RESEARCH ON THE SEQUENTIAL QUADRATIC PROGRAMMING DIFFERENTIAL EVOLUTION, JAYA, AND JAYA'S EVOLUTIONARY ALGORITHM

Huynh Van Sa Ren[1*], Thuan Lam-Phat[2]

Abstract – *This study focuses on researching and comparing optimization algorithms such as Sequential Quadratic Programming, Differential Evolution, Jaya, and Jaya's evolutionary algorithm. Sequential Quadratic Programming is an optimization method based on mathematical programming, where constraints and objective functions are represented by convex and differentiable functions. Differential Evolution is a combinatorial evolutionary algorithm that utilizes genomic manipulators including recombination plus alteration generate new generations of individuals. Jaya constitutes a single enhancement procedure founded upon continuous improvement of the population, where individuals are revised predicated on the prevailing optimal resolutionare. This study focuses on the specific application of the Jaya evolutionary algorithm (iJaya) compared to other algorithms and compares the performance of these algorithms in solving optimization problems through real-world examples of fiber orientation optimization in stiffened composite plates. Experiments on popular optimization problems and measure factors such as runtime, accuracy, and the ability to search for optimal solutions were conducted. The research results provide an overview of the performance and advantages of each algorithm, thereby providing recommendations for selecting the appropriate algorithm for corresponding problems in the construction field.*

Keywords: differential evolution, evolutionary algorithm of Jaya, iJaya, Jaya, optimization algorithm, sequential quadratic programming.

[1]Tra Vinh University, Vietnam
[2]GACES, HCMC University of Technology and Education, Vietnam
*Corresponding author: saren@tvu.edu.vn
Received date: 16th June 2023; Revised date: 20th October 2023; Accepted date: 30th November 2023

I. INTRODUCTION

Optimization is a promising field that has captured the intrigue of numerous scholars. Throughout recent decades, myriad optimal techniques have been explored, devised, refined, and put into practical use across various domains. These methodologies can be categorized into two primary factions: those grounded in gradients and those aligned with popular approaches. The gradient-based cluster holds the virtue of swift attainment of optimal solutions; however, it bears the drawback of susceptibility to confinement within local extrema. Additionally, their applicability is confined to challenges featuring continuous constraints and objective functions. Eminent members within the gradient-based family encompass sequential linear programming (SLP) [1, 2], sequential quadratic programming (SQP) [3, 4], the steepest descent method, and Newton's method. In contrast, population-based techniques adeptly navigate both continuous and discrete variables, rendering them easy to implement. Furthermore, they exhibit an aptitude for evading local extrema, thereby achieving global optimization. Notable representatives of the population-based approach include the genetic algorithm (GA) [5], particle swarm optimization (PSO) [6], differential evolution (DE) [7], the artificial bee colony (ABC) [8], and the Jaya algorithm, among others.

Among the various algorithms, the recently introduced Jaya algorithm by Rao [9] and its derivatives have emerged as one of the most productive and potent methods. The Jaya approach has been applied in many engineering optimization problems and has obtained incredibly good computational efficiency [10–14]. Nonetheless, the practice of the traditional Jaya algorithm for opting for the next generation set of entities through pairwise evaluations could potentially result in the exclusion of promising persons when compared against stronger counterparts and consequently hinder the rate of convergence. Conversely, exclusively relying upon the foremost members across the whole demographic to shape the subsequent lineage population would curtail the algorithm exploration capacity, possibly resulting in premature convergence. To attain the picking of adept candidates whilst upholding stability amidst explorative and exploitative aspects, this document presents an enhanced rendition of the Jaya algorithm labeled as iJaya. Within the iJaya mechanism, the subsequent generation populace is extracted from two dissimilar groups via dissimilar assortment techniques. This improvement within the assortment procedure strives to synchronize the algorithm's potential for exploration and exploitation, thereby amplifying the pace of convergence, and concurrently preserving the guarantee of superior solution excellence.

II. LITERATURE REVIEW

The literature review on SQP, DE, Jaya, and iJaya includes examining previous studies related to these optimization algorithms. Previous studies have focused on analyzing and evaluating the performance, advantages, and applications of these algorithms in solving optimization problems. The SQP algorithm [3, 4] is an optimization method based on mathematical programming, which uses linear and quadratic approximations to search for optimal solutions. Previous research has focused on analyzing the computational complexity, performance, and advantages of SQP in solving constrained optimization problems. The DE optimization algorithm [7] is a combination of evolutionary operations, using genetic operators such as crossover and mutation to generate new generations of individuals. Previous studies have focused on analyzing and evaluating the performance, advantages, and applications of DE in solving optimization problems. Jaya [9] is an optimization algorithm based on continuous improvement of the population, where individuals are updated based on the current best solution. Previous studies have focused on analyzing and evaluating the performance, advantages, and applications of Jaya in solving optimization problems. The iJaya algorithm is an improved version of Jaya, aimed at enhancing the search capability and improving the convergence speed of the algorithm. Previous studies have focused on analyzing and evaluating the performance, advantages, and applications of the Jaya evolutionary algorithm in solving optimization problems.

This research is focused on a comprehensive examination of the SQP, DE, Jaya, and iJaya algorithms to assess and contrast their performance and benefits in addressing optimization challenges. Experimental trials are conducted on well-established problem scenarios, encompassing the measurement and comparison of variables, including runtime, precision, and the algorithms' capability to locate optimal solutions. The outcomes of this research endeavor may provide an overview of the performance of each algorithm and recommend the most suitable approach for specific optimization issues within the field.

III. RESEARCH METHODS

A. Brief introduction on reinforced composite panel

Composite panels that have been reinforced are formed by integrating a composite sheet alongside one or multiple composite girders, as illustrated in Figure 1.

In Figure 1, the distance of e creates a division between the equilibrium plane of the girder and that of the panel. The composite girder utilized in this configuration follows the Timoshenko beam theory and operates as a strengthening component.

In the context of static examination, the conduct of the reinforced composite panel is regulated by the global equations $[K]\Delta = F$ and can be referenced in [15].

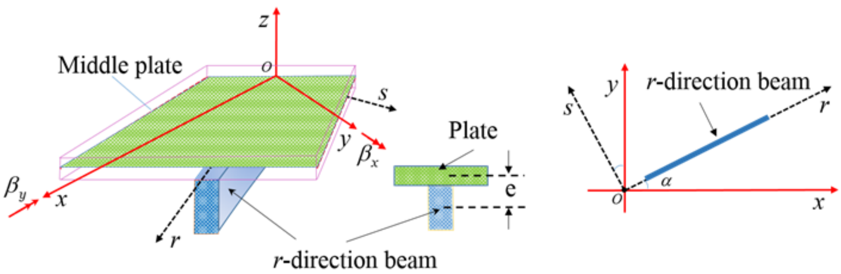

Fig. 1: Representation of a reinforced composite panel model

The FEM simulation of the beam-plate arrangement is established using a collection of junction points. Every junction point]in the plate segment possesses five axes of autonomy and represents symbolized via the vectorial representation $d = [u, v, w, \beta_x, \beta_y]^T$, where u, v, w and β_x, β_y are the middle point of the panel experiencing displacements, alongside twists along the y-axis and x-axis, correspondingly. Furthermore, each junction point within the girder exhibits the same degrees of freedom $d_{st} = [u_r, u_s, u_z, \beta_r, \beta_s]^T$, where, u_r, u_s, u_z constitute centroid shifts of the girder and β_r, β_s constitute the girder's rotational movements about r-axis and s-axis.

B. The initial Jaya technique

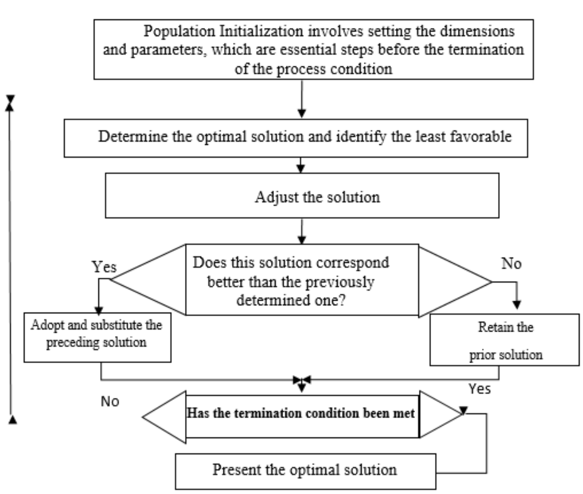

Fig. 2: Diagram illustrating the original Jaya algorithm's process

Jaya is a straightforward, yet effective collective-oriented optimization technique initially introduced in 2016 by Rao [9]. It can be applied to address both restricted and open optimization challenges. This methodology is grounded on the concept that the optimal resolution to a specific issue without any algorithm-specific control settings; only needs the standard control parameters. Initialization, modification, and selection are the three straightforward processes that make up the Jaya algorithm. Figure 2 depicts pertaining to the fundamental Jaya algorithm's flowchart. Initially, a group of Np entities is arbitrarily commenced within the exploration domain. Every contender represents $x = (x_1, x_2, \ldots, x_n)$ encompassing values within the superior and inferior limits:

$$x_{j,i} = x^l_{j,i} + rand[0,1] \times \left(x^u_{j,i} - x^l_{j,i}\right), j = (1,2,\ldots,n), i = (1,2,\ldots,N_p) \quad (1)$$

where $x^l_{j,i}$ and $x^u_{j,i}$ represent the maximum and minimum values pertaining to the configuration parameter x_j; rand[0, 1] produces a stochastic figure within the spectrum from [0, 1].

While f(x) stands as the optimization problem's goal function, the efficacy of every individual within the demographic is assessed based on the f(x) valuations. The finest (x_{best}) and the poorest (x_{worst}) contenders represent the individuals possessing the most exceptional and the least desirable fitness function values across the entire population, correspondingly. If $x_{j,i,k}$ denotes the magnitude of the j_{th} parameter belonging to the

i_{th} participant during the k_{th} cycle, then a fresh vector $x'_{j,i,k}$ is generated through a random process altering $x_{j,i,k}$ according to the subsequent process:

$$x'_{j,i,k} = x_{j,i,k} + r_{1,j,k} \times \left(x_{j,best,k} - \left|x_{j,i,k}\right|\right) - r_{2,j,k} \times \left(x_{j,worst,k} - \left|x_{j,i,k}\right|\right) \quad (2)$$

where $x_{j,best,k}$ and $x_{j,worst,k}$ represent the magnitudes of the j_{th} variable aligned with the optimal and the least effective aspirant throughout the entire demographic at the k_{th} cycle. $r_{1,j,k}$ and $r_{2,j,k}$ constitute stochastic figures within the spectrum of [0,1]. The term $r_{1,j,k} \times (x_{j,best,k} - |x_{j,i,k|})$ signifies the inclination of the design variable approaching nearer to the superior solution and the expression $r_{2,j,k} \times (x_{j,worst,k} - |x_{j,i,k|})$ exhibits the propensity of the solution diverging from the inferior one.

In the concluding phase, the determinant for selecting which candidate is retained for the succeeding generation is the objective function values computed from A and B. All candidates that are approved after each iteration are preserved and utilized as the input for the ensuing iteration.

$$x_{i,k+1} = \begin{cases} x'_{i,k} & \text{if } f(x'_{i,k}) \le f(x_{i,k}) \\ x_{i,k} & \text{otherwise} \end{cases} \quad (3)$$

C. Formulation of the Jaya technique

In the native Jaya technique, pairs of individuals are compared (paired comparison) according to the values of their fitness function to determine the population of the following generation. As a result, when contrasted with a more capable individual within the populace, a decent person is rejected. This implies that a person who does worse in one pair than the winner in another can nonetheless exceed them.

At the same time, restricting the choice of the succeeding generation's populace to merely the finest individuals would limit the algorithm's exploration, which could lead to an early convergence. A new 2-step version of the optimization technique is suggested aimed at selecting proficient individuals whilst ensuring equilibrium between discovery and utilization.

The demographic of the subsequent generation is initially segmented into two categories. A third of the populace (N/3) belongs to the first group, and the remaining (2N/3) are in the second group. In order to choose the best prospects for the coming generation, various selection strategies are used for each category of the population. The decision stage of the procedure, the first group of people is used to choose one-third of the members in the following population set employing a method of paired comparison, like that found within the foundational Jaya methodology. The remainder of the subsequent population is then chosen using an elitist approach from the second group [16].

The top-performing candidates from both the parental and offspring groups are distinctly combined to constitute two-thirds of the total population, and their selection for the ensuing generation is guided by their fitness evaluations. Figure 3 offers a visual representation of how adjustments can be made to the selection phase. This technique can effectively enhance the algorithm's convergence rate and maintain equilibrium between its exploration and exploitation capacities.

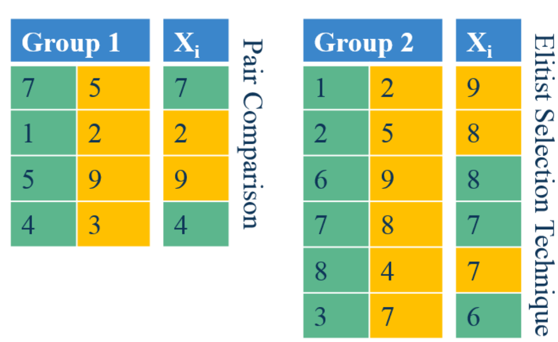

Fig. 3: Enhancement of the selection stage in the initial Jaya algorithm

IV. RESULTS AND DISCUSSION

To showcase the efficiency and accuracy of the method, two problems related to stiffened composite plates is addressed using an enhanced variant of the Jaya algorithm known as iJaya. In the initial scenario, the iJaya algorithm was utilized to determine the most favorable fiber orientation for the stiffened composite plate, to attain the lowest possible value of the strain energy objective function. The second challenge focuses on reducing the weight of the fortified composite plate. Here, the key design variable is the thickness of both the composite plate and the support beam. The algorithm's performance, precision, and rate of convergence is evaluated by comparing its outcomes with established reference solutions.

Following is a quick presentation of models and stiffened composite plate parameters to aid in the examination of example problems:

The present section examines two categories of plates: square plates and oblong plates enhanced along the X, Y, or X-Y orientations. All plate configurations adhere to the condition of being simply supported.

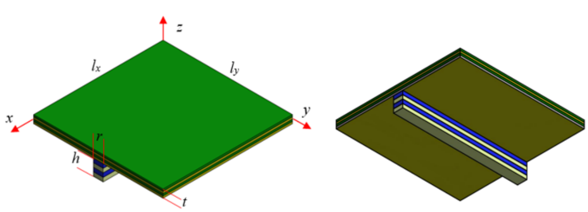

Fig. 4: Stiffened composite plate parameters

The amalgamated beam possesses transverse measurements of h, x, r, with h representing the depth and r indicating the breadth of the beam. The beam lengths in the enhancement scenarios of this segment are denoted as l_x and l_y, in accordance with instances cases of fortification in the X and Y orientations.

Figure 4 presents a depiction of an X-oriented strengthened composite panel, featuring fundamental dimensional attributes. The composite plate possesses a thickness denoted as t, while its dimensions in the X and Y directions are indicated as l_x and l_y, correspondingly.

A. Fiber alignment enhancement for reinforced composite panel utilizing iJaya

This segment investigates the optimization of fiber orientation for two variants of stiffened composite plate models that are square and rectangular in order to assess the precision and efficiency of the iJaya algorithm. Both times, the stiffeners are positioned as shown in Figure 5 in the X direction. The flexural energy of the panel is chosen as the issue's primary target function.

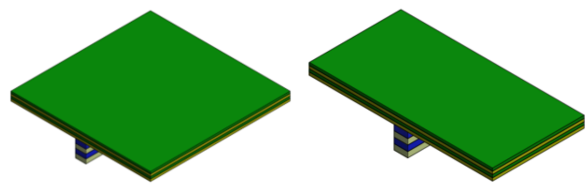

Fig. 5: Diagrams depicting quadrangular and oblong reinforced composite panel configurations

The enhancement conundrum may be delineated in the subsequent manner:

$$\begin{cases} \min\limits_{\theta} \quad \mathbf{U} = \frac{1}{2}\mathbf{d}^T\mathbf{K}\mathbf{d} \\ \text{subject to} \quad 0 \le \theta_i \le 180, i = 1,\ldots,4 \end{cases} \tag{4}$$

$\min\limits_{\theta}$ is the initial minimum strain energy

U is strain energy

ϕ_i is fiber orientation of i_{th} layer

Table 1 presents the fiber direction's optimal outcomes for both scenarios. Both continuous and discrete variables (integers) are used as design variables in the problem.

The results indicate a notable concurrence between the outcomes derived through the iJaya algorithm and those yielded by the SQP [4], DE [17], and Jaya methods. In two specific cases, the strain energy acquired from iJaya surpasses that of SQP in Nguyen's study [4], spanning from 6183.1 to 6363.8 for the square plate and 30,366 to 31,471 for the rectangular plate.

The iJaya technique exhibits superior performance compared to the DE and Jaya approaches concerning computational duration, especially when considering rectangular plates. In this particular scenario, iJaya demonstrates a computational expenditure of merely 4,462 seconds, contrasting with 6,217 and 7,762 seconds for Jaya and DE.

In addition, it is obvious from comparing the results produced with continuous and discrete variables that discrete variables have a substantially faster computational time than continuous variables. When using discrete variables instead of continuous variables, the computing time for a rectangular plate dropped by around 35%, from 4,462 seconds to 2,896 seconds. This amply demonstrated the iJaya method's efficacy and precision.

Table 1: The best solutions to two issues involving continuous and discrete variables

Type	Methods	Optimal Angle [° Degree]				Strain energy	CPU time (s)
		θ_1	θ_2	θ_3	θ_4		
Square (a = b = 254 mm)	SQP [4]	134.5	47.4	0	179.8	6363.8	129
	DE [17]	134.5	46.6	0.25	179.5	6364.0	2626
	Jaya	135.0	47.9	0	180	6183.1	1564
	iJaya	**134.9**	**47.8**	**0**	**180**	**6183.1**	**1386**
Rectangular (a = 254 mm, b = 508 mm)	SQP [4]	160.3	35.6	0	179.8	31471	154
	DE [17]	159.9	37.1	0	0	30366	7762
	Jaya	159.2	37.0	0	180	30300	6217
	iJaya	**159.9**	**37.1**	**0**	**180**	**30366**	**4462**

B. Optimization of stiffened composite plate thickness

The task of enhancing the performance of a composite plate with composite girders is taken into account, as illustrated in Figure 5, under the conditions of a single-span state. The details of the problem are described as follows: the plate has a dimension of length a = 254 mm and a thickness of tp, the strengthened beam's cross-section exhibits a breadth of cx = 6.35 mm and a depth of tb. The enhancement analysis is conducted employing two instances: a quadrilateral plate (b = 254 mm) and an oblong plate (b = 508 mm).

Each of the panel and girder features a balanced four-tier composition. The orientation of fibers in each stratum on the panel aligns sequentially at [90 45 45 90], while the fibrous layer of the girder follows an arrangement of [180 0 0 180]

Girders and panel are constructed using identical substances, characterized by parameters E_1 = 144.8 GPa, $E_2 = E_3 = 9.65$, $G_{12} = G_{13} = 4.14$ GPa, G_{23} = 3.45 GPa, $v_{12} = v_{13} = v_{23} = 0.3$. The panel experiences a uniform load bearing the magnitude of f = 0.6895 (N/mm^2).

The issue may be depicted in the Equation (5).

$$\begin{cases} \min\limits_{t_p, t_b} & Weight(h, d_x) \\ \text{subject to} & \text{Disp} \le 1 \\ & \sigma_{Tsai-wu} \le 1 \end{cases} \quad (5)$$

Specifically, the primary concerns the weight of the reinforced composite plate, which serves as the optimization objective. This mass is subject to two limitations: the deflection of the strengthened plate must remain under 1, and the Tsai-Wu stress should also not surpass 1.

The thickness enhancement challenges are addressed across three distinct scenarios: an oblong panel reinforced in the X-direction (R-X), a quadrilateral panel reinforced in the X-direction (S-X), and a square plate featuring dual stiffeners in the X-Y direction (S-XY). The optimal outcomes are displayed in Table 2. The findings indicate that, in the scenario of the quadrilateral panel, the prime results derived from the iJaya algorithm align closely with those of DE and Jaya for all three cases: S-X, R-X, and S-XY. This substantiates the precision of the iJaya technique.

The iJaya algorithm outperformed the two DE, Jaya methods in terms of computational expenses, particularly when dealing with square plates. IJaya's computation time for S-X was cut in half when in contrast with DE and the initial Jaya method. While employing the iJaya technique, the computation time for R-X is also decreased by up to 13%.

The results further demonstrate that the configuration involving a square plate and two reinforced beams in the X-Y direction (S-XY) achieves the minimum value of the objective function. This phenomenon can be attributed to the enhanced structural stability offered by two reinforcing beams, enabling a reduction in plate thickness and overall building weight. Despite the marginal difference of around 10% compared to the S-X scenario, the incorporation of a dual-beam reinforcement design provides superior structural equilibrium.

The structural support efficacy of the stiffener for the plate is decreased in the instance of R-X because the beam is strengthened along the X axis (the axis of the lesser dimension). To ensure that the entire structure could withstand the loading, the thickness of the plates was increased as a result. As a result, the structure's overall weight has also greatly increased. One conclusion drawn from the results above is that the best choice is typically made by combining the fewest number of stiffened beams with the shortest plate thickness.

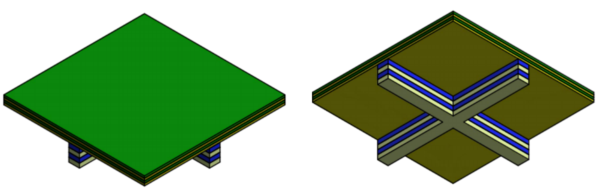

Fig. 6: Rectilinear composite panel reinforced with dual beams

Table 2: Thickness refinement of square and rectangular reinforced composite plates

Type	Method	Thickness		Weight	CPU's time
		t_p	t_b		
S-X (a = b = 254 mm)	DE [17]	13	83	1.351135507	2545
	Jaya	13	83	1.351135507	2684
	iJaya	**13**	**83**	**1.351135507**	**1469**
R-X (a = 254 mm b = 508 mm)	DE [17]	18	20	3.271406038	2984
	Jaya	18	20	3.271406038	2903
	iJaya	**18**	**20**	**3.271406038**	**2565**
S-XY (a = b = 254 mm)	DE	10	66	1.192046584	1803
	Jaya	10	66	1.191838584	7496
	iJaya	**10**	**66**	**1.192046584**	**1393**

The DE, Jaya, and iJaya algorithms were implemented using the MATLAB programming language to compute the results in this paper, and the computer used for generating all the results possessed the subsequent setup as follow:

Central Processing	Intel Core I5-2420M CPU @ 2.4 GHz
Random Access Memory (RAM)	4.00 GB (3.9 GB usable)
Operating System Architecture	64-bit Operating System, x64-based processor

There are two main factors leading to the variation in the CPU time in Table 2.

Different Parameter Optimization: The DE, Jaya, and iJaya algorithms are optimized or adjusted to operate more efficiently on a range of data. This optimization can lead to different results on the CPU. When algorithms are fine-tuned or optimized for specific problems, they effectively leverage CPU resources to perform computations faster and more efficiently.

Different Algorithms: Each algorithm (DE, Jaya, iJaya) operates differently and has distinct computational approaches. This variation can result in differences in CPU time and computational performance. One algorithm may work more efficiently on the CPU in a specific situation, while another algorithm may be better suited for a different scenario.

C. Convergence of the iJaya algorithm

Based on the findings displayed in Table 2, the iJaya algorithm consistently surpasses both the DE and Jaya methods in both computational efficiency and precision. This superiority is further evident from the convergence curves of the S-X case, as illustrated in Figure 7. Notably, while the Jaya and DE algorithms necessitate over 300 structural analyses to achieve the specified value, the iJaya algorithm accomplishes this within fewer than 200 iterations.

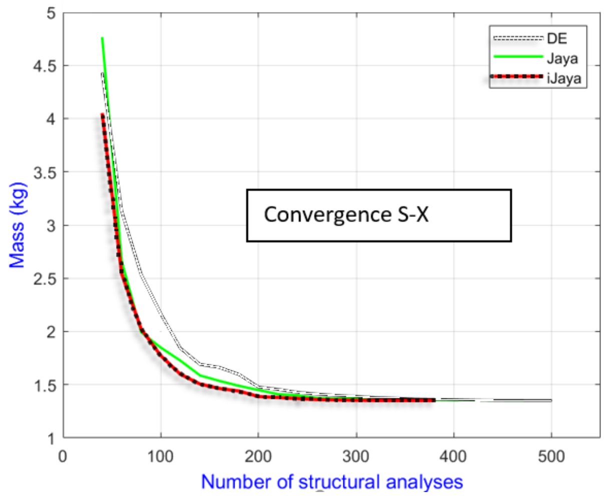

Fig. 7: Graphs depicting the convergence progression pertaining to the case of a square panel augmented in the X-axis

V. CONCLUSION

In this study, an enhanced version of the Jaya algorithm, referred to as iJaya, has been presented. The modification primarily pertains to choosing stage of the initial algorithm, resulting in the development of iJaya. This algorithm achieves a delicate equilibrium between exploration and exploitation by segregating the population into distinct clusters and employing divergent selection mechanisms, namely the paired comparison and elitist selection techniques.

The application of the iJaya methodology extends to optimizing the fiber orientation angle and the structural density of fortified composite panels. These two optimization challenges were successfully addressed using the iJaya approach, yielding results that strongly underscore the algorithm's precision and computational efficiency.

REFERENCES

[1] Lamberti L, Pappalettere C. Comparison of the numerical efficiency of different sequential linear programming-based algorithms for structural optimisation problems. *Computers & Structures*. 2000;76(6): 713–728. https://doi.org/10.1016/S0045-7949(99)00185-6.

[2] Lamberti L, Pappalettere C. Improved sequential linear programming formulation for structural weight minimization. *Computer Methods in Applied Mechanics and Engineering*. 2004;193(33–35): 3493–3521. https://doi.org/10.1016/j.cma.2003.12.040.

[3] Sedaghati R. Benchmark case studies in structural design optimization using the force method. *International Journal of Solids and Structures*. 2005;42(21–22): 5848–5871. https://doi.org/10.1016/j.ijsolstr.2005.03.030.

[4] Nguyen TT, Ho HV, Dang TH, Bui XT, Lam PT. Optimization analysis of stiffened composite plate by sequential quadratic programming. *Journal of Science and Technology*. 2013;51(1B): 156–165.

[5] Koza JR. *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, MA: MIT Press; 1992.

[6] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95 – International Conference on Neural Networks, Perth, WA, Australia*. USA: IEEE; 1995. p.1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

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

[8] Basturk B, Karaboga D. An artificial bee colony (ABC) algorithm for numeric function optimization. In: *Proceedings of the IEEE Swarm Intelligence Symposium*. USA: IEEE; 2006. p.127– 135.

[9] Rao R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*. 2016;7(1): 19–34. https://doi.org/10.5267/j.ijiec.2015.8.004.

[10] Dinh CD, Ho HV, Vo DT, Ngo THQ, Nguyen TT. Efficiency of Jaya algorithm for solving the optimization-based structural damage identification problem based on a hybrid objective function. *Engineering Optimization*. 2018;50(8): 1233–1251. https://doi.org/10.1080/0305215X.2017.1367392.

[11] Dinh CD, Vo DT, Ho HV, Nguyen TT. Damage assessment in plate-like structures using a two-stage method based on modal strain energy change and Jaya algorithm. *Inverse Problems in Science and Engineering*. 2019;27(2): 166–189. https://doi.org/10.1080/17415977.2018.1454445.

[12] Rao RV, More KC. Design optimization and analysis of selected thermal devices using self-adaptive Jaya algorithm. *Energy Conversion and Management*. 2017;140: 24–35. https://doi.org/10.1016/j.enconman.2017.02.068.

[13] Rao RV, More KC. Optimal design and analysis of mechanical draft cooling tower using improved Jaya algorithm. *International*

Journal of Refrigeration. 2017;82: 312–324. https://doi.org/10.1016/j.ijrefrig.2017.06.024.

[14] Rao RV, Rai DP. Optimisation of welding processes using quasi-oppositional-based Jaya algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*. 2017;29(5): 1099–1117. https://doi.org/10.1080/0952813X.2017.1309692.

[15] Kolli M, Chandrashekhara K. Finite element analysis of stiffened laminated plates under transverse loading. *Composites science and technology*. 1996;56(12): 1355–1361. https://doi.org/10.1016/S0266-3538(96)00086-3.

[16] Padhye N, Bhardawaj P, Deb K. Improving differential evolution through a unified approach. *Journal of Global Optimization*. 2013;55: 771–799. https://doi.org/10.1007/s10898-012-9897-0.

[17] Lam PT, Nguyen HS, Ho HV, Nguyen TT. Optimization of stiffened composite plate using adjusted different evolution algorithm. In: *Proceeding of the 7^th^ International Conference on Computational Methods (ICCM2016)*. Berkeley, CA, USA; 2016. p.76–85.