

## Research Article

# Solving Interval Quadratic Programming Problems by Using the Numerical Method and Swarm Algorithms

M. A. Elsisy <sup>1</sup>, D. A. Hammad,<sup>1</sup> and M. A. El-Shorbagy <sup>2,3</sup>

<sup>1</sup>Basic Engineering Sciences Department, Benha Faculty of Engineering, Benha University, Benha 13512, Egypt

<sup>2</sup>Department of Mathematics, College of Science and Humanities in Al-Kharj, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

<sup>3</sup>Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Shebin El-Kom, Egypt

Correspondence should be addressed to M. A. Elsisy; mohamedelsisy2@gmail.com

Received 20 May 2020; Revised 21 July 2020; Accepted 5 August 2020; Published 30 September 2020

Guest Editor: Edgar Cristian Díaz González

Copyright © 2020 M. A. Elsisy et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we present a new approach which is based on using numerical solutions and swarm algorithms (SAs) to solve the interval quadratic programming problem (IQPP). We use numerical solutions for SA to improve its performance. Our approach replaced all intervals in IQPP by additional variables. This new form is called the modified quadratic programming problem (MQPP). The Karush–Kuhn–Tucker (KKT) conditions for MQPP are obtained and solved by the numerical method to get solutions. These solutions are functions in the additional variables. Also, they provide the boundaries of the basic variables which are used as a start point for SAs. Chaotic particle swarm optimization (CPSO) and chaotic firefly algorithm (CFA) are presented. In addition, we use the solution of dual MQPP to improve the behavior and as a stopping criterion for SAs. Finally, the comparison and relations between numerical solutions and SAs are shown in some well-known examples.

## 1. Introduction

Nonlinear programming has been appeared in solving many real-world problems. Solving interval programming problems is a hot issue in the research area. Interval programming problems are divided into interval linear programming and interval nonlinear programming [1–12].

Interval nonlinear programming problems are used in modeling and solving many real applications such as planning of waste management activities [13]. The mathematical model and the proofs of interval analysis can be found easily in [14]. Many researchers and authors solve the interval nonlinear programming problems by different methods [15–18], but all these methods try to get the optimal solution under some specific conditions. For example, in [8, 9], Hladík divided the problem into subclasses which can be reduced to easy problems. He put a condition for solving these problems that they must be convex quadratic programming. Jiang et al. [11] suggested a method to solve the nonlinear interval number programming problem with uncertain coefficients both in

nonlinear objective function and nonlinear constraints. Liu and Wang [17] presented a numerical method to interval quadratic programming. Li and Tian [18] generalized Liu and Wang's method [17] to solve interval quadratic programming. Their proposed method requires less computing compared with Liu and Wang's method.

As mentioned above, there are many approaches for solving IQPP, but the most common one is dividing the interval problem into two problems. In the first problem, the optimal solution of the lower objective function on the largest feasible region is found, while in the second one, the optimal solution of the upper objective function on the lowest feasible region is obtained. So, the solution value of the interval problem is between values of the lower objective function and the upper objective function. As is known, this process is very difficult in many applications which lead to the difficulty in reaching the lowest value of the objective function of the problem.

KKT conditions are first-order necessary conditions for solving quadratic programming problems. KKT conditions

are optimality conditions for the optimization problems with interval-valued objective functions and real-valued constraint functions investigated and discussed in [19–22]. Wolfe’s duality theorems, strong duality theorems, and duality gap in interval-valued optimization problems are discussed in a good mathematical view in [12]. Chalco-Cano et al. [19] introduced a new concept of a stationary point for an interval-valued function based on the  $g^H$  derivative.

SAs are an important concept in computer science [23, 24]. SAs can be described as a population of agents or individuals interacting with each other and with their environment and working under very few rules. The inspiration often comes from nature, especially biological systems. They are successfully applied in real-life applications. Examples of SAs are ant colony optimization [24, 25], particle swarm optimization (PSO) [26, 27], firefly algorithm (FA) [28], glowworm algorithm [29], krill herd algorithm [30], monkey algorithm [31], and grasshopper optimization algorithm [32].

On the other hand, most of the researchers proposed hybrid algorithms to improve the solution quality, to benefit from their advantages, and to overcome any deficiencies. For example, a gaining sharing knowledge-based algorithm for solving optimization problems over the continuous space was proposed in [33]. Yulian Cao et al. presented a comprehensive learning particle swarm optimizer (CLPSO) embedded with local search (LS) which has the strong global search capability of CLPSO and fast convergence ability of LS [34]. In [35], the authors presented an adaptive particle swarm optimization with supervised learning and control (APSO-SLC) for the parameter settings and diversity maintenance of particle swarm optimization (PSO) to adaptively choose parameters, while improving its exploration competence. A new hybrid PSO algorithm that introduces opposition-based learning (OBL) into PSO variants for improving the latter’s performance is proposed in [36]. In [37], a surrogate-assisted PSO with Pareto active learning was proposed to solve the multiobjective optimization problem with high computational cost. Finally, the historical memory-based PSO (HMPSO) is proposed in [38] which used an estimation of distribution algorithm to estimate and preserve the distribution information of particles’ historical promising  $p$  bests.

In this paper, a new approach is suggested to solve the interval quadratic programming problem (IQPP). IQPP is converted into the modified quadratic programming problem (MQPP) by replacing all intervals by additional variables. KKT conditions of MQPP are derived and solved by a numerical method. The numerical method provides the boundaries of the basic variables which are used as starting points in SAs. The solutions of KKT conditions are obtained by using the Mathematica program. CPSO and CFA are used to solve these problems to give the decision maker (DM) a fast view about the position of the optimal solution in the intervals. The dual of MQPP is discussed. The solutions of this problem are used to improve the behavior of the proposed approach and as a stopping criterion for this approach.

## 2. Interval Quadratic Programming Problem (IQPP)

The interval quadratic programming problem (IQPP) is an interval nonlinear programming problem [9–11]. The objective function is the quadratic function, and the constraints are linear functions. IQPP can be defined as

$$\begin{aligned} \min \quad & \sum_{i=1}^k y_i^f f_i(x) \\ \text{subject to:} \quad & \sum_{i=1}^l y_{ij}^c g_{ij}(x) \leq y_j^R, \quad j = 1, 2, \dots, m, \end{aligned} \quad (1)$$

where  $\sum_{i=1}^k y_i^f f_i(x)$  is an interval-valued function,  $y_i^f = [y_i^{fL}, y_i^{fU}] \forall i$ ,  $f_i(x)$  is a quadratic function,  $g_{ij}(x)$ ,  $i = 1, \dots, l$ ,  $j = 1, \dots, m$  are the linear functions,  $y_{ij}^c = [y_{ij}^{cL}, y_{ij}^{cU}]$ , and  $y_j^R = [y_j^{RL}, y_j^{RU}]$ . The feasible region is supposed to be nonempty and fixed.

The optimal solution of the interval programming problem cannot be defined exactly because at each value belonging to interval coefficients in the objective function and/or the constraints, there may be a new optimal solution. So, no one can define the exact optimal solution. Many researchers desried the optimal solution of IQPP by the objective function values. In [4, 15], the authors defined the optimal solution for the interval linear programming problem. For example, Garajová and Hladík [4] defined the optimal set of the interval linear programming problem and examined sufficient conditions for its closedness, boundedness, connectedness, and convexity. So, we defined the optimal solution as the union of all optimal solutions of IQPP. We explored the whole feasible region to get all possible optimal solutions of IQPP. In our approach, by numerical methods, we tried to get all optimal solutions in the feasible region, while by SAs, we found the best objective value in the whole interval.

## 3. The Solution of IQPP

The idea of our new vision to solve IQPP starts by replacing all intervals by additional variables and converting IQPP to the modified interval nonlinear problem (MQPP). KKT conditions of MQPP are obtained and solved by a numerical method. The solutions of the numerical method are functions in the additional variables. These solutions are providing the boundaries of the basic variables which are used as start points for SA. The dual of MQPP is presented and solved. CPSO and CFA are used to solve MQPP and its dual form. Furthermore, the solution of dual MQPP is used as a stopping criterion for our approach and to improve its performance. The proposed approach leads to explore the whole feasible region to get the optimal solution anywhere in the intervals.

*3.1. Numerical Methodology.* KKT conditions are a system of equations solved by two methods, with additional variables and with interval coefficients. Mathematica is used to solve this system program. In the first method, the equations are solved as algebraic equations where the solutions can be expressed as a

function of the additional variables. These solutions are very helpful for DM if the optimal solution at certain values of interval coefficients is required. We use the Newton method in the second method if we want to know the boundary of the variables. The solution of this method is used as an initial stage of CPSO, and CFA improved their ability to find the solution in short time than using the whole space of the variables.

The following theorems are used for solving the system of nonlinear equations with interval coefficients. Let MQPP have a continuous function  $G: A_0 \subseteq \mathbb{IR}^n \rightarrow \mathbb{IR}^n$  which has a zero  $y^*$  in a given subset  $A$  of  $A_0$ , i.e., a vector  $y^* \in A \subseteq A_0$  exists such that  $G(y^*) = 0$ , where  $\mathbb{IR}^n$  is the set of real intervals'  $n$  vectors. Let  $R^n$  be the set of real  $n$  vectors,  $R^{n \times n}$  be the set of  $n \times n$  matrices,  $\tilde{y}$  be an element of an interval vector  $y$ ,  $B^H$  be a hull inverse of the  $n \times n$  interval matrix  $B$ ,  $\text{int}(A)$  be an interior of the  $m \times n$  interval matrix  $A$ ,  $IA$  be the set  $\{y \in \mathbb{IR}^n \mid y \subseteq A\}$ ,  $\text{int}(y) \equiv ]y, \bar{y}[$  be the interior of an interval  $y$ , and  $\text{vol}(y)$  be a volume-reducing property of the Newton iteration.

**Theorem 1** (see [38]). *Let  $G: A_0 \subseteq R^n \rightarrow R^n$  be Lipschitz continuous on  $A \subseteq A_0$  and let  $B$  be a regular Lipschitz set on  $A$ , then*

- (i) *For every  $a^* \in R^n$ , the equation  $G(y^*) = a^*$  has at most one solution  $y^* \in A$*
- (ii) *The inverse function  $G^{-1}: G^*(A) \rightarrow R^n$  defined on the range  $G^*(A) := \{G(\tilde{y}) \mid \tilde{y} \subseteq A\}$  by  $G^{-1}(\tilde{a}) = \tilde{y} \iff G(\tilde{y}) = \tilde{a}$  is Lipschitz continuous, and  $B^{-1}$  is a Lipschitz matrix for  $G^{-1}$  on  $G^*(A)$*
- (iii) *If  $a^* \in G^*(A)$ , then for every  $\tilde{y}^0 \in A$ ,  $G^{-1}(a^*) \in y^0 + B^H(a^* - G(\tilde{y}^0))$*
- (iv) *If  $A$  is compact and there is a point  $\tilde{y}^0 \in \text{int}(A)$  such that  $G(\tilde{y}) \neq \lambda G(\tilde{y}^0) + (1 - \lambda)a^*$  for all  $\tilde{y} \in \partial A$   $\lambda \in (0, 1]$ , then  $a^* \in G^*(A)$ , i.e., the equation  $G(y^*) = a^*$  has a unique solution  $y^* \in A$*

**Theorem 2** (see [38]). *Under the assumption of Theorem 1 above, if  $\tilde{y} \in y \in IA$ , then every  $y' \in \mathbb{IR}^n$  satisfying  $N(y, \tilde{y}) := \tilde{y} - B^H G(\tilde{y}) \subseteq y'$  has the following three properties:*

- (i) *Every zero  $y^* \in y$  of  $G$  satisfies  $y^* \in y'$*
- (ii) *If  $y' \cap y = 0$ , then  $G$  contains no zero in  $y$*
- (iii) *If  $y^* \in \text{int}(y)$  and  $y' \subseteq y$ , then  $G$  contains a unique zero in  $y$  (and hence in  $y'$ )*

Since  $\tilde{y}$ ,  $y^* \in y$  implies  $y^* \in N(y, \tilde{y})$ , it is natural to consider the general Newton iteration [38]:

$$y^0 := y, y^{l+1} := N_I(y^l, \tilde{y}^l) \cap y^l, \quad \text{for } l = 0, 1, 2, \dots \quad (2)$$

With the general Newton operator:

$$N_I(y, \tilde{y}) := \tilde{y} - (CB)^l (CG(\tilde{y})). \quad (3)$$

**Theorem 3** (see [38]). *Let  $B$  be a strongly regular Lipschitz matrix on  $y \in IA_0$  for  $G: A_0 \subseteq R^n \rightarrow R^n$ . Let  $C \in R^{n \times n}$  be*

*such that  $CB$  is regular and let  $(CB)^l$  be an inverse of  $CB$ . If  $(CB)^l$  is regular, then the Newton iteration (6) is strongly convergent for every choice of  $\tilde{y}^l \in y^l$ . Moreover, for all  $l \geq 0$ , we have either*

$$\tilde{y}^l \notin y^{l+1} \text{ or } y^{l+1} = \tilde{y}^l \text{ and } G(y^l) = 0. \quad (4)$$

**Corollary 1** (see [38]). *If, for some  $C \in R^{n \times n}$ ,  $CB$  is an  $M$ -matrix or  $I - CB < (1/2)$ , then the optimal Newton iteration (6) is strongly convergent for every choice of  $\tilde{y}^l \in y^l$  and the relations (8) and  $\text{vol}(y^{l+1}) \leq (1/2)\text{vol}(y^l)$  hold.*

$\text{vol}(y)$  is defined as

$$\text{vol}(y) := (\bar{y}_1 - \underline{y}_1)(\bar{y}_2 - \underline{y}_2) \cdots (\bar{y}_n - \underline{y}_n), \quad y \in \mathbb{IR}^n. \quad (5)$$

**3.2. Swarm Algorithm (SA).** Chaos theory (CT) is used to improve the performance of many SAs [39], where the high randomness of the chaotic sequence improves the convergence and diversity of the solutions. CT is considered as irregular behavior in nonlinear systems due to using the chaotic maps. These maps are worked as particles which move in a small range of nonlinear dynamic systems without knowing the traveling path of these particles. Many researchers proposed combinations between CT and meta-heuristic algorithms to improve the solution quality such as hybrid chaos-PSO [40], chaotic genetic algorithm [41], combined evolutionary algorithm with chaos [42], chaotic whale optimization algorithm [43], and chaotic artificial neural networks [44].

**3.2.1. Chaotic Firefly Algorithm (CFA).** FA is an evolutionary computation technique [28]. The main advantages of FA are exploitation and exploration. The improved FA with CT which is called the chaotic firefly algorithm (CFA) is applied to solve IQPP. The details of the main steps of CFA are described as follows:

**Step 1. Initialization.** A population of random  $N$  fireflies (solutions) is initialized  $t = 0$ , where  $T$  is the total number of iterations. The position of the  $i$ -th firefly in an  $n$ -dimensional space is denoted as  $x_i$  and represented as  $x_i^t = (x_{i1}, x_{i2}, \dots, x_{in})$ .

**Step 2. Evaluation.** Evaluating the fitness value (the light intensity  $I(X_i^K) \forall i = 1, 2, \dots, N$  of each firefly in the population or simply  $I(x_i^t) = f(x_i^t) \forall i = 1, 2, \dots, N$ .

**Step 3. Determination of Best Solution.** For minimization problems, the firefly that has minimum light intensity is the best solution  $x_b$ .

**Step 4. Updating Positions of Fireflies.** For every firefly  $i = 1, 2, \dots, N$  and for every firefly  $j = 1, 2, \dots, N$  do the following: if  $I(x_i^t) < I(x_j^t)$ , the  $i$ -th firefly is attracted to the firefly  $j$ , and its position  $X_i^K$  is updated according to the following equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha \varepsilon_G, \quad (6)$$

$$r_{ij} = |x_i^t - x_j^t| = \sqrt{\sum_{d=1}^d (x_{id}^t - x_{jd}^t)^2}, \quad (7)$$

where  $\beta_0$  the attractiveness at  $r_{ij}$  is 0,  $\gamma$  is the light absorption coefficient,  $r_{ij}$  is the Cartesian distance between the two fireflies  $i$  and  $j$ ,  $\alpha$  is a step size factor controlling the step size, and  $\varepsilon_G$  is a vector drawn from a Gaussian or other distribution. If  $I(x_j^{t+1}) < I(x_i^t)$ , then  $x_i^{t+1} = x_j^{t+1}$  otherwise  $x_i^{t+1} = x_i^t$ .

*Step 4-1.* Chaotic repairing of the new position  $x_i^{t+1}$ .

*Step 4-2.* Updating the best solution  $x_b$ : if the new position of the  $i$ -th firefly  $x_i^{t+1}$  is better than the best solution  $x_b$ , i.e.,  $I(x_i^{t+1}) < I(x_b)$ , then  $x_b = x_i^{t+1}$ .

*Step 5. Stopping Condition.* If a prespecified stopping criterion is satisfied, stop the run; otherwise, go to *Step 4*.

**3.2.2. Chaotic Particle Swarm Optimization Algorithm (CPSO).** PSO can solve many difficult optimization problems. It has a faster convergence on some problems in comparison [45]. The idea of PSO is that several random particles are placed in the search domain of the optimization problem. At its current location, each particle evaluates the objective function. After that, each particle determines the direction of movement in the search domain by combining some aspects of the history of its own current and best locations with particles located nearby in the swarm, but with some random disturbance. The next iteration takes place after all particles have been moved. Eventually the swarm, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function. The  $i$ -th particle is described by an  $n$ -dimensional vector as  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ , while its velocity is represented as  $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$ . The best position of the particle in its memory that it visited is denoted as  $p_i^{\text{best}} = (p_{i1}, p_{i2}, \dots, p_{in})$ . The best position in the swarm is denoted as  $g^{\text{best}} = (g_1, g_2, \dots, g_n)$ . The steps of the CPSO algorithm are described as follows:

*Step 1. Initialization.*

- Initializing randomly the positions of all particles
- Initializing randomly the velocities of all particles
- Setting  $t = 1$ , where  $t$  is the increment of time and  $T$  is the total number of iterations

*Step 2. Optimization.*

- Evaluating the objective function value  $f_i^t$
- If  $f_i^t \leq f_i^{\text{best}}$ , then  $f_i^{\text{best}} = f_i^t$  and  $p_i^{\text{best}} = x_i^t$
- If  $f_i^t \leq f_g^{\text{best}}$ , then  $f_g^{\text{best}} = f_i^t$  and  $g_i^{\text{best}} = x_i^t$
- If the stopping criterion is satisfied go to Step 3
- All velocities ( $v_i^t \forall i$  the particle) are updated according to the following equation:

$$v_i^{t+1} = wv_i^t + c_1 \times r_1 \times (p_i^{\text{best}} - x_i^t) + c_2 \times r_2 \times (g_i^{\text{best}} - x_i^t), \quad (8)$$

where  $w$  is an inertia term,  $c_1$  and  $c_2$  are the positive constants, and  $r_1$  and  $r_2$  are the random numbers belonging to  $(0, 1)$ .

- All positions ( $x_i^t \forall i$  the particle) are updated according to the following equation:

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \quad (9)$$

- Chaotic repairing of the new position  $x_i^{t+1}$

- $t = t + 1$

- Go to Step 2(a)

*Step 3. Termination.* If a prespecified stopping criterion is satisfied, stop the run; otherwise, go to Step 2.

**3.2.3. Chaotic Repairing of Infeasible Solution.** If the new position  $x_i^{t+1}$  is infeasible, it is repaired according to the following equation:

$$x_i^{t+1} = \phi \times x_i^{t+1} + (1 - \phi) \times \text{FS}. \quad (10)$$

If  $x_i^{t+1}$  is still infeasible,  $x_i^{t+1}$  is repaired according to the following equation:

$$x_i^{t+1} = \phi \times \text{FS} + (1 - \phi) \times x_i^{t+1}, \quad (11)$$

where FS is any feasible solution in the search space and  $\phi$  is a chaotic number generated by the following logistic map:

$$\phi_{m+1} = c\phi_m(1 - \phi_m), \quad (12)$$

where  $m$  is the age of the infeasible solution,  $c = 4$ ,  $\phi_0 \in (0, 1)$ , and  $\phi_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$ .

**3.3. The Proposed Approach.** In this section, we discuss the proposed approach. The following steps describe the proposed approach clearly:

*Step 1.* Replacing all intervals in IQPP by additional variables which is called MQPP and obtaining the dual form of MQPP.

*Step 2.* Constructing KKT for MQPP, and solving KKT equations by the numerical algorithm.

*Step 3.* Using the solutions of the numerical algorithm as a start point of CPSO and CFA.

*Step 4.* Solving MQPP and its dual form by CPSO and CFA.

*Step 5.* The values of the objective function which are obtained from solving the problem by SA and its dual form are compared. If their values are the same, the global optimal solution of our problem is found. If there is a difference between the outputs from the problem and its dual, we solve the problem and its dual form again until the difference between them is  $\varepsilon$ , where  $\varepsilon$  can be computed as

$$\varepsilon = \frac{\delta}{\text{the optimal value of the problem}}, \quad (13)$$

where  $\delta$  is the difference between the optimal value of the problem and the optimal value of its dual problem. This solution is a local optimal solution. This comparison is used as a new stopping criterion.

The suggested method is suitable for convex and nonconvex problems. The steps of the proposed approach are illustrated in Figure 1.

## 4. Results and Discussion

The proposed algorithm is tested by solving three problems taken from the literature. Each problem was independently run 30 times. The proposed algorithm was programmed in MATLAB (R2016b) and implemented on the PC with P4 CPU 3.00 GHz, 1 GB RAM with an i5 processor, Windows 7 operating system. The proposed algorithm, as any nontraditional optimization algorithms, involves a number of parameters that affect the performance of the algorithm. The parameters adopted in the implementation of CFA and CPSO are listed in Table 1.

*4.1. Problem 1.* This problem is formulated as follows [10]:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + [1, 2] \\ \text{subject to:} \quad & [1, 6]x_1 + [1, 2]x_2 \geq [1, 12], \quad x_1, x_2 \geq 0. \end{aligned} \quad (14)$$

By replacing all intervals by additional parameters, the problem becomes

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + a \\ \text{subject to:} \quad & b_1x_1 + b_2x_2 \geq b_3, \quad x_1, x_2 \geq 0, \end{aligned} \quad (15)$$

where  $a = [1, 2]$ ,  $b_1 = [1, 6]$ ,  $b_2 = [1, 2]$ , and  $b_3 = [1, 12]$ .

The dual form of problem (15) is

$$\begin{aligned} \max \quad & x_1^2 + x_2^2 + a - u_1(b_1x_1 + b_2x_2 - b_3) - u_2x_1 - u_3x_2, \\ & u_1 \geq 0, u_2 \geq 0, u_3 \geq 0. \end{aligned} \quad (16)$$

KKT conditions of problem (15) are

$$\begin{aligned} 2x_1 - u_1b_1 - u_2 &= 0, \\ 2x_2 - u_1b_2 - u_3 &= 0, \\ u_1(b_1x_1 + b_2x_2 - b_3) &= 0, \\ u_2x_1 &= 0, \\ u_3x_2 &= 0, \\ b_1x_1 + b_2x_2 &\geq b_3, \\ x_1, x_2 &\geq 0, \\ u_1 &\geq 0, \\ u_2 &\geq 0, \\ u_3 &\geq 0. \end{aligned} \quad (17)$$

In [8, 9], problem (14) can be divided into two problems. The first problem is

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + 1 \\ \text{subject to:} \quad & 6x_1 + 2x_2 \geq 1, \quad x_1, x_2 \geq 0. \end{aligned} \quad (18)$$

Its solution is  $(x_1, x_2) = (0.15, 0.05)$  and  $f(x) = 1.025$ . The second problem is

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + 2 \\ \text{subject to:} \quad & x_1 + x_2 \geq 12, \quad x_1, x_2 \geq 0. \end{aligned} \quad (19)$$

Its solution is  $(x_1, x_2) = (6, 6)$  and  $f(x) = 74$ . The solutions of KKT conditions in (18) can be expressed as

$$\text{solutions set} = \left\{ \left( \frac{b_1b_3}{b_1^2 + b_2^2}, \frac{b_2b_3}{b_1^2 + b_2^2} \right) \right\}. \quad (20)$$

In addition, the numerical solution provides the boundaries of the basic variables as  $x_1 = [0.025, 36]$  and  $x_2 = [0.025, 12]$ . The comparison between different types of SAs in solving problem (1) is shown in Table 2.

*4.2. Problem 2.* This problem is formulated as follows [1]:

$$\begin{aligned} \min \quad & [2, 3]x_1^2 + 2x_2^2 - 2x_1x_2 + [-5, -3]x_1 + [1, 2]x_2, \\ \text{subject to:} \quad & [1, 2]x_1 + x_2 \leq [2, 4], [2, 3]x_1 + [-1, -0.5]x_2 \\ & \leq [3, 4], \quad x_1, x_2 \geq 0. \end{aligned} \quad (21)$$

By replacing all intervals by additional parameters, the problem becomes

$$\begin{aligned} \min \quad & a_1x_1^2 + 2x_2^2 - 2x_1x_2 + a_2x_1 + a_3x_2 \\ \text{subject to:} \quad & b_1x_1 + x_2 \leq b_2b_3x_1 + b_4x_2 \leq b_5, \quad x_1, x_2 \geq 0, \end{aligned} \quad (22)$$

where  $a_1 = [2, 3]$ ,  $a_2 = [-5, -3]$ ,  $a_3 = [1, 2]$ ,  $b_1 = [1, 2]$ ,  $b_2 = [2, 4]$ ,  $b_3 = [2, 3]$ ,  $b_4 = [-1, -0.5]$ , and  $b_5 = [3, 4]$ .

The dual form of problem (22) is

$$\begin{aligned} \max \quad & a_1x_1^2 + 2x_2^2 - 2x_1x_2 + a_2x_1 + a_3x_2 + u_1(b_1x_1 + x_2 - b_2) \\ & + u_2(b_3x_1 + b_4x_2 - b_5) - u_3x_1 - u_4x_2 \\ \text{subject to:} \quad & u_1 \geq 0, u_2 \geq 0, u_3 \geq 0, u_4 \geq 0. \end{aligned} \quad (23)$$

The KKT conditions of problem (22) are

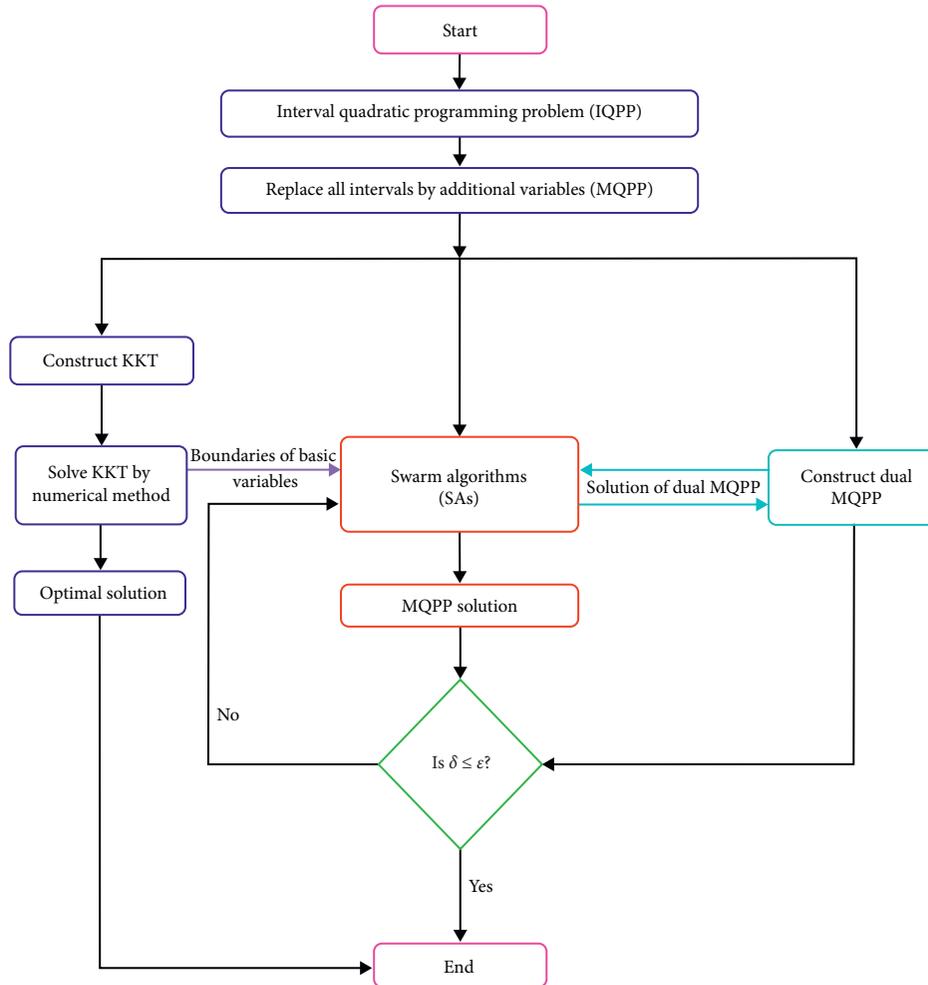


FIGURE 1: The flowchart of the proposed approach.

TABLE 1: The proposed algorithm parameters.

CFA parameters		CPSO parameters	
The swarm size ( $m$ )	20	The swarm size	20
Number of iteration ( $T$ )	200	Number of iteration ( $T$ )	200
Initial attractiveness ( $\beta_0$ )	1	Acceleration coefficients	$c_1$ 2.8
The light absorption coefficient ( $\gamma$ )	1		$c_2$ 1.3
The step size factor ( $\alpha$ )	0.95	The inertia weight ( $w$ )	0.6
Chaos search repairing iteration ( $m$ )			1E02
$\varepsilon$			1E-6

TABLE 2: The results of problem (1) by using SAs.

	PSO	FA	CPSO	CFA
$(x_1, x_2)$	(0.37931, 0.21558)	(0.20087, 0.12304)	(0.1504, 0.0489)	(0.149462, 0.0516)
$f(x)$	1.6209	1.5252	1.0250	1.0250

$$\begin{aligned}
2a_1x_1 - 2x_2 + a_2 + u_1b_1 + u_2b_3 - u_3 &= 0, \\
4x_2 - 2x_1 + a_3 + u_1 + u_2b_4 - u_4 &= 0, \\
u_1(b_1x_1 + x_2 - b_2) &= 0, \\
u_2(b_3x_1 + b_4x_2 - b_5) &= 0, \\
u_3x_1 &= 0, \\
u_4x_2 &= 0, \\
b_1x_1 + x_2 &\leq b_2 \\
b_3x_1 + b_4x_2 &\leq b_5x_1, \\
x_2 &\geq 0, \\
u_1 &\geq 0, \\
u_2 &\geq 0, \\
u_3 &\geq 0, \\
u_4 &\geq 0.
\end{aligned} \tag{24}$$

In [8, 9], problem (21) is divided into two problems. The first problem is

$$\begin{aligned}
\min \quad & 2x_1^2 + 2x_2^2 - 2x_1x_2 - 5x_1 + x_2, \\
\text{subject to: } & x_1 + x_2 \leq 4, 2x_1 - x_2 \leq 4, \quad x_1, x_2 \geq 0.
\end{aligned} \tag{25}$$

Its solution is  $(x_1, x_2) = (1.5, 0.5)$  and  $f(x) = -3.5$ . The second problem is

$$\begin{aligned}
\min \quad & 3x_1^2 + 2x_2^2 - 2x_1x_2 - 3x_1 + 2x_2 \\
\text{subject to: } & 2x_1 + x_2 \leq 2, 3x_1 - 0.5x_2 \leq 3, \quad x_1, x_2 \geq 0.
\end{aligned} \tag{26}$$

Its solution is  $(x_1, x_2) = (0.5, 0)$  and  $f(x) = -0.75$ . The solutions of KKT conditions in (24) can be expressed as

$$= \left\{ \begin{array}{l} \left( \frac{2a_2 + a_3}{2 - 4a_1}, \frac{a_2 + a_1a_3}{2 - 4a_1} \right), \left( -\frac{a_2}{2a_1}, 0 \right), \left( \frac{-b_2b_4 + b_5}{b_3 - b_1b_4}, \frac{b_2b_3 - b_1b_5}{b_3 - b_1b_4} \right), \left( \frac{b_5}{b_3}, 0 \right), \left( \frac{b_2}{b_1}, 0 \right), \\ \left( \frac{2b_2(2b_1 + 1) - a_2 + a_3b_1}{2a_1 + 4b_1(b_1 + 1)}, \frac{a_2b_1 - a_3b_1^2 + 2b_2(a_1 + b_1)}{2a_1 + 4b_1(b_1 + 1)} \right) \end{array} \right\}. \tag{27}$$

In addition, the numerical solution provides the boundaries of the basic variables as  $x_1 = [0.4, 4.08333]$  and  $x_2 = [-1.6, 3.6]$ . The results of problem (2) by using SAs are shown in Table 3.

4.3. *Problem 3.* This problem is formulated as follows [46]:

$$\begin{aligned}
\min \quad & [2, 3]x_1^2 + 2x_2^2 - 2x_1x_2 + [-5, -3]x_1 + [1, 2]x_2 \\
\text{subject to: } & [1, 2]x_1 + x_2 \leq [2, 4], [2, 3]x_1 + [-1, -0.5]x_2 \\
& \leq [3, 4], [4, 5]x_1 + [-8, -7]x_2 = [1, 1.5], \quad x_1, x_2 \geq 0.
\end{aligned} \tag{28}$$

By replacing all intervals by additional parameters, the problem becomes

$$\begin{aligned}
\min \quad & a_1x_1^2 + 2x_2^2 - 2x_1x_2 + a_2x_1 + a_3x_2 \\
\text{subject to: } & b_1x_1 + x_2 \leq b_2, b_3x_1 + b_4x_2 \\
& \leq b_5, b_6x_1 + b_7x_2 = b_8, \quad x_1, x_2 \geq 0,
\end{aligned} \tag{29}$$

where  $a_1 = [2, 3]$ ,  $a_2 = [-5, -3]$ ,  $a_3 = [1, 2]$ ,  $b_1 = [1, 2]$ ,  $b_2 = [2, 4]$ ,  $b_3 = [2, 3]$ ,  $b_4 = [-1, -0.5]$ ,  $b_5 = [3, 4]$ ,  $b_6 = [4, 5]$ ,  $b_7 = [-8, -7]$ , and  $b_8 = [1, 1.5]$ .

The dual form of problem (29) is

$$\begin{aligned}
\max \quad & a_1x_1^2 + 2x_2^2 - 2x_1x_2 + a_2x_1 + a_3x_2 + u_1(b_1x_1 + x_2 - b_2) \\
& + u_2(b_3x_1 + b_4x_2 - b_5) + \lambda(b_6x_1 + b_7x_2 - b_8) \\
& - u_3x_1 - u_4x_2, \\
u_1 &\geq 0, u_2 \geq 0, u_3 \geq 0, u_4 \geq 0.
\end{aligned} \tag{30}$$

The KKT conditions of problem (28) are

$$\begin{aligned}
2a_1x_1 - 2x_2 + a_2 + u_1 + u_1b_1 + u_2b_3 + \lambda b_6 - u_3 &= 0, \\
4x_2 - 2x_1 + a_3 + u_1 + u_2b_4 + \lambda b_7 - u_4 &= 0, \\
u_1(b_1x_1 + x_2 - b_2) &= 0, \\
u_2(b_3x_1 + b_4x_2 - b_5) &= 0, \\
\lambda(b_6x_1 + b_7x_2 - b_8) &= 0, \\
u_3x_1 &= 0, \\
u_4x_2 &= 0, \\
b_1x_1 + x_2 &\leq b_2, \\
b_3x_1 + b_4x_2 &\leq b_5, \\
b_6x_1 + b_7x_2 &= b_8, \\
x_1, x_2 &\geq 0, \\
u_1 &\geq 0, \\
u_2 &\geq 0, \\
u_3 &\geq 0, \\
u_4 &\geq 0.
\end{aligned} \tag{31}$$

TABLE 3: The results of problem (2) by using SAs.

	PSO	FA	CPSO	CFA
$(x_1, x_2)$	(1.217548, 0.42732)	(1.23733, 0.39811)	(0.1504, 0.0489)	(1.4964, 0.4970)
$f(x)$	-3.2759	-3.3355	-3.3355	-3.5

TABLE 4: The results of problem (2) by using SAs.

	PSO	FA	CPSO	CFA
$(x_1, x_2)$	(1.07113, 0.41978)	(1.07116, 0.41973)	(0.7754, 0.2002)	(1.4998, 0.5624)
$f(x)$	-1.9297	-3.1872	-2.7046	-3.4922

TABLE 5: The statistical results of SAs over the 30 runs.

			Algorithm	CPU time(s)	Mean	SD	Worst	Best
Problem 1	Range of $f(x)$	[1.025, 74]	PSO	5.45	1.6783	0.0419	1.7024	1.6209
			FA	4.56	1.5337	0.0065	1.5379	1.5252
			CPSO	3.76E-002	1.0250	0	1.0250	1.0250
			CFA	2.95E-002	1.0250	0	1.0250	1.0250
Problem 2	Range of $f(x)$	[-3.5, -0.75]	PSO	12.05	-3.2244	0.0458	-3.1846	-3.2759
			FA	7.96	-3.3230	0.0066	-3.3257	-3.3355
			CPSO	2.54E-002	-3.4999	0	-3.4999	-3.4999
			CFA	1.51E-002	-3.5	0	-3.5	-3.5
Problem 3	Range of $f(x)$	[-3.4922, 0.5217]	PSO	22.45	-1.8795	0.0304	-1.8750	-1.9297
			FA	10.38	-3.1498	0.0487	-3.0907	-3.1872
			CPSO	9.41E-002	2.7037	0.0013	-2.7005	-2.7046
			CFA	8.95E-002	-3.4922	0	-3.4922	-3.4922

In [8, 9], problem (28) is divided into two problems. The first problem is

$$\begin{aligned} \min & 2x_1^2 + 2x_2^2 - 2x_1x_2 - 5x_1 + x_2 \\ \text{subject to: } & x_1 + x_2 \leq 4, 2x_1 - x_2 \leq 4, 4x_1 - 8x_2 = 1.5, \quad x_1, x_2 \geq 0. \end{aligned} \quad (32)$$

Its solution is  $(x_1, x_2) = (1.5, 0.5625)$  and  $f(x) = -3.4922$ . The second problem is

$$\begin{aligned} \min & 3x_1^2 + 2x_2^2 - 2x_1x_2 - 3x_1 + 2x_2 \\ \text{subject to: } & 2x_1 + x_2 \leq 2, 3x_1 - 0.5x_2 \leq 3, 5x_1 + -7x_2 = 1, \quad x_1, x_2 \geq 0. \end{aligned} \quad (33)$$

Its solution is  $(x_1, x_2) = (0.3268, 0.0906)$  and  $f(x) = -0.5217$ .

The solutions of KKT conditions in (31) can be expressed as

$$\text{solutions set} = \left\{ \begin{aligned} & \left( \frac{2a_2 + a_3}{2 - 4a_1}, \frac{a_2 + a_1a_3}{2 - 4a_1} \right), \left( -\frac{a_2}{2a_1}, 0 \right), \left( \frac{b_4b_8 + b_5b_7}{b_4b_6 - b_3b_7}, \frac{b_5b_6 + b_3b_8}{b_4b_6 - b_3b_7} \right), \left( \frac{b_5}{b_3}, 0 \right), \left( \frac{b_8 + b_2b_7}{b_6 - b_1b_7}, \frac{b_2b_6 + b_1b_8}{b_6 - b_1b_7} \right), \\ & \left( \frac{b_5 - b_2b_4}{b_3 - b_1b_4}, \frac{b_2b_3 - b_1b_5}{b_3 - b_1b_4} \right), \left( \frac{b_5}{b_3}, 0 \right), \left( \frac{2b_2(2b_1 + 1) - a_2 + a_3b_1}{2a_1 + 4b_1(b_1 + 1)}, \frac{a_2b_1 - a_3b_1^2 + 2b_2(a_1 + b_1)}{2a_1 + 4b_1(b_1 + 1)} \right), \left( \frac{b_2}{b_1}, 0 \right) \\ & \left( \frac{a_3b_6b_7 - a_2b_7^2 + 2b_8(2b_6b_7)}{4b_6^2 + 4b_6b_7 + 2a_1b_7^2}, \frac{-a_3b_6^2 + a_2b_6b_7 + 2b_8(b_6 + a_1b_7)}{4b_6^2 + 4b_6b_7 + 2a_1b_7^2} \right), \left( \frac{b_2}{b_1}, 0 \right) \end{aligned} \right\}. \quad (34)$$

In addition, the numerical solution provides the boundaries of the basic variables  $x_1 = [0.28289, 9.8]$  and  $x_2 = [-5.4, 5.8]$ . In Table 4, the results of problem (2) by using SAs are listed.

In addition, the statistical results obtained, by original PSO, original FA, CPSO, and CFA, over the 30 runs are summarized in terms of CPU time, mean value, standard deviation, and worst and best values in Table 5.

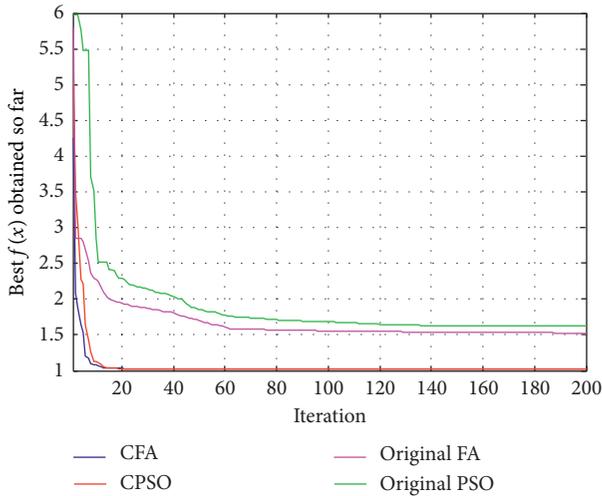


FIGURE 2: The convergence curve of the best  $f(x)$  obtained so far by original PSO, original FA, CPSO, and CFA for problem 1.

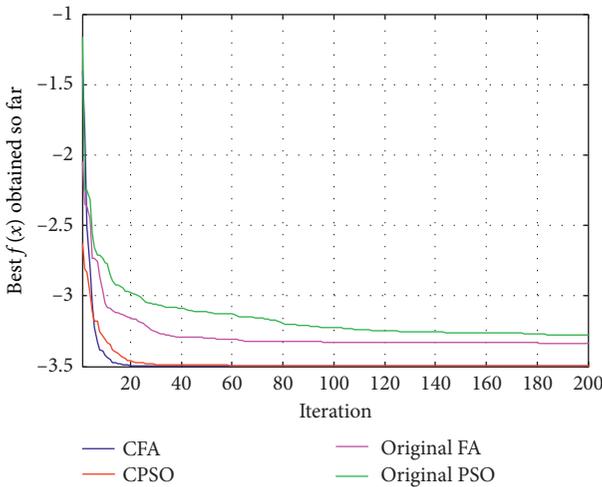


FIGURE 3: The convergence curve of the best  $f(x)$  obtained so far by original PSO, original FA, CPSO, and CFA for problem 2.

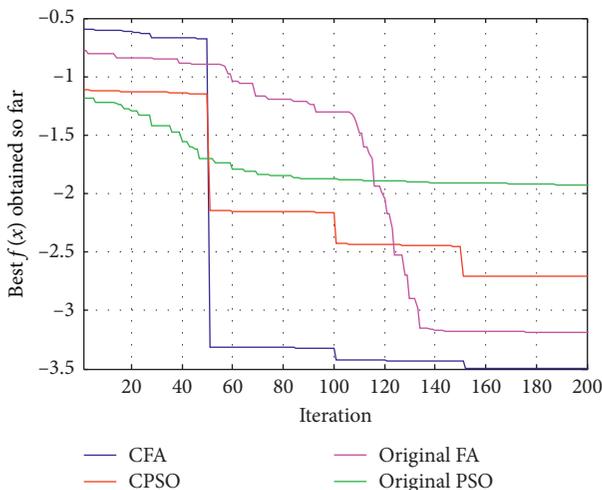


FIGURE 4: The convergence curve of the best  $f(x)$  obtained so far by original PSO, original FA, CPSO, and CFA for problem 3.

Furthermore, Figures 2–4 show the convergence curve of the best  $f(x)$  obtained so far by original PSO, original FA, CPSO, and CFA for the 3 problems.

Results show that the proposed SAs (CFA and CPSO) outperform the other original algorithms in terms of the optimality. In addition, these results prove that the proposed SAs can solve IQPP effectively with low computational cost where the CPU time is less than the other original algorithms as shown in Table 5. In other words, the solutions, of the test problems, of CPSO and CFA are the same as the solutions of previous methods, but they are very fast without any effort of computation. On the other hand, the numerical approach gives the solution as a general formula in the additional variables, where we can obtain, by this formula, the solution at any values inside the intervals. In addition, the numerical solution provides the boundaries of the basic variables which are used in the step of initialization in SAs. Finally, we can say that our approach, as any SAs, is more generalized and suitable for real applications than traditional methods.

## 5. Conclusion

This paper deals with a new approach to solve IQPP. We aim to explore the feasible region to get the optimal solution anywhere. All intervals were replaced by additional variables. The new form with additional variables is MQPP. KKT conditions for MQPP were solved numerically to get the solutions as a function in the additional variables and provide the boundaries of the basic variables. The solutions are used as start points for SAs. CPSO and CFA are used to solve MQPP and its dual form. The advantages of our procedure are (1) the solution of the numerical method is more general than previous methods, (2) giving the decision maker a very fast view of the optimal solution inside the intervals, (3) using the optimal solution of the dual problem as a stopping criterion for SAs is more suitable than other criteria, and (4) its effectiveness is verified as compared with other studies. Also, we compare PSO, FA, CPSO, and CFA with each other. Real applications of interval nonlinear programming problems should be conducted in the future. In addition, we are planning to use this vision to solve multiobjective linear programming with interval coefficients. Also, we aim to discuss the GSK algorithm to solve IQPP.

## Data Availability

All data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that this article content has no conflicts of interest.

## References

- [1] A. K. Bhurjee and G. Panda, "Efficient solution of interval optimization problem," *Mathematical Methods of Operations Research*, vol. 76, no. 3, pp. 273–288, 2012.

- [2] M. Chen, S.-G. Wang, P. P. Wang, and X. Ye, "A new equivalent transformation for interval inequality constraints of interval linear programming," *Fuzzy Optimization and Decision Making*, vol. 15, no. 2, pp. 155–175, 2016.
- [3] J. W. Chinneck and K. Ramadan, "Linear programming with interval coefficients," *The Journal of the Operational Research Society*, vol. 51, no. 2, pp. 209–220, 2000.
- [4] E. Garajová and M. Hladík, "On the optimal solution set in interval linear programming," *Computational Optimization and Applications*, vol. 72, no. 1, pp. 269–292, 2019.
- [5] M. Hladík, "Robust optimal solutions in interval linear programming with forall-exists quantifiers," *European Journal of Operational Research*, vol. 254, no. 3, pp. 705–714, 2016.
- [6] M. Hladík, "How to determine basis stability in interval linear programming," *Optimization Letters*, vol. 8, no. 1, pp. 375–389, 2014.
- [7] M. Hladík, "Weak and strong solvability of interval linear systems of equations and inequalities," *Linear Algebra and Its Applications*, vol. 438, no. 11, pp. 4156–4165, 2013.
- [8] M. Hladík, "Optimal value bounds in nonlinear programming with interval data," *TOP*, vol. 19, no. 1, pp. 93–106, 2011.
- [9] M. Hladík, "Optimal value range in interval linear programming," *Fuzzy Optimization and Decision Making*, vol. 8, no. 3, pp. 283–294, 2009.
- [10] C. Jiang, Z. G. Zhang, Q. F. Zhang, X. Han, H. C. Xie, and J. Liu, "A new nonlinear interval programming method for uncertain problems with dependent interval variables," *European Journal of Operational Research*, vol. 238, no. 1, pp. 245–253, 2014.
- [11] C. Jiang, X. Han, G. R. Liu, and G. P. Liu, "A nonlinear interval number programming method for uncertain optimization problems," *European Journal of Operational Research*, vol. 188, no. 1, pp. 1–13, 2008.
- [12] H.-C. Wu, "On interval-valued nonlinear programming problems," *Journal of Mathematical Analysis and Applications*, vol. 338, no. 1, pp. 299–316, 2008.
- [13] X. Y. Wu, G. H. Huang, L. Liu, and J. B. Li, "An interval nonlinear program for the planning of waste management systems with economies-of-scale effects-A case study for the region of Hamilton, Ontario, Canada," *European Journal of Operational Research*, vol. 171, no. 2, pp. 349–372, 2006.
- [14] R. E. Moore, *Methods and Applications of Interval Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1979.
- [15] M. Allahdadi and H. Mishmast Nehi, "The optimal solution set of the interval linear programming problems," *Optimization Letters*, vol. 7, no. 8, pp. 1893–1911, 2013.
- [16] H. Ishibuchi and H. Tanaka, "Multiobjective programming in optimization of the interval objective function," *European Journal of Operational Research*, vol. 48, no. 2, pp. 219–225, 1990.
- [17] S.-T. Liu and R.-T. Wang, "A numerical solution method to interval quadratic programming," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1274–1281, 2007.
- [18] W. Li and X. Tian, "Numerical solution method for general interval quadratic programming," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 589–595, 2008.
- [19] Y. Chalco-Cano, R. Osuna-Gómez, B. Hernández-Jiménez, and H. Román-Flores, *A Note on Optimality Conditions to Interval Optimization Problems*, Atlantis Press, Paris, France, 2015.
- [20] L. Stefanini and B. Bede, "Generalized Hukuhara differentiability of interval-valued functions and interval differential equations," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 3-4, pp. 1311–1328, 2009.
- [21] H.-C. Wu, "The Karush-Kuhn-Tucker optimality conditions in an optimization problem with interval-valued objective function," *European Journal of Operational Research*, vol. 176, no. 1, pp. 46–59, 2007.
- [22] J. Zhang, S. Liu, L. Li, and Q. Feng, "The KKT optimality conditions in a class of generalized convex optimization problems with an interval-valued objective function," *Optimization Letters*, vol. 8, no. 2, pp. 607–631, 2014.
- [23] K. G. Dhal, S. Ray, A. Das, and S. Das, "A survey on nature-inspired optimization algorithms and their application in image enhancement domain," *Archives of Computational Methods in Engineering*, vol. 26, pp. 1607–1638, 2018.
- [24] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., pp. 36–39, Springer, Boston, MA, USA, 2010.
- [25] Abd el-wahed, F. Waiel, A. A. Mousa, and M. A. Elsisy, "Solving economic emissions load dispatch problem by using hybrid ACO-MSM approach," *The Online Journal on Power and Energy Engineering (OJPEE)*, vol. 1, no. 1, pp. 31–35, 2009.
- [26] M. A. El-Shorbagy and A. E. Hassanien, "Particle swarm optimization from theory to applications," *International Journal of Rough Sets and Data Analysis*, vol. 5, no. 2, pp. 1–24, 2018.
- [27] M. A. El-Shorbagy, "Hybrid particle swarm algorithm for multi-objective optimization," M.S. thesis, Menoufiya University, Al Minufya, Egypt, 2010.
- [28] S. Verma and V. Mukherjee, "Firefly algorithm for congestion management in deregulated environment," *Engineering Science and Technology, an International Journal*, vol. 19, no. 3, pp. 1254–1265, 2016.
- [29] M. Marinaki and Y. Marinakis, "A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands," *Expert Systems with Applications*, vol. 46, pp. 145–163, 2016.
- [30] A. L. a. Bolaji, M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and L. M. Abualigah, "A comprehensive review: krill Herd algorithm (KH) and its applications," *Applied Soft Computing*, vol. 49, pp. 437–446, 2016.
- [31] Y. Zhou, X. Chen, and G. Zhou, "An improved monkey algorithm for a 0-1 knapsack problem," *Applied Soft Computing*, vol. 38, pp. 817–830, 2016.
- [32] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimization algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
- [33] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 7, pp. 1501–1529, 2020.
- [34] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, "Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 718–731, 2018.
- [35] W. Dong and M. Zhou, "A supervised learning and control method to improve particle swarm optimization algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1135–1148, 2016.
- [36] Q. Kang, C. Xiong, M. Zhou, and L. Meng, "Opposition-based hybrid strategy for particle swarm optimization in noisy environments," *IEEE Access*, vol. 6, pp. 21888–21900, 2018.

- [37] Z. Lv, L. Wang, Z. Han, J. Zhao, and W. Wang, "Surrogate-assisted particle swarm optimization algorithm with Pareto active learning for expensive multi-objective optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 838–849, 2019, p.
- [38] A. Neumaier, *Interval Methods for Systems of Equations*, Vol. 37, Cambridge University Press, Cambridge, UK, 1990.
- [39] D. Yang, G. Li, and G. Cheng, "On the efficiency of chaos optimization algorithms for global optimization," *Chaos, Solitons & Fractals*, vol. 34, no. 4, pp. 1366–1375, 2007.
- [40] M. A. E. Shorbagy and A. A. Mousa, "Chaotic particle swarm optimization for imprecise combined economic and emission dispatch problem," *Review of Information Engineering and Applications*, vol. 4, no. 1, pp. 20–35, 2017.
- [41] M. A. El-Shorbagy, A. A. Mousa, and S. M. Nasr, "A chaos-based evolutionary algorithm for general nonlinear programming problems," *Chaos, Solitons & Fractals*, vol. 85, pp. 8–21, 2016.
- [42] J. Xiao, *Improved Quantum Evolutionary Algorithm Combined with Chaos and its Application*, Springer, Berlin, Germany, 2009.
- [43] D. Oliva, M. Abd El Aziz, and A. Ella Hassanien, "Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm," *Applied Energy*, vol. 200, pp. 141–154, 2017.
- [44] Z. Aram, S. Jafari, J. Ma, J. C. Sprott, S. Zندهrouh, and V.-T. Pham, "Using chaotic artificial neural networks to model memory in the brain," *Communications in Nonlinear Science and Numerical Simulation*, vol. 44, pp. 449–459, 2017.
- [45] M. A. El-Shorbagy, M. Elhoseny, A. E. Hassanien, and S. H. Ahmed, "A novel PSO algorithm for dynamic wireless sensor network multiobjective optimization problem," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 11, 2018.
- [46] D. Ghosh, D. Ghosh, S. K. Bhuiya, and L. K. Patra, "A saddle point characterization of efficient solutions for interval optimization problems," *Journal of Applied Mathematics and Computing*, vol. 58, no. 1-2, pp. 193–217, 2018.