Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

A chaotic-based improved many-objective Jaya algorithm for many-objective optimization problems

Sandeep U. Mane[a*] **and M. R. Narsingrao**[a]

[a]*Department of of CSE, KLEF Deemed to be University, Vaddeswaram, Guntur Dist, AP, India*

C H R O N I C L E

Article history:
Received September 16 2020
Received in Revised Format October 7 2020
Accepted October 7 2020
Available online October, 7 2020

Keywords:
Chaotic sequence
Exploration and Exploitation
C-MaOJaya algorithm
Many-objective optimization problems
DTLZ benchmark functions

A B S T R A C T

The Jaya algorithm is a recently developed novel population-based algorithm. The proposed work presents the modifications in the existing many-objective Jaya (MaOJaya) algorithm by integrating the chaotic sequence to improve the performance to optimize many-objective benchmark optimization problems. The MaOJaya algorithm has exploitation more dominating, due to which it traps in local optima. The proposed work aims to reduce these limitations by modifying the solution update equation of the MaOJaya algorithm. The purpose of the modification is to balance the exploration and exploitation, improve the divergence and avoid premature convergence. The well-known chaotic sequence - a logistic map integrated into the solution update equation. This modification keeps the MaOJaya algorithm simple as well as, preserves its parameter-less feature. The other component of the existing MaOJaya algorithm, such as non-dominated sorting, reference vector and tournament selection scheme of NSGA-II is preserved. The decomposition approach used in the proposed approach simplifies the complex many-objective optimization problems. The performance of the proposed chaotic based many-objective Jaya (C-MaOJaya) algorithm is tested on DTLZ benchmark functions for three to ten objectives. The IGD and Hypervolume performance metrics evaluate the performance of the proposed C-MaOJaya algorithm. The statistical tests are used to compare the performance of the proposed C-MaOJaya algorithm with the MaOJaya algorithm and other algorithms from the literature. The C-MaOJaya algorithm improved the balance between exploration and exploitation and avoids premature convergence significantly. The comparison shows that the proposed C-MaOJaya algorithm is a promising approach to solve many-objective optimization problems.

1. Introduction

The optimization problems inherently exist in various scientific and engineering domains. The constrained and unconstrained single, bi-objective and multi-objective optimization problems are addressed by researchers from different domains. The multi-objective optimization problems with more than three objective functions become more challenging than addressing single or bi-objective optimization problems. The optimization problems with more than three objectives are referred by researchers as many-objective optimization problems (Zhang et al., 2016). The increased objective functions posed challenges in developing the evolutionary approaches to solve such problems. The challenges include visualization of obtained solutions, identification and comparison of best solutions, use of suitable performance metrics, difficult for decision making, etc. (Meneghini et al., 2020, Mane & NarsingRao, 2017). Over the last few years (more precisely since 2005), researchers have paid more attention to solving the optimization problems with more than three objectives. Authors have used existing as well as developed new algorithms to solve the many-objective optimization problems. The researchers have developed various

* Corresponding author
E-mail: manesandip82@gmail.com (S. U. Mane)

2021 Growing Science Ltd.
doi: 10.5267/j.ijiec.2020.10.001

traditional as well as nature-inspired optimization methods to solve different optimization problems. These approaches are most suitable to solve single and multi-objective optimization problems. To solve the many-objective optimization problems, the existing approaches are not found suitable. The researchers are working to develop different approaches to solve many-objective optimization problems. The recently developed many-objective optimization algorithms are categorised in various categories. It includes diversity-based, indicator-based, relaxed-dominance-based, preference-based, aggregation-based, reference-set-based, and dimensionality-reduction-based (Taha et al., 2017). The algorithms, belonging to these categories have pros and cons. In decomposition-based approaches, the multiple objective problems are divided into single-objective optimization problems and each single-objective optimization problem is optimized (Cheng et al., 2016, Wang et al., 2016). The Pareto-dominance based approaches aim to maximize populations' separability as well as maximize populations' internal diversity. The indicator-based approaches mainly use the hypervolume and R2 indicator to develop many-objective evolutionary algorithms. The reference-set-based approaches use the reference set to generate the solution (Taha, 2020, Mane & NarsingRao, 2017).

The decomposition-based many-objective evolutionary algorithms are one of the ways to solve many-objective optimization problems. The decomposition-based approach with reference vector guided, adjusting weight vector, Pareto adaptive scalarizing, information feedback models developed (Zhang et al., 2020). The widely used decomposition methods are the weighted sum approach, the Tchebycheff approach, and the Penalty-based Boundary Intersection approach. These three approaches are used to construct the aggregation function to be optimized (Mane et al., 2018). In the weighted sum approach, each decomposed objective function weights are assigned either it will be equal for all objectives or it may be based on the importance of each objective. These weighted objective functions are added and optimized. Another popular approach is Tchebycheff, it computes the distance from reference points. The Penalty-based boundary intersection approach introduced penalty in constraint while computing the distance (Zhang et al., 2020; Mane et al., 2018).

Researchers have presented a review of multi-objective and many-objective optimization algorithms and problems. The challenges faced while solving many-objective optimization problems have been reported by researchers in their review. Also, authors have presented the pros and cons of the existing as well as newly developed methods to address the many-objective optimization problems (Taha, 2020, Mane & NarsingRao, 2017).

The nature-inspired algorithms are widely used by researchers to address optimization problems from various domains. Some of these applications can found in (Rajeswari et al., 2017, Bewoor et al., 2017, Rajakumar et al., 2017, Kumar & Venkatesan, 2019, Reddy et al., 2018, Raveendra & Vinothkanna, 2019, Pawar & Prasanth, 2017, Ramgouda & Chandraprakash, 2019, Mane & NarsingRao, 2019). One of the challenges while developing the evolutionary algorithm to solve any type of optimization problem is to balance the exploration and exploitation behaviour of the algorithm (Črepinšek et al., 2013). The solutions obtained to optimization problems using the approaches inspired by the natural phenomena are largely affected by the exploration and exploitation strategies used in that algorithm. The nature-inspired algorithms require to tune properly the parameters which are affecting its exploration and exploitation behaviour. The crossover and mutation operators' probability needs to choose intelligently so the algorithm does not converge quickly as well as it should not be stuck locally, in the case of genetic algorithms. The pheromone evaporation rate in ant colony optimization needs to choose wisely. The inertia weight, social and cognitive parameters are important for particle swarm optimization (Rao, 2019). The success of nature-inspired or evolutionary and swarm-based algorithms largely depends on how the algorithm balances the exploration and exploitation by intelligently choosing the algorithm-specific parameters. The well-adjusted exploration and exploitation of an algorithm help to obtain the optimum solution for selected optimization problems with optimum computational efforts (Wu & He, 2020).

To avoid choosing algorithm-specific parameters, the TLBO, Jaya and Rao's algorithm was developed by Rao and his colleagues (Rao et al., 2012, Rao, 2016, Rao, 2020). To solve the optimization problem using these approaches, one does not require tune the algorithm-specific parameters. These algorithms make use of only common control parameters. The researchers have used these algorithms to solve various optimization problems. Also, different variations of the TLBO and Jaya algorithm developed by researchers (Rao, 2016, Sarzaeim, 2018, Rao, 2019). The Jaya algorithm is developed in such a way that, it pushes the fitness function value to the search space with an improved or better value than the earlier iteration. The Jaya algorithm tries to grow towards the finest possible value of a fitness function, so the exploitation will be more dominating (Ingle & Jatoth, 2020). The Jaya algorithm is a very simple algorithm to use for solving the optimization problem, but its performance is degraded due to trapping in local optima as well as weak exploration ability. The reason behind performance degradation discussed in (Wu & He, 2020) is that the solution is updated using best and worst values from the earlier iteration. As the Jaya algorithm growing towards the best possible fitness value, so the exploitation is more dominating (Ingle & Jatoth, 2020).The Many Objective Jaya (MaOJaya) Optimization algorithm developed to solve the many-objective optimization problems (Mane et al., 2018). The MaOJaya algorithm is a decomposition-based approach that makes use of the non-domination concept found in NSGA-II as well as the reference point set is used to compute the crowding distance. The tournament selection scheme used to select better solutions to improve the obtained solutions in the next iterations. The proposed work modifies the existing MaOJaya algorithm to solve many-objective optimization problems. The solution update equation of the MaOJaya algorithm is modified using a chaotic learning method.

The main contributions of this paper are as follows. The improved MaOJaya algorithm is presented to improve the exploration of the existing MaOJaya algorithm. The proposed approach developed using the logistic map based chaotic learning method, adapted from (Yu et al., 2017). The proposed algorithm's performance is evaluated using well-known DTLZ unconstrained benchmark functions. The existing MaOJaya and Chaotic-based MaOJaya algorithms with other many-objective optimization algorithms performance are compared using the statistical tests.

The remaining paper is organized as follows: Section 2 discusses the current scenario of many-objective optimization algorithmic development. Section 3 presents the basic Jaya algorithm and its variations. The proposed methodology is presented in Section 4. The obtained results and analysis are presented in section 5. Section 6 outlines the conclusion and future research directions.

2. Current Scenario about Many-objective Optimization Algorithmic Development

This section briefly reviews the current development of many-objective optimization algorithms and different problems addressed by researchers. The literature selected for study in this section is recently published. Various authors have worked to design and develop many-objective optimization algorithms to solve either standard benchmark problems or to address the many-objective optimization problems from different domains of engineering and science. The RD-EMO, a many-objective evolutionary algorithm developed to address the standard DTLZ and WFG benchmark problems. It is a region-division based decomposition approach for evolutionary many-objective optimization, where the objective space is divided into a set of regions using angle bisectors. Authors have compared the proposed approach with well-known decomposition-based approaches (Liu et al., 2020). The reference points and intuitionistic fuzzy dominance-based particle swarm algorithm presented to solve DTLZ and WFG test problems. As compared to the basic particle swarm optimization algorithm, this algorithm uses a double search strategy to update the velocity and position of particles. This strategy improved exploration and exploitation (Yang et al., 2020). The Pareto dominance criteria are widely used while obtaining solutions in multi or many-objective optimization problems. When the dimensional space increases, it becomes difficult to compare the individuals. To overcome this problem, the angle-dominance criteria are integrated into the NSGA-II algorithm. The authors have evaluated the approach using DTLZ and WFG test functions (Liu et al., 2020). In another study, authors have reported that the Pareto dominance becomes ineffective when the number of objectives is more than three. In such cases the diversity estimator part becomes dominating and it affects the performance of the algorithm. The solution moves away from Pareto-front. To overcome this problem, authors have proposed pre-processing and in the latter stage used a penalty mechanism. The penalty mechanism balances diversity and convergence. The proposed approach founds effective when tested with DTLZ, WFG, and MaF problem set (Liu et al., 2020). The angle-based and shift-based density estimation strategies are integrated to develop the many-objective evolutionary algorithm named as AnD. These two strategies remove weak individuals from the population at the selection step. It does not use Pareto-dominance relation, weight vectors or reference points, and indicators. This algorithm tested with DTLZ and WFG test functions for five, ten, and fifteen objectives. The advantage of this approach is its simple framework (Liu et al., 2020). The decomposition-based approach requires the reference point set. The decomposition-based many-objective evolutionary algorithms face the problem of generating a random number of equispaced reference points. The authors have used Das and Dennis's structured approach to generate reference points for any number of dimensions in any direction. As the selected approach is too structured, it does not generate a random number of points, for that purpose Riesz s-Energy Method is used with Das and Dennis's approach (Blank et al., 2020). The recently developed many-objective optimization algorithms focus on solving small scale many-objective optimization problems. The improved NSGA-III algorithm developed by incorporating the information feedback model. This model is used to select individuals with a fixed way and random way. Based on the information feedback model, authors have developed six versions of the improved NSGA-III algorithm. The performance is evaluated using 9 Large-scale many-objective optimization functions (Gu & Wang, 2020). The cellular genetic algorithm and multi-agent complex network-based multi-objective multi-agent complex network optimization algorithm proposed to solve unconstrained multi and many-objective benchmark optimization problems. Authors have proposed a local-global genetic operator and a chaotic based mutation operator is used to achieve exploration and exploitation (Li & Zhang, 2020). The convergence with generating a diverse set of solutions is a challenging task in the Pareto-dominance approach to solving many-objective optimization problems. Authors have proposed the use of the adaptive dominance principle and objective reduction approach, to reduce the effect of Pareto-dominance. After the predefined number of iterations, three objectives were selected randomly to optimize. The proposed modifications are performed on NSGA-II and multi-objective particle swarm optimisation algorithm (Helbig & Engelbrecht, 2020). The meta-objective method is used by researchers to transform the many-objective optimization problem into a new problem. The new problem becomes easy to solve using Pareto-dominance based algorithm. The authors have integrated the meta-objective strategy with NSGA-II and SPEA2 algorithms for solving DTLZ and WFG test problems (Gong et al., 2020). Some of the researchers have used the performance indicator to develop many-objective optimization algorithms. The hypervolume and reference vector guided, a hybrid approach is used to propose the many-objective evolutionary algorithm (H-RVEA). The approach is tested with standard as well as real-life many-objective optimization problems. The mutation and recombination operators maintain exploration and exploitation (Dhiman et al., 2020). Another performance indicator-based approach is S Metric Selection

Evolutionary Multi-Objective Algorithm. As the hypervolume is the computationally expensive performance metric, the hypervolume based many-objective evolutionary algorithm can be developed in a parallel fashion. The parallelization can reduce exponentially the computation time when the number of objectives increased. The parallel island model for hypervolume-based many-objective optimization algorithms found suitable when tested for DTLZ and WFG test problems (Gomez et al., 2020). Another nature-inspired algorithm to address many-objective optimization problem is the pigeon-inspired optimization algorithm. It uses the velocity operator to explore the search space while crossover and mutation operators are used to enhance the quality of the solution. The external archive is used to store the best solutions from where the top 10% solutions are selected (Cui et al., 2020). The angle-based crowding degree estimation approach is used to replace the distance-based crowding degree estimation in a meta-objective optimization-based bi-goal evolution approach. The proposed approach helps to reduce the effect of dominance resistant solutions and improves the exploration (Xue et al., 2020). The transfer matrix-based objective reduction approach and the Kriging model is used to develop the many-objective evolutionary algorithm. The transfer matrix is used to reduce redundant objectives while preserving the properties of objectives. The kriging model is applied to reduced objectives to generate the solutions (Ma et al., 2020). Exploration and exploitation is one of the challenge in evolutionary algorithms. The Pareto explorer tool developed to perform a global and local search to find the solution for many-objective optimization problems. The global search is performed using any multi-objective optimization algorithms while to perform local exploration, the local search approaches are used. The complexity is increased with the Pareto explorer tool when the number of objectives increased (Schütze et al., 2020).

The many-objective optimization problems attracted researchers to develop different evolutionary algorithmic strategies to address these problems. Earlier the researchers working to develop the many-objective optimization algorithms to test on standard benchmark optimization problems. As the many-objective optimization problems exist in different domains, researchers are trying to solve them. Some of the applications addressed by researchers are presented in brief here. The multi-objective pigeon-inspired optimization approach is applied to solve the unmanned aerial vehicle flocking control with obstacle avoidance, a many-objective optimization problem (Qiu & Duan, 2020). As future work, authors have suggested performing the convergence analysis. Another unmanned aerial vehicle problem is presented with six objectives. The decomposition strategy based self-adaptive meta-heuristic algorithm is presented to solve this problem (A-MnOMH/D) (Champasak et al., 2020). The transportation of fresh food products as early as is necessary to avoid the decaying of it. The increased use of transportation vehicles causes different pollutions. To avoid pollutions, the green vehicle is the solution and due to this green vehicle routing problem has attracted researchers. The green vehicle routing problem is presented as a four-objective optimization problem. The many-objective gradient evolution algorithm is developed to address the green vehicle routing problem for perishable food products. The proposed approach uses vector updating, jumping and refreshing operators for exploring the search space (Zulvia et al., 2020). The many-objective WSN energy balance optimization problem tackled using a clustering-based approach LEACH-ABF by integrating the adaptive balance function strategy, genetic operation and penalty-based boundary selection intersection scheme (Wu et al., 2020). The VAR planning method to enhance the voltage stability of wind energy power is a real-life many-objective optimization problem. This problem is addressed using the Adaptive NSGA-III-LHS algorithm. The modified approach uses the adaptive mutation rate and crossover operator from the differential evolution algorithm to balance the exploitation and exploration respectively (Chi et al., 2020). The rolling schedule optimization problem is modelled as a five-objective optimization problem. These objectives are equal relative power margin, slippage prevents, good flatness, total energy consumption and energy consumption per ton. This problem solved using a many-objective evolutionary algorithm (Hu et al., 2020). The green coal production is influenced by several factors. These factors are coal economic, energy, ecological, coal gangue economic and social benefits. To achieve a balance between these factors, a green coal production problem is modelled as a five-objective optimization problem. It is addressed by developing a hybrid many-objective particle swarm optimization algorithm. The environmental selection, an evolutionary operator is incorporated for hybridization purposes (Cui et al., 2020). The partitioning of a water distribution system (WDS) into district metered areas (DMAs) is a many-objective optimization problem. This problem is addressed using the Borg multi-objective evolutionary algorithm. The Borg algorithm is a hybrid approach, it integrates ε-dominance, ε-progress, randomized restarts, and auto-adaptive multi-operator recombination operator (Liu & Lansey, 2020). The major research work in many-objective optimization algorithmic development is more focused on standard benchmark problems, the less attention is paid towards the solving real-time problems. The study presented in (Fritsche & Pozo, 2020) is an analysis of cooperative hyper-heuristic many-objective optimization algorithm to solve the five-objective constrained wind turbine design optimization problem. The problem consists of 32 design variables and 22 constraints. The many-objective optimization problems with large-scale design variables add more complexity and increase the difficulty level of the problem. Due to the large-scale design variables, the many-objective optimization problems are still challenging to solve it using existing multi-objective evolutionary algorithms. The scalable small subpopulations based covariance matrix adaptation evolution strategy (S^3-CMA-ES) proposed to solve the large-scale many-objective optimization problem. The time-complexity analysis of the proposed approach presented while solving the LSMOP1 – LSMOP9 with 5, 8, 10 and 15 objective functions. The number of decision variables is nearly equal to 100 * number of objective functions (Chen et al., 2020). The real-world multi/many-objective optimization problems with a different number of objectives, the shape of Pareto-front and, the number of design variables are presented in (Tanabe &

Ishibuchi, 2020). The collection contains 16 real-world constrained problems. Authors have also provided the source code in Java, C, and Matlab tools. The issues related to the evaluation of solutions obtained for many-objective optimization problems are presented in (Mohammed et al., 2020). The collection of benchmark problems, widely used to assess the performance of many-objective optimization algorithms is also presented.

From the literature study, it is observed that researchers are working to improve the performance of many-objective evolutionary algorithms. Various strategies are adopting to improve performance. Also, researchers are focusing on real-life many-objective optimization problems.

3. Introduction to Jaya Algorithm and Its Variations

The Jaya algorithm is a population-based algorithm proposed to solve constrained and unconstrained optimization problems. The merits of the Jaya algorithm are, it is a parameter-less algorithm, the function evaluations required to obtain a solution are less as compared to the teaching-learning based optimization (TLBO) algorithm, it always tries to remove the worst solutions and iterates towards the best solution search space. Like other population-based approaches, the Jaya algorithm has limitations also. The Jaya algorithm gets trapped into local optimal solutions, the exploitation is more dominating than exploration.

The basic Jaya algorithm developed by Rao (2016) to reduce the pressure on researchers to tune the algorithm-specific parameters. The basic Jaya algorithm has the following steps.

a. Initialize population size, the number of design variables, and the termination criterion.

b. Evaluate the objective function and identify the best and worst solutions.

c. Generate new solutions using the best and worst solution with the equation,

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}[(X_{j,best,i}) - |(X_{j,k,i})|] - r_{2,j,i}[(X_{j,worst,i}) - |(X_{j,k,i})|]$$

d. Compare the new solution with the previous iteration's solution. If a new solution is better than the previous solution then replace the old solution otherwise keep the old solution.

e. If termination criteria are not satisfied then go to step c otherwise stop.

f. Report the optimum solution.

Various researchers have used the Jaya algorithm to solve different real-time and standard benchmark functions with diverse properties. The different variations of the Jaya algorithm are also reported in the literature. This section presents some of the variations of the Jaya algorithm from the literature.

Ingale and Jatoth (2020) proposed the Lévy flight (LF) and a greedy selection scheme based Jaya algorithm to solve the non-linear channel equalization problem. The Levy flight and greedy selection scheme improved the exploration ability of the basic Jaya algorithm. The proposed approach evaluated using unimodal and multimodal test functions as well as non-linear channel equalization problem. The self-adaptive weight, experience-based learning strategy, and chaotic elite learning-based improved Jaya algorithm presented in (Yu et al., 2017) to identify the parameters of a photovoltaic model. The integrated strategies improve the balance between exploration and exploitation. The modified Jaya algorithm is presented to optimize the material costs and electric-thermal performance of an underground power cable system. The modified Jaya algorithm identifies the 3 best solutions instead of identifying a single best value. The single best solution will be selected randomly based on certain criteria and used to generate a new solution (Ocłoń et al., 2018). Rao and Keesari (2018) used multiple teams to modify the basic Jaya algorithm. The multiple teams use the same population with six different solution update equations. The proposed approach is applied for wind farm layout optimization. The multi-team approach improves the exploration and exploitation capability of the basic Jaya algorithm (Rao & Keesari, 2018). The adaptive multi-team perturbation guiding Jaya algorithm proposed to solve multi-objective solar dish Stirling heat engine system and a multi-objective optimization Stirling heat pump optimization In the proposed approach, the number of teams gets reduced as function evaluation travels towards the pre-defined number (Rao et al., 2019). Another modification proposed by incorporating the linear decreasing inertia weight, neighbourhood search, and use of elitism to balance the exploration and exploitation (Elitist–Jaya). The improved Elitist–Jaya algorithm used to execute load flow and network reconfiguration problem (Raut & Mishra, 2019). The hybrid Jaya algorithm is proposed to solve the set-union knapsack problem. The exploration is improved by combining the Jaya algorithm with the differential evolution algorithm. The exploitation is improved using the Cauchy mutation operator (Wu & He, 2020). Another hybrid approach is Jaya and Bat algorithm. It is used to minimize power consumption in a cognitive radio network (Kaur et al., 2019). The chaotic systems parameter identified using a modified Jaya algorithm. The modification is done to improve the performance using a one-step K-means clustering mechanism and a new updated equation for the best-so-far solution (Chen et al., 2018). The multi-start Jaya algorithm for multi-objective software module clustering problem is proposed. It uses the scaling factor, multi-start adaptive, and elitism scheme to improve the performance of the proposed approach (Zamli et al., 2018). The two group strategy is used to modify the basic Jaya algorithm, where the best individuals and worst individuals are grouped into two groups. The mean of each group is computed and used to generate new solutions (Gong, 2017). The performance guided evolution strategy is presented to enhance the performance of the basic Jaya algorithm. The solution quality is improved using a self-adaptive chaotic equation. The proposed approach is used to estimate the parameters of a PV cell (Yu et al., 2019). Another variation of the basic Jaya algorithm is JayaX, developed using the XOR logic gate (Aslan et al., 2019). The chaotic based mutation strategy is used to modify the existing Jaya algorithm (Farah & Belazi, 2018). The proposed approach evaluated using unconstrained optimization benchmark problems.

The developer of the Jaya algorithm has reported the recent literature about the use of the Jaya algorithm and its variations at https://sites.google.com/site/jayaalgorithm. To understand the working of the basic Jaya algorithm readers may refer to this website. From the literature, it is observed that researchers are working to improve the exploration and exploitation ability of basic Jaya algorithm to solve various real-time and benchmark problems. The different variations of Jaya algorithm presented here mainly focus on overcoming the limitation of the basic Jaya algorithm.

4. Chaotic-based Improved MaoJaya Algorithm

This section presents the proposed chaotic-based improved many-objective Jaya optimization algorithm.

The Jaya algorithm's limitations have reported in the literature. The balance in exploration and exploitation is one of the parameters of success of any nature-inspired algorithm. The working principle of Jaya algorithm is that it drives the objective function value towards the best solution space. Due to which the exploitation is more powerful than exploration. It results in trapping in local minima and getting less diversified solutions. The Jaya algorithm uses only the best and worst solutions from earlier iterations to update the solution, it leads to premature convergence and has an impact on the quality of solutions. Another limitation reported in the literature is that the basic Jaya algorithm is weak in exchanging information among individuals. If the algorithm is stuck into the local minima, there is no mechanism to come out of the local minima (Ingle & Jatoth, 2020, Wu & He, 2020, Zamli et al., 2018). Researchers have suggested different approaches to be incorporated into the Jaya algorithm to improve performance while balancing exploration and exploitation. The authors of this paper are motivated to use and test the suggested approaches to improve the performance of the MaOJaya algorithm to solve many-objective optimization problems. The proposed study presents the use of chaotic sequence a well-known logistic map to modify the existing MaOJaya algorithm.

4.1 Chaotic Mechanism

The Chaotic mechanism is a well-known logistic map. The Chaos mechanism is reported in the literature, used with TLBO and basic Jaya algorithm. The Chaotic sequence has characteristics such as ergodicity and randomicity. These properties help the algorithm to come out from the local optimal solution. It results in improving the quality of solutions (Rao & Keesari, 2018, Yu et al., 2016, Yu et al., 2017). The Chaotic sequence is a logistic map and generated using Eq. (1).

$$c_{m+1} = 4c_m(1-c_m) \tag{1}$$

where *m* is the iteration number, the c_m is the value of a chaotic sequence at mth iteration. The initial value of c_0 is generated randomly in the range [0, 1].

4.2 Proposed modifications in MaOJaya Algorithm

The MaOJaya algorithm is developed to solve many-objective optimization problems (Mane et al., 2018). It is developed by incorporating several components from existing many-objective evolutionary algorithms. It includes the Tchebycheff – a decomposition approach, a non-dominated sorting scheme with a reference point mechanism and tournament selection from the NSGA-II algorithm. The MaOJaya algorithm combines the solutions from the current iteration and previous iteration and sorts in ascending order. The first N (population size) best solutions are selected from the sorted list using a tournament selection scheme. From these N (population size) best solutions, the best and worst solutions are selected to update the individuals in the next iteration. This approach removes the worst solutions. It results in searching at the local region and leads to premature convergence. Also, less diversified solutions will be generated.

To overcome the limitation of the MaOJaya algorithm, the solution update equation is modified by incorporating the chaotic sequence. The random number used in the solution update equation of the MaOJaya algorithm is replaced with a chaotic sequence. The modified solution update equation is presented in Eq. (2),

$$X'_{j,k,i} = X_{j,k,i} + c_{m,j,i}\left(X_{j,best,i} - |X_{j,k,i}|\right) - c_{m,j,i}\left(X_{j,worst,i} - |X_{j,k,i}|\right). \tag{2}$$

The newly generated solution will be accepted if it is better than the earlier one. Based on the above-discussed modifications, the Chaotic based MaOJaya algorithm's pseudo-code is presented in Algorithm 1.

Algorithm 1: Pseudo-code of Chaotic based MaOJaya Algorithm

```
Begin
Initialize Number of objectives ($M$), Maximum number of generations ($Gmax$), Population size ($N$), Reference points ($R^*$), Weight vector ($W$);
Generate initial population randomly;
Evaluate the objective function value for each individual ($P_0$, $W$, $d_m$);
While $t < Gmax$ do
Find the Best = best ($P_t$) and Worst = worst ($P_t$), individuals;
Generate the new solution ($Q_t$) using a modified solution update equation (Eq.2);
Evolve ($Q_t$, $W$, $R$);
Merge the parent and offspring solutions as, $U_t = P_t \cup Q_t$;
Perform non-dominated sorting ($U_{tsort}$) on ($U_t$, $R^*$);
Rank the sorted solution ($U_{tsort}$);
Use a tournament selection scheme to select first $N$ individuals for the next iteration ($P_{t+1}$);
Evaluate the new solution ($Q_t$, $f(x)$);
$Gmax+1$;
Accept the new solution if it better than the parent solution;
End While
End
```

Fig. 1 presents the flowchart of the proposed Chaotic-based MaOJaya algorithm. The important steps of the proposed Chaotic based MaOJaya algorithm are as follows:

Step 1: Initialize the algorithm's control parameters such as population size (N) and the maximum number of iterations ($Gmax$). Also initialize the problem-specific parameters such as a number of objectives (M), reference vector (R^*), and weight vector (W).

Step 2: Generate the initial solution randomly and evaluate it for each individual. Identify the best and worst solutions.

Step 3: Find the new solution for all the individuals in a population using the modified solution update equation.

Step 4: Evaluate the modified solution using Non-dominated sorting with a reference point vector and ranking method.

Step 5: Combine the new solution with the old one. Sort the combined solution in ascending order. Use the tournament selection scheme to select the first N individuals. Identify the best and worst from the selected solution to update the solution in the next iteration.

Step 6: Update the global best solution by comparing the old global best solution with the new best solution. Update the reference point vector.

Step 7: If the number of iterations reached maximum iterations then stop the procedure and report the global best solution otherwise repeat the procedure from **Step 3**.

5. Computational Results and Analysis

This section presents the evaluation of the proposed chaotic based MaOJaya algorithm. The proposed chaotic based many-objective Jaya algorithm is implemented in C programming and experimentations are performed on PC with 2.2GHz Intel(R) Core i7 processor with 8GB of RAM. The proposed approach is tested with DTLZ1-DTLZ5, well-known many-objective benchmark functions by considering the 3, 5, 8, and 10 objectives. The results obtained by the chaotic based MaOJaya algorithm are compared with the results found in the literature using well-known statistical tests.

The DTLZ functions Pareto front have diverse characteristics', which makes the challenging to solve these problems using many-objective evolutionary algorithms. These problems have characteristics such as, separable, multimodal, linear, concave, mixed, and biased (Deb et al., 2005). As per the suggestions given by Deb et al. (2005), the total number of decision variables computed. It is obtained using (M+K-1), where M presents the number of objectives and the value of K is suggested as 3 for DTLZ1 and DTLZ2 and 10 for DTLZ3 to DTLZ5 test functions.

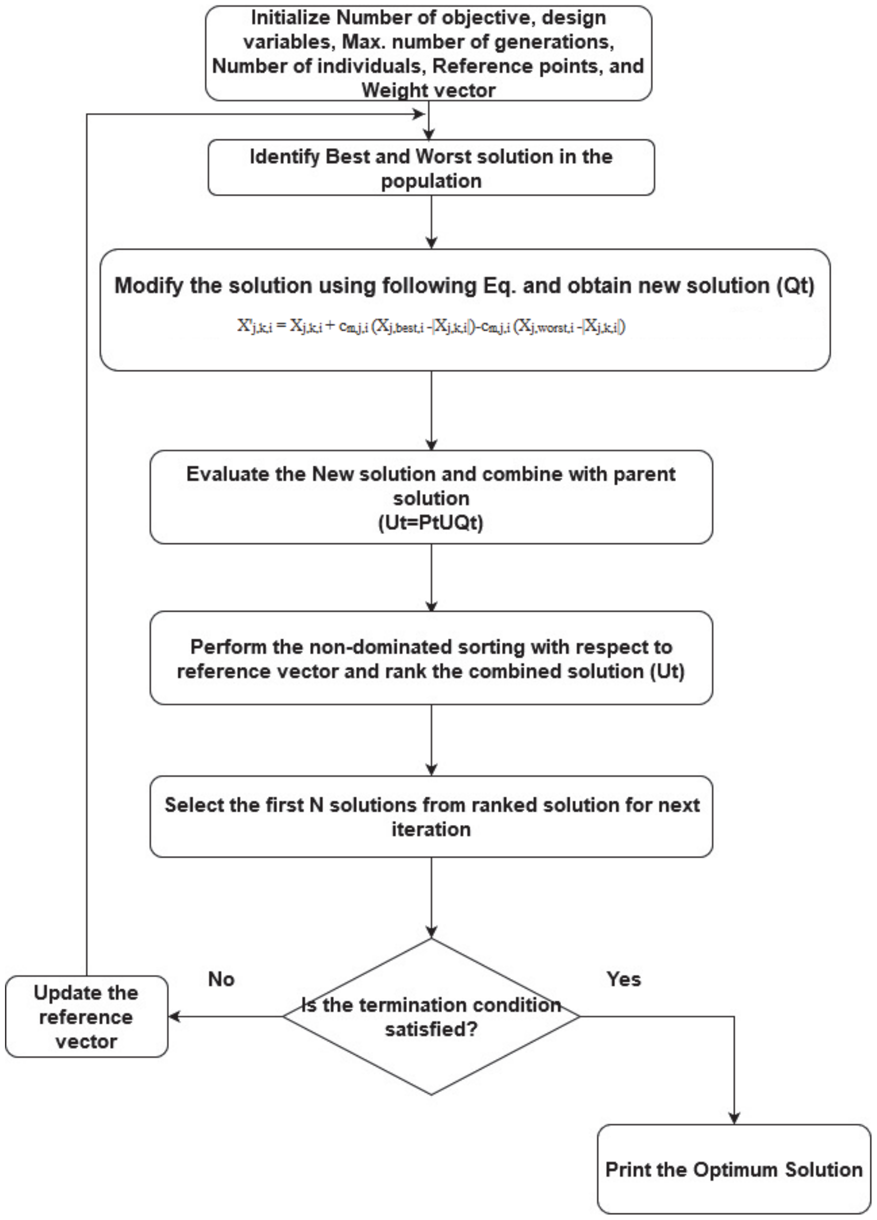
Initialize Number of objective, design variables, Max. number of generations, Number of individuals, Reference points, and Weight vector
Identify Best and Worst solution in the population
Modify the solution using following Eq. and obtain new solution (Qt)
$X'_{j,k,i} = X_{j,k,i} + c_{m,j,i} (X_{j,best,i} - |X_{j,k,i}|) - c_{m,j,i} (X_{j,worst,i} - |X_{j,k,i}|)$
Evaluate the New solution and combine with parent solution (Ut=PtUQt)
Perform the non-dominated sorting with respect to reference vector and rank the combined solution (Ut)
Select the first N solutions from ranked solution for next iteration
Is the termination condition satisfied?
No
Yes
Update the reference vector
Print the Optimum Solution

Fig. 1. Flowchart of Chaotic based MaOJaya Algorithm

The proposed chaotic based many-objective Jaya algorithm is an extension to well-known parameter less Jaya algorithm, so it requires only common controlling parameters. The following common controlling parameters are used to perform experiments.

- Population size: 200
- Maximum iterations: 500

The proposed chaotic based MaOJaya algorithm is compared with the RD-EMO, NSGA-III, MOEA/D, MOEA/DD, RVEA, and MOEA/D-M2M algorithm from (Liu et al., 2020) and the MaOJaya algorithm from (Mane et al., 2018). R. Liu et al. (2020) have used different population sizes for each of the selected algorithms for each test function. It ranges from 91 to 276. The function evaluations are computed for chaotic based MaOJaya by multiplying population size with the maximum iteration number. It is performed as per the discussion by Rao et al., (2017).

5.1 Performance Metrics

The proposed approach uses two popular performance metrics used to evaluate the performance of many-objective evolutionary algorithms. These are inverted generational distance (IGD) and hypervolume (Liu et al., 2020). The IGD performance metric measures the diversity of solution and convergence to the Pareto-front. Mathematically it is represented as Eq. (3),

$$IGD = \frac{\sqrt{\sum_{i=1}^{I} D_i^2}}{I} \tag{3}$$

The variable 'I' represents the number of sample points on the Pareto front and D_i represents the Euclidian distance between reference points and the obtained Pareto set. The minimum value of IGD indicates the better performance of many-objective optimization algorithm. Another performance metric used is the hypervolume (HV). It measures the closeness of the obtained solutions as well as diversity among solutions. The maximum value of HV indicates a better performing algorithm. The maximum value of the hypervolume is 1. The HV is biased, so the ratio of obtained Pareto front (Q) and best known Pareto front (P*) is computed (HVR). The HV is computed as Eq. (4),

$$HV = \text{volume}\left(\bigcup_{i=1}^{|Q|} v_i\right) \tag{4}$$

The HVR is computed as Eq. (5),

$$HVR = \frac{HV(Q)}{HV(P^*)} \tag{5}$$

The proposed chaotic based MaOJaya algorithm is executed 20 times for the selected test function and the corresponding number of objectives. Table 1 and Table 2 present the comparison of IGD values for DTLZ1 to DTLZ5 test functions obtained using chaotic based MaOJaya and MaOJaya algorithms. The IGD values are presented in the form of best, median and worst. The MaOJaya algorithm uses the solution update equation as it is from the basic Jaya algorithm (Mane et al., 2018). The common controlling parameters used for the MaOJaya algorithm and chaotic based MaOJaya algorithm are the same. From the results presented in Table 1, it is observed that the C-MaOJaya algorithm gives better "best" value for DTLZ3 and DTLZ4 test functions for selected objectives (3, 5, 8, and 10). It gives better "best" value for 5, 8, and 10 objectives of DTLZ1 and DTLZ2 test functions.

Table 1
Comparison of IGD values obtained using C-MaOJaya and MaOJaya Algorithm

Function	M	ST	C-MaOJaya	MaOJaya	Function	M	ST	C-MaOJaya	MaOJaya
DTLZ1	3	Best	5.40E-03	**0.00E+00**	DTLZ2	3	Best	9.73E-02	**1.000E-3**
		Median	3.77E-01	**0.00E+00**			median	7.28E-02	**5.279E-3**
		Worst	6.45E-01	**0.00E+00**			Worst	1.01E-01	**1.012E-2**
	5	Best	**0.00E+00**	3.00E-6		5	Best	**2.02E-04**	7.719E-4
		median	1.55E-01	**1.55E-5**			median	2.69E-02	**2.576E-3**
		worst	7.35E+00	**7.35E-4**			worst	5.09E-01	**4.189E-3**
	8	Best	**0.00E+00**	4.56E-3		8	Best	**2.19E-04**	3.031E-3
		median	2.30E+00	**2.10E-2**			median	1.32E-02	**5.171E-3**
		worst	**5.22E-03**	5.28E-2			worst	1.95E+00	**4.187E-3**
	10	Best	**0.00E+00**	4.17E-3		10	Best	**1.59E-04**	2.019E-3
		median	3.66E-01	**7.15E-3**			median	7.50E-03	**4.497E-3**
		worst	**1.19E-03**	3.74E-2			worst	2.14E-01	**6.091E-3**
DTLZ3	3	Best	**4.02E-04**	5.519E-3	DTLZ4	3	Best	**2.20E-05**	2.545E-3
		median	3.95E-02	**3.949E-2**			median	3.11E-02	**3.108E-2**
		worst	**1.02E-01**	1.024E-1			worst	2.50E-01	**8.056E-2**
	5	Best	**1.40E-05**	4.648E-3		5	Best	**2.04E-04**	4.055E-4
		median	**1.08E-02**	1.310E-2			median	1.07E-02	**1.244E-3**
		worst	7.73E-02	**1.929E-2**			worst	7.01E-02	**2.189E-3**
	8	Best	**3.10E-05**	6.250E-4		8	Best	**2.46E-04**	2.620E-4
		median	**6.76E-03**	8.306E-3			median	**4.86E-03**	6.079E-3
		worst	1.05E+00	**2.879E-3**			worst	4.17E-01	**3.607E-2**
	10	Best	**5.04E-04**	8.060E-4		10	Best	**1.27E-04**	2.760E-4
		median	**9.99E-03**	9.991E-3			median	**7.88E-03**	7.884E-3
		worst	4.65E-01	**2.931E-3**			worst	1.28E-01	**1.022E-1**

Best results are indicated in boldface. ST = Statistical Test

Table 2 presents the IGD values obtained using C-MaOJaya and MaOJaya algorithm for the DTLZ5 test function. The MaOJaya algorithm performs better for the DTLZ5 test problem. The C-MaOJaya gives better values for the "median" test for the DTLZ5's 8 and 10 objectives. Table 3 to Table 6 present the analysis of the comparative results between the proposed chaotic based MaOJaya algorithm with other state of art algorithms result taken from the literature (R. Liu, J. Liu, R. Zhou, et al., 2020). The results are presented in the form of best, mean and worst IGD values. Table 3 shows the comparison between IGD values obtained using the C-MaOJaya algorithm and other state-of-art algorithms from the literature for the DTLZ1 function. From the results, it is observed that the proposed approach gives better "best" values than the RD-EMO algorithm for five, eight, and ten objectives. The C-MaoJaya algorithm performs better than NSGA-III, MOEA/D, RVEA and MOEA/D-M2M algorithm for all the selected instances of DTLZ1problem. The RD-EMO algorithm and MOEA/DD have better results than the C-MaOJaya algorithm for three objectives. The C-MaOJaya performs better than the MOEA/DD algorithm for the rest of the instances of the DTLZ1 problem.

Table 2
Comparison of IGD values obtained using C-MaOJaya and MaOJaya Algorithm

Function	M	Statistical Test	C-MaOJaya	MaOJaya
DTLZ5	3	Best	7.03E-01	**4.05E-02**
		median	6.70E-01	**5.70E-02**
		worst	1.33E+00	**1.04E-01**
	5	Best	3.51E-01	**2.09E-02**
		median	3.85E-01	**4.18E-02**
		worst	7.24E+00	**6.60E-01**
	8	Best	3.50E-01	**5.27E-02**
		median	**3.58E-01**	7.88E-01
		worst	3.15E+00	**1.01E+00**
	10	Best	1.16E-01	**1.16E-02**
		median	**3.59E-01**	3.59E-01
		worst	6.18E+00	**1.084**

Best results are indicated in boldface

Table 3
Comparison of IGD values of DTLZ1 Function

Function	M	ST	C-MaOJaya	Results Taken From (R. Liu, J. Liu, R. Zhou, et al., 2020)					
				RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M
DTLZ1	3	B	5.40E-03	**2.96E-04**	8.77E-04	4.07E-04	3.15E-04	5.51E-04	6.53E-02
		M	3.30E-01	1.25E-03	2.47E-03	2.51E-03	**9.40E-04**	2.27E-03	4.34E-01
		W	6.45E-01	6.60E-03	5.12E-03	5.51E-03	**2.52E-03**	8.84E-03	9.20E-01
	5	B	**0.00E+00**	1.00E-04	6.22E-04	4.57E-04	2.51E-04	5.98E-04	1.98E-01
		M	9.52E-01	**2.15E-04**	1.20E-03	8.16E-04	3.75E-04	1.05E-03	8.18E-01
		W	7.35E+00	**5.88E-04**	2.25E-03	1.52E-03	8.48E-04	2.11E-03	2.25E+00
	8	B	**0.00E+00**	1.46E-03	2.12E-03	3.66E-03	1.54E-03	4.39E-03	2.88E-01
		M	**2.29E-03**	2.39E-03	4.96E-03	5.68E-03	2.84E-03	6.14E-03	6.80E-01
		W	5.22E-03	**3.67E-03**	2.43E-02	8.03E-03	6.17E-03	9.74E-03	1.44E+00
	10	B	**0.00E+00**	1.36E-03	2.43E-03	2.38E-03	1.63E-03	3.71E-03	1.49E-01
		M	4.49E-03	2.03E-03	3.67E-03	3.56E-03	2.94E-03	4.74E-03	3.38E-01
		W	**1.19E-03**	2.54E-03	5.80E-03	5.07E-03	5.54E-03	5.82E-03	6.71E-01

Best results are indicated in boldface. ST= Statistical Test, B=Best, M= Mean, W=Worst

Table 4 presents the IGD values obtained for the DTLZ2 test function. The RD-EMO performs better than C-MaOJaya algorithm for three objectives. The best value obtained by C-MaOJaya algorithm for five, eight and ten objectives is better than RD-EMO algorithm. Other algorithms also perform better than the proposed approach for the DTLZ2 problem. From the results, it is observed that there is scope to improve the performance of C-MaOJaya algorithm.

Table 4
Comparison of IGD values of DTLZ2 Function

Function	M	ST	C-MaOJaya	Results Taken From (R. Liu, J. Liu, R. Zhou, et al., 2020)					
				RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M
DTLZ2	3	B	9.73E-02	**3.8096E-04**	1.1680E-03	5.3890E-04	6.0836E-04	1.7398E-03	2.4088E-02
		M	6.35E-02	**5.7305E-04**	1.5088E-03	6.5878E-04	8.2044E-04	2.5228E-03	2.8681E-02
		W	1.01E-01	**7.0330E-04**	2.8249E-03	8.1124E-04	1.0324E-03	3.3615E-03	3.5624E-02
	5	B	**2.02E-04**	5.7026E-04	3.8544E-03	1.3645E-03	1.1851E-03	3.6916E-03	1.2757E-01
		M	2.70E-02	**7.0769E-04**	4.5554E-03	1.5202E-03	1.3483E-03	4.2637E-03	1.4020E-01
		W	5.09E-01	**8.6502E-04**	5.3582E-03	1.7343E-03	1.6948E-03	4.9227E-03	1.5705E-01
	8	B	**2.19E-04**	1.1994E-03	1.2720E-02	2.7114E-03	2.9286E-03	1.0492E-02	2.6614E-01
		M	1.74E-02	**1.6610E-03**	1.6153E-02	3.2387E-03	3.2460E-03	1.2219E-02	2.7901E-01
		W	1.95E+00	**2.3950E-03**	2.1178E-02	4.7974E-03	3.8672E-03	1.4102E-02	3.0256E-01
	10	B	**1.59E-04**	1.4871E-03	1.3781E-02	2.2361E-03	3.1735E-03	1.1734E-02	2.6155E-01
		M	9.41E-03	**1.7217E-03**	1.5855E-02	2.4812E-03	3.6803E-03	1.2932E-02	2.7744E-01
		W	2.14E-01	**1.9500E-03**	1.8276e-02	2.8031E-03	4.4048E-03	1.4586E-02	2.8737E-01

Best results are indicated in boldface. ST= Statistical Test, B=Best, M= Mean, W=Worst

The DTLZ3 test functions result in selected objectives presented in Table 5. The C-MaOJaya algorithm performs better than the RD-EMO algorithm in terms of the best value obtained for five, eight, and ten objectives. The RD-EMO gives better mean and worst values than the C-MaOJaya algorithm. The proposed C-MaOJaya algorithm performs significantly better than other algorithms presented in Table 5. It gives better best value than NSGA-III, MOEA/D, RVEA, MOEA/DD and MOEA/D-M2M algorithms for three objectives. The result obtained for the DTLZ4 test function presented in Table 6. The result indicates that the proposed C-MaOJaya algorithm performs significantly better than NSGA-III, MOEA/D, MOEA/DD, RVEA, and MOEA/D-M2M algorithms for three, five, eight, and ten objectives. The proposed approach obtains better best values for three, eight, and ten objectives than the RD-EMO algorithm. The MOEA/D and MOEA/D-M2M performs poor than the C-MaOJaya algorithm for five, eight, and ten objectives in terms of best, mean and worst value. For a few cases, the RVEA performs better than the C-MaOJaya algorithm.

Table 5
Comparison of IGD values of DTLZ3 Function

Function	M	ST	C-MaOJaya	Results Taken From (R. Liu et al., 2020)					
				RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M
DTLZ3	3	B	4.02E-04	**3.6801E-04**	1.1342E-03	1.2431E-03	6.9944E-04	2.5684E-03	3.1479E-01
		M	4.38E-02	**2.4768E-03**	4.4643E-03	6.2928E-03	3.0240E-03	1.2222E-02	3.5763E+00
		W	1.02E-01	6.8003E-03	9.4649E-03	1.5861E-02	6.2711E-03	3.3278E-02	9.39E+00
	5	B	**1.40E-05**	2.9604E-04	2.6134E-03	1.0975E-03	6.4016E-04	2.5903E-03	1.78E+00
		M	1.12E-02	**1.1896E-03**	6.6809E-03	3.4035E-03	1.7240E-03	6.4479E-03	1.0892E+01
		W	7.73E-02	**2.7809E-03**	1.3183E-02	1.0296E-02	6.6136E-03	1.4841E-02	2.07E+01
	8	B	**3.10E-05**	2.0604E-03	1.3800E-02	9.5163E-03	3.6338E-03	1.4851E-02	4.98E+00
		M	1.44E-02	**4.2423E-03**	6.5299E-02	1.4828E-01	8.4512E-03	2.2796E-02	1.6380E+01
		W	1.05E+00	**9.7472E-03**	6.6335E-01	9.0249E-01	2.1469E-02	3.6916E-02	3.12E+01
	10	B	**5.04E-04**	1.2750E-03	9.5568E-03	2.1509E-03	2.8805E-03	9.2338E-03	6.2451E-01
		M	1.01E-02	**1.7636E-03**	1.5083E-02	6.2201E-02	4.2235E-03	1.2914E-02	3.3605E+00
		W	4.65E-01	**2.2182E-03**	2.3234E-02	8.9993E-01	7.7519E-03	1.5671E-02	4.9751E+00

Best results are indicated in boldface. ST= Statistical Test, B=Best, M= Mean, W=Worst

Table 6
Comparison of IGD values of DTLZ4 Function

Function	M	ST	C-MaOJaya	Results Taken From (R. Liu et al., 2020)					
				RD-EMO	NSGA-III	MOEA/D	MOEA/DD	RVEA	MOEA/D-M2M
DTLZ4	3	B	**2.20E-05**	8.2711E-05	2.2066E-04	9.9498E-05	1.1632E-04	4.8315E-04	9.3162E-03
		M	3.91E-02	**9.7714E-05**	1.5995E-01	4.4432E-01	1.4525E-04	6.3219E-04	1.5791E-02
		W	2.50E-01	**1.1185E-04**	5.3103E-01	9.5034E-01	2.0977E-04	9.7108E-04	2.1010E-02
	5	B	2.04E-04	**6.1397E-05**	4.5997E-04	1.1177E-04	9.6365E-05	1.4161E-03	3.8815E-02
		M	1.07E-02	**7.6481E-05**	6.3652E-04	1.8913E-01	1.2963E-04	1.6645E-03	5.6774E-02
		W	7.01E-02	**9.5655E-05**	8.7450E-04	3.4368E-01	1.6087E-04	2.7465E-03	7.3294E-02
	8	B	**2.46E-04**	7.3815E-04	2.8941E-03	2.2144E-01	4.6608E-04	7.1811E-03	1.3241E-01
		M	1.13E-02	1.1932E-03	4.1315E-03	3.9531E-01	**7.0076E-04**	8.0438E-03	1.4931E-01
		W	4.17E-01	5.2271E-03	5.5001E-03	5.8165E-01	**1.1483E-03**	9.2406E-03	1.7280E-01
	10	B	**1.27E-04**	9.0461E-04	3.7977E-03	2.0763E-03	1.3599E-03	6.3194E-03	1.4249E-01
		M	1.54E-02	**1.2414E-03**	4.3800E-03	2.7921E-01	1.6265E-03	7.8140E-03	1.5362E-01
		W	1.28E-01	**1.4410E-03**	5.6771E-03	6.1552E-01	2.1604E-03	1.1935E-02	1.6421E-01

Best results are indicated in boldface. ST= Statistical Test, B=Best, M= Mean, W=Worst

Table 7 presents the hypervolume (HV) values obtained using the C-MaOJaya algorithm for DTLZ1 to DTLZ5 test functions. The best HV values obtained for DTLZ1 to DTLZ4 test functions for three, five, eight, and ten objectives are close to 1. It indicates that the proposed C-MaOJaya succeed to obtain diverse solutions as well as the algorithm does not converge prematurely. The best HV values obtained for the DTLZ5 test function are not significant. The DTLZ1 and DTLZ3 are separable, multimodal functions.

Table 7
HV values of DTLZ1 – DTLZ5 Functions obtained using C-MaOJaya Algorithm

Function	Statistical Test	M			
		3	5	8	10
DTLZ1	Best	0.93714	0.96126	0.92962	0.98914
	Mean	0.80973	0.82429	0.79295	0.86806
	Worst	0.68147	0.67091	0.67279	0.76709
	SD	0.12784	0.14587	0.12921	0.11269
DTLZ2	Best	0.99545	0.98998	0.97894	0.99998
	Mean	0.64746	0.69188	0.83168	0.69700
	Worst	0.03426	0.65449	0.75172	0.73066
	SD	0.68958	0.36738	0.07692	0.38567
DTLZ3	Best	0.99740	0.97093	0.95842	0.98916
	Mean	0.91598	0.83749	0.85635	0.91483
	Worst	0.67321	0.68266	0.61301	0.78091
	SD	0.77667	0.12832	0.10111	0.78514
DTLZ4	Best	0.98018	0.96271	0.98159	0.95537
	Mean	0.94650	0.66301	0.88911	0.87180
	Worst	0.52108	0.51324	0.72471	0.70903
	SD	0.37393	0.58778	0.10406	0.64984
DTLZ5	Best	0.71541	0.80469	0.75278	0.84065
	Mean	0.49275	0.59901	0.61720	0.72186
	Worst	0.39005	0.65391	0.55991	0.59156
	SD	0.25433	0.50074	0.08199	0.14459

The Pareto front is in linear and concave shaped for DTLZ1 and DTLZ3 respectively. The DTLZ2 and DTLZ4 are separable, unimodal and the Pareto front has a concave shape, however, the solution points on the Pareto front in DTLZ4 are biased. There is a need to perform extensive experimentation by carefully analysing the C-MaOJaya algorithm. Also, the mean value obtained for DTLZ2 and DTLZ4 is poor. It requires improvement. Overall the C-MaOJaya algorithm is a better choice to solve the many-objective optimization problem as it does not require to tune the algorithm-specific parameters.

The proposed chaotic based many-objective Jaya (C-MaOJaya) algorithm is a promising approach to solve many-objective optimization problems. Its performs is significantly better in terms of best IGD value for DTLZ1 to DTLZ4 benchmark functions for five, eight, and ten objectives than the existing MaOJaya algorithm. The following points help to achieve these results.

- In the basic Jaya, as well as the MaOJaya algorithm, the exploitation is more dominating. The proposed C-MaOJaya algorithm was modified to balance exploration and exploitation by reducing the dominance of exploitation behavior.
- The use of chaotic sequence number improves the convergence rate and balances the exploitation and exploration.
- The use of reference point helps to perform the guided search.
- The use of the Tchebychef – decomposition approach reduces the complexity of many-objective optimization problems.

6. Conclusion

In this work, we have proposed an improved chaotic based many-objective Jaya algorithm to solve many-objective optimization problems. The proposed approach improves the existing MaOJaya algorithm by introducing the 2D logistic based chaotic sequence. The solution update equation of the existing MaOJaya algorithm is modified by integrating the chaotic sequence. The modifications are mainly performed to reduce the dominance of exploitation in the MaOJaya algorithm. The proposed modification enhances the searchability of the Many-Objective Jaya algorithm. The chaotic sequence is introduced to improve the exploration and balance between exploitation and exploration in the MaOJaya algorithm. The proposed C-MaOJaya algorithm's efficiency is tested using a well-known DTLZ test suit. The performance is measured using IGD and HV, a performance metric. The results obtained by the proposed approach compared with the best-known results found in the literature. After comparison, it is observed that C-MaOJaya performs better than the MaOJaya algorithm. Also, its performance is relatively good, when compared with state-of-art algorithms from the literature. Though the proposed C-MaOJaya algorithm improved the balance between exploration and exploitation, there is still scope to improve its performance. As future work, the proposed approach can be evaluated using other many-objective benchmark datasets as well as real-time many-objective optimization problems from different domains.

References

Aslan, M., Gunduz, M., & Kiran, M. S. (2019). JayaX: Jaya algorithm with xor operator for binary optimization. *Applied Soft Computing*, *82*, 105576.

Bewoor, L. A., Chandra Prakash, V., & Sapkal, S. U. (2017). Evolutionary hybrid particle swarm optimization algorithm for solving NP-hard no-wait flow shop scheduling problems. *Algorithms*, *10*(4), 121.

Blank, J., Deb, K., Dhebar, Y., Bandaru, S., & Seada, H. (2020). Generating well-spaced points on a unit simplex for evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*.

Champasak, P., Panagant, N., Pholdee, N., Bureerat, S., & Yildiz, A. R. (2020). Self-adaptive many-objective meta-heuristic based on decomposition for many-objective conceptual design of a fixed wing unmanned aerial vehicle. *Aerospace Science and Technology*, *100*, 105783.

Chen, F., Ding, Z., Lu, Z., & Zeng, X. (2018). Parameters identification for chaotic systems based on a modified Jaya algorithm. *Nonlinear Dynamics*, *94*(4), 2307-2326.

Chen, H., Cheng, R., Wen, J., Li, H., & Weng, J. (2020). Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations. *Information Sciences*, *509*, 457-469.

Cheng, R., Jin, Y., Olhofer, M., & Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *20*(5), 773-791.

Chi, Y., Xu, Y., & Zhang, R. (2020). Many-objective Robust Optimization for Dynamic VAR Planning to Enhance Voltage Stability of a Wind-Energy Power System. *IEEE Transactions on Power Delivery*.

Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys (CSUR)*, *45*(3), 1-33.

Cui, Z., Zhang, J., Wang, Y., Cao, Y., Cai, X., Zhang, W., & Chen, J. (2019). A pigeon-inspired optimization algorithm for many-objective optimization problems. *Sci. China Information Science*, *62*(7), 70212-1.

Cui, Z., Zhang, J., Wu, D., Cai, X., Wang, H., Zhang, W., & Chen, J. (2020). Hybrid many-objective particle swarm optimization algorithm for green coal production problem. *Information Sciences*, *518*, 256-271.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization* (pp. 105-145). Springer, London.

Dhiman, G., Soni, M., Pandey, H. M., Slowik, A., & Kaur, H. (2020). A novel hybrid hypervolume indicator and reference vector adaptation strategies based evolutionary algorithm for many-objective optimization. *Engineering with Computers*, 1-19.

Di Wu, S. G., Cai, X., Zhang, G., & Xue, F. (2020). A many-objective optimization WSN energy balance model.

Farah, A., & Belazi, A. (2018). A novel chaotic Jaya algorithm for unconstrained numerical optimization. *Nonlinear Dynamics*, *93*(3), 1451-1480.

Fritsche, G., & Pozo, A. (2020). The Analysis of a Cooperative Hyper-Heuristic on a Constrained Real-world Many-objective Continuous Problem, *IEEE Congress on Evolutionary Computation (CEC),* Glasgow, United Kingdom, pp. 1-8.

Gómez, R. H., Coello, C. A. C., & Alba, E. (2020). A Parallel Island Model for Hypervolume-Based Many-Objective Optimization. In *High-Performance Simulation-Based Optimization* (pp. 247-273). Springer, Cham.

Gong, C. (2017). An enhanced Jaya algorithm with a two group Adaption. *International Journal of Computational Intelligence Systems*, *10*(1), 1102-1115.

Gong, D., Liu, Y., & Yen, G. G. (2020). A meta-objective approach for many-objective evolutionary optimization. *Evolutionary computation*, *28*(1), 1-25.

Gu, Z. M., & Wang, G. G. (2020). Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization. *Future Generation Computer Systems*, *107*, 49-69.

Helbig, M., & Engelbrecht, A. (2020, March). Partial Dominance for Many-Objective Optimization. In *Proceedings of the 2020 4th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence* (pp. 81-86).

Hu, Z., Wei, Z., Ma, X., Sun, H., & Yang, J. (2020). Multi-parameter deep-perception and many-objective autonomous-control of rolling schedule on high speed cold tandem mill. *ISA transactions*.

Ingle, K. K., & Jatoth, R. K. (2020). An efficient JAYA algorithm with lévy flight for non-linear channel equalization. *Expert Systems with Applications*, *145*, 112970.

Kaur, A., Sharma, S., & Mishra, A. (2019). A Novel Jaya-BAT Algorithm Based Power Consumption Minimization in Cognitive Radio Network. *Wireless Personal Communications*, *108*(4), 2059-2075.

Kumar, A. S., & Venkatesan, M. (2019). Multi-Objective Task Scheduling Using Hybrid Genetic-Ant Colony Optimization Algorithm in Cloud Environment. *Wireless Personal Communications*, *107*(4), 1835-1848.

Li, X., & Zhang, H. (2020). A multi-agent complex network algorithm for multi-objective optimization. *Applied Intelligence*, 1-28.

Liu, J., & Lansey, K. E. (2020). Multiphase DMA design methodology based on graph theory and many-objective optimization. *Journal of Water Resources Planning and Management*, *146*(8), 04020068.

Liu, R., Liu, J., Zhou, R., Lian, C., & Bian, R. (2020). A region division based decomposition approach for evolutionary many-objective optimization. *Knowledge-Based Systems*, 105518.

Liu, Y., Zhu, N., & Li, M. (2020). Solving Many-Objective Optimization Problems by a Pareto-Based Evolutionary Algorithm With Preprocessing and a Penalty Mechanism. *IEEE Transactions on Cybernetics*.

Liu, Y., Zhu, N., Li, K., Li, M., Zheng, J., & Li, K. (2020). An angle dominance criterion for evolutionary many-objective optimization. *Information Sciences*, *509*, 376-399.

Liu, Z. Z., Wang, Y., & Huang, P. Q. (2020). AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation. *Information Sciences*, *509*, 400-419.

Ma, L., Wang, R., Chen, S., Cheng, S., Wang, X., Lin, Z., ... & Huang, M. (2020). A novel many-objective evolutionary algorithm based on transfer matrix with Kriging model. *Information Sciences*, *509*, 437-456.

Mane, S. U., & Rao, M. N. (2019). Large-Scale Compute-Intensive Constrained Optimization Problems: GPGPU-Based Approach. In *Soft Computing: Theories and Applications* (pp. 579-589). Springer, Singapore.

Mane, S., & Rao, M. N. (2017). Many-objective optimization: Problems and evolutionary algorithms–a short review. *International Journal of Applied Engineering Research*, *12*(20), 9774-9793.

Mane, S., Narsingrao, M., & Patil, V. (2018). A many-objective Jaya algorithm for many-objective optimization problems. *Decision Science Letters*, *7*(4), 567-582.

Meneghini, I. R., Alves, M. A., Gaspar-Cunha, A., & Guimarães, F. G. (2020). Scalable and customizable benchmark problems for many-objective optimization. *Applied Soft Computing*, *90*, 106139.

Mohammed, R. T., Yaakob, R., Zaidan, A. A., Sharef, N. M., Abdullah, R. H., Zaidan, B. B., & Dawood, K. A. (2020). Review of the research landscape of multi-criteria evaluation and benchmarking processes for many-objective optimisation methods: coherent taxonomy, challenges and recommended solution. *International Journal of Information Technology & Decision Making*.

Ocłoń, P., Cisek, P., Rerak, M., Taler, D., Rao, R. V., Vallati, A., & Pilarczyk, M. (2018). Thermal performance optimization of the underground power cable system by using a modified Jaya algorithm. *International Journal of Thermal Sciences*, *123*, 162-180.

Pawar, S. S., & Prasanth, Y. (2017). Multi-Objective Optimization Model for QoS-Enabled Web Service Selection in Service-Based Systems. *New Review of Information Networking*, *22*(1), 34-53.

Qiu, H., & Duan, H. (2020). A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles. *Information Sciences*, *509*, 515-529.

Rajakumar, R., Amudhavel, J., Dhavachelvan, P., & Vengattaraman, T. (2017). GWO-LPWSN: Grey wolf optimization algorithm for node localization problem in wireless sensor networks. *Journal of Computer Networks and Communications*.

Rajeswari, M., Amudhavel, J., Pothula, S., & Dhavachelvan, P. (2017). Directed bee colony optimization algorithm to solve the nurse rostering problem. *Computational intelligence and neuroscience*, *2017*.

Ramgouda, P., & Chandraprakash, V. (2019). Constraints handling in combinatorial interaction testing using multi-objective crow search and fruitfly optimization. *Soft Computing*, *23*(8), 2713-2726.

Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, *7*(1), 19-34.

Rao, R. (2016). Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decision science letters*, *5*(1), 1-30.

Rao, R. (2020). Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems. *International Journal of Industrial Engineering Computations*, *11*(1), 107-130.

Rao, R. V. (2019). *Jaya: an advanced optimization algorithm and its engineering applications*. Cham: Springer International Publishing.

Rao, R. V., & Keesari, H. S. (2018). Multi-team perturbation guiding Jaya algorithm for optimization of wind farm layout. *Applied Soft Computing*, *71*, 800-815.

Rao, R. V., Keesari, H. S., Oclon, P., & Taler, J. (2019). Improved multi-objective Jaya optimization algorithm for a solar dish Stirling engine. *Journal of Renewable and Sustainable Energy*, *11*(2), 025903.

Rao, R. V., Savsani, V. J., & Balic, J. (2012). Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. *Engineering Optimization*, *44*(12), 1447-1462.

Rao, S. S. (2019). *Engineering optimization: theory and practice*. John Wiley & Sons.

Raut, U., & Mishra, S. (2019). An improved elitist–jaya algorithm for simultaneous network reconfiguration and dg allocation in power distribution systems. *Renewable Energy Focus*, *30*, 92-106.

Raveendra, K., & Vinothkanna, R. (2019). Hybrid ant colony optimization model for image retrieval using scale-invariant feature transform local descriptor. *Computers & Electrical Engineering*, *74*, 281-291.

Reddy, M. S., Ratnam, C., Rajyalakshmi, G., & Manupati, V. K. (2018). An effective hybrid multi objective evolutionary algorithm for solving real time event in flexible job shop scheduling problem. *Measurement*, *114*, 78-90.

Sarzaeim, P., Bozorg-Haddad, O., & Chu, X. (2018). Teaching-learning-based optimization (TLBO) algorithm. In *Advanced Optimization by Nature-Inspired Algorithms* (pp. 51-58). Springer, Singapore.

Schütze, O., Cuate, O., Martín, A., Peitz, S., & Dellnitz, M. (2020). Pareto Explorer: a global/local exploration tool for many-objective optimization problems. *Engineering Optimization*, *52*(5), 832-855.

Taha, K. (2020). Methods That Optimize Multi-Objective Problems: A Survey and Experimental Evaluation. *IEEE Access*, *8*, 80855-80878.

Tanabe, R., & Ishibuchi, H. (2020). An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, *89*, 106078.

Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., & Zhang, T. (2016). Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation, 22*(1), 3-18.

Wu, C., & He, Y. (2020). Solving the set-union knapsack problem by a novel hybrid Jaya algorithm. *Soft Computing*, *24*(3), 1883-1902.

Xue, Y., Li, M., & Liu, X. (2020, April). Angle-Based Crowding Degree Estimation for Many-Objective Optimization. In *International Symposium on Intelligent Data Analysis* (pp. 574-586). Springer, Cham.

Yang, W., Chen, L., Wang, Y., & Zhang, M. (2020). A reference points and intuitionistic fuzzy dominance based particle swarm algorithm for multi/many-objective optimization. *Applied Intelligence*, *50*(4), 1133-1154.

Yu, K., Liang, J. J., Qu, B. Y., Chen, X., & Wang, H. (2017). Parameters identification of photovoltaic models using an improved JAYA optimization algorithm. *Energy Conversion and Management*, *150*, 742-753.

Yu, K., Qu, B., Yue, C., Ge, S., Chen, X., & Liang, J. (2019). A performance-guided JAYA algorithm for parameters identification of photovoltaic cell and module. *Applied Energy*, *237*, 241-257.

Yu, K., Wang, X., & Wang, Z. (2016). An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. *Journal of Intelligent Manufacturing*, *27*(4), 831-843.

Zamli, K. Z., Alsewari, A., & Ahmed, B. S. (2018). Multi-Start Jaya Algorithm for Software Module Clustering Problem. *Azerbaijan Journal of High Performance Computing*, *1*, 87-112.

Zhang, Y. H., Gong, Y. J., Zhang, J., & Ling, Y. B. (2016, July). A hybrid evolutionary algorithm with dual populations for many-objective optimization. In *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 1610-1617). IEEE.

Zhang, Y., Wang, G. G., Li, K., Yeh, W. C., Jian, M., & Dong, J. (2020). Enhancing MOEA/D with information feedback models for large-scale many-objective optimization. *Information Sciences*.

Zulvia, F. E., Kuo, R. J., & Nugroho, D. Y. (2020). A many-objective gradient evolution algorithm for solving a green vehicle routing problem with time windows and time dependency for perishable products. *Journal of Cleaner Production*, *242*, 118428.