

A many-objective Jaya algorithm for many-objective optimization problems

Sandeep U. Mane^{a*}, M. R. Narsingrao^a and Vrushali C. Patil^b

^aDepartment of CSE, K L University, Vaddeswaram, Guntur Dist., AP, India

^bDepartment of CSE, Rajarambapu Institute of Technology, Rajaramnagar, Sangli Dist., MH, India

CHRONICLE

Article history:

Received April 28, 2017

Received in revised format:

October 20, 2017

Accepted November 19, 2017

Available online

November 19, 2017

Keywords:

Many-objective Jaya (MaOJaya) algorithm

Many-objective optimization problems

DTLZ benchmark functions

ABSTRACT

The proposed work presents the design and application of many-objective Jaya (MaOJaya) algorithm to optimize many-objective benchmark optimization problems. The basic Jaya algorithm is modified by introducing non-dominated sorting and tournament selection scheme of NSGA-II. The reference point mechanism is introduced to traverse algorithm towards the best solutions. The basic Jaya algorithm is modified while preserving its essential properties. The Tchebycheff – a decomposition based approach is used to simplify the complex MaOPs. The proposed MaOJaya algorithm is tested on DTLZ benchmark functions with objectives ranging from three to ten to measure its applicability and effectiveness to solve many-objective optimization problems. The IGD and Hypervolume performance metrics are used to evaluate the performance of proposed MaOJaya algorithm. The obtained IGD and Hypervolume values compared with the best known results and it is observed that, the proposed MaOJaya algorithm gives competitive or better results than known best results.

© 2018 Growing Science Ltd. All rights reserved.

1. Introduction

Obtaining the globally best values for a problem which involves more than three or four conflicting objectives found in industrial, engineering, scientific, etc. applications. Such problems evaluated by multiple criteria; this multi criterion formulation known as Multi-objective optimization problems (MOPs) (Zhang et al., 2016). Objectives of MOPs have the conflicting nature with each other, so there is no single solution exist that can optimize collaborative all objectives. There is set of trade-off solutions which evaluate all objectives. Many objective optimization problems (MaOPs) are a special class of MOPs, where more than three objectives considered. In the practical world, optimization problems are mostly belonging to the multiple objectives or criteria's (Gong et al., 2016, Zhang et al., 2016, Ibrahim et al., 2016, Deb & Jain, 2014; Bhoje et al., 2016).

* Corresponding author. Tel.: +91-9403510475
E-mail address: manesandip82@gmail.com (S. U. Mane)

A many objective optimization problem with minimization objectives represented as (Chand & Wagner, 2015),

$$\begin{aligned} \min f(x) &= (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{subject to} \quad &x \in X \subseteq R^n \end{aligned} \quad (1)$$

where, $x = (x_1 \dots x_n)$ are the n number of decision variables in the decision space X and $m =$ number of objectives; since $m > 3$.

Two ultimate goals exist while solving the MaOPs. First is diversity, where distribution of solution is maximized and the second is convergence, where distance to Pareto-front is minimized (Wang et al., 2016, Li et al., 2015). To achieve these two goals, recently number of multiobjective evolutionary algorithms (MOEAs) proposed by researchers. These MOEAs mainly classified in to four major categories, viz., decomposition based approach, Pareto dominance, Multi-population co-evolution and Indicator based approach (Wang et al., 2016).

- Decomposition based approach
 - By using the aggregation function; MOP is divided into a number of single objective problems (SOP). By solving each SOP, optimal value of each SOP can be obtained (Asafuddoula et al., 2015, Cheng et al., 2016, Wang et al., 2016).
- Pareto-dominance relation based approach
 - For selection criteria the Pareto-dominance relation used. Each individual is ranked using dominance relation (Wang et al., 2016, Murata et al., 2001).
- Multi-population co-evolution based approach
 - This approach considers multiple populations while solving the MOP. The information is shared between these multiple populations by using ant mechanism (Wang et al., 2016).
- Indicator based approach
 - According to the population distribution, fitness evolution function is constructed as a performance indicator (Wang et al., 2016, Li et al., 2015).

While solving many-objective optimizations problems using multi-objective optimization evolutionary algorithms various challenges faced by researchers. Some of them are described below.

- To identify and compare the best solutions, most MOEAs use the Pareto-dominance concept. As the number of objectives get increased solutions become non-dominated to each other. If the numbers of non-dominated solutions are more, then these solutions do not give a chance to every generation for creating new solution (Li et al., 2015, Wang et al., 2016, Cheng et al., 2016).
- A number of objectives get increased many objective optimization algorithm becomes computationally expensive. While using the performance indicators, more computational power is required (Hypervolume) (Li et al., 2015, Cheung et al., 2016).
- Important role is played by recombination operators in MOEAs. But when the dimension space is large more solutions are distinct from each other (Cheng et al., 2016, Wang et al., 2016, Li et al., 2015).
- Visualization is the important part of decision making in optimization. Visualizing the objective space is more difficult when the number of objectives increases more than three. Visualizing may not have an effect on the optimization process, but it has an impact on decision making (Cheung et al., 2016, Li et al, 2015).

Most MOEAs give the poor performance while handling MaOPs. Recently proposed many-objective evolutionary algorithms (MaOEAs) to handle MaOPs are categorized into the seven different

approaches, viz., diversity-based, relaxed dominance-based, the reference set based, indicator based, preference based, aggregation based and dimensionality reduction based approaches (Li et al, 2015).

The population based algorithms is basically categorized into the swarm intelligence and evolutionary algorithms. These algorithms are successfully applied to solve the many real life optimization problems from various domains. The success of such kind of algorithms depends on the tunings of algorithm specific parameters with common controlling parameters. The algorithm specific parameters have various purposes like, genetic algorithm has the mutation operator to achieve the exploitation and to achieve the exploration crossover operator is used. The algorithm specific parameters affect the convergence rate, diversity, efficiency, scalability, exploitation and exploration in solution space. Proper tuning of algorithm specific parameters is important for success of population based algorithms. To control the algorithm specific parameters; researcher requires the extra effort and time. Rao and his colleagues recently developed an algorithm which is free from algorithm specific parameters, known as teaching-learning based optimization (TLBO) algorithm. It mimics the teaching-learning process. With success of TLBO, Jaya – another algorithm similar to TLBO introduced. It also requires the tuning of common control parameters such as population size, number of generations, and design variables (Rao, 2016; Pandey, 2016; Rao et al., 2011).

The Jaya algorithm is an evolutionary optimization algorithm and initially proposed to solve single objective constrained and unconstrained optimization problems. It requires a single phase to update the solutions from earlier generations while algorithms like TLBO, Artificial Bee Colony (ABC) and Differential Evolution (DE), etc. requires two phases to update the solution. In Jaya algorithm, solutions always move toward the best solution and remove the worst solutions obtained for a given problem (Rao 2016; Rao et al., 2017).

The Jaya algorithm is applied to solve problems from different domains. Some of the applications found in literature are summarized below.

Carbon fibre-reinforced polymer (CFRP) composites are widely used in the aerospace, vehicles and the defense systems. The machining or the turning of such composites is challenging problem. As the machining process is the crucial and complex task Kumar Abhishek et al., (2016) have been proposed solution for selecting the optimal process parameter configurations while the machining or turning of carbon fibre-reinforced composites. The proposed work integrates the nonlinear regression modeling and fuzzy inference system (FIS) with the Jaya algorithm. The selected problem is multi-response optimization problem. The performance of Jaya algorithm is compared with TLBO, genetic algorithms (GA) and ICA. The Jaya algorithm has provided reliable results with less computational efforts and time. Warid et al. (2016) have been proposed the meta-heuristic optimization approach to solve the optimal power flow problem (OPF) using the Jaya algorithm. Authors have selected three objectives viz., generation cost minimization, voltage stability improvement, and real power loss reduction while addressing the OPF problem. Efficiency of the Jaya algorithm is evaluated using modified IEEE 30-bus and IEEE 118-bus networks and the reliability checked with the statistical analysis. The obtained solutions are also compared with other stochastic algorithms. Singh et al. (2017) have been proposed the Jaya algorithm based PID controller for AGC, which interconnects power system in two areas. Designing the controller aspect of the AGC of multi-area power system is very complex and it has various conflicting objectives. The selected problem with three objectives is converted into a single objective optimization problem using the AHP method and obtained a solution using Jaya algorithm. Indrajit N. Trivedi et al. (2016) proposed Jaya algorithm to solve Combined Economic Emission Dispatch (CEED) problem. Results obtained using Jaya algorithms are compared with the results of Gradient Method (GM), Ant Colony Optimization (ACO) and Particle Swarm Optimizer (PSO). In comparison, the Jaya algorithm gives better results than other methods. Rao and Waghmare (2017) investigated the performance and efficiency of Jaya algorithm to solve the complex constrained design optimization problem. The constraints are handled using Deb's heuristic method, which adds the penalty factor in the objective function value to get the unconstrained problem from constrained. The performance and efficiency of the Jaya algorithm tested on the four design problems of mechanical

field namely, robot gripper, clutch break with multiple discs, rolling element in bearing, and hydrostatic trust in bearing. The dimensional optimization of a micro-channel heat sink is addressed using Jaya algorithm. The Jaya algorithm is evaluated with two case studies. The problem formulated is multi-objective optimization problem with objectives, pumping power and thermal resistance. The performance of proposed Jaya algorithm is compared with hybrid MOEA and TLBO algorithm. Results obtained using Jaya algorithm are better or equally optimal as compared to hybrid MOEA and TLBO algorithm in case study 1, and founds better for case study 2. The authors have suggested future work to solve MaOPs of thermal engineering devices and systems by using extended Jaya algorithm (Rao et al., 2016). Another application of Jaya algorithm is in parameter optimization of nano-finishing process. The Jaya algorithm is compared with the desirability function approach and genetic algorithms. The Jaya algorithm gives better performance for selected problem (Rao et al., 2017). The distributed static compensator (DSTATCOM) is the custom power device which is used to improve the power quality issues like high reactive power burdening and load unbalancing. Mishra and Ray (2016) optimized the coefficients of proportional plus integral controller and filter parameters of photovoltaic fed distributed static compensator (PV-DSTATCOM). Authors have been used Jaya algorithm to solve the selected optimization problems. Results obtained by Jaya algorithm are compared with the results obtained using grenade explosion method (GEM) and teaching learning based optimized (TLBO) approaches. The result obtained from the simulations and the experiments show that (PV-DSTATCOM) optimized by the Jaya algorithm shows a satisfactory result than GEM and TLBO.

The literature reveals that, the Jaya algorithm is another algorithm which is free from algorithm-specific parameters similar to TLBO algorithm. As compared to few other evolutionary algorithms, it requires single phase to update the solution. The recently solved applications are mainly single objective and few are multi-objective optimization problems.

There are few variations of Jaya algorithms found in literature. The variations are multi-objective Jaya algorithm, Binary Jaya algorithm, Improved Jaya Algorithm, elitist Jaya algorithm, and Self-Adaptive multi-population based Jaya (SAMP-Jaya) algorithm (Rao et al., 2016; Prakash et al., 2017; Azizipanah-Abarghooee et al., 2016; Rao & Saroj, 2016; Rao & Saroj, 2017).

The Multi-objective Jaya algorithm is proposed to optimize the modern machining processes. Authors have been selected three objectives viz., micro electro-discharge machining, electro-discharge machining and plasma arc machining. The regression models developed using the actual experimental data of the machining process. To solve the MOPs, Jaya algorithm embedded with the crowding distance and dominance based ranking approach. Also the Jaya algorithms posterior version is used to solve MOPs in simulation of a single run (Rao et al., 2016).

Tapan Prakash et al. (2017) have been proposed the new Binary Jaya algorithm to solve the optimal placement of Phasor measurement units. In Binary Jaya Algorithm, the candidate solutions are represented in binary form (0, 1). The new updated solution is formed by Greedy selection approach.

The improved gradient based Jaya algorithm for solving MOPs has been proposed (Azizipanah-Abarghooee et al., 2016). In the proposed work, the economic dispatch problem with the multi-area bi-objective to determine the power output and transmission power-flow of the unit is considered. The gradient search method improves Jaya algorithm to obtain the set of Pareto optimal solutions related to the operation cost. The exploitation and exploration of Jaya algorithm improved using the new mutation operator. The proposed Pareto-based bi-objective approach works well with multi-modal problems. While doing the experimentations they verified the scalability, flexibility, and effectiveness of proposed gradient based improved Jaya algorithm by doing the convergence and complexity analysis to solve the MOPs.

Rao and Saroj (2016) introduced new elitist Jaya algorithm for solving the MOPs. Authors have selected heat exchangers design problem, where two conflicting objectives considered – maximize the effectiveness of the heat exchanger and minimize the total cost. Authors used Jaya algorithm's a priori

approach to solve the MOPs. In the elitist value approach, at the end of every iteration worst solutions of Jaya algorithm are replaced or exchanged with elitist value. The local trapping is avoided by randomly selecting the parameters of duplicate solutions. Proposed Elitist Jaya algorithm gives the better performance for solving MOPs of heat exchanger design as compared to GA and TLBO.

Rao and Saroj (2017) proposed new multi-population based Jaya algorithm, known as Self-Adaptive multi-population based Jaya (SAMP-Jaya) algorithm which has multi-population and self-adaptive strategy to solve various unconstrained and constrained engineering optimization problems. In the proposed Jaya algorithm search mechanism is modified by using the search schema as a multi-population. The population is divided into sub-population by using self-adaptive schema, which controls the exploitation and exploration rate. The proposed algorithm is tested on 15 unconstrained CEC'15 benchmark function as well as 10 constrained benchmark functions from literature. After evaluating performance, the SAMP-Jaya obtained better optimal results as compared to ANS, MEMDE and SL-PSO in more than 65% cases. Due to the multiple sub-populations; solutions are uniformly distributed in search space. Number of sub-populations is not fixed; it is modified by monitoring the change in exploitation and exploration so that sub-populations are decreased or increased in the search process. The proposed SAMP-Jaya algorithm is tested to minimize the entropy generation unit of plate-fin heat exchanger.

In the current work, a Many Objective Jaya (MaOJaya) Optimization algorithm is proposed. While designing Jaya algorithm to solve many-objective optimization problem, the properties of Jaya algorithm are preserved. The Tchebycheff – a decomposition method used to design Many-objective Jaya algorithm to solve the MaOPs (Wang et al., 2017). The Non-dominated sorting approach with a Reference point is used to find dominance relation. To perform Non-dominated sorting, the population from the previous iteration is combined with the updated population in the current iteration. With respect to the Reference Point, Non-dominated sorting is performed according to the Euclidean distance between the each solution of the front and Reference points. The obtained solutions are sorted in ascending order. The tournament selection approach is used to select the first N solution for the next iteration.

The remaining of the paper is organized as follows. Section 2 briefly describes the many-objective optimization algorithms applications and decomposition strategies to design many-objective optimization algorithms. Section 3 outlines the Jaya algorithm and its working. The modifications in Jaya algorithm to solve many-objective optimization problems discussed in section 4. Section 5 presents results and discussion. The conclusions of study with future research directions are presented in section 6.

2. Many-objective Optimization Problems

Many-objective optimization problems exist in various industrial, engineering and scientific domains. To solve many-objective optimization problems, earlier researchers have preferred MOEAs. There are various design techniques found in literature to develop many-objective evolutionary algorithms (MaOEAs). This section presents many-objective evolutionary algorithms developed to solve MaOPs. Also the decomposition techniques discussed.

Zhuansun et al. (2016) have been proposed PSO algorithm with Pareto Entropy to solve the MaOPs of reactive power model. This model has four objectives viz., maximize capability of the power supply, maximize the stability of static voltage, minimize deviation of node voltage and minimize loss of active power. To map the Pareto-front (PF) and to adopt the loose in relationship of Pareto-dominance, authors used the Pareto Entropy with many-objective PSO algorithm. Proposed algorithm has the capability to balance the diversity and convergence of approximate PF. The selection strategy is designed to select the global best solution from Pareto optimal solution and to assess the individual fitness value of Pareto optimal solution; individual's density and intensity of cell dominant are used. The IEEE14-bus system has been used to check the efficiency of algorithm and effectiveness of reactive power model.

The performance of MaOPSO and NSGA-III algorithm is investigated to design the optical network in (Figueiredo et al., 2016). The goal of designing a network problem is to simultaneously minimize the capital expenditures, probability of network blocking, network robustness and energy consumption while finding the specification of devices and physical topology. With respect to diversity and convergence of the solution, it is found that NSGA-III algorithm outperformed by MaOPSO algorithm.

The set-based GA has been proposed to solve the interval MaOPs. These interval MaOPs contain more than 3 objectives, where one objective must be subjected to interval uncertainty. Original MaOPs are first converted into the deterministic bi-objective problem, in which these new objectives are imprecision and hyper-volume. Then NSGA-II algorithm is modified by using Pareto-dominance relation with set-based approach. The performance is evaluated using 39 benchmark functions of interval MaOPs and also the design problem of car cab. Results show the superiority of the proposed algorithm (Gong et al., 2016).

Ibrahim et al. (2015) proposed a new elite NSGA-III algorithm to solve the MaOPs, which is an improved version of NSGA-III algorithm. NSGA-III fails to store the elite population as a new population that affects the convergence and diversity of the solution; this issue is overcome in the elite NSGA-III algorithm. The elite NSGA-III algorithm stores the elite population archive to preserve the previously generated elite solutions. The 11 (DTLZ and WFG) many-objective test functions with 3 to 15 objectives are used to test the performance of the elite NSGA-III algorithm.

The Tree-of-Heaven Inspired Approach (THIA) is developed to solve the MaOPs. To maintain the balance between diversity and convergence rate of the solution, natural resources are more used. By adapting the change in light, this invasive tree is rapidly spread. Authors considered two strategies to spread the tree; in the first, to maintain the territory it spreads its branches towards the ideal points (SUN) where the crowded region is more while in the second, to expand territory (diversity) where the crowded region is less and then grow taller afterwards. Techniques that are used in MOEA/D and NSGA-III algorithm are combined into THIA to get benefit and to maintain the convergence and diversity. Performance of THIA was tested on the four DTLZ (DTLZ1 to DTLZ4) functions with up to 15 objectives (Khosravi & Akbarzadeh, 2015).

Zi-Jia Wang et al. (2016) have proposed the evolutionary algorithm with parallel multi-strategy (PMEA) to solve MaOPs. PMEA contains three different populations in parallel. In which, individuals are selected based on three approaches: as density estimation by using set-based algorithm, indicator-based approach and decomposition-based approach. After selection of these three strategies, information is shared using Message Passing Interface (MPI). Individuals from the first population are selected and evaluated from the local view while next individuals are selected from two populations and evaluated from a global view. Due to three populations, PMEA improves the diversity of the population and increases the exploration ability of the algorithm. For experimentation, they used the five DTLZ functions and nine WFG functions. Results show the competitive and better performance as compared to other peer algorithms.

2.1. Decomposition-based approach for many-objective optimization problems

The decomposition is one of the methods in which MaOPs are divided into the number of single objective sub-problems and each single objective is evaluated so that the optimum value of every sub-problem can be obtained (Wang et al., 2016). There are three widely used decomposition methods to define an aggregation function, viz., the weighted sum approach, the Tchebycheff approach, and the boundary intersection (BI) approach (Zhang et al., 2016, Wang et al., 2016).

2.1.1. Weighted sum approach

Decomposition-based approaches are mainly classified into two approaches. First is to decompose the MaOPs into the sub-MOPs. For instance, to manage the convergence and diversity of candidate

solutions NSGA-III uses the set of evenly distributed reference points (Deb & Jain, 2014). Second approach is to decompose the MaOPs into the group of single objective problems. For instance, to find the set of Pareto-optimal solutions MOGA uses the variable weights, weighted aggregation based approach (Murata et al., 2001).

To solve the MaOPs Asafuddoula et al., (2015) introduced the evolutionary algorithm with decomposition-based approach. Systematic sampling is used to generate the reference directions that are used in the decomposition process; the NSGA-III (Deb & Jain, 2014) strategy adopted to generate the reference points. The simple prioritized schema is used to control and balance the association for comparison.

Wang et al. (2016) proposed the MOEA with the Local Weighted Sum Method (LWS). The weighted sum approach is applied in local manner and the Hypercones used to generate the neighborhood size of the LWS. By using these Hypercones, each weighted sum method restricted within it.

2.1.2. Tchebycheff method

The Tchebycheff method is one of the commonly used decomposition method. This method is always close to ideal points and gives better convergence than the Pareto dominance relation based selection. In terms of change in the objective space, it performs satisfactory. It is also used for convex and non-convex regions (Zhang et al., 2016; Wang et al., 2016; Wang et al., 2017).

Mathematically Tchebycheff method is represented as,

$$\text{minimization } (g^{tch}(x, w, z^*) = \max \{W_i | f_i(x) - Z_i^* | \}) \quad (2)$$

where, $z^* \in R^m$ = reference points, $w = (w_1 \dots w_m)$ is weight vector (which positive).

2.1.3. Reference set based approach

To measure the quality of the solution this approach uses a set of reference points. Due to the reference points search process gets the direction. While using the reference set two key issues found, how the reference set is constructed and how reference sets are used to measure the quality of solutions. To construct the reference set NSGA-III uses the virtual points while TAA algorithm selects reference sets from current or historical population (Deb & Jain, 2014; Li et al., 2015).

2.1.4. Preference based approach

Instead of focusing on entire PF or solution space, subset of solution or PF according to user's preference can be focused. There are two main issues while using user preference based approach. First which preference model is chosen and when integrate the user preference based information. The various models of user preference are present such as objective ranking, trade-off between objectives, goal specification and so on. And the second issue is when to integrate the user preference information in algorithm, there are three classes defined; integrated approach, priori approach and posterior approach (Cheng et al., 2016).

Ran Cheng et al. (2016) have been proposed the Reference Vector Guided Evolutionary Algorithm (REVA). Authors used the reference points for decomposition as well as to target the subset of entire Pareto-Front. By using the reference points, objective space is divided into small subspaces. To balance the diversity and convergence, angle penalized distance approach is used. According to objective functions scaling, reference vectors are distributed dynamically. To adjust distribution the adoption strategy is proposed. To handle the irregular Pareto-Fronts, the reference vector regeneration strategy is used.

There are mainly four categories of EMO to solve MaOPs; these are decomposition based approach, Pareto dominance, multi-population co-evolution approach, and indicator based approach. As found in literature there are various strategies used to modify the Jaya algorithm. Some of these strategies are posterior approach, binary Jaya, a priori approach, gradient search method and multi-objective Jaya.

To design MaOJaya algorithm, the decomposition approach, reference point, Non-dominated sorting, and Tournament selection scheme used in this paper to solve MaOPs.

3. Basic Jaya Algorithm

The Jaya algorithm is another algorithm proposed by Rao (2016) which is free from algorithm-specific parameters; it requires only common controlling parameters such as population size, number of generations, and design variables. The Jaya algorithm has a single phase and is simple to apply any constrained and un-constrained optimization problems.

3.1. Steps of Jaya algorithm

The Jaya algorithm consists of following steps.

1. Initialize population, design variable, number of iterations.
2. Evaluate objective Function.
3. Identify best and worst solutions in the population.
4. Modify the solutions based on best and worst solution using Eq. (3).

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}[(X_{j,best,i}) - |(X_{j,k,i})|] - r_{2,j,i}[(X_{j,worst,i}) - |(X_{j,k,i})|] \quad (3)$$

5. Compare new solution with previous solution and continue with the best solutions and skip the worst solution.
6. If termination criteria are not satisfied then go to step 3
7. If termination criterion is satisfied then stop and report the optimal solution.

For more details about Jaya algorithm, readers may refer to, <https://sites.google.com/site/jayaalgorithm>.

4. MaOJaya Algorithm for Many-objective Optimization Problems

This section describes the proposed many-objective Jaya algorithm to solve many-objective optimization problem.

In the proposed work, entire population is divided into different levels of non-domination in the Non-dominated sorting approach adopted from NSGA-II. In this approach, population is sorted in ascending order by using its non-domination level. Solutions present at the non-dominated level 1 are known as the best non-dominated solutions (Seshadri, 2006). Selection pressure is determined by using tournament value; selection pressure is more when tournament value is bigger. The tournament value 2 is suggested in NSGA-II algorithm (Wang, 2016). To perform Non-dominated sorting of population, the current population and modified population are combined into the new population and the new population is sorted using Non-dominated sorting approach. The tournament selection method is used to select individuals for the next iteration. The Tehebycheff - a decomposition based method is used to solve the MaOPs. The dominance relation is determined using the Non-dominated sorting approach with a reference point. To perform the Non-dominated sorting, population from the earlier iteration is combined with the updated population. Afterwards with respect to the reference point, non-dominated sorting is performed according to the Euclidean distance between the each solution of the front and

reference points. Then these solutions are sorted in ascending order. The tournament selection scheme is used to select first N solutions for next iteration.

Algorithm 1: Pseudo code of MaOJaya algorithm

- 1) Input: Number of objectives M , number of generation G_{max} , number of individuals N , reference points R^* , weight vector W .
- 2) Output: find population $P_{g_{max}}$, fitness $f_i(x)$ function value
- 3) Algorithm
 - 1) Initialize: initial population p_0 generated randomly with N individuals, initialization of W , R^* is done, initialize number of generations, initialize the best and worst
 - 2) Evaluate (P_0, W, d_m)
 - 3) While $t < G_{max}$ do
 - a) Find Best =best(P_t) and Worst= worst (P_t)
 - b) Q_t = Generate modified individuals (Best , Worst)
 - c) Evolve (Q_t, W, R)
 - d) Evaluate ($Q_t, f(x)$)
 - e) $U_t = P_t \cup Q_t$
 - f) $U_{t_{sort}} =$ Non-dominated sorting(U_t, R^*)
 - g) Ranking ($U_{t_{sort}}$)
 - h) By using tournament selection select N individuals for next iteration P_{t+1} and repeat.
 - 4) End while

Fig. 1. Pseudo code for MaOJaya algorithm

MaOJaya algorithm's main steps are outlined below.

Step 1 initialization: Initial random population P_0 with N individuals is generated. Reference points and weight vectors are initialized.

Step 2 Evolutionary cycle of MaOJaya: To find better solutions, population is searched (best and worst values), evolved, evaluated and modified by using Non-dominated sorting and ranking method.

Step 3 MaOJaya search the best and worst solutions from the earlier iteration's population (P_{t+1}) and these solutions are used by evolve function with decomposition-approach and Jaya algorithm's population modification formula to generate the new updated population Q_t . The evaluate function evaluates the evolved population Q_t to get the set of solutions.

The evaluated population Q_t is combined with P_t and the combined solutions U_t are sorted using Non-dominated sorting approach. The reference points are used to guide the algorithm sorting towards the users' reference points. From sorted solutions $U_{t_{sort}}$ the N - number of solutions (P_{t+1}) are selected by using tournament selection method for next iteration.

This process repeatedly works till the termination criteria fulfills.

5. Results and Discussion

In this section, the MaOJaya algorithm's performance and ability to deal with many-objective optimization problems is evaluated using many-objective benchmark functions. The proposed MaOJaya algorithm is implemented for DTLZ1 – DTLZ5 functions with three, five, eight and ten objectives. The proposed MaOJaya algorithm's framework makes use of Tchebycheff method and uses non-dominated sorting approach, so the results obtained are compared with the NSGA-III and MOEA/D approaches given by Deb and Jain (2014).

The DTLZ functions possess different characteristics. These are listed below:

Table 1

Many-objective benchmark functions characteristics (Zhang et al., 2016)

| Benchmark functions | Characteristics |
|---------------------|--------------------------------------|
| DTLZ1 | Separable, multimodal, linear shape |
| DTLZ2 | Separable, unimodal, concave |
| DTLZ3 | Separable, multimodal, concave |
| DTLZ4 | Separable, unimodal, concave, biased |

The total number of decision variables is calculated using $(M+K-1)$, where M is number of objectives and value of K used is, K=3 for DTLZ1 and DTLZ2 and K=10 for DTLZ3 to DTLZ5 benchmark function. The Deb and Jain (2014) have solved DTLZ1 to DTLZ4 test problems, this paper has addressed DTLZ5 test problem also. The common controlling parameters for MaOJaya algorithm such as population size and termination conditions used for experimentation are presented.

- Population Size: 200
- Termination Condition: 500 iterations
- Reference point $RP^* = \{RP_1^* \dots, RP_m^*\}$ in objective space are determined by Pareto-optimal solution. In this paper, $RP^* = (2, 2, 2 \dots 2)$ is used for selected DTLZ functions.

Deb and Jain (2014) have used varying population size for both NSGA-III and MOEA/D approaches, it ranges from 92 to 276 and 91 to 275 respectively. The Ran Cheng et al., (2016) have also used different population size as number of objectives changes from three to ten, they have used population size from 105 to 275.

Rao et al., (2017) calculated number of function evaluations by multiplying population size with number of generations for a single simulation run. When the algorithm is executed more than one time, that factor is also multiplied to compute total function evaluations. The number of function evaluations required for the proposed MaOJaya algorithm computed in similar fashion.

5.1. Performance indicator

To calculate the performance of MOEAs, several number of quality indicators have been proposed. Mostly used quality indicators are inverted generational distance (IGD), hypervolume and the R2 indicator (Chand & Wagner, 2015).

The IGD performance indicator used to measure the diversity (the spread of the solutions) and convergence to the Pareto-front (Chand & Wagner, 2015). The IGD performance indicator calculates the how long the elements are in the Pareto optimal set from those in the set of non-dominated vectors founds. If the value of IGD equal to zero, means that all elements are on the Pareto front. IGD measures the average distance of the element to Pareto front (Lwin et al., 2014).

Mathematically it is represented as,

$$IGD = \frac{\sqrt{\sum_{i=1}^I D_i^2}}{I}, \quad (4)$$

where, i is equal to number of sample points on Pareto-optimal front and D_i is Euclidean distance between obtained solution and sample points on the Pareto-Optimal front.

Hypervolume (HV) is another metric to measure the closeness and diversity of the obtained solution. The hypervolume is calculated as,

$$HV = \text{volume} \left(\bigcup_{i=1}^{|Q|} v_i \right) \quad (5)$$

The calculated HV value is biased towards least-scaled objective function. It is eliminated by calculating HVR, as the ratio of obtained Pareto front (Q) and best known Pareto front (P*).

$$HVR = \frac{HV(Q)}{HV(P^*)} \quad (6)$$

The large (maximum) value of hypervolume ratio (HVR) is one, when the problems have all objectives of type minimization (Lwin et al., 2014).

The proposed MaOJaya algorithm is executed 20 times for each test function and its associated number of objectives. The results obtained in terms of selected performance metric are used to compare with the known results from the literature. The Table 2 to Table 5 show the comparison between IGD values obtained using proposed MaOJaya algorithm and the IGD values taken from Deb and Jain (2014); the IGD values are best, median and worst.

Table 2

Comparison of IGD values of DTLZ1 Function

| Function | M | Statistical Test | MaOJaya | Results Taken From (Deb & Jain, 2014) | | | |
|----------|----|------------------|-----------------|---------------------------------------|-----------------|------------|-----------|
| | | | | NSGA-III | MOEA/D-PBI | MOEA/D-TCH | MOEA/D-DE |
| DTLZ1 | 3 | Best | 0.00E+00 | 4.880E-4 | 4.095E-4 | 3.296E-2 | 5.470E-3 |
| | | Median | 0.00E+00 | 1.308E-3 | 1.495E-3 | 3.321E-2 | 1.778E-2 |
| | | Worst | 0.00E+00 | 4.880E-3 | 4.743E-3 | 3.359E-2 | 3.394E-1 |
| | 5 | Best | 3.00E-6 | 5.116E-4 | 3.179E-4 | 1.124E-1 | 2.149E-2 |
| | | median | 1.55E-5 | 9.799E-4 | 6.372E-4 | 1.129E-1 | 2.489E-2 |
| | | Worst | 7.35E-4 | 1.979E-3 | 1.635E-3 | 1.137E-1 | 3.432E-2 |
| | 8 | Best | 4.56E-3 | 2.044E-3 | 3.914E-3 | 1.729E-1 | 3.849E-2 |
| | | median | 2.10E-2 | 3.979E-3 | 6.106E-3 | 1.953E-1 | 4.145E-2 |
| | | Worst | 5.28E-2 | 8.721E-3 | 8.537E-3 | 2.094E-1 | 4.815E-2 |
| | 10 | Best | 4.17E-3 | 2.215E-3 | 3.872E-3 | 2.072E-1 | 4.253E-2 |
| | | median | 7.15E-3 | 3.462E-3 | 5.073E-3 | 2.147E-1 | 4.648E-2 |
| | | Worst | 3.74E-2 | 6.869E-3 | 6.130E-3 | 2.400E-1 | 4.908E-2 |

Best results are indicated in bold face

Table 2 shows comparison between results obtained using proposed MaOJaya algorithm and existing results taken from Deb and Jain (2014) of DTLZ1 function for 3 to 10 objectives. The MaOJaya algorithm performs better than decomposition based approaches viz., MOEA/D-PBI, MOEA/D-TCH, and MOEA/D-DE for selected objective functions. For eight and ten objectives the NSGA-III algorithms performs better than MaOJaya algorithm. The proposed approach is executed with 200 population size and 500 iterations, whereas Deb and Jain (2014) varied population size from 92 to 276 and number of iterations used are from 400 to 1000. The performance of MaOJaya is better than MOEA/D-TCH for all objectives of DTLZ1 test function. As number of objectives increases the MOEA/D-TCH algorithm's performance deteriorates, though MaOJaya is based on Tchebycheff approach its performance is better than MOEA/D-TCH. The NSGA-III performs better than MaOJaya algorithm for eight and ten objectives.

The IGD values obtained for DTLZ2 test function presented in Table 3. The IGD values given for MOEA/D-PBI are better than MaOJaya algorithm for three objectives. The median and worst values obtained for five and ten objectives are better than MaOJaya algorithm, while the best values obtained for five, eight and ten objectives are better than values given by Deb and Jain (2014).

The DTLZ3 problem has number of local Pareto-front, so it's difficult to obtain global Pareto-front. The MaOJaya algorithm obtains better IGD values for three to ten objectives than MOEA/D-TCH and MOEA/D-DE algorithms, as shown in Table 4. The MOEA/D-PBI has better values for five objectives and NSGA-III has better values for three objectives.

Table 3
Comparison of IGD values of DTLZ2 Function

| Function | M | Statistical Test | MaOJaya | Results Taken From (Deb & Jain, 2014) | | | |
|----------|----|------------------|-----------------|---------------------------------------|-----------------|------------|-----------|
| | | | | NSGA-III | MOEA/D-PBI | MOEA/D-TCH | MOEA/D_DE |
| DTLZ2 | 3 | Best | 1.000E-3 | 1.262E-3 | 5.432E-4 | 7.499E-2 | 3.849E-2 |
| | | Median | 5.279E-3 | 1.357E-3 | 6.406E-4 | 7.574 | 4.562E-2 |
| | | Worst | 1.012E-2 | 2.114E-3 | 8.006E-4 | 7.657 | 6.069E2 |
| | 5 | Best | 7.719E-4 | 4.254E-3 | 1.219E-3 | 2.935E-1 | 1.595-1 |
| | | Median | 2.576E-3 | 4.982E-3 | 1.437E-3 | 2.945E-1 | 1.820E-1 |
| | | Worst | 4.189E-3 | 5.862E-3 | 1.727E-3 | 2.953E-1 | 1.935E-1 |
| | 8 | Best | 3.031E-3 | 1.371E-2 | 3.097E-3 | 5.989E-1 | 3.003E-1 |
| | | Median | 5.171E-3 | 1.571E-2 | 3.763E-3 | 6.301E-1 | 3.194E-1 |
| | | Worst | 4.187E-3 | 1.811E-2 | 5.198E-3 | 6.606E-1 | 3.481E-1 |
| | 10 | Best | 2.019E-3 | 1.350E-2 | 2.474E-3 | 7.002E-1 | 2.629E-1 |
| | | Median | 4.497E-3 | 1.528E-2 | 2.778E-3 | 7.266E-1 | 3.194E-1 |
| | | Worst | 6.091E-3 | 1.697E-2 | 3.235E-3 | 7.704E-1 | 3.337E-1 |

Best results are indicated in bold face

Table 4
Comparison of IGD values of DTLZ3 Function

| Function | M | Statistical Test | MaOJaya | Results Taken From (Deb & Jain, 2014) | | | |
|----------|----|------------------|-----------------|---------------------------------------|-----------------|------------|-----------|
| | | | | NSGA-III | MOEA/D-PBI | MOEA/D-TCH | MOEA/D-DE |
| DTLZ3 | 3 | Best | 5.519E-3 | 9.751E-4 | 9.773E-4 | 7.602E-2 | 5.610E-2 |
| | | Median | 3.949E-2 | 4.007E-3 | 3.426E-3 | 7.658E-2 | 1.439E-1 |
| | | Worst | 1.024E-1 | 6.665E-3 | 9.113E-3 | 7.764E-2 | 8.887E-1 |
| | 5 | Best | 4.648E-3 | 3.086E-3 | 1.129E-3 | 2.938E-1 | 1.544E-1 |
| | | Median | 1.310E-2 | 5.960E-3 | 2.213E-3 | 2.948E-1 | 2.115E-1 |
| | | Worst | 1.929E-2 | 1.196E-2 | 6.147E-3 | 2.956E-1 | 8.152E-1 |
| | 8 | Best | 6.250E-4 | 1.244E-2 | 6.459E-3 | 6.062E-1 | 2.607E-1 |
| | | Median | 8.306E-3 | 2.375E-2 | 1.948E-2 | 6.399E-1 | 3.321E-1 |
| | | Worst | 2.879E-3 | 9.649E-2 | 1.123 | 6.808E-1 | 3.923 |
| | 10 | Best | 8.060E-4 | 8.849E-3 | 2.791E-3 | 7.174E-1 | 2.549E-1 |
| | | Median | 9.991E-3 | 1.188E-2 | 4.319E-3 | 7.398E-1 | 2.789E-1 |
| | | Worst | 2.931E-3 | 2.083E-2 | 1.010 | 8.047E-1 | 2.998E-1 |

Best results are indicated in bold face

The IGD values comparison for DTLZ4 test function is presented in Table 5. The MaOJaya algorithm performs better than MOEA/D-PBI, MOEA/D-TCH, and MOEA/D-DE algorithms for three to ten objectives. The NSGA-III has better IGD values for three objectives. The performance of MOEA/D-TCH is always poor than MaOJaya algorithm for three to ten objectives of DTLZ4 function.

Table 5
Comparison of IGD values of DTLZ4 Function

| Function | M | Statistical Test | MaOJaya | Results Taken From (Deb & Jain, 2014) | | | |
|----------|----|------------------|-----------------|---------------------------------------|------------|------------|-----------|
| | | | | NSGA-III | MOEA/D-PBI | MOEA/D-TCH | MOEA/D-DE |
| DTLZ4 | 3 | Best | 2.545E-3 | 2.915E-4 | 2.929E-1 | 2.168E-1 | 3.276E-2 |
| | | Median | 3.108E-2 | 5.970E-4 | 4.280E-1 | 3.724E-1 | 6.049E-2 |
| | | Worst | 8.056E-2 | 4.286E-1 | 5.234E-1 | 4.421E-1 | 3.468E-1 |
| | 5 | Best | 4.055E-4 | 9.849E-4 | 1.080E-1 | 3.846E-1 | 1.090E-1 |
| | | Median | 1.244E-3 | 1.255E-3 | 5.787E-1 | 5.527E-1 | 1.479E-1 |
| | | Worst | 2.189E-3 | 1.721E-3 | 7.348E-1 | 7.491E-1 | 4.116E-1 |
| | 8 | Best | 2.620E-4 | 5.079E-3 | 5.298E-1 | 6.676E-1 | 2.333E-1 |
| | | Median | 6.079E-3 | 7.054E-3 | 8.816E-1 | 9.036E-1 | 3.333E-1 |
| | | Worst | 3.607E-2 | 6.051E-1 | 9.723E-1 | 1.035 | 7.443E-1 |
| | 10 | Best | 2.760E-4 | 5.694E-3 | 3.966E-1 | 7.734E-1 | 2.102E-1 |
| | | Median | 7.884E-3 | 6.337E-3 | 9.203E-1 | 9.310E-1 | 2.885E-1 |
| | | Worst | 1.022E-1 | 1.076E-1 | 1.077 | 1.039 | 6.422E-1 |

Best results are indicated in bold face

The Table 6 presents IGD values of DTLZ5 test function obtained for three to ten objectives by proposed MaOJaya algorithm. The values presented are best, median and worst. The DTLZ5 have irregular and degenerate Pareto front nature. The results are obtained for three to ten objectives of DTLZ5 test function. From the results it is observed that, the MaOJaya algorithm's performance degrades as number of objectives increased for many-objective problems with irregular Pareto front. It is necessary to do the certain modifications so that the performance of MaOJaya will improved.

Table 6
IGD values of DTLZ5 Function

| Function | Algorithm | Statistical Test | M | | | |
|----------|-----------|------------------|----------|----------|----------|----------|
| | | | 3 | 5 | 8 | 10 |
| DTLZ5 | MaOJaya | Best | 4.053E-2 | 2.087E-2 | 5.273E-2 | 1.160E-2 |
| | | Median | 5.695E-2 | 4.179E-2 | 7.879E-1 | 3.590E-1 |
| | | Worst | 1.044E-1 | 6.596E-1 | 1.008 | 1.084 |

Deb and Jain (2014) have tested and compared NSGA-III algorithm with other decomposition based approaches for DTLZ1 to DTLZ4 and two WFG test functions. The present article also test the MaOJaya algorithm for DTLZ5 test function. Table 7 shows the comparison of HV mean and standard deviation (SD) values obtained for three, eight and ten objectives of DTLZ1 to DTLZ4 test functions and results presented in (Ran Cheng et al., 2016).

Table 7
HV values of DTLZ1 – DTLZ4 Function

| Function | M | Statistical Test | MaOJaya | (Taken from Ran Cheng et al. 2016) | | | |
|----------|----|------------------|-----------------|------------------------------------|-----------------|----------|-----------------|
| | | | | RVEA | MOEA/DD | NSGA-III | MOEA/D-PBI |
| DTLZ1 | 3 | Mean | 0.980110 | 0.992299 | 0.992321 | 0.992275 | 0.992217 |
| | | SD | 0.000002 | 0.000027 | 0.000005 | 0.000017 | 0.000068 |
| | 8 | Mean | 0.980007 | 0.999999 | 0.999984 | 0.999993 | 0.999998 |
| | | SD | 0.004231 | 0.000000 | 0.000016 | 0.000022 | 0.000001 |
| | 10 | Mean | 0.984308 | 0.999999 | 0.999995 | 0.999992 | 0.999999 |
| | | SD | 0.005430 | 0.000000 | 0.000005 | 0.000027 | 0.000000 |
| DTLZ2 | 3 | Mean | 0.995446 | 0.926994 | 0.927292 | 0.927012 | 0.926808 |
| | | SD | 0.000756 | 0.000041 | 0.000002 | 0.000032 | 0.000042 |
| | 8 | Mean | 0.988943 | 0.999338 | 0.999330 | 0.998223 | 0.999333 |
| | | SD | 0.008156 | 0.000096 | 0.000079 | 0.002551 | 0.000024 |
| | 10 | Mean | 0.984355 | 0.999912 | 0.999920 | 0.999804 | 0.999917 |
| | | SD | 0.005430 | 0.000040 | 0.000022 | 0.000222 | 0.000011 |
| DTLZ3 | 3 | Mean | 0.997415 | 0.924421 | 0.926921 | 0.925650 | 0.924153 |
| | | SD | 0.000223 | 0.001273 | 0.000316 | 0.000664 | 0.001546 |
| | 8 | Mean | 0.998422 | 0.999350 | 0.999338 | 0.683011 | 0.997552 |
| | | SD | 0.007509 | 0.000090 | 0.000069 | 0.452841 | 0.004021 |
| | 10 | Mean | 0.996856 | 0.999919 | 0.999914 | 0.588230 | 0.883623 |
| | | SD | 0.002085 | 0.000027 | 0.000033 | 0.507488 | 0.195053 |
| DTLZ4 | 3 | Mean | 0.970075 | 0.926922 | 0.927295 | 0.926947 | 0.761109 |
| | | SD | 0.000892 | 0.000049 | 0.000001 | 0.000035 | 0.188313 |
| | 8 | Mean | 0.980695 | 0.999359 | 0.999374 | 0.999016 | 0.987237 |
| | | SD | 0.000918 | 0.000043 | 0.000057 | 0.000055 | 0.013487 |
| | 10 | Mean | 0.987951 | 0.999915 | 0.999916 | 0.999844 | 0.998947 |
| | | SD | 0.000018 | 0.000036 | 0.000019 | 0.000023 | 0.000615 |

Best results are indicated in bold face

The statistical results are obtained by executing the MaOJaya algorithm for 20 times independently. The RVEA, MOEA/DD, NSGA-III and MOEA/D-PBI algorithm's result given by Ran Cheng et al. (2016) are used to compare with MaOJaya algorithm. From the results, it is observed that the results given in (Ran Cheng et al., 2016) have better HV values than proposed MaOJaya algorithm for most

of the test functions. There is no significant difference between results obtained by MaOJaya algorithm and results used for comparison. The results are encouraging of all benchmark test function for three to ten objectives. The MaOJaya algorithm obtains better values for three objective test instance of DTLZ2, DTLZ3 and DTLZ4 function. The DTLZ2 is simple test function than DTLZ1 function. The DTLZ1 and DTLZ3 are multimodal test functions whereas in DTLZ4 test function, the solution points on true Pareto front are biased. From the results it is observed that, by performing modifications in proposed MaOJaya algorithm or by changing the common parameters settings, performance of MaOJaya can be evaluated.

Table 8

HV values of DTLZ5 Function

| Function | M | Statistical Test | MaOJaya | (Taken from Ran Cheng et al. 2016) | |
|----------|----|------------------|-----------------|------------------------------------|----------|
| | | | | RVEA* | RVEA |
| DTLZ5 | 3 | Mean | 0.725409 | 0.730371 | 0.725655 |
| | | SD | 0.000024 | 0.000086 | 0.000524 |
| | 8 | Mean | 0.635480 | 0.640410 | 0.581582 |
| | | SD | 0.008509 | 0.009400 | 0.122095 |
| | 10 | Mean | 0.637655 | 0.643795 | 0.609602 |
| | | SD | 0.000281 | 0.002908 | 0.002982 |

Best results are indicated in bold face

The Table 8 shows the Hypervolume values comparison between MaOJaya, RVEA* and RVEA algorithms for DTLZ5 test function. The results obtained are competitive with existing results and are not significantly different than RVEA* and RVEA algorithms.

The proposed many-objective Jaya (MaOJaya) algorithm gives better results due to the following reasons.

- In basic Jaya algorithm solutions always move towards the best solution and remove the worst solution obtained for selected problem.
- The reference point used gives the direction to the algorithm to move towards best result.
- Instead of approximating the entire Pareto front, the proposed work focus on subset of PF by using reference point so more focused search is performed.
- Complexity of the MaOPs is reduced by using decomposition approach and solving multiple single objective problems in collaboration manner, which improves the efficiency of proposed algorithm. The decomposition-based approach maintains both diversity and convergence.

6. Conclusion

This work is to design many-objective optimization algorithm to solve MaOPs. The many-objective Jaya (MaOJaya) algorithm is designed to solve MaOPs. The basic Jaya algorithm is modified while preserving its important properties. The Tchebycheff – a decomposition based approach is used to simplify the complex MaOPs. The basic Jaya algorithm is modified by introducing non-dominated sorting from NSGA-II and tournament selection scheme, with tournament size 2. The reference point mechanism is introduced to traverse algorithm towards the best solutions.

The proposed MaOJaya algorithm is tested on DTLZ benchmark functions to measure its applicability and effectiveness to solve many-objective optimization problems. The IGD performance metric is used to measure the performance as it measures both convergence and diversity of obtained solutions. The obtained IGD values compared with the best known results. After comparison it is observed that, the proposed MaOJaya algorithm gives comparatively better results than known best results for DTLZ1, DTLZ3 and DTLZ4 test functions and competitive results for other test functions. The HV performance metric is also used to measure the closeness and diversity of solutions. The HV values obtained using MaOJaya algorithm does not have significant difference than best known results taken from (Ran

Cheng et al., 2016). As per best knowledge of the authors of this paper, the design of many-objective Jaya algorithm is not reported in literature. The proposed MaOJaya algorithm can be implemented by incorporating other algorithmic development strategies such as, preference information, decomposition schemes, indicator based approach objective reduction and hybrid approach in future. This article has evaluated the performance of proposed MaOJaya algorithm using DTLZ1 to DTLZ5 test problems, in future scaled DTLZ problems, constrained DTLZ problems and WFG test problems can be tested to assess the performance of MaOJaya algorithm. As future research work, the proposed MaOJaya algorithm can be customized to solve the real-time applications from engineering, scientific or any other domain with more than three objectives.

References

- Abhishek, K., Kumar, V. R., Datta, S., & Mahapatra, S. S. (2016). Application of JAYA algorithm for the optimization of machining performance characteristics during the turning of CFRP (epoxy) composites: comparison with TLBO, GA, and ICA. *Engineering with Computers*, 1-19.
- Asafuddoula, M., Ray, T., & Sarker, R. (2015). A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3), 445-460.
- Azizipanah-Abarghooee, R., Dehghanian, P., & Terzija, V. (2016). Practical multi-area bi-objective environmental economic dispatch equipped with a hybrid gradient search method and improved Jaya algorithm. *IET Generation, Transmission & Distribution*, 10(14), 3580-3596.
- Bhoye, M., Pandya, M. H., Valvi, S., Trivedi, I. N., Jangir, P., & Parmar, S. A. (2016, April). An emission constraint economic load dispatch problem solution with microgrid using JAYA algorithm. In *Energy Efficient Technologies for Sustainability (ICEETS), 2016 International Conference on* (pp. 497-502). IEEE.
- Chand, S., & Wagner, M. (2015). Evolutionary many-objective optimization: a quick-start guide. *Surveys in Operations Research and Management Science*, 20(2), 35-42.
- 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.
- Cheung, Y. M., Gu, F., & Liu, H. L. (2016). Objective extraction for many-objective optimization problems: Algorithm and test problems. *IEEE Transactions on Evolutionary Computation*, 20(5), 755-772.
- Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation*, 18(4), 577-601.
- Figueiredo, E. M., Araújo, D. R., Bastos Filho, C. J., & Ludermir, T. B. (2016, October). Physical Topology Design of Optical Networks Aided by Many-Objective Optimization Algorithms. In *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on* (pp. 409-414). IEEE.
- Gong, D. W., Sun, J., & Miao, Z. (2016). A Set-based Genetic Algorithm for Interval Many-objective Optimization Problems. *IEEE Transactions on Evolutionary Computation*.
- Ibrahim, A., Rahnamayan, S., Martin, M. V., & Deb, K. (2016, July). EliteNSGA-III: An improved evolutionary many-objective optimization algorithm. In *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 973-982). IEEE.
- Khosravi, S., & Akbarzadeh-T, M. R. (2015, November). A tree-of-heaven inspired approach for solving many-objective optimization problems. In *Technology, Communication and Knowledge (ICTCK), 2015 International Congress on* (pp. 104-111). IEEE.
- Li, B., Li, J., Tang, K., & Yao, X. (2015). Many-objective evolutionary algorithms: A survey. *ACIM Computing Surveys (CSUR)*, 48(1), 13.
- Lwin, K., Qu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, 24, 757-772.
- Mishra, S., & Ray, P. K. (2016). Power quality improvement using photovoltaic fed DSTATCOM based on JAYA optimization. *IEEE Transactions on Sustainable Energy*, 7(4), 1672-1680.
- Murata, T., Ishibuchi, H., & Gen, M. (2001, March). Specification of genetic search directions in cellular multi-objective genetic algorithms. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 82-95). Springer Berlin Heidelberg.

- Pandey, H. M. (2016, January). Jaya a novel optimization algorithm: What, how and why?. In *Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference* (pp. 728-730). IEEE.
- Prakash, T., Singh, V. P., Singh, S. P., & Mohanty, S. R. Binary Jaya Algorithm Based Optimal Placement of Phasor Measurement Units for Power System Observability.
- 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. V., & Saroj, A. (2016). Multi-objective design optimization of heat exchangers using elitist-Jaya algorithm. *Energy Systems*, 1-37.
- Rao, R. V., & Saroj, A. (2017). A self-adaptive multi-population based Jaya algorithm for engineering optimization. *Swarm and Evolutionary Computation*.
- Rao, R. V., & Waghmare, G. G. (2017). A new optimization algorithm for solving complex constrained design optimization problems. *Engineering Optimization*, 49(1), 60-83.
- Rao, R. V., More, K. C., Taler, J., & Ocoń, P. (2016). Dimensional optimization of a micro-channel heat sink using Jaya algorithm. *Applied Thermal Engineering*, 103, 572-582.
- Rao, R. V., Rai, D. P., & Balic, J. (2017). A new optimization algorithm for parameter optimization of nano-finishing processes. *Scientia Iranica. Transaction E, Industrial Engineering*, 24(2), 868.
- Rao, R. V., Rai, D. P., & Balic, J. (2017). A multi-objective algorithm for optimization of modern machining processes. *Engineering Applications of Artificial Intelligence*, 61, 103-125
- Rao, R. V., Rai, D. P., Ramkumar, J., & Balic, J. (2016). A new multi-objective Jaya algorithm for optimization of modern machining processes. *Advances in Production Engineering & Management*, 11(4), 271.
- Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303-315.
- Seshadri, A. (2006). Multi-objective optimization using evolutionary algorithms (MOEA). *Matlab Website: <http://www.Mathworks.com/matlabcentral/fileexchange/10429>, by, 19.*
- Singh, S. P., Prakash, T., Singh, V. P., & Babu, M. G. (2017). Analytic hierarchy process based automatic generation control of multi-area interconnected power system using Jaya algorithm. *Engineering Applications of Artificial Intelligence*, 60, 35-44.
- Wang, R. (2016). An Improved Nondominated Sorting Genetic Algorithm for Multiobjective Problem. *Mathematical Problems in Engineering*, 2016.
- Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., & Zhang, T. (2016). Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation*.
- Wang, W., Ying, S., Li, L., Wang, Z., & Li, W. (2017). An improved decomposition-based multiobjective evolutionary algorithm with a better balance of convergence and diversity. *Applied Soft Computing*, 57, 627-641.
- Wang, Z. J., Zhan, Z. H., & Zhang, J. (2016, December). Parallel multi-strategy evolutionary algorithm using message passing interface for many-objective optimization. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on* (pp. 1-8). IEEE.
- Warid, W., Hizam, H., Mariun, N., & Abdul-Wahab, N. I. (2016). Optimal Power Flow Using the Jaya Algorithm. *Energies*, 9(9), 678.
- Zhang, X., Tian, Y., & Jin, Y. (2015). A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(6), 761-776.
- 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.
- Zhuansun, X., Zhu, A., Wu, J., Han, T., & Chen, Y. (2016, October). Many-objective reactive power optimization using Particle Swarm Optimization algorithm based on Pareto entropy. In *Power and Energy Engineering Conference (APPEEC), 2016 IEEE PES Asia-Pacific* (pp. 923-928). IEEE.

