

## AN IMPROVED EVAPORATION RATE-WATER CYCLE ALGORITHM BASED GENETIC ALGORITHM FOR SOLVING GENERALIZED RATIO PROBLEMS

C. VEERAMANI<sup>1</sup> AND S. SHARANYA<sup>2</sup>

**Abstract.** This paper presents an efficient metaheuristic approach for optimizing the generalized ratio problems such as the sum and multiplicative of linear or nonlinear ratio objective function with affine constraints. This paper focuses on the significance of hybrid techniques, which are implemented by using GA and ER-WCA to increase efficiency and robustness for solving linear and nonlinear generalized ratio problems. Initially, GA starts with an initial random population and it is processed by genetic operators. ER-WCA will observe and preserve the GAs fittest chromosome in each cycle and every generation. This Genetic ER-WCA algorithm is provided with better optimal solutions while solving constrained ratio optimization problems. Also, the effectiveness of the proposed genetic ER-WCA algorithm is analyzed while solving the large scale ratio problems. The results and performance of the proposed algorithm ensures a strong optimization and improves the exploitative process when compared to the other existing metaheuristic techniques. Numerical problems and applications are used to test the performance of the convergence and the accuracy of the approached method. The behavior of this Genetic ER-WCA algorithm is compared with those of evolutionary algorithms namely Neural Network Algorithm, Grey Wolf Optimization, Evaporation Rate - Water Cycle Algorithm, Water Cycle Algorithm, Firefly algorithm, Cuckoo search algorithm. The evaluated results show that the proposed algorithm increases the convergence and accuracy more than other existing algorithms.

**Mathematics Subject Classification.** 90C32, 90C26, 90C59.

Received April 12, 2019. Accepted April 28, 2020.

### 1. INTRODUCTION

The linear and non-linear problems are solved with various types of traditional and metaheuristic approaches. But in real-life problems can be linear or non-linear problems. Finding solutions to such non-linear problems is more complicated than linear problems. The nonlinear problems are formulated as the nonlinear function with either linear or nonlinear constraints. The fractional problem also called nonlinear problems. So nonlinear programming problems can be either nonlinear or fractional function. This paper discussed the optimum solution for the fractional problem, which has linear and nonlinear functions in numerator and denominator. The purpose of fractions is “the area we need in the aggregate”. For example, inventory/sales, actual cost/standard cost, output/employee.

---

*Keywords.* Optimization Problems, Evolutionary Algorithms, Genetic Algorithm, Water Cycle Algorithm.

<sup>1</sup> Department of Applied Science, PSG College of Technology, Coimbatore 641 004, India.

<sup>2</sup> Research Scholar, Department of Applied Science, PSG College of Technology, Coimbatore 641 004, India.

\*Corresponding author: [ssharanya015@gmail.com](mailto:ssharanya015@gmail.com)

The generalized ratio problems are very useful to formulate all types of real-world problems. This is because, for example, ratios are used to solve all problems, such as finding the profit for a particular product from the total profit in a company. So, this fractional problem is applied to various areas like economics, engineering, business, finance, etc. In the 1978s, the theory of fractional programming problem was initially demonstrated by S. Schaible and was used in many applications. After that, Craven proposed the next monograph in 1988. Jamali *et al.* [3] proposed an approach, hybrid Improved Cuckoo Search algorithm (ICS) and Genetic Algorithm (GA) to solve the Markov-Modulated Demand problem. Veeramani *et al.* [15] proposed a solution procedure to solve fuzzy linear fractional programming problems.

Over the past years, various solution methods have been developed to solve the generalized fractional programming problems. The generalized linear ratio problems, such as Sum or product ratio problems, are solved using a novel unified method proposed by Phuong *et al.* [9]. Similarly, to solve such type of problems, Shen *et al.* [13] proposed a range division and linearization algorithm. Shen *et al.* [11] developed a regional division and cut algorithm for solving the sum of linear ratio problems by transformed an equivalent optimization problem. The outcome range reduction procedure was proposed by Jiao *et al.* [5] for solving the SAR problem. The process of this method is the new outcome space for denominator was range reduction and branching operations with the help of the branch and bound algorithm. He also proposed a novel method for solving the generalized affine multiplicative programming problem with polynomial constraints by using branch and bound method [4]. A new method of polynomial theory using Dinkelbachis algorithm was proposed for solving fractional polynomial problems, by Pizzo *et al.* [8].

Jong [6] presented a new method for solving nonlinear sum of ratio problems in various applications. He also proved that theoretically and numerically the sum of ratio problems is given global optimal solutions by using the method. A generalized fractional programming problem has been solved using an efficient unified method that was developed by Shen *et al.* [12]. This method is developed by using a two-part linearization method, a sequence of linear programming relaxations that are embedded in a branch-and-bound algorithm. Chun-Feng *et al.* [2] solving a sum of geometric fractional functions under geometric constraints using an efficient branch and bound algorithm in various applications. This method was using an equivalent transformation and a new linear relaxation technique to found global solutions. Shen *et al.* [14] proposed a practicable contraction algorithm to solve the sum of the generalized polynomial ratios problem. Liu *et al.* [7] proposed a new global optimization algorithm, which can better solve a class of linear fractional programming problems on a large scale by using A output-space branch and bound algorithm.

This paper presents a Genetic ER-WCA Algorithm, which combines the Genetic Algorithm and Evaporation Rate based Water Cycle Algorithm to solve the constrained ratio Problems. In the recommended hybridization technique, initially, GA works with genetic operators and produces new offspring that are the good populations of the ER-WCA algorithm. Meanwhile, the local optimal solutions produced by GA are verified using the boundary strategy, that is, whether the solutions are still constrained by lower and upper boundaries. After that, the ER-WCA is implemented through the generated offsprings of GA. This process is repeated until a set of solutions arrive.

The rest of the work is organized as follows: Section 2 explains the formulation of various fractional programming problems. Section 3 designs an algorithm for the proposed method. Section 4 illustrates the proposed procedure through small scale numerical examples. Section 5 interprets the formulation of large scale ratio problems and also compares the results with the existing methods for small scale numerical problems and large scale ratio problems and Section 6 discusses a mathematical formulation of application ratio problem and the results are proposed and Section 7 draws the conclusion of the paper.

## 2. PROBLEM FORMULATION

The ratio problems are expressed by ratios of linear and nonlinear functions which are named nonlinear programming problems (or) fractional programming problems. The general formulation of the generalized fractional

programming problem is as follows [1]:

$$\begin{aligned}
 &\text{Max/Min } F(x) = \frac{N(x)}{D(x)} \\
 &\text{Subject to} \\
 &A(x) \left\{ \begin{array}{l} \leq \\ = \\ \geq \end{array} \right\} b, \quad x > 0,
 \end{aligned} \tag{2.1}$$

where  $F(x)$  be a real-valued, concave function defined on a desirable convex compact set. That is, the numerator and denominators are concave or convex, differentiable and non-negative functions with the constraints are also convex or concave. The solution set  $S$  is denoted as the feasible solution set of the ratio problem, which is on convex set

$$S = \{x \in R^n : A(x) \leq b, x > 0\} \text{ and } D_i(x) \neq 0.$$

Unfortunately, most of the real-world problems have one or more fractions with distinct types of formulations. So, the fractional programming problem is categorized as Multiple Objective Ratios, Multiplicative Ratios, Sum-of-ratios and Fractional polynomial Optimization. This paper will be focused on the Sum and Product of ratio problems. The formulation of these problems are as follows:

**2.1. Sum-of-ratios problem (SRP)**

$$\begin{aligned}
 &\text{Max/Min } \phi(x) = \sum_{q=1}^r \left( \frac{\sum_{j=1}^n c_{qj}x_j + \alpha_i}{\sum_{j=1}^n d_{qj}x_j + \beta_i} \right)^\gamma \\
 &\text{Subject to} \\
 &\sum_{j=1}^n a_{ij}x_j \left\{ \begin{array}{l} \leq \\ = \\ \geq \end{array} \right\} b_i, \quad x > 0,
 \end{aligned} \tag{2.2}$$

where  $\phi(x)$  is a concave function defined throughout  $R^n$ . Also assume that the set  $S$  is non-empty and compact convex set. That is,  $N_i(x) = \sum_{j=1}^n c_{qj}x_j + \alpha_i$ ,  $D_i(x) = \sum_{j=1}^n d_{qj}x_j + \beta_i$ ,  $x, n, r \in R^n$ , and  $\alpha_i, \beta_i, \gamma \in R$ .

**2.2. Multiplicative ratios problem (MRP)**

$$\begin{aligned}
 &\text{Max/Min } \psi(x) = \prod_{i=1}^k \left( \frac{\sum_{j=1}^n c_{qj}x_j + \alpha_i}{\sum_{j=1}^n d_{qj}x_j + \beta_i} \right)^\gamma \\
 &\text{Subject to} \\
 &\sum_{j=1}^n a_{ij}x_j \left\{ \begin{array}{l} \leq \\ = \\ \geq \end{array} \right\} b_i, \quad x > 0.
 \end{aligned} \tag{2.3}$$

Similar to the SRP problems,  $\psi(x)$  be a continuously differentiable concave function defined on convex set. Where  $N_i(x) = \sum_{j=1}^n c_{qj}x_j + \alpha_i$ ,  $D_i(x) = \sum_{j=1}^n d_{qj}x_j + \beta_i$ ,  $x, n, r \in R^n$ , and  $\alpha_i, \beta_i, \gamma \in R$ .

The SRP and MRP problems have many applications in worldwide, such as fuzzy pricing, cluster analysis, cargo problems, portfolio optimization, production planning, etc. Hence, this paper will be focused on SRP, MRP, Large-scale ratio problems. Further, a production planning area application problem also will be solved.

**3. PROPOSED ALGORITHM**

Generally, non-linear programming problems are resolved with traditional and metaheuristic methods. But, traditional and evolutionary optimization could not achieve global optimum while solving non-linear programming problems. Therefore, many hybrid mechanisms are combined and developed in two or more evolutionary algorithms to achieve nearby global optimum. Those algorithms were executed one after another. As such, the optimal solution of an algorithm was provided as input for the other method.

### 3.1. Genetic algorithm

A genetic algorithm is a search-based optimization method that starts with a solution set (chromosomes) called population ( $P_{ga}$ ). After the evaluation of the fitness function, the mechanism obtains the fitness value for each individual. The GA follows the following structure, the evaluation, Selection, Crossover, and Mutation. In the selection stage, parents are produce offspring that come into the parents features. If the parents have better fitness, their offspring will be better than the parents and have a better chance to survive. In the Crossover stage, mated each pair of parents, which is the second part of the first parent will be exchanged with the second part of the second parent. In the Mutation stage, the elected individuals have reversed alleles that are if the allele is 0, then it is changed to 1 and vice versa. The structure of stages is reiterated to produce new individuals in each generation which are better than the preceding generations.

### 3.2. Bounding strategy

The proposed method is developed by connecting GA and ER-WCA algorithms (Genetic ER-WCA) for attaining the global optimal solution. Similar to other evolutionary algorithms, our hybrid method starts with fixed boundaries of random numbers according to the problem dimension. In this mechanism, try to update the boundaries of the variables after the evolution process of GA. Then, the initial population of ER-WCA, performs the boundary updating every iteration.

After reaching the local optimal population, the population is likely to move out of its borders. Therefore, the boundary is checked after receiving the local optimal of GA. Initially, the fitness of the problem is found by applying the values between the fixed LB and the UB. The process of a genetic algorithm is implemented using these fitness values and achieving local optimal solutions. Now simply select good the raindrops in the population by dint of  $\alpha * N_{pop}$ , where  $\alpha$  is the pre-determinant constant referred to as the selecting fraction, and  $0 < \alpha < 1$ . The search space of selected individuals depends on the initially defined LB and UB. The selecting fraction  $\alpha$  is used to determine the number of individuals that should be selected based on their fitness values. Since the selecting fraction can affect the performance and convergence of the proposed method, which is set close to 0. if it is nearer to 1, the updated boundary of the population will be close to the existing boundary of the population. These updated boundaries being too narrow and may increase the possibility of missing the true optimal solution. The value of  $\alpha$  is chosen to be 0.2 for all problems, which means 20% of the individuals are selected.

In this paper, the updated boundaries of the selected population is established with the help of multivariate gaussian distribution. The mean and std deviation are found by using the following equations:

$$\begin{aligned}\mu_i &= \frac{\sum_{j=1}^Q x_{ij}}{Q}, \\ \sigma_i &= \sqrt{\frac{\sum_{j=1}^Q (x_{i,j} - \mu)^2}{Q}},\end{aligned}\tag{3.1}$$

where  $\mu$  is the mean for  $i$ th variable and  $\sigma_i$  is the standard deviation for the  $i$ th variable. The probabilistic distribution  $D$  for the  $i$ th variable is as follows:

$$D = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}}.\tag{3.2}$$

Based on this probabilistic distribution  $D$ , the upper and lower bound of a variable  $x_i$  are updated as follows:

$$\begin{aligned}\text{flagUB}_i &= \mu_i + b * [\max\{\text{UB}_{\text{fix}}, \text{UB}_D - \mu_i\}], \\ \text{flagLB}_i &= \mu_i - b * [\mu_i - \min\{\text{LB}_{\text{fix}}, \text{LB}_D\}],\end{aligned}\tag{3.3}$$

where  $\text{UB}_i$  and  $\text{LB}_i$  are the updated upper and lower bound of the variable  $x_i$ ,  $\mu_i$  is the mean value of the variable  $x_i$ ,  $\text{UB}_{\text{fix}}$  and  $\text{LB}_{\text{fix}}$  are the fixed boundaries of the variables,  $\text{UB}_D$  and  $\text{LB}_D$  are obtained probabilistic distribution boundaries and  $b$  is the bounding factor, which is set to be grater than 1, but close to 1.

### 3.3. Evaporation rate-water cycle algorithm

The procedure of the water cycle is the main theme of the WCA. Initially, WCA creates an initial population named, such as raindrops. Then by evaluating the cost function, the cost of each individual (*i.e.*, raindrops) is obtained. The evaluated individuals are ordered as the sea, river, and streams by using the cost function value. Then assigning the summation of the number of rivers and a single sea is in  $N_{sr}$ . The nominated streams elected for each river and sea ( $N_{sr}$ ) are selected by an intensity flow. That is, the chosen streams flow either to the rivers or directly flow to the sea and, similarly, the nominated rivers flow to the sea. However, the solution established by a stream is better than its connecting river, the locations of the river and stream are swapped. Likewise, it happens for both the river and the sea. Finally, the evaporation will be performed. The evaporation process happens both rivers and the streams, which is nearer to the sea. If the evaporation process is satisfied, then the raining procedure should be performed. Otherwise, the value of  $d_{max}$  adaptively decreases. Then, some new streams in different positions are created. This process will continue until a set of the solution has arrived. Thus the WCA leads to an indirect move towards the best solution.

### 3.4. Hybrid genetic ER-WCA

The main advantage of genetic ER-WCA is to stimulate the exploitation and exploration process using GA and ER-WCA. Generally, GA works, to find new solutions with the help of crossover and mutation operations, which can improve the initial solutions to get optimal solutions. Because genetic operations produce offsprings better than parents. The offsprings are generated by combining parent’s information. Similarly, WCA also has the ability to explore in search space by using the concept of stream and river that update their position towards the sea and the bounding strategy leads to the global optimal solution from the local optimal solution by constraint the boundaries. By considering these advantages of algorithms, the hybrid genetic ER-WCA is proposed to find a guaranteed global solution. In this model, GA performs first, after that WCA will be performed. The genetic ER-WCA algorithm processes are summarized as follows:

**Step 1:** Genetic ER-WCA has been started to set initial population ( $N_P$ ), maximum iteration (max.it), number of streams ( $P_{sr}$ ), crossover percentage ( $C_{per}$ ), mutation percentage ( $M_{per}$ ), Lower bound (LB) and Upper bound (UB) for the problems.

**Step 2:** Initialize the random population (chromosome,  $P$ ) for GA. A random population consists of a matrix of the population with the size of  $P \times N$ , where  $N$  is a problem dimension (*i.e.*, Number of decision variable) and  $P$  is population size.

$$\text{A candidate} = [x_1^i, x_2^i, \dots, x_N^i]$$

**Step 3:** Calculate the fitness value for each chromosomes. Using the below equation,

$$C_i^{ga} = f(X_1^i, X_2^i, X_3^i, \dots, X_{P_{ga}}^i), \quad i = 1, 2, 3, \dots, P_{ga}, \tag{3.4}$$

then the chromosomes are performed gradually through selection and crossover and mutation.

**Step 4:** Apply roulette wheel selection on  $C_i^{ga}$ .

$$P_{ga}^{new} = \frac{C_i^{ga}}{\sum_{i=1}^{C_{ga}} C_i^{ga}}. \tag{3.5}$$

**Step 5:** Perform crossover and mutation operations on  $P_{ga}^{new}$  to produce GA’s best evaluated population ( $P_{ga}^{local}$ ) with the help of  $C_{per}$  and  $M_{per}$ .

**Step 6:** Save the better population of  $P_{ga}^{local}$  into  $P_{er-wca}$ , where  $P_{er-wca}$  is an initial population for ER-WCA. Here, the better population is formed by combining the crossover and mutation offsprings. Doing this, ER-WCA takes the advantages of GA, which leads to reaching the global optimal solution of the problem.

**Step 7:** Check the boundaries of the new population  $P_{er-wca}$  using bounding mechanism.

**Step 8:** Compute fitness value for  $P_{er-wca}$  using the equation

$$C_i^{er-wca} = f(X_1^i, X_2^i, X_3^i, \dots, X_{P_{er-wca}}^i), \quad i = 1, 2, 3, \dots, P_{er-wca}, \tag{3.6}$$

where  $P_{er-wca}$  is a dimension of optimization problem.

**Step 9:** The raindrop can be represented the size of  $N_p \times P_{er-wca}$ . Form initial Sea, river and streams in  $P_{er-wca}$  using the below equations

$$\text{Total population} = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \dots & x_{P_{er-wca}}^1 \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_{P_{er-wca}}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{N_p} & x_2^{N_p} & x_3^{N_p} & \dots & x_{P_{er-wca}}^{N_p} \end{bmatrix} = \begin{bmatrix} \text{Sea } (C_1^{er-wca}) \\ \text{Riv1 } (C_2^{er-wca}) \\ \text{Riv2 } (C_3^{er-wca}) \\ \text{Riv3 } (C_4^{er-wca}) \\ \vdots \\ \text{Stre } P_{sn} + 1 (C_{P_{sn}+1}^{er-wca}) \\ \text{Str } P_{sn} + 2 (C_{P_{sn}+2}^{er-wca}) \\ \vdots \\ \text{Str } N_p (C_{N_p}^{er-wca}) \end{bmatrix} \tag{3.7}$$

$$\begin{aligned} P_{sn} &= \text{number of rivers} + 1 \\ P_{strs} &= P_{er-wca} - P_{sn}. \end{aligned} \tag{3.8}$$

The sea, which has a minimum value of the total population and the better values of the number of rivers are chosen as river and also the remaining values (*i.e.*, good values) are chosen as the streams.

**Step 10:** find intensity flow for rivers and sea using

$$C_i = C_n^{er-wca} - C_{P_{sn}+1}^{er-wca} \quad i = 1, 2, \dots, P_{sn}, \tag{3.9}$$

$$P_{sr} = \text{round} \left| \frac{C_i}{\sum_{i=1}^{P_{sr}} C_i} \right| * P_{strs} \quad i = 1, 2, \dots, P_{sn}, \tag{3.10}$$

where  $P_{sr}$  is the number of streams that flow to the specific rivers or sea. From precipitation, most of the raindrops are generated. Each raindrops to flow various directions towards the sea. By using this intensity flow, the raindrops are assigned to flow the rivers and sea. Equation (3.11) interprets the streams, which tend to move towards the rivers and the sea-based on their magnitude of flow. Hence, equations (3.10) and (3.11) will be able to control and posses more streams.

**Step 11:** Find the distance  $V$  between streams and rivers randomly  $V \in (0, c \times d)$ ,  $1 < c < 2$  and  $d$  is the current distance. Hence, the value of  $V$  is a random number between 0 to  $c \times d$ .

**Step 12:** Find the new position of the streams and rivers using the following equations,

$$\vec{V}_S^{i+1} = \vec{V}_S^i + \text{rand} * C * (\vec{V}_R^i - \vec{V}_S^i), \tag{3.11}$$

$$\vec{V}_S^{i+1} = \vec{V}_S^i + \text{rand} * C * (\vec{V}_O^i - \vec{V}_S^i), \tag{3.12}$$

$$\vec{V}_R^{i+1} = \vec{V}_R^i + \text{rand} * C * (\vec{V}_O^i - \vec{V}_R^i), \tag{3.13}$$

where  $\vec{V}_S^i, \vec{V}_R^i$  and  $\vec{V}_O^i$  respectively be the  $i$ th position of the stream, river and sea in the search space. The new position of the raindrops at  $(i + 1)$ th generation is updated using equations (3.11) to (3.13). similarly, rand is the uniformly distributed number between 0 to 1 and  $C$  is the acceleration coefficient, which is 1 to 2 (near to 2). The new position of streams flowing to rivers is represented in equation (3.11). Similarly, the new location of streams flowing to sea is represented in equation (3.12). Also, equation (3.13), represent the new position of

rivers flowing to the sea. Here, the new position means the rivers/streams that exchange their positions with the sea and also rivers that exchange their positions with the streams, when one solution is better than the other one.

**Step 13:** Calculate the evaporation rate for only streams and rivers using

$$EVR = \frac{\text{Sum}(P_{sr})}{P_{sn}-1} \times \text{rand} \quad n = 2, \dots, P_{sn}. \tag{3.14}$$

**Step 14:** Check the evaporation condition among rivers and streams, also find the newly formed streams locations using below pseudo code,

$$\begin{aligned} &\text{for } i = 2 : P_{sn} - 1 \\ &\quad \text{if}(\exp(-k/\text{max\_it}) < \text{rand}) \& (P_{S_i} < EVR) \\ &\quad \quad V_S^{\text{new}}(t + 1) = LB + \left(\frac{\text{rand}}{\text{max\_it}}\right)^{\text{rand}} \times (UB - LB) \\ &\quad \text{end} \\ &\text{end} \end{aligned} \tag{P1}$$

where LB and UB are upper bound and lower bound of the problem. An evaporation rate is an important factor for surface runoff water. This help to improve our research towards an optimal solution. The Pseudocode (P1) represents if the rivers and their corresponding streams have a low evaporation rate (i.e quality solution is less), then the raining process will help to move or flow better search area. That is, if the given condition in (P1) is satisfied for any river, the river with its corresponding stream will be removed (i.e evaporated). After that the new streams equal to the number of previous streams and river will be created in new positions using equation  $V_s^{\text{new}}$ .

**Step 15:** Similarly, check the evaporation condition between sea and streams/rivers, also find the newly formed streams and rivers using the following pseudocode (P2) and (P3) respectively:

$$\begin{aligned} &\text{if} \|V_O - V_S\| < d_{\text{max}} \\ &\quad V_S^{\text{new}}(t + 1) = V_O(t) + \sqrt{\mu} \times \text{rand}n(1, N) \\ &\quad \text{end} \end{aligned} \tag{P2}$$

$$\begin{aligned} &\text{if} \|V_O - V_R\| < d_{\text{max}} \text{ or } \text{rand} < 0.1 \\ &\quad V_S^{\text{new}}(t + 1) = LB + \left(\frac{\text{rand}}{\text{max\_it}}\right)^{\text{rand}} \times (UB - LB) \\ &\quad \text{end} \end{aligned} \tag{P3}$$

where the control parameter  $d_{\text{max}}$  is a small value. Because it encourages the search intensity near the sea. While it is large values, it prevents more searches. Also,  $\mu$  is a range coefficient, which shows that the nearer region to the sea.

**Step 16:** Minimize  $d_{\text{max}}$  and check stopping criteria. if it is satisfied, the algorithm will be stopped, otherwise return to Step 10.

$$d_{\text{max}}(t + 1) = d_{\text{max}}(t) - \frac{d_{\text{max}}(t)}{\text{max\_it}}. \tag{3.15}$$

#### 4. NUMERICAL PROBLEMS

The applicability of the proposed hybrid algorithm in solving ratio functions [10] and application problem is validated in this section. Basically, in this section four types of problems are discussed. That is the Generalized ratio problem, Sum-of-Ratio problem, Multiplicative Ratio Problem, Large-scale ratio problems and an application of Production problem.

**Generalized Ratio Problems (GRPs):**

$$\text{N1: } \text{Max } Z = \frac{4x+2y+10}{x+2y+5} \quad \text{subject to } x + 3y \leq 30; -x + 2y \leq 5; x, y \geq 0.$$

$$\text{N2: } \text{Max } Z = \frac{8x+7y-2.33(9x^2+4y^2)^{0.5}}{20x+12y-2.33(3x^2+2xy+4y^2)^{0.5}}$$

$$\text{subject to } 2x + y \leq 18; x + 2y \leq 16; x, y \geq 0.$$

$$\text{N3: } \text{Min } Z = \frac{x+y+1}{2x-y+3} \quad \text{subject to } 0 \leq x \leq 1; 0 \leq y \leq 1.$$

**Sum-of-Ratio Problems:**

$$\text{N4: } \text{Max } Z = \frac{-x^2+3x-y^2+3y+3.5}{x+1} + \frac{y}{x^2-2x+y^2-8y+20}$$

$$\text{subject to } 2x + y \leq 6; 3x + y \leq 8; -x + y \geq -1; x, y \geq 0.$$

$$\text{N5: } \text{Max } Z = \frac{2x+y}{x} + \frac{2}{y}$$

$$\text{subject to } 2x + y \leq 6; 3x + y \leq 8; -x + y \geq -1; x, y \geq 0.$$

$$\text{N6: } \text{Max } Z = \frac{37x+73y+13}{13x+13y+13} + \frac{63x-18y+39}{13x+26y+13}$$

$$\text{subject to } 5x + 3y = 3; 1.5 \leq x \leq 3; x, y \geq 0.$$

$$\text{N7: } \text{Max } Z = \frac{-x^2y^{0.5}+2xy^{-1}-y^2+2.8x^{-1}y+7.5}{xy^{1.5}+1} + \frac{y+0.1}{-x^2y^{-1}-3x^{-1}+2xy^2+9y^{-1}+12}$$

$$\text{subject to } 2x^{-1} + xy \leq 4; x + 3x^{-1}y \leq 5; x^2 - 3y^3 \geq 2; 1 \leq x \leq 3; 1 \leq y \leq 3.$$

**Multiplicative Ratio Problems (MRP):**

$$\text{N8: } \text{Min } Z = \left(\frac{x+y+1}{x+y+2}\right)^{1.5} \times \left(\frac{x+y+2}{x+y+3}\right)^{2.1}$$

$$\text{subject to } x^{1.1}y^{1.6} - x^{1.2}y^{1.5} \leq 5; 1 \leq x, y \leq 2.$$

$$\text{N9: } \text{Max } Z = \left(\frac{3x+y-2z+0.8}{2x-y+z}\right) \times \left(\frac{4x-2y+z}{7x+3y-z}\right)$$

$$\text{subject to } x + y - z \leq 1; -x + y - z \leq -1; 12 + 5y + 12z \leq 34.8$$

$$12x + 12y + 7z \leq 29.1; -6x + y + z \leq -4.1; x, y, z \geq 0.$$

$$\text{N10: } \text{Min } Z = \left(\frac{-x+2y+2}{3x-4y+5}\right)^{0.5} \times \left(\frac{4x-3y+4}{-2x+y+3}\right)^{0.2}$$

$$\text{subject to } x + y - z \leq 1; x + y = 1.5; x \leq y; 0 \leq x, y \leq 1.$$

**5. RESULTS AND DISCUSSION**

This section has been discussed about the results of numerical problem, the performance comparison between proposed with other existing algorithm, convergence rate achieved by Cuckoo Search Algorithm, ER-WCA, Firefly Algorithm, Grey Wolf Optimization (GWO), Water Cycle Algorithm (WCA), Neural Network (NN) and the proposed hybrid Genetic ER-WCA algorithm for numerical test problems. The numerical test problems are solved using the proposed algorithm which has been executed in MATLAB R2015. Table 1 shows that the parameters, which are used in the proposed algorithm.

The generalized ratio problems are solved using the proposed Genetic ER-WCA and the Obtained results are compared with the existing algorithms as shown in the below Table 2. From Table 2, the proposed Genetic ER-WCA gives a better solution is proved. From problem N1, the proposed algorithm proved that gives more accuracy and the problem N2, the Genetic ER-WCA is better than other algorithms. Similarly, the problem N3, also produce a global solution. Hence, the results of the Generalized Ratio Problems are given more accuracy while using the proposed algorithm.



TABLE 1. Parameters of the Genetic ER-WCA algorithm.

Genetic ER-WCA parameters	Values
Pop size	100
Initial pop	100 × 100
max <sub>it</sub>	100
C <sub>per</sub>	0.7
M <sub>per</sub>	0.3
X <sub>rate</sub>	0.4
Mut <sub>rate</sub>	0.1
d <sub>max</sub>	1E-5
N <sub>sr</sub>	4

TABLE 2. Performance comparison of the proposed algorithm and other existing algorithms for the Generalized ratio problems.

Problem no	N1(Max)	N2(Max)	N3(Min)
GWA	3.71428	0.33604	0.3333
(X1; X2)	(29.9998,0)	(1,1)	(0,0)
WCA	3.71428	0.33603	0.3333
(X1; X2)	(29.9999,0)	(1,1.0000303)	(0,0)
NN	3.7142	0.33603	0.3333
(X1; X2)	(29.9999,0)	(1,1.0000303)	(0,0)
ER-WCA	3.71428	0.33606	0.3333
(X1; X2)	(29.999,0)	(1,1.000061)	(0,0)
CUCKOO	3.7141	0.33605	0.3333
(X1; X2)	(29.983,0)	(1,1.0001)	(0,0)
FF	3.7142	0.3759	0.3333
(X1; X2)	(29.995311,0)	(1.2763,1.27707)	(0,0)
Genetic ER-WCA	3.7148	0.3383	0.3333
(X1; X2)	(29.99,0.0026)	(0.1008,0.5645)	(0,0)

Similarly, from Figure 1 we can analyze the proposed algorithm starts with the best solutions. Because ER-WCA’s initial population is the solution of the genetic algorithm. Figure 1 is presenting the graphical view of the convergence of all algorithms on ratio problems in which it can be clearly seen that the GA-WCA algorithm is nearest to the global optima of generalized ratio problem among algorithms viz. GWA, WCA, NN, ER-WCA, Cuckoo and FF and it also shows the fastest convergence of all. In Figure 1, NN and FF illustrate poorer convergence than the other algorithms in the initial iterations.

Figure 2, Cuckoo Algorithm, and NN depict poorer convergence than the other algorithms in the initial iterations. However, the search process is progressively accelerated during iterations for these algorithms. This shows that the performance of ER-WCA can be boosted by the Genetic algorithm in terms of not only exploration but also exploitation. Hence the proposed algorithm is always performed with the best solutions and also must reach global solutions.

The sum-of ratio problem results are compared with the existing algorithms as shown in the below Table 3. From Table 3, the proposed Genetic ER-WCA gives a better solution is proved. That is, from problems N4 and N5, the proposed algorithm proved that gives more accuracy and the problem N6 and N7, the Genetic ER-WCA is better than other algorithms and also produce global solutions. Hence, the results of the Sum of Ratio Problems are given more accuracy while using the proposed algorithm.

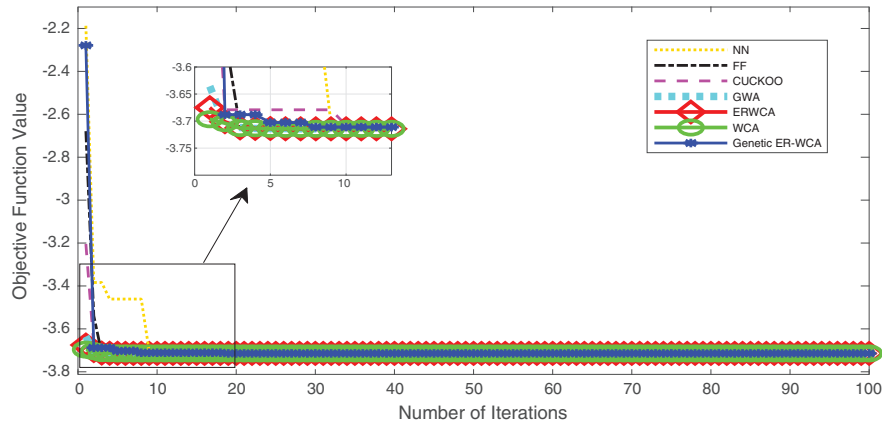


FIGURE 1. Convergence curve of the proposed algorithm for problem N1.

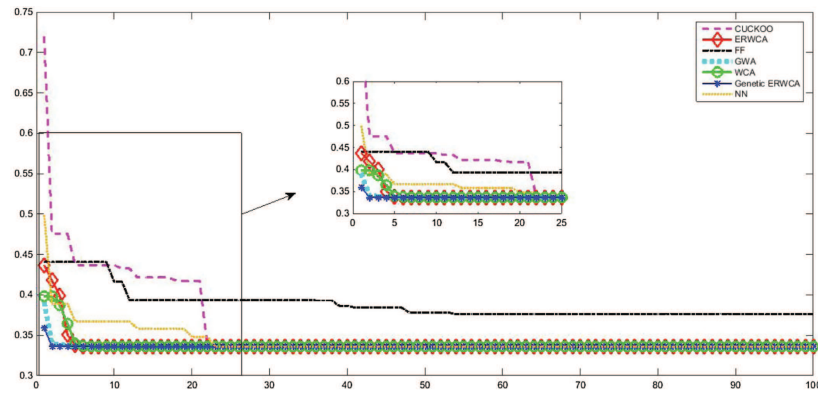


FIGURE 2. Convergence curve of the proposed algorithm for problem N2.

TABLE 3. Performance comparison of the proposed algorithm and other existing algorithms for Sum of Ratio Problems.

Problem no	N4(Max)	N5(Max)	N6(Max)	N7(Max)
GWA	5.3992	6.49999	4.0608	
(X1; X2)	(0,0.8576)	(1,3.9999)	(1,1.7438)	5.2079 (1,1)
WCA	5.5181	6.49999	4.06081916	
(X1; X2)	( $3.5e^{-04}$ )	(1,3.9999)	(1,1.74382)	5.20789 (1,1)
NN	5.5584	6.499993	4.060819	
(X1; X2)	(0,0.9836)	(1,3.999927)	(1,1.743815)	5.20789 (1,1)
ER-WCA	5.3417	6.49999	4.060819	
(X1; X2)	(0,1)	(1,3.9999)	(1,1.74382)	5.20789 (1,1)
CUCKOO	5.5753	6.9999	4.0608	
(X1; X2)	(0.00067,1)	(1,3.9999)	(1,1.7438)	5.2079 (1,1)
FF	5.3668	6.49856	4.0608191	
(X1; X2)	(0.03835,0.89998)	(1,3.99835)	(1,1.74379)	5.20789 (1,1)
Genetic ER-WCA	5.5769	6.6530	4.07606568	
(X1; X2)	(0,1)	(1,4.3652)	(0.03615,1.69547)	5.20789 (1,1)

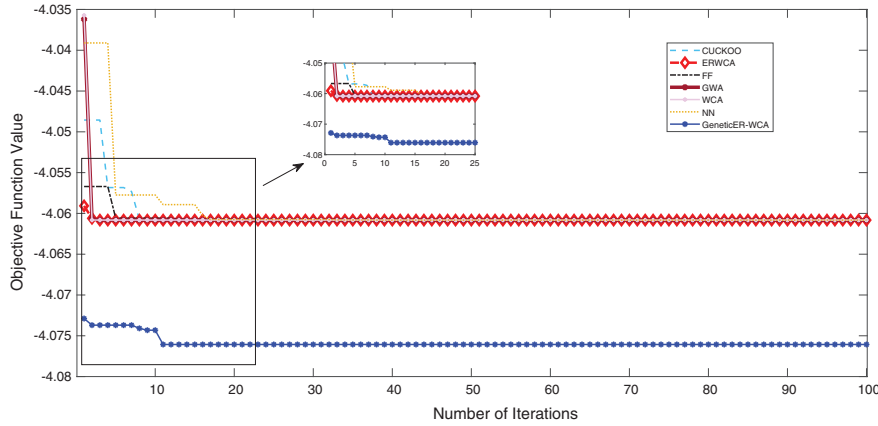


FIGURE 3. Convergence curve of the proposed algorithm for problem N6.

TABLE 4. Performance comparison of the proposed algorithm and other existing algorithms for Multiplicative ratio problems.

Problem no	N8	N9(Max)	N10
GWA	0.40652	1.08514	0.66991
(X1; X2)	(1,1)	(0.9994,0.00032,0.0009408)	(0,0)
WCA	0.4074526	1.0857137	0.7782046
(X1; X2)	(1,1.01)	(0.999998,0,0.0000011)	(0.2999998,0.3)
NN	0.41573394	1.08560537	0.66991183
(X1; X2)	(1,1.1)	(0.99992958,0,0.000181759)	(0.0000009999,0)
ER-WCA	0.496676	1.0857125710	0.8561892
(X1; X2)	(1,1)	(0.999996,0,0.000003734)	(0.5,0.500000067)
CUCKOO	0.42024205	1.0840441122	0.74105822
(X1; X2)	(1,1.15)	(0.99886,0,0.002807)	(0.1999977,0.2)
FF	0.416187695	1.085271258	0.85621654
(X1; X2)	(1,1.1050001)	(0.99918,0,0.0009156)	(0.5,0.50004665)
Genetic ER-WCA	0.406519017	1.086128	0.5358867
(X1; X2)	(1,1)	(0.799457,0.224592,0.224592)	(0.3750,0.6250)

Similarly, from Figure 3 we can analyze the proposed algorithm starts with the best solutions. Since ER-WCAs initial population is the solution of the genetic algorithm the Genetic ER-WCA is performed well. Similarly, Figure 3, WCA Algorithm and GWA exhibit poorer convergence than the other algorithms in the initial iterations. Genetic ER-WCA and ER-WCA algorithms are given the better and best convergence respectively. Hence the proposed algorithm is always performed with the best solutions and also must reach global solutions.

The Multiplicative Ratio problems are solved with the proposed algorithm Genetic ER-WCA and the evaluated results are compared with the existing algorithms as shown in the below Table 4. From Table 4, the proposed Genetic ER-WCA is proved to getting a better solution. That is, from problems N8 and N9, the proposed algorithm gives more accuracy and the problem N10, the Genetic ER-WCA is better than other algorithms and also produce global solutions. Hence, the results of the Multiplicative Ratio Problems are given more accuracy while using the proposed algorithm.

Table 5, shows statistical optimization results for the Generalized Ratio Problems, Sum-of ratio problems, Multiplicative ratio problems using the Genetic ER-WCA and other algorithms. Table 5, displays the worst,

TABLE 5. Comparative analysis of the statistical values of the Genetic ER-WCA algorithm with some existing algorithms for generalized ratio problems.

	Methods	Best	Average	Worst	SD
N1 (Max)	Genetic ER-WCA	3.71428	3.71428	3.69779	0.001843
	GWO	3.714285	3.7119579	3.486810	0.022744
	ERWCA	3.714285	3.71385959	3.6757663	0.003853
	NN	3.714285	3.632658	2.173466	0.3258489
	WCA	3.7142854	3.71328815	3.6190408	0.009542
	CUCKOO	3.714185	3.641729	2	0.320982
	FF	3.7142288	3.7099266	3.3704698	0.0345039
N2 (Min)	Genetic ER-WCA	0.3360370	0.3369841	0.3810072	0.0061905
	GWO	0.336040	0.3370357	0.415519	0.00810462
	ERWCA	0.336067	0.33823731	0.495007	0.0163466
	NN	0.3360376	0.3388646	0.445139	0.014004667
	WCA	0.336037	0.3373923	0.4654116	0.01293436
	CUCKOO	0.3360585	0.3316366	0.35213134	0.0066144
	FF	0.375984	0.33477380	0.5706844	0.03134298
N3 (Max)	Genetic ER-WCA	0.3333	0.3333	0.3333	3.905356 e-16
	GWO	0.3333	0.333462	0.346202	0.00128695
	ERWCA	0.3333	0.3333	0.3343	3.906856 e-15
	NN	0.3333	0.333747	0.336131	0.00188918
	WCA	0.3333	0.3333	0.33450	3.905855 e-15
	CUCKOO	0.3333	0.3336357	0.35850	0.00258335
	FF	0.3333	0.334096	0.36394	0.003800379
N4 (Max)	Genetic ER-WCA	5.5769	5.556542	5.4954382	6.248570e-15
	GWO	5.3992	5.398145	5.300458	0.00986746
	ERWCA	5.3417	5.245820	5.1908321	0.0591139
	NN	5.5584	5.546395	5.2716373	0.0474975
	WCA	5.5181	5.518062	5.508252	1.33897 e-04
	CUCKOO	5.5753	5.574849	5.5344946	0.0040762
	FF	5.3668	5.3631110	4.999637	0.0367144

average, better solutions and standard deviation (SD) of numerical functions using various methods. Further, by observing Table 5, the genetic ER-WCA can identify the best Standard Deviation by comparing various methods. That is, the solutions to the problems are nearer to the mean value of the solutions. So the Genetic ER-WCA starts from near to the global solution. Hence, Genetic ER-WCA is more effective than other algorithms. Solving large scale problems is the very complexity of the evolutionary algorithms. Not only that, these large scale problems are used to finding the effectiveness of the algorithms. Because these problems are multidimensional, solving this using an evolutionary method takes much time and does not achieve global optimum.

### 5.1. Large scale ratio problem

The evolutionary algorithms are easy to optimize the small-scale problems, but the large-scale optimization problems are more challenging to optimize. The large-scale optimization problems are very difficult to get the global optimum, since it may have a large number of local minimum, and each of them has non-isolated and singular. It is represented by a huge number of decision variables with constraints. In this section, the computational results of large scale problems are discussed to demonstrate the effectiveness of the proposed algorithm.

TABLE 5. continued.

	Methods	Best	Average	Worst	SD
N5 (Max)	Genetic ER-WCA	6.49999	6.48010784	5.36529372	0.125113539
	GWO	6.49999	6.476003181	4.2967931	0.220445276
	ERWCA	6.49999	6.47311019	5	0.170083473
	NN	6.499943	6.1929878119	4.26007576	0.62531494
	WCA	6.49999	6.47251095	5	0.19177515
	CUCKOO	6.4486093	6.134551866	4.09452778	0.60582439
	FF	6.4985547	6.4510421138	4.730529045	0.21499959
N6 (Max)	Genetic ER-WCA	4.0760656	4.0758370	4.072884325	7.004161071e-4
	GWO	4.06081916	4.0607368080	4.05258417	8.234982e-4
	ERWCA	4.0608191608	4.0607373581	4.0526484	8.170666971e-4
	NN	4.06081916	4.054757311	3.95904349	0.020443091
	WCA	4.06081916	4.060751670	4.054106	7.71258377e-4
	CUCKOO	4.0608191608	4.059986759	4.01438495	0.005856244
	FF	4.0608191571	4.058598021	3.96279830	0.01255398
N7 (Max)	Genetic ER-WCA	5.207894736	5.207894836	5.207894736	8.926528984e-15
	GWO	5.20789716924	5.199771692	4.395590294	0.081230444
	ERWCA	5.207894736	5.207894736	5.207894736	8.926528984e-15
	NN	5.2078947368	5.186291057	4.376662316	0.11171227745
	WCA	5.2078947368	5.207894736	5.207894736	8.926528984e-15
	CUCKOO	5.207994736	5.113827001	3.1919413716	0.373931050
	FF	5.20789473	5.191854723	4.082753199	0.1137534548
N8 (Min)	Genetic ER-WCA	0.4065190179	0.465196	0.4065191178	3.9053564e-16
	GWO	0.40652	0.40653042	0.4065190179	1.146278508e-4
	ERWCA	0.4966769833	0.49667613	0.496680983	6.1369886e-16
	NN	0.415733943	0.4162651728	0.429372790	0.00261658085
	WCA	0.4074526401	0.4074526461	0.40745264617	6.69489673e-16
	CUCKOO	0.42024205133	0.4204105328	0.4322666297	0.001249486
	FF	0.4161876954	0.4164470865	0.4284863406	0.001557657
N9 (Max)	Genetic ER-WCA	1.08612875	1.08473460	1.00781569	0.005212032
	GWO	1.085146108	1.0832542437	1.03535061	0.0068537959
	ERWCA	1.0046821245	1.083738918	1.085712571	0.01086988575
	NN	1.0522981125	1.0750492592	1.0856053717	0.01317977763
	WCA	1.020821231914	1.0836486027	1.0857137520	0.009411251939
	CUCKOO	1.0180040112	1.06947202814	1.0840441122	0.021207428
	FF	0.9665088507	1.076655573345	1.0852412585	0.0233152913
N10 (Min)	Genetic ER-WCA	0.53588673	0.53669820	0.561848926	0.0031903817
	GWO	0.7005291586	0.6702181949	0.669911977933	0.003261718
	ERWCA	0.856189206	0.8564891467	0.881299307	0.0025412460
	NN	0.6699119980	0.67370654062	0.78201423624	0.0196236431
	WCA	0.7782046328	0.77833229810	0.787300507742	9.528786318e-4
	CUCKOO	0.741058220	0.7444056809	0.78966544049	0.009721399377
	FF	0.85621654084	0.86369442959	0.897452747	0.0198955321037

Problem N11, N12 and N13, N14, we consider the four types of randomly generated ratio problems, which are Sum-of-Linear Ratio Problem (SLRP), Sum-of-Nonlinear Ratio Problem (SNRP), Multiplicative Linear Ratio Problem (MLRP) and Multiplicative Nonlinear Ratio Problem (MNRP).

The elements of random developed matrix are  $a_{ij}, c_{qj}, d_{qj}, \alpha_q, \beta_q, b_i \in R$ . The co-efficient  $a_{ij}, c_{qj}, d_{qj}$  of the problem are yield at interval  $[0, 1]$ . The constant co-efficient of numerator and denominator is the same random number which is provoked at  $[1, 100]$ . Also, the elements of  $b_i$  are fixed for all is 1. The randomly induced

TABLE 6. considered various types of large scale problems.

Case	$(r, c, n)$	Type of problem
1	(5,30,30)	Non-linear
2	(10,30,30)	Non-linear
3	(5,50,50)	Linear
4	(5,100,100)	Linear
5	(10,50,50)	Linear
6	(10,100,100)	Linear

elements are fixed for all cases. The elements of the randomly generated problems will be randomly generated by using MATLAB R2015. The number of ratios, number of constraints and number of variables can be denoted by  $r, c, n$  respectively. Hence for given  $r, c, n$ , we test 6 instances, then discuss the objective values about the randomly developed instance. For sum-of linear ratio problem and Multiplicative linear ratio problem, the exponent  $\gamma$  value is 1 and for sum-of nonlinear ratio problem and Multiplicative nonlinear ratio problem, the exponent of  $\gamma$  is fixed randomly at 0.3. From the Table 6, the considered different type of randomly generated large scale problems are shown below.

**Large-Scale Linear Ratio Problem (LS-SLRPs):**

$$\text{N11: } \quad \text{Min } Z = \sum_{q=1}^r \left( \frac{\sum_{j=1}^n c_{qj} x_j + \alpha_i}{\sum_{j=1}^n d_{qj} x_j + \beta_i} \right) \quad \text{Subject to } \quad \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad x > 0.$$

$$\text{N12: } \quad \text{Min } Z = \prod_{q=1}^r \left( \frac{\sum_{j=1}^n c_{qj} x_j + \alpha_i}{\sum_{j=1}^n d_{qj} x_j + \beta_i} \right) \quad \text{Subject to } \quad \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad x > 0.$$

**Large-Scale Non-Linear Ratio Problem (LS-SLRPs):**

$$\text{N13: } \quad \text{Min } Z = \sum_{q=1}^r \left( \frac{\sum_{j=1}^n c_{qj} x_j + \alpha_i}{\sum_{j=1}^n d_{qj} x_j + \beta_i} \right)^{0.3} \quad \text{Subject to } \quad \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad x > 0.$$

$$\text{N14: } \quad \text{Min } Z = \prod_{q=1}^r \left( \frac{\sum_{j=1}^n c_{qj} x_j + \alpha_i}{\sum_{j=1}^n d_{qj} x_j + \beta_i} \right)^{0.3} \quad \text{Subject to } \quad \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad x > 0.$$

Example N11 and N12 is solved into four different cases. They are as follows: Case 1 has problem 5 ratios, 50 variables under 50 constraints, and Case 2 problem has 5 ratios, with 100 variables under 100 constraints. Similarly, Case 3 has problem 10 ratios, 50 variables under 50 constraints, and Case 4 problem has 10 ratios, with 100 variables under 100 constraints. The computational statistical results for four cases of Example N11 are shown in Table 7. From Table 7, it is clear that the results are proclaimed by the proposed hybrid algorithm in case 1 and 4 violate other existing algorithms.

Similar to N11, N12 is resolved in the same four distinctive cases. Such as, 5 and 10 multiplicative ratios are solved with 50 and 100 variables subject to 50 and 100 constraints, respectively. The computational statistical results for four cases of Example N12 are shown in Table 8. From Table 8, according to the obtained statistical results of four cases, the proposed method performs better than other algorithms. In cases 1 and 2, ER-WCA and GWO algorithms are given the worst performance by comparing other algorithms, respectively. The WCA algorithm in the 3rd case and the GWO algorithm in the 4th case perform very poorly by comparing the other methods.

Also different type of randomly developed large scale Sum and Multiplicative Non-linear problems are considered. Therefore the exponent of general Non-linear ratio problems is fixed at 0.3. The randomly procreated Non-linear ratio problems are solved into two different cases. They are as follows: Case 1 has 5 ratios, 30 variables with subject to 30 constraints, and Case 2 has 10 ratios, with 30 variables with subject to 30 constraints.

TABLE 7. Computational statistical results for Large Scale Sum-of-Linear Ratio Problem (LS-SLRP).

Min N11 ( $r, c, n$ )	Algorithms	Best	Avg.	Worst	SD
Case 1-(5,50,50)	Genetic ERWCA	4.75949045	4.79132373	4.82285732	0.008204056
	GWO	4.914125892	4.806985208	4.977942433	0.016954249
	ERWCA	4.950003507	4.983158883	4.994803854	0.005111145
	NN	4.949466454	4.983977246	4.999486756	0.008357551
	WCA	4.964289485	4.8649389	5.015999901	0.013095756
	CUCKOO	4.992858335	4.996638874	4.99696514	0.008812481
	FA	4.891677401	4.904157552	4.967529799	0.017304634
Case 2-(5,100,100)	Genetic ERWCA	4.775115116	4.82321435	4.88403335	0.025180731
	GWO	4.969193365	4.70901309	5.678448591	0.049817923
	ERWCA	4.98986075	4.982581923	5.02789447	0.025433168
	NN	4.979950473	5.007207815	5.059693696	0.020885091
	WCA	4.984169837	5.01209352	5.055795009	0.016134886
	CUCKOO	4.9845881	4.986271028	4.999373165	0.026141826
	FA	4.979632703	5.002426179	5.133157796	0.03333708
Case 3-(10,50,50)	Genetic ERWCA	9.790821583	9.8540596	9.94807745	0.0146544758
	GWO	9.937053791	9.639267133	9.625860204	0.04183607
	ERWCA	9.954456313	9.97065902	10.0949875	0.020573737
	NN	9.936541178	9.992379038	10.06374005	0.036966727
	WCA	9.96468867	10.02747273	10.03807178	0.049706742
	CUCKOO	9.991954072	9.99653549	10.00975337	0.004785722
	FA	9.967757325	9.798714253	9.937579442	0.020491913
Case 4-(10,100,100)	Genetic ERWCA	9.04734680	9.12501414	9.223736063	0.03169365
	GWO	9.398950816	9.363710276	9.807190527	0.086210787
	ERWCA	9.920335072	9.978707447	10.03689899	0.033999055
	NN	9.843224343	9.989837198	10.07267487	0.036686917
	WCA	9.870204556	9.969282453	10.0201175	0.053396825
	CUCKOO	9.752531558	9.761688065	9.82476604	0.04266981
	FA	9.578878801	9.618758218	9.858935513	0.058449337

The computational statistical results for two cases of Example N13 are shown in Table 9. From Table 9, according to the obtained comprehensive statistical results of two cases, the proposed method performs better than other algorithms. In both cases, the NN algorithm is give very worst performance by comparing the other methods.

Similar to N13, N14 is resolved in the same two cases. Such as, 5 and 10 multiplicative ratios are solved with 30 variables subject to 30 constraints, respectively. The computational statistical results for two cases of Example N14 are shown in Table 10. Similar to the N13, the proposed algorithm is well performed by comparing the statistical results of others algorithm, while solving the N14. The cuckoo search algorithm in the 1st case and the NN algorithm in 2nd case perform very poorly by comparing the other methods.

### 6. APPLICATION PROBLEM

A management has discontinued production of a certain unprofitable product line and this has created considerable excess production capacity. Management is considering to devote this excess capacity to produce one or more of three products 1, 2 and 3 in their 4 branches. The available excess capacity on the machines,

TABLE 8. Computational statistical results for Large Scale Multiplicative-of-Linear Ratio Problem (LS-MLRP).

Min N12 ( $r, c, n$ )	Algorithms	Best	Avg.	Worst	SD
Case 1-(5,50,50)	Genetic ERWCA	0.7168701864	0.73652151	0.77047998	0.01155075
	GWO	0.814986654	0.822342196	0.924444331	0.01660128
	ERWCA	0.94591658	0.977893403	1.002735944	0.069732004
	NN	0.948870514	0.988548001	1.014281084	0.012086218
	WCA	0.955903989	1.012036036	1.03797346	0.016299212
	CUCKOO	0.977091205	1.01089893	1.033514422	0.01279304
	FA	0.960780292	0.971867324	1.028260568	0.015446073
Case 2-(5,100,100)	Genetic ERWCA	0.84064856	0.88745025	0.943283114	0.008327130
	GWO	0.928927155	0.933362434	1.167029629	0.042658614
	ERWCA	1.014496343	1.048532766	1.114715792	0.021433234
	NN	0.987348657	1.027604985	1.146678683	0.042205023
	WCA	0.943305026	0.980017161	1.049613142	0.018960256
	CUCKOO	0.985239721	0.985835634	0.990286213	0.009060665
	FA	0.928880601	0.943647902	1.077474865	0.036241651
Case 3-(10,50,50)	Genetic ERWCA	0.809436225	0.86351842	0.947479700	0.01420033
	GWO	0.97951223	1.00734198	1.131869898	0.040513646
	ERWCA	0.934166538	0.958885183	1.061864952	0.024044671
	NN	0.936818293	0.994314307	1.116100448	0.043418737
	WCA	0.924829928	0.9973134	1.186208934	0.06367028
	CUCKOO	0.993758176	0.995543998	1.006084087	0.021649532
	FA	0.988466502	1.00931327	1.132126453	0.032220389
Case 4-(10,100,100)	Genetic ERWCA	0.711088990	0.767999301	0.825544737	0.013501334
	GWO	0.940828674	0.969655431	1.170123488	0.050662549
	ERWCA	0.98211964	1.083322633	1.129462571	0.030744941
	NN	0.91287193	0.980629076	1.085780066	0.035347594
	WCA	0.943736596	1.019844176	1.075485759	0.021976432
	CUCKOO	0.968795716	0.97545524	1.004353917	0.03889192
	FA	0.980587436	1.009398569	1.170565054	0.043955399

TABLE 9. Computational statistical results for Large Scale Sum-of-Non-Linear Ratio Problem (LS-SNLRP).

Min N13 ( $r, c, n$ )	Algorithms	Best	Avg.	Worst	SD
Case 1-(5,30,30)	Genetic ERWCA	4.61844553	4.62784041818	4.63809480	0.000647993
	GWO	4.985218753	4.976470201	5.015193101	0.004010578
	ERWCA	4.980991744	4.991948574	5.003329381	0.002749425
	NN	4.991819141	4.996841183	5.008596379	0.005441922
	WCA	4.990707204	4.992691023	5.008844902	0.003733282
	CUCKOO	4.997609722	4.998193903	4.999971256	0.002891612
	FA	4.96601759	4.969506707	4.989779008	0.005082032
Case 2-(10,30,30)	Genetic ERWCA	122.1021036	123.2354031	124.5571823	0.801257881
	GWO	137.7638095	137.9840018	141.4854711	0.967172908
	ERWCA	147.3466601	172.5538975	180.0787519	6.451363023
	NN	146.1484807	172.9422209	186.1285575	11.63059348
	WCA	145.1047471	175.6778528	195.1121517	7.077757174
	CUCKOO	136.6981421	137.8161121	140.8049775	0.855013089
	FA	136.6288281	137.0692078	140.2831956	0.70902999



TABLE 10. Computational statistical results for Multiplicative -of- Non-Linear Ratio Problem (MNLRP).

Min N14 (r,c,n)	Algorithms	Best	Avg.	Worst	SD
Case 1-(5,30,30)	Genetic ERWCA	0.928173518	0.936480405	0.947578667	0.005188756
	GWO	0.95542134	0.956540527	0.98336519	0.003813135
	ERWCA	0.990478271	0.99830095	1.013581952	0.005426799
	NN	0.992309519	0.99753644	1.007913492	0.004613779
	WCA	0.992101654	0.994411821	1.009126581	0.003320177
	CUCKOO	0.970084435	0.975703215	0.994572899	0.007582201
	FA	0.96560466	0.970015961	0.989367967	0.006200266
Case 2-(10,30,30)	Genetic ERWCA	123.75867287	125.2460363	127.8554381	1.048661682
	GWO	141.0822458	160.1024172	171.8114009	2.606540748
	ERWCA	137.6633521	162.5440464	169.5129236	5.114714415
	NN	141.6937485	161.8923548	175.5594349	9.425663766
	WCA	139.5578837	163.0024834	171.6135463	5.90074494
	CUCKOO	145.168135	146.425487	149.2409333	2.507357165
	FA	146.6032319	147.0686147	150.6370088	1.73732626

which might limit output is summarized in the following table:

Machine type	Available excess capacity (in machine hour per week)
M1	10
M2	10
M3	10

The number of machine-hours require for each unit of the respective product is given below.

Machine type	Capacity requirement in machine hours per unit		
	P1	P2	P3
M1	2	1	5
M2	1	6	3
M3	5	9	2

Also the management has a total man power 10 persons. In this 9 persons, 7 persons and 3 persons are used to produce the products respectively. The per unit contribution of each branches is as follows:

Branches	Per unit contribution in branches (in cost)		
	P1	P2	P3
A	4	3	3
B	3	0	4
C	1	2	5
D	1	5	4

Moreover the total time needed to produce each product in each branches is as follows:

Branches	Per unit contribution in branches (in cost)		
	P1	P2	P3
A	0	3	3
B	4	4	5
C	1	5	5
D	0	5	4

The management also has a fixed establishment cost and time of Rs. 50/- and 50 h for each branches respectively. The management wants to maximize the sum of all branches profit on time. This leads to a sum of linear fractional programming problem and the formulated sum of linear fractional programming problem is as follows:

$$\begin{aligned}
 \text{Max } Z &= \frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} + \frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50} + \frac{x_1+8x_2+5x_3+50}{x_1+5x_2+5x_3+50} + \frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50} \\
 &\text{subject to } 2x_1 + x_2 + 5x_3 \leq 10 \\
 &\quad x_1 + 6x_2 + 3x_3 \leq 10 \\
 &\quad 5x_1 + 9x_2 + 2x_3 \leq 10 \\
 &\quad 9x_1 + 7x_2 + 3x_3 \leq 10, \quad x_1, x_2, x_3 \geq 0.
 \end{aligned} \tag{6.1}$$

The application Ratio problem is solved with the proposed algorithm of Genetic ER-WCA. The obtained results are compared with the other existing algorithms as shown in Table 11. From Table 11, the proposed Genetic ER-WCA is proved to getting a better solution. That is, from the application problem, the proposed algorithm gives more accuracy and also produce global solutions. Hence, the results of the application Problem gives more accuracy and better solutions while using the proposed algorithm.

Similarly, from Figure 4 we can observe that proposed algorithm starts with the best solutions. Since ER-WCAs initial population is the solution of the genetic algorithm the Genetic ER-WCA is performed well. Figure 4, NN depicts poorer convergence than the other algorithms in the initial iterations. The firefly algorithm represents the moderate convergence than other methods in the initial iteration. Cuckoo and Genetic ER-WCA

TABLE 11. Performance comparison of the proposed algorithm and other existing algorithms of the application problem.

Methods	Solutions
GWA	4.080956
(X1; X2)	(1.111111110934,0,0)
WCA	4.090702945
(X1; X2)	(1.111111076979,0,0)
NN	4.0907029476
(X1; X2)	(1.11111064993,0,0)
ER-WCA	4.090702947
(X1; X2)	(1.11111111066156,0,0)
CUCKOO	4.0815
(X1; X2)	(1,0,0)
FF	4.0815
(X1; X2)	(1,0,0)
Genetic ER-WCA	4.1308113
(X1; X2)	(0.3823,0,0)

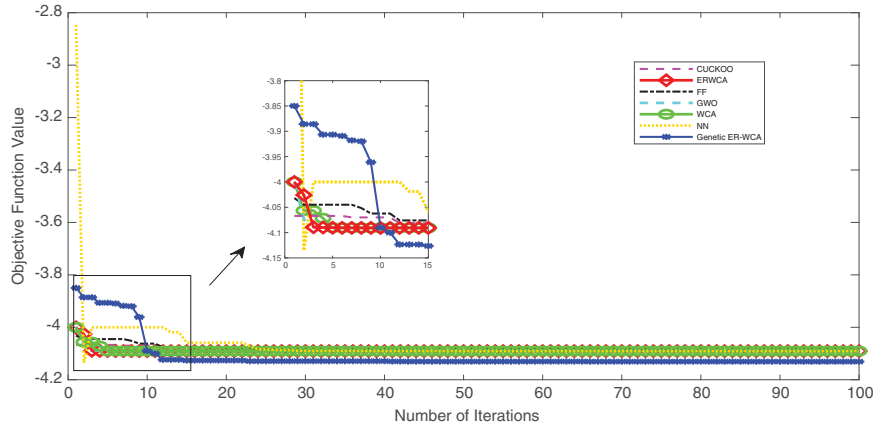


FIGURE 4. Convergence curve of the proposed algorithm for application problem.

TABLE 12. Comparative analysis of the statistical values of the Genetic ER-WCA algorithm with some existing algorithms for application.

	Methods	Best	Average	Worst	SD
Application (Max)	Genetic ER-WCA	4.1308113	4.10874596	3.849818061	0.06526036
	GWO	4.0907029471	4.089562071	4	0.0092115895
	ERWCA	4.090702947	4.08883622166377	4	0.0127563326
	NN	4.0358355	4.065067647	2.846182477	0.12658785703
	WCA	4.0907029455	4.08891477	4	0.0103463557
	CUCKOO	4.08148148148	4.0807535850	4.063365871	0.0074230238
	FA	4.08148148148	4.0790190182838	4.03191517394	0.00695109838

algorithms are given the better and best convergence respectively. Hence the proposed algorithm is always performed with the best solutions and also must reach the global solutions.

Table 12, shows statistical optimization results for the Application ratio problem using the Genetic ER-WCA and different methods. Table 12, displays the worst, average, better solutions and standard deviation (SD) of problem using various methods. Further, by observing Table 12, the genetic ER-WCA can identify the best Standard Deviation by comparing different methods. It conveys, the solutions of the problems are close to the mean value of the solutions. So the Genetic ER-WCA starts from near to the global solution. Hence, the Genetic ER-WCA is more effective than other algorithm.

### 7. CONCLUSION

This paper proposes a hybrid algorithm, by using GA and ER-WCA for solving the generalized linear and nonlinear ratio problems. In real life, solving the generalized ratio of linear and nonlinear problems are most complex. Because of this complexity, the highly nonlinear problems are solved using meta-heuristic evolutionary algorithms, but which are not guaranteed to reach the global convergence. So we proposed the hybrid algorithm using GA and ER-WCA algorithms to reach the appropriate global optimal solution and which is compared to an exhaustive search algorithms. In the proposed algorithm, the worst raindrops in ER-WCA are replaced with the best chromosomes in GA. These fine chromosomes have been Inspected using a boundary mechanism and then the process of ER-WCA is implemented. After running an iteration, the poorer chromosomes of GA are replaced with the better raindrops of ER-WCA. The communication strategy of proposed algorithm provides

the information flow for the raindrops to communicate in WCA with the chromosomes in GA. The performance of Genetic ER-WCA algorithm is better than other algorithms in terms of convergence and accuracy. Finally, proved the global convergence of the proposed algorithm using several numerical example and also verify the superiority of the proposed hybrid algorithm.

## REFERENCES

- [1] H.P. Benson, Concave minimization: theory, applications and algorithms. In: Handbook of Global Optimization. Springer, Boston, MA (1995) 43–148.
- [2] W. Chun-Feng, L. San-Yang and S. Pei-Ping, Global optimization for sum of geometric fractional functions. *Appl. Math. Comput.* **216** (2010) 2263–2270.
- [3] G. Jamali, S.S. Sana and R. Moghdani, Hybrid improved cuckoo search algorithm and genetic algorithm for solving markov-modulated demand. *RAIRO: OR* **52** (2018) 473–497.
- [4] H.W. Jiao, S.Y. Liu and Y.F. Zhao, Effective algorithm for solving the generalized linear multiplicative problem with generalized polynomial constraints. *Appl. Math. Model.* **39** (2014) 7568–7582.
- [5] H. Jiao, S. Liu, J. Yin and Y. Zhao, Outcome space range reduction method for global optimization of sum of affine ratios problem. *Open Math.* **14** (2016) 736–746.
- [6] Y.-C. Jong, An Efficient Global Optimization Algorithm for Nonlinear Sum-of-Ratios Problem (2012) 1–21.
- [7] X. Liu, Y.L. Gao, B. Zhang and F.P. Tian, A new global optimization algorithm for a class of linear fractional programming. *Mathematics* **7** (2019) 867.
- [8] A. Pizzo, A. Zappone and L. Sanguinetti, Solving fractional polynomial problems by polynomial optimization theory. *IEEE Signal Process. Lett.* **25** (2018) 1540–1544.
- [9] N.T.H. Phuong and H. Tuy, A unified monotonic approach to generalized linear fractional programming. *J. Glob. Optim.* **26** (2003) 229–259.
- [10] O.A. Raouf and I. M. Hezam, Solving fractional programming problems based on swarm intelligence. *J. Ind. Eng. Int.* **10** (2014) 56.
- [11] P.-P. Shen and T. Lu, Regional division and reduction algorithm for minimizing the sum of linear fractional functions. *J. Inequal. Appl.* **2018** (2018) 63.
- [12] P.P. Shen and C.F. Wang, Global optimization for sum of generalized fractional functions. *J. Comput. Appl. Math.* **214** (2008) 1–12.
- [13] P. Shen, B. Huang and L. Wang, Range division and linearization algorithm for a class of linear ratios optimization problems. *J. Comput. Appl. Math.* **350** (2019) 324–342.
- [14] P. Shen, Z. Zhu and X. Chen, A practicable contraction approach for the sum of the generalized polynomial ratios problem. *Eur. J. Oper. Res.* **278** (2019) 36–48.
- [15] C. Veeramani and M. Sumathi, Fuzzy mathematical programming approach for solving fuzzy linear fractional programming problem. *RAIRO: OR* **48** (2014) 109–122.