(x)=$(4.6684)^2$+ $(26.8983)^2$ = 745.3132 for candidate 1. f(x) = $(59.5498)^2$ + $(54.6855)^2$ = 6536.6840 for candidate 2. New x_1, x_2 & f(x) for both candidate inserted into table 2.

TABLE 2: NEW VALUE OF VARIABLES

Candidate	X_1	X_2	f(x)	Status
1 (First)	4.6684	26.8983	745.3132	better
2 (Second)	59.5498	54.6855	6536.6840	

Now values of f(x) from table 1 and table 2 are compared and select candidates having the best value of f(x). Insert the selected candidate in table 3. From table 1 candidate first having f(x) = 842.7391 while table 2 candidate having a value of f(x)= 745.3132, hence the candidate first from table 2 is selected and inserted into table 3 as the first candidate. Similarly, the second candidate from table 2 having f(x) = 6536.6840 which is better than the second candidate of table 1 hence the second candidate from table 2 is selected as a better candidate and inserted into table 3. The first iteration is over now.

TABLE 3: UPDATED VALUE OF THE VARIABLES

Candidate	X_1	X_2	f(x)	Remark	Status
1-(First)	4.6684	26.8983	745.3132	From table 2	better
2-(Second)	59.5498	54.6855	6536.684	From table 2	

Second Iteration:

The same calculation is repeated for the second iteration. New values of x_1 and x_2 for first and second candidates are calculated as per equation 1. From table 3 (as initial solution for second iteration) first candidate having better f(x) = 745.3132. Hence the better solution is located corresponding to the first candidate and marked as a 'better' solution. Corresponding to the first candidate having a value of the variable x_1= 4.6684 and variable x_2= 26.8983 hence $x_{1Better}$ =4.6684 for x_1 variable, $x_{2Better}$=26.8983 for variable x_2, x_{1Mean} = [$|x_{(1,1)}|$+$|x_{(1,2)}|$]/2 i.e. (|4.6684| + |59.5498|)/2 = 32.1091, x_{2Mean} = [$|x_{(1,2)}|$ + $|x_{(2,2)}|$] /2 i.e. (|26.8983| + |54.6855|) / 2= 40.7919. Consider r_1 = 0.7568 for x_1 and r_2 = 0.4415 for x_2 as random numbers within the range of [-1, 1].

For the first candidate, the new value of variables x_1 calculated as $A'_{(1,1)}$= 4.6684+ 0.7568*(4.6684-32.1091) = -16.0987; For First candidate, new value of variable x_2 as $A'_{(1,2)}$=26.8983+ 0.4415*(26.8983-40.7919)= 20.7643; For second candidate, new value of variable x_1 as $A'_{(2,1)}$= 59.5498+ 0.7568*(4.6684-32.1091)= 38.7827; For second candidate, new value of variable x_2 as $A'_{(2,2)}$ = 54.6855 + 0.4415*(26.8983-40.7919)= 48.5515. Now calculate the value of function for two candidate set. Insert new value of x_1 and x_2 for both candidates and corresponding objective function in table 4.

TABLE 4: NEW VALUE OF VARIABLES FOR THE SECOND GENERATION

Candidate	X_1	X_2	f(x)	Status
1 (First)	-16.0987	20.7643	690.324	
2 (Second)	38.7827	48.5515	3861.342	

Now compare table 3 and table 4 and insert the candidate having a better value of f(x) into table 5. The second iteration is over now.

TABLE 5: THE UPDATED VALUE OF THE VARIABLES (COMPARE TABLE 3 AND TABLE 4)

Candidate	X_1	X_2	f(x)	Remark	Status
1 (First)	-16.099	20.7643	690.324	From table 4	better
2 (Second)	38.7827	48.5515	3861.342	From table 4	

Repeat the same procedure till 'q' iteration i.e. 50 iterations. Table 6 shows the result after 50 iterations. Now there are minor correction observed if continues to the next iteration, hence we stop here and select a better candidate that would be input for the second phase.

TABLE 6: AT THE END OF 50 ITERATIONS

Candidate	X_1	X_2	f(x)	Remark
1 (First)	-11.5036	14.4653	341.5798	better
2 (Second)	-9.9444	15.5834	341.7334	

B. Second Phase

Consider a better solution from the first phase Insert as the first candidate and generate the second candidate randomly within the upper bound and lower bound.

TABLE 7: INITIAL POPULATION FOR THE SECOND PHASE

Candidate	X_1	X_2	f(x)	Status
1(First) (better candidate from last phase)	-11.503	14.4653	341.5798	better
2(Second) (random generated)	93.9877	31.6288	9834.065	

Thus repeating the same procedure as per the first iteration. This is the first iteration for the second phase (overall 'q+1'

iteration) and repeats for 'q' numbers of iteration (total 'q+q' iterations required at the end of the second phase). After the end of the second phase again select the best candidate as input for the third phase while the second candidate is randomly generated. Repeat procedure for 'p' numbers of phases. The suggested value of 'p' should be between 20 to 50. Total 'p*q' iterations are required to find the optimum solution.

A MATLAB program is written to implement the Ananya algorithm and the total CPU time required 19.257 seconds for 2500 iteration while at 589th iteration represents x_1=0, x_2=0, and f(x)=0. The convergence graph is shown in figure 2.

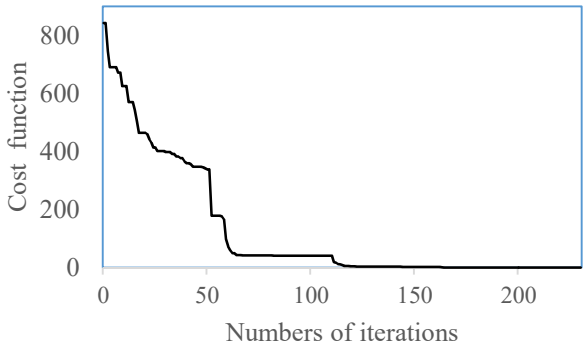

Fig. 2: Convergence graph for sphere function

IV.

V. BENCHMARK FUNCTIONS:

C. Beale function

Minimize

$$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2 \quad (4)$$

Beale function is generally evaluated for the range [-4.5, 4.5] for x_1 and x_2 and global minimum value of f(x)= 0, at x_1=3 and x_2=0.5. Beale function is tested using the Ananya algorithm and convergence diagram shown in figure 3. Total CPU time 35.939 seconds required for 2500 iteration while the system produces optimum values at 630th iteration.

D. Goldstein Price Function

Minimize

$$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)][30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 32x_1x_2 + 27x_2^2)] \quad (5)$$

Goldstein Price Function has several local minima and global minimum value of f(x)=3 for at x_1=0 and x_2=-1. This function generally evaluated for the range [-2, 2] for x_1 and x_2. The function is tested for input domain range [-2, 2] using the Ananya algorithm. The convergence graph is shown in figure 4.2. The CPU time required 39.237 seconds for 2500 iterations while at 525th iteration x_1= 0 and x_2= -1, f(x)=3.

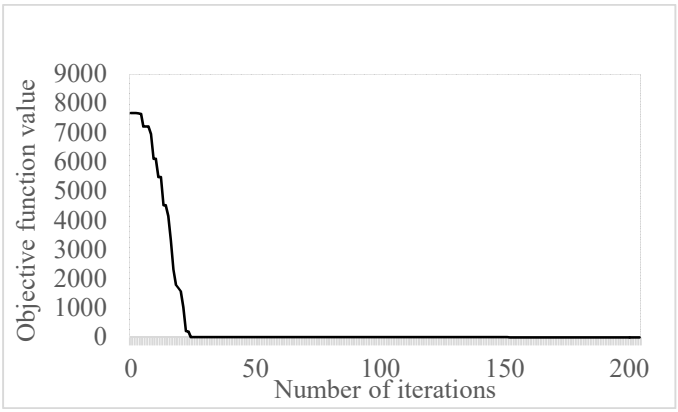

Fig 3: Convergence graph for Beale function

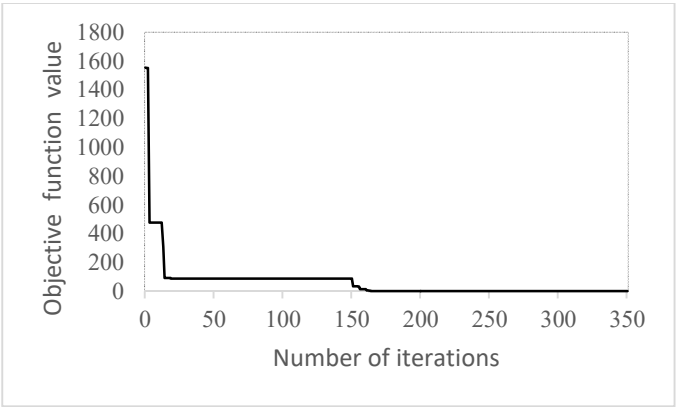

Fig 4: Convergence graph for Goldstein Price Function

E. Booth Function

The booth function is usually evaluated for the range [-10, 10] for global minima of function f(x)=0, at x_1=1 and x_2=3.The booth function is solved by the Ananya algorithm. At the end of the 440th iteration value of x_1=1 and x_2= 3 achieved and corresponding f(x)=0. CPU time taken by the Ananya algorithm is 35.327 seconds for 2500 iteration while the result is achieved in just 440 iterations. The convergence graph is shown in figure 5.

F. Matyas Function

Matyas function has global minima f(x)=0 at x_1=0 and x_2=0. Matyas function usually evaluated a range of [-10, 10] for x_1 and x_2. Matyas function does not have any local minima. Matyas function is solved using the Ananya algorithm took CPU time 32.856 seconds for 2500 iteration while the result is achieved in the 700th iteration. The convergence graph is shown in figure 6.

Minimize $f(x) = 0.26(x_1{}^2 + x_2{}^2) - 0.48x_1x_2$ (6)

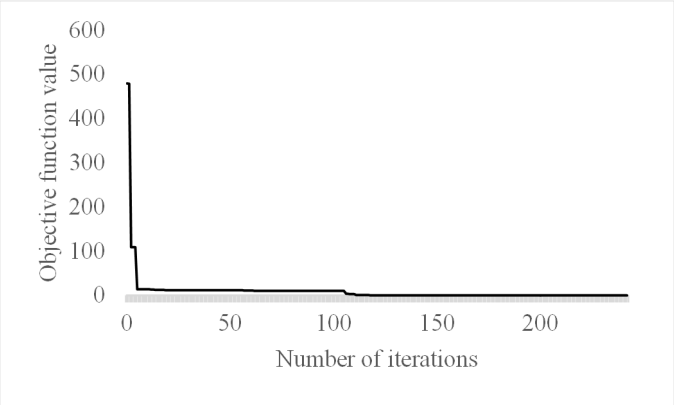

Fig 5: Convergence graph for Booth function

Fig 6: Convergence graph for Matyas function

G. Material Removal Rate to maximize

Titanium alloy is machined with electrical discharge machining. The mathematical modeling for the material removal rate (MRR) is given in equation (7) with four control parameters as peak current (Ip), Pulse on time (Ton), Voltage (V), Duty factor (t). The objective is to maximize MRR

$$MRR = 1.276+0.3099*Ip-0.01435*V-0.01417*Ton+0.721*t-0.02767t*t+0.001893*Ip*Ton-0.01660*Ip*t+0.000464*Ton*t \quad (7)$$

This problem is optimized using Ananya Algorithm. Maximum MRR is achieved as 4.6848 mg/min, with optimum control parameters are as Ip=8, Ton=150, t=11.8856, V=40. The same result is achieved using the PSO algorithm.

H. Comparison to other algorithms

The result of the Ananya algorithm is compared to other well-known optimization algorithms for benchmark objective functions [12]. The Ananya algorithm, PSO, and GA are run 20 times for each benchmark function. Ananya algorithm runs for a total of 2500 iteration. To find an optimal solution, the number of times a function is needed for evaluation is known as mean function evaluation (MFE). The standard deviation (SD) and MFE are calculated for every function as shown in table 8. PSO and GA is the most popular optimization algorithm among the researcher but a little bit complicated for implementation. These algorithms involve algorithm-specific parameters while the Ananya algorithm does not require these parameters. Ananya algorithm uses a very simple step for application furthermore, it has superior performance over the two algorithms with a faster convergence rate.

TABLE 8: COMPARISON OF ANAYA ALGORITHM, PSO, AND GA FOR DIFFERENT BENCHMARK FUNCTION [12]

Function	Algorithm	SD	MFE	Optimum cost
Sphere function	Ananya Algorithm	0	758	0
	Particle swarm optimization (PSO)	7.1998E+04	610	0
	Genetic Algorithm	1.6876E+03	1210	0
Beale function	Ananya Algorithm	0	1260	0
	PSO	0	962	0
	Genetic Algorithm	0	897	0
Goldstein function	Ananya Algorithm	0	1054	3
	PSO	0	957	3
	Genetic Algorithm	0	1357	3
Booth function	Ananya Algorithm	0	880	0
	PSO	0	1212	0
	Genetic Algorithm	0	765	0
Matyas function	Ananya Algorithm	0	1400	0
	PSO	0	843	0
	Genetic Algorithm	0	1257	0

VI. CONCLUSION

Ananya algorithm is successfully applied for benchmark function optimization. It has the potential to solve the optimization problem. There are many ways to represent the result. Someone presents the simplest things in a complicated manner. Here a complicated optimization algorithm is represented in a very simple way. Authors are not claiming that the Ananya algorithm is the 'best' algorithm, but it has a high potential to solve constrained and unconstrained optimization problems in a very simple manner. Furthermore, it is a very simple optimization algorithm. The study had been conducted for thirty benchmark functions to optimize. Ananya algorithm does not require a large number of parameters. It considers only two general parameters like mean and best variable value and gets optimum value in few iterations. The result of the Ananya algorithm is compared to PSO and GA. Ananya algorithm is found better than PSO and GA.

No one is perfect in this world. Ananya algorithm also has some limitations. The convergance rate maybe improve. The researcher can overcome this limitation in the future. Furthermore, it can be applied in constrained optimization also.

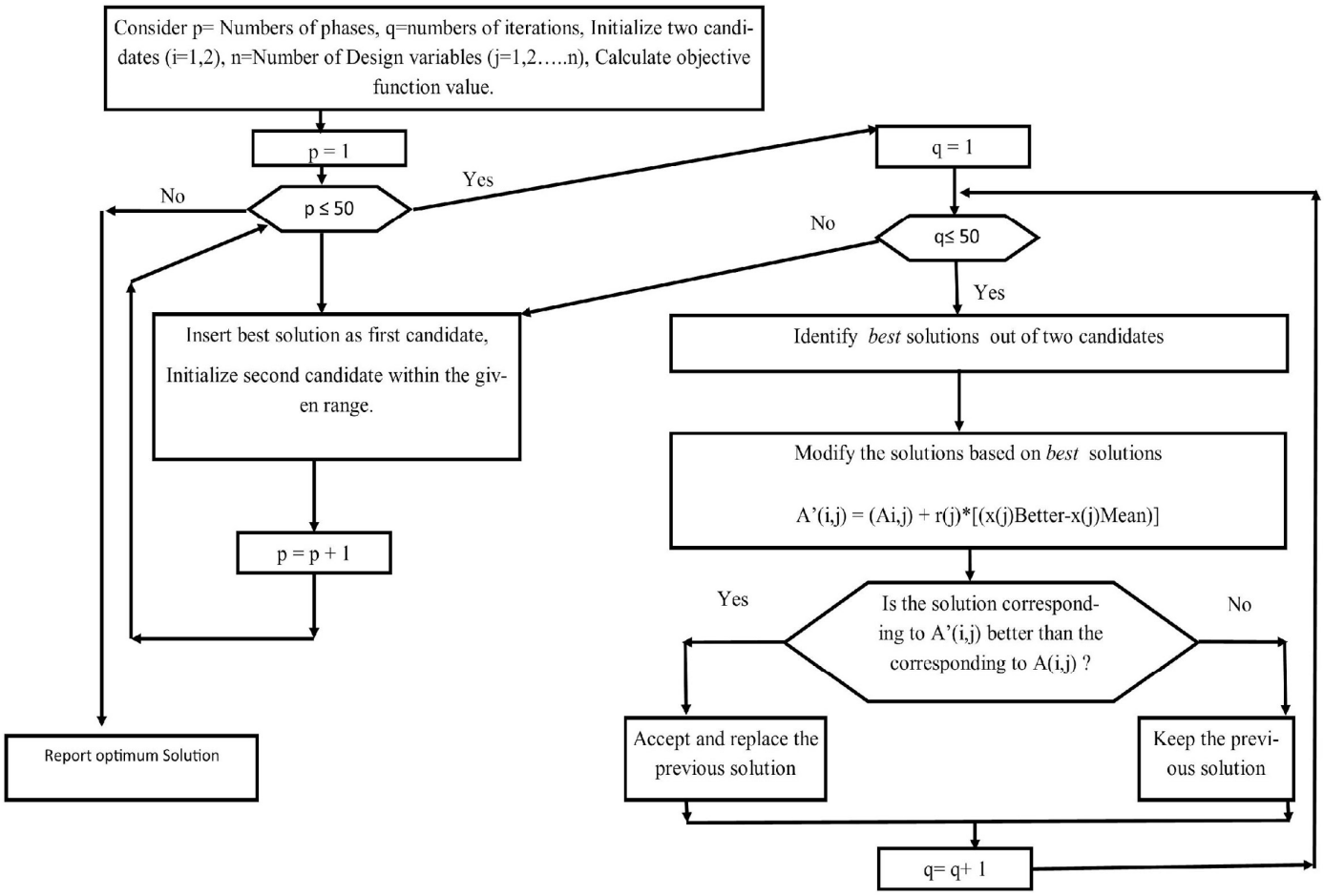

Fig 1: Flow diagram of Ananya Algorithm

REFERENCES

[1] A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis, Engineering Optimization: Methods and Applications: Second Edition. 2007.

[2] X.-S. Yang, "Engineering optimization: an introduction with metaheuristic applications." 2010.

[3] X. S. Yang, "Review of meta-heuristics and generalised evolutionary walk algorithm," Int. J. Bio-Inspired Comput., vol. 3, no. 2, pp. 77–84, 2011, doi: 10.1504/IJBIC.2011.039907.

[4] C. García, F. Herrera, and T. Stützle, "A review on the ant colony optimization metaheuristic: Basis, models and new trends," Mathw. soft Comput., vol. 9, no. 2 [-3], 2002.

[5] D. Whitley, A Genetic Algorithm Tutorial. 1993.

[6] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," Chemom. Intell. Lab. Syst., vol. 149, pp. 153–165, 2015, doi: 10.1016/j.chemolab.2015.08.020.

[7] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 1447, pp. 591–600, 1998, doi: 10.1007/bfb0040810.

[8] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice," Comput. Methods Appl. Mech. Eng., vol. 194, no. 36–38, pp. 3902–3933, 2005, doi: 10.1016/j.cma.2004.09.007.

[9] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Adv. Eng. Softw., vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[10] [P. . J. M. van Laarhoven and E. H. L. Aarts, "Simulated annealing," Simulated Annealing Theory Appl., p. 7, 1987, [Online]. Available: https://link-springer-com.ezproxy2.library.colostate.edu/content/pdf/10.1007%2F978-94-015-7744-1_2.pdf.

[11] [S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," Soft Comput., vol. 23, no. 3, pp. 715–734, 2019, doi: 10.1007/s00500-018-3102-4.

[12] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, "A comprehensive review of swarm optimization algorithms," PLoS One, vol. 10, no. 5, pp. 1–36, 2015, doi: 10.1371/journal.pone.0122827.