Available online at http://scik.org
Adv. Fixed Point Theory, 2026, 16:16
https://doi.org/10.28919/afpt/9837
ISSN: 1927-6303

FIXED POINTS OF SET-VALUED MAPPINGS IN QUASILINEAR METRIC SPACES VIA SWARM INTELLIGENCE

NATASHA CLARISSA MAHARANI, MOHAMAD MUSLIKH*, ABDUL ROUF ALGHOFARI

Department of Mathematics, University of Brawijaya, Malang 65145, Indonesia

Copyright © 2026 the author(s). 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.

Abstract. Fixed point theory in quasilinear metric spaces plays an important role in programming and data science. However, determining the existence and approximation of fixed points becomes particularly challenging when dealing with set-valued mappings. In this article, we introduce the use of swarm intelligence algorithms as an alternative approach for approximating fixed points in quasilinear metric spaces ($\mathbb{R}^n$). To evaluate their effectiveness, five promising swarm intelligence algorithms are employed and analyzed. The results indicate that swarm intelligence provides a robust and competitive framework for fixed point approximation in quasilinear metric spaces.

Keywords: fixed point; set-valued functions; quasilinear metric space; optimization algorithm; swarm intelligent.
2020 AMS Subject Classification: 47H10, 39B52, 26E25, 65K10.

1. Introduction

Fixed point theorems have been a central topic in mathematical research for several decades. Many mathematical problems from various branches of mathematics can be formulated as fixed point problems. This theory is particularly useful for addressing existence and uniqueness issues of solutions in nonlinear differential and integral equations [17]. Moreover, fixed point theory

*Corresponding author
E-mail address: mslk@ub.ac.id
Received February 21, 2026

plays an important role in data science by providing a natural framework for advanced convex optimization methods as well as modern nonlinear techniques [4].

The existence of fixed point theorem was first introduced by the mathematician L. E. J. Brouwer in 1912 [3]. Brouwer's contribution not only laid the foundation for the development of fixed point theory but also influenced many other fields, such as game theory, economics, and dynamical systems[2, 8]. For instance, in economics, this theorem is used to prove the existence of market equilibrium, where prices and quantities converge to a stable fixed point.

Despite its significant impact, Brouwer's fixed point theorem has several limitations. One major limitation is that it applies only to finite-dimensional spaces $\mathbb{R}^n$. These limitations motivated mathematicians to seek broader generalizations of fixed point theorems that could be applied to infinite-dimensional spaces. This effort culminated in the 1930s when Juliusz Schauder established a fixed point theorem valid in infinite-dimensional spaces [12]. This result, known as the Schauder fixed point theorem, marked an important step toward extending fixed point theory to more complex settings.

The study of fixed points for set-valued mappings originated from the seminal work of Shizuo Kakutani, who extended Brouwer's fixed point theorem to collections of sets in finite-dimensional spaces [10]. This generalization marked a fundamental advancement in fixed point theory by allowing mappings to assign sets rather than single points. However, Kakutani's theorem has an important limitation, as it cannot, in general, be applied to infinite-dimensional spaces due to the increased structural complexity of such spaces, which complicates the application of classical fixed point principles [10].

Subsequent research further broadened the scope of fixed point theory for set-valued mappings. In particular, Nadler [9] introduces the existence of fixed point theorem for set-valued mappings in metric spaces. In Nadler's theorem, the contraction condition is extended to apply to the set-valued images of the mapping using Hausdorff metric. This result opened new avenues for investigating the existence of fixed points in a more general framework, as it is no longer restricted to single-valued mappings.

Set-valued mappings have attracted considerable attention due to their greater generality compared to single-valued functions and their wide applicability in fields such as economics, optimization, and data processing in programming [5]. Despite this progress, existing studies have not yet fully explored an even more general framework, such as quasilinear metric spaces[1], which naturally accommodate both single-valued and set-valued mappings. Quasilinear metric spaces are frequently employed in various aspects of programming, particularly for improving algorithmic efficiency by exploiting their structural properties [19]. Moreover, fixed point theorems in such spaces play a crucial role in ensuring the convergence of iterative algorithms and accelerating model training processes [4].

Finding fixed points in quasilinear metric spaces is highly challenging due to their complexity and nonlinear characteristics. Conventional analytical methods are often considered insufficient, making numerical and heuristic approaches more suitable. Swarm intelligence, which mimics the collective behavior of natural systems such as bird flocks or ant colonies, offers flexible solutions for optimization problems [14]. Algorithms such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) have proven effective in exploring search spaces without relying on derivatives or strict mathematical conditions [20]. Consequently, swarm-intelligence-based approaches provide innovative and adaptive solutions to the challenges of fixed point approximation in quasilinear metric spaces.

In this study, we investigate the existence of fixed points for mappings defined on quasilinear metric spaces and explore the approximation of fixed points using swarm intelligence methods. The primary focus is on advancing fixed point theory within the framework of quasilinear metric spaces, representing a significant contribution to applied mathematics. The primary contributions of this study are as follows:

(1) To propose of new fixed point and contraction concepts for set-valued mappings in quasilinear metric spaces, together with corresponding existence results.
(2) To derive of sufficient conditions guaranteeing the existence of fixed points in quasilinear metric spaces for set-valued functions.
(3) To develop of a swarm intelligence–based algorithm for approximating fixed points of set-valued functions in quasilinear metric spaces.
(4) To evaluate of the proposed approach through three representative numerical examples.

The remainder of this article is organized as follows. After the introduction, the second section presents the theoretical background required for the analysis, including fundamental concepts of fixed point theory in quasilinear metric spaces and the formulation of fixed point problems for set-valued mappings. This section establishes the analytical foundation for the subsequent results.

The third section introduces swarm intelligence–based optimization methods for fixed point approximation and describes the implementation of five different algorithms within a unified framework. Rather than focusing on a single method, this section emphasizes the comparative aspects of the algorithms and outlines the common adaptation strategy employed to address fixed point problems in quasilinear metric spaces.

The fourth section presents the results, in which a new fixed point theorem for the considered space is introduced. This section also discusses three illustrative examples. For each example, the proposed fixed point theorem is first established theoretically, and the corresponding fixed points are then approximated using swarm intelligence algorithms. Numerical results are provided to demonstrate the effectiveness of the proposed theorem and to evaluate the performance of the swarm intelligence methods in terms of convergence behavior and approximation accuracy.

2. PRELIMINARIES

In this section, we discuss the theoretical foundations that support the results presented in the subsequent sections.

Definition 2.1: [18] *Let* $T:(X,d)\to(X,d)$ *be a mapping. The mapping* T *is said to be* Lipschitz *if there exists a constant* $c>0$*, called the* Lipschitz constant *of* T*, such that*

$$d(T(u),T(v))\leq c\,d(u,v),\quad \forall u,v\in X.$$

If $c<1$*, then the Lipschitz mapping* T *is called a* contraction*.*

This notion of contraction plays a central role in Banach's approach to fixed point theory and is used to establish the existence and uniqueness of fixed points. We recall the Banach Contraction Principle below.

Theorem 2.2: [18] *Let X be a metric space and let $T : X \to X$ be a contraction mapping. Then T has exactly one fixed point.*

After discussing the fixed point principle for single-valued mappings, we now turn to the set-valued case. To this end, we first introduce distance notions between sets, which will serve as essential tools in the subsequent analysis.

In what follows, $\mathscr{P}_0(X)$ denotes the collection of all nonempty subsets of X.

Definition 2.3: [6] *Let (X,d) be a metric space. The distance function $\mathscr{H}$ with*

$$\mathscr{H} : \mathscr{P}_0(X) \times \mathscr{P}_0(X) \to \mathbb{R},$$

is defined as follows. For all $A,B \in \mathscr{P}_0(X)$, the function

$$\mathscr{H}_+(A,B) = \sup_{x\in B} d(x,A) = \sup_{x\in B} \inf_{y\in A} d(x,y)$$

is called the upper Hausdorff distance*, and*

$$\mathscr{H}_-(A,B) = \sup_{x\in A} d(x,B) = \sup_{x\in A} \inf_{y\in B} d(x,y)$$

is called the lower Hausdorff distance*. The* Hausdorff distance *between A and B is defined by*

$$\mathscr{H}(A,B) = \max\{\mathscr{H}_+(A,B), \mathscr{H}_-(A,B)\}.$$

Proposition 2.4: *Let (X,d) be a metric space and let $\mathscr{CB}(X)$ denote the collection of all nonempty closed bounded subsets of X. Then the Hausdorff distance $\mathscr{H}$ on $\mathscr{CB}(X)$ is a metric.*

Proof. Let $A,B,C \in \mathscr{CB}(X)$. We verify that $\mathscr{H}(A,B)$ satisfies the axioms of a metric.

(1) **Non-negativity.** Since $d(x,y) \geq 0$ for all $x,y \in X$, it follows that

$$\mathscr{H}_+(A,B) \geq 0 \quad \text{and} \quad \mathscr{H}_-(A,B) \geq 0.$$

Hence,

$$\mathscr{H}(A,B) = \max\{\mathscr{H}_+(A,B), \mathscr{H}_-(A,B)\} \geq 0.$$

(2) **Identity of indiscernibles.** If $A = B$, then

$$\mathscr{H}_+(A,B) = \sup_{x\in B} d(x,A) = 0 \quad \text{and} \quad \mathscr{H}_-(A,B) = \sup_{x\in A} d(x,B) = 0.$$

Therefore,

$$\mathscr{H}(A,B) = 0.$$

Conversely, if $\mathscr{H}(A,B) = 0$, then every point of A lies arbitrarily close to B and vice versa, which implies $A = B$ since both sets are compact.

(3) **Symmetry.** By definition,

$$\mathscr{H}_{+}(A,B) = \sup_{x \in B} d(x,A) = \mathscr{H}_{-}(B,A),$$

and

$$\mathscr{H}_{-}(A,B) = \sup_{x \in A} d(x,B) = \mathscr{H}_{+}(B,A).$$

Thus,

$$\mathscr{H}(A,B) = \mathscr{H}(B,A).$$

(4) **Triangle inequality.** By the triangle inequality of d, for all $a \in A$, $b \in B$, and $c \in C$,

$$d(a,b) \leq d(a,c) + d(c,b).$$

Hence,

$$\inf_{b \in B} d(a,b) \leq d(a,c) + \inf_{b \in B} d(c,b) \leq d(a,c) + \mathscr{H}_{+}(C,B).$$

Taking the infimum over $c \in C$ and then the supremum over $a \in A$, we obtain

$$\mathscr{H}_{+}(A,B) \leq \mathscr{H}_{+}(A,C) + \mathscr{H}_{+}(C,B).$$

Similarly,

$$\mathscr{H}_{-}(A,B) \leq \mathscr{H}_{-}(A,C) + \mathscr{H}_{-}(C,B).$$

Therefore,

$$\mathscr{H}(A,B) \leq \mathscr{H}(A,C) + \mathscr{H}(C,B).$$

Thus, $\mathscr{H}$ satisfies all metric axioms and is a metric on $\mathscr{C}\mathscr{B}(X)$. □

Theorem 2.5: *If the metric space (X,d) is complete, then the metric space $(\mathscr{C}\mathscr{B}(X), \mathscr{H})$ is also complete.*

We now consider fixed point results for set-valued mappings. In particular, we focus on the fixed point theorem introduced by Nadler[9]. Prior to presenting the theorem, we introduce the notion of a fixed point for set-valued functions.

Definition 2.6: [9] *Let (X,d) be a metric space and let $F : X \to \mathscr{CB}(X)$, x is a fixed point of $F(x)$ if $x \in F(x)$.*

Definition 2.7: [9] *Let (X,d) be metric spaces and $E \subset X$. The set-valued function $F : E \to \mathscr{CB}(X)$ is a lipschitz if*

$$\mathscr{H}(F(x),F(y)) \leq \alpha d(x,y), \forall x,y \in E,$$

where $\alpha > 0$ is a lipschitz constant. If $\alpha < 1$ than F is a contraction.

Example 2.8: *let $I = [0,1] \subset \mathbb{R}$ and let $f : I \to I$ be a mapping defined by*

$$f(x) = \begin{cases} \frac{1}{2}x + \frac{1}{2}, & 0 \leq x \leq \frac{1}{2}, \\ -\frac{1}{2}x + 1, & \frac{1}{2} < x \leq 1. \end{cases}$$

Define a set-valued mapping $F : I \to \mathscr{P}_0(I)$ by

$$F(x) = \{0\} \cup \{f(x)\}.$$

Take arbitrary $x,y \in I$ such that

$$F(x) = \{0\} \cup \{f(x)\} = \{0, f(x)\} \quad \textit{and} \quad F(y) = \{0\} \cup \{f(y)\} = \{0, f(y)\}.$$

Then we obtain

$$\begin{aligned} \mathscr{H}_+(F(x),F(y)) &= \sup_{a \in F(y)} \inf_{b \in F(x)} d(a,b) \\ &= \sup\{0, |f(x) - f(y)|\} = |f(x) - f(y)|, \end{aligned}$$

and

$$\begin{aligned} \mathscr{H}_-(F(y),F(x)) &= \sup_{a \in F(x)} \inf_{b \in F(y)} d(a,b) \\ &= \sup\{0, |f(x) - f(y)|\} = |f(x) - f(y)|. \end{aligned}$$

Hence,

$$\mathscr{H}(F(x),F(y)) = \max\{\mathscr{H}_+(F(x),F(y)), \mathscr{H}_-(F(x),F(y))\} = |f(x) - f(y)|.$$

Next, we determine the Lipschitz constant.

If $x,y \in \left[0, \frac{1}{2}\right]$, *then*

$$|f(x) - f(y)| = \left|\frac{1}{2}x + \frac{1}{2} - \left(\frac{1}{2}y + \frac{1}{2}\right)\right| = \frac{1}{2}|x-y|.$$

If $x,y \in \left(\frac{1}{2}, 1\right]$, *then*

$$|f(x) - f(y)| = \left|-\frac{1}{2}x + 1 - \left(-\frac{1}{2}y + 1\right)\right| = \frac{1}{2}|x-y|.$$

If $x \in \left[0, \frac{1}{2}\right]$ *and* $y \in \left(\frac{1}{2}, 1\right]$, *then*

$$|f(x) - f(y)| = \left|\frac{1}{2}x + \frac{1}{2} - \left(-\frac{1}{2}y + 1\right)\right| \leq \frac{1}{2}|x+y-1| \leq \frac{1}{2}|x-y|.$$

If $x \in \left(\frac{1}{2}, 1\right]$ *and* $y \in \left[0, \frac{1}{2}\right]$, *then*

$$|f(x) - f(y)| = \left|-\frac{1}{2}x + 1 - \left(\frac{1}{2}y + \frac{1}{2}\right)\right| \leq \frac{1}{2}|1-x-y| \leq \frac{1}{2}|x-y|.$$

From all four possible cases, it follows that

$$|f(x) - f(y)| \leq \frac{1}{2}|x-y|,$$

and therefore

$$\mathscr{H}(F(x), F(y)) = |f(x) - f(y)| \leq \frac{1}{2}|x-y|.$$

Thus, the Lipschitz constant is $\alpha = \frac{1}{2}$, *and hence the mapping* F *is a contraction.*

Theorem 2.9: [9] *Let* (X,d) *be a complete metric space and let* $F : X \to \mathscr{CB}(X)$ *be a set-valued contraction. Then* F *has a fixed point.*

3. METHODOLOGY

In this section, we describe the main problem addressed in this study and the methodological framework adopted to solve it. We begin by formulating the fixed point problem for a set-valued mapping, where a point x is said to be a fixed point if $x \in F(x)$. Although this definition is mathematically well established, directly verifying or computing such fixed points is challenging from a computational perspective, particularly when using optimization-based algorithms. The main difficulty arises from the set-valued nature of $F(x)$, which prevents a straightforward evaluation of equality or residual error as in single-valued fixed point problems.

To address this issue, we reformulate the fixed point problem as an optimization problem. Instead of directly enforcing the condition $x \in F(x)$, we define an objective function based on the distance between x and the elements of the set $F(x)$. For each candidate solution x, the minimum distance between x and the members of $F(x)$ is computed, and the optimization process seeks to minimize this distance. In this formulation, a fixed point is approximated by a point for which the distance value approaches zero. This transformation enables the fixed point problem for set-valued mappings to be effectively handled using population-based optimization algorithms. A more detailed construction of the objective function and its practical implementation are presented in the Results section through three representative examples.

In this methodology, several metaheuristic optimization algorithms are employed, Seagull Optimization Algorithm (SOA), Nu Modified Seagull Optimization Algorithm (SOA), Particle Swarm optimization Algorithm (PSO), Whale Optimization Algorithm (WOA), Grey Wolf Optimization Algorithm (GWO). However, this section does not aim to provide a detailed description of each algorithm. Instead, we briefly introduce these algorithms and cite the relevant literature, focusing on their role as optimization tools for minimizing the proposed distance-based objective function.

To evaluate the performance of the proposed approach and the comparative algorithms, the Mean Squared Error (MSE) is adopted as the primary evaluation metric. The MSE provides a quantitative measure of the discrepancy between the candidate solutions and the fixed point condition induced by the set-valued mapping, allowing for a consistent and fair comparison across all experiments.

To provide a concise overview of the experimental procedure, a flowchart summarizing the main stages of the study is presented in Figure 1. The flowchart illustrates the sequential steps of the experimental workflow, beginning with the definition of the objective and fitness functions, followed by the selection of optimization algorithms and parameter settings. The algorithms are then executed under a uniform termination condition defined by a maximum number of iterations. Finally, the generated solutions are evaluated using the MSE metric, and the process is repeated over multiple trials to ensure robustness. This visualization clarifies the overall experimental pipeline and complements the methodological details presented in this section.

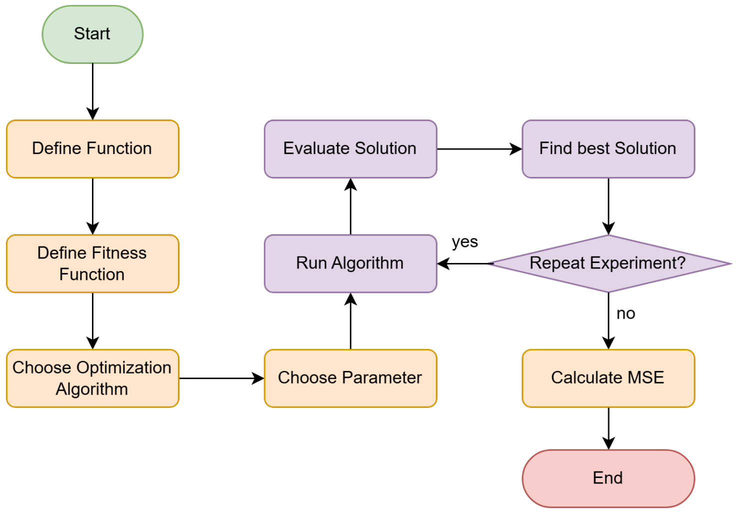


FIGURE 1. Research Steps

3.1. Seagull Optimization Algorithm. The Seagull Optimization Algorithm (SOA), proposed by Dhiman and Kumar [7], is a population-based metaheuristic inspired by the migratory and hunting behaviors of seagulls. SOA balances exploration and exploitation through two main phases: diversification (migration) and intensification (attack).

During the diversification phase, collision avoidance and movement toward the best solution are modeled as

$$\vec{Z}_s = M \cdot \vec{Y}_s(t), \tag{1}$$

where

$$M = f_{rc} - t \cdot \frac{f_{rc}}{Max_{iter}}. \tag{2}$$

The direction toward the best seagull is defined by

$$\vec{D}_s = E \cdot (\vec{Y}_{fs} - \vec{Y}_s(t)), \quad E = 2M^2 \cdot Rand. \tag{3}$$

Seagulls adjust their proximity based on the best position:

$$\vec{P}_s = |\vec{Z}_s + \vec{D}_s| \tag{4}$$

In the intensification phase, seagulls follow a spiral attack pattern:

$$x = r\cos\theta, \quad y = r\sin\theta, \quad z = r\theta, \tag{5}$$

with

$$r = \nu e^{\omega_s \theta}. \tag{6}$$

The velocity of each seagull $\vec{V}_s$ is updated using:

$$\vec{V}_s(t+1) = \vec{P}_s \cdot x \cdot y \cdot z \tag{7}$$

The position update rule is given by

$$\vec{Y}_s(t+1) = \vec{P}_s \cdot xyz + \vec{Y}_{fs}(t). \tag{8}$$

The main variables used in SOA are:

- $\vec{Y}_s(t)$: position of the s-th seagull at iteration t.
- $\vec{Y}_s(t+1)$: updated position of the s-th seagull.
- $\vec{P}_s$: proximity vector combining collision avoidance and movement toward the best seagull.
- $\vec{Z}_s$: intermediate position vector after collision avoidance.
- $\vec{D}_s$: direction vector toward the best seagull.
- $\vec{Y}_{fs}$: position of the best-performing seagull.
- M: motion coefficient for collision avoidance.
- E: coefficient controlling movement toward the best seagull.
- f_{rc}: initial motion coefficient used in computing M.
- Max_{iter}: maximum number of iterations.
- t: current iteration.
- $Rand$: uniformly distributed random number in $[0,1]$.
- x, y, z: coordinates for the spiral attack.
- θ: control parameter for spiral movement.
- r: spiral radius.
- ν: spiral scaling parameter; dynamically updated in NuM-SOA.

- ω_s: constant controlling spiral expansion.
- $\vec{V}_s$: velocity of the s-th seagull.

3.2. Nu Modified Seagull Optimization Algorithm. The modification of the Seagull Optimization Algorithm employed in this study follows our previously developed Nu-modified Seagull Optimization Algorithm (NuM-SOA), which has been submitted for publication, preprint [13]. In this work, the algorithm is adopted without redefining its core structure, and only its implementation details are summarized for completeness. As introduced in [13], the control parameter ν is updated dynamically to improve convergence:

$$\nu_t = \nu\left(1 - \frac{t}{Max_{iter}}\right). \tag{9}$$

The overall procedure of the NuM-SOA is summarized in Algorithm 1.

3.3. Whale Optimization Algorithm (WOA). WOA [15] uses encircling or spiral updating:

$$\begin{cases} \vec{X_W}(t+1) = \vec{X}^*(t) - A\vec{D}_W & \text{if } p_{ran} < 0,5 \\ \vec{X}_W(t+1) = \vec{D}'_W e^{b\ell} \cdot \cos(2\pi\ell) + \vec{X}^*(t) & \text{if } p_{ran} \geq 0,5 \end{cases} \tag{10}$$

where

$$\vec{D_W} = |C\vec{X}^*(t) - \vec{X}_W(t)|, A = 2ar - a, C = 2r$$

and

$$D'_W = |\vec{X}^*_W(t) - \vec{X}_W(t)|$$

The main variables used are:

- $\vec{X}_W$: position of the W-th whale in the search space.
- $\vec{X}^*_W$: the best solution (whale) found so far.
- A, C: coefficient vectors controlling the movement and encircling behavior.
- b: spiral parameter controlling the shape of the logarithmic spiral.
- ℓ: random number in $[-1,1]$ used in the spiral updating.
- a: a number linearly decreased from 2 to 0 over the course of iterations
- p_{ran}: probability parameter used to switch between the encircling and spiral updating mechanisms, randomly generated in the interval $[0,1]$. .

Algorithm 1 Nu Modified in Seagull Optimization Algorithm (NuM-SOA)

Input: Population of seagulls $\bar{Y}_s$, $s = 1,\dots,n$; Maximum iterations Max_{itr}
Output: Global best seagull position $\bar{Y}_{fs}$
Initialize population of n seagulls $\bar{Y}_s$
Initialize control parameters M and E
Set $f_c \leftarrow 2$, $\nu \leftarrow 0.03$, $w \leftarrow 0.02$
for $t \leftarrow 1$ **to** Max_{itr} **do**
  Evaluate fitness value of each seagull
  Find the best seagull $\bar{Y}_{fs}$ based on fitness
  for each seagull s **do**
    Generate control parameter $\theta \in [0, 2\pi]$
    Compute ν_t using Eq. (9)
    Generate spiral behavior using Eq. (6)
    Compute distance $\bar{D}_s$ using Eq. (3)
    Update speed using Eq. (7)
    Update position using Eq. (8)
  end for
  Recalculate fitness of all seagulls
  Update $\bar{Y}_s$ based on best fitness
end for
return $\bar{Y}_{fs}$

3.4. Grey Wolf Optimizer (GWO).

GWO [16] updates positions based on alpha ($\vec{X}_\alpha$), beta ($\vec{X}_\beta$), and delta ($\vec{X}_\delta$) wolves:

$$A_i = 2a_i \cdot r_1 - a_i, \qquad C_i = 2 \cdot r_2, \tag{11}$$

$$\vec{D}_\alpha = |C_1 \cdot \vec{X}_\alpha - \vec{X}_i|, \qquad \vec{X}_a = \vec{X}_\alpha - A_1 \cdot \vec{D}_\alpha, \tag{12}$$

$$\vec{D}_\beta = |C_2 \cdot \vec{X}_\beta - \vec{X}_i|, \qquad \vec{X}_b = \vec{X}_\beta - A_2 \cdot \vec{D}_\beta, \tag{13}$$

$$\vec{D}_\delta = |C_3 \cdot \vec{X}_\delta - \vec{X}_i|, \qquad \vec{X}_c = \vec{X}_\delta - A_3 \cdot \vec{D}_\delta, \tag{14}$$

$$\vec{X}_G(t+1) = \frac{\vec{X}_a + \vec{X}_b + \vec{X}_c}{3}. \tag{15}$$

The main variables in GWO are:

- $\vec{X}_G$: position of the G-th grey wolf.
- $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$: positions of the three best wolves guiding the rest.
- A_i, C_i with $i = 1,2,3$: coefficient vectors controlling exploration and exploitation.
- $\vec{X}_a, \vec{X}_b, \vec{X}_c$: intermediate positions computed relative to alpha, beta, and delta wolves.

3.5. Particle Swarm Optimization (PSO).

PSO [11] updates velocity and position as:

$$\vec{V}_P(t+1) = \omega_\rho \vec{V}_P(t) + c_1 r_1 (\vec{X}_P^p - \vec{X}_P(t)) + c_2 r_2 (\vec{G} - \vec{X}_P(t)) \tag{16}$$

$$\vec{X}_P(t+1) = \vec{X}_P(t) + \vec{V}_P(t+1). \tag{17}$$

The main variables in PSO are:

- $\vec{X}_P$: position of the P-th particle in the search space.
- $\vec{V}_P$: velocity of the P-th particle.
- $\vec{X}_P^p$: personal best position of the P-th particle.
- $\vec{X}^*$: global best position found by the swarm.
- ω_ρ: inertia weight controlling exploration/exploitation.
- c_1, c_2: acceleration coefficients for personal and social influence.
- r_1, r_2: random numbers in $[0,1]$.

3.6. Cross Validation and Parameter Selection.

To maximize the performance of all algorithms, each algorithm was subjected to a cross-validation procedure. The parameters considered in this process are listed in Table 1. All possible combinations of these parameters were evaluated, and each configuration was executed five independent times to reduce the influence of stochastic effects inherent in metaheuristic algorithms. The mean squared error (MSE) obtained from these runs was calculated and used as the performance indicator, providing a more reliable and representative assessment of each parameter setting.

The evaluation was carried out using three example problems, where for each example, the existence of a fixed point was first established before computing an approximate fixed point using the algorithms. After selecting the optimal parameters from cross-validation, all algorithms were subsequently run using 2500 iterations for the final evaluation, ensuring a consistent and fair comparison across all methods. For Problem 2, both the MSE and the number of iterations required to reach zero error were recorded to provide additional insight into the convergence behavior of each algorithm.

Algorithm	Parameters	value
SOA	v	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1
	ω_s	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1
	Num of seagulls(s)	10,20, 30, 40, 50
Num-SOA	v	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1
	ω_s	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1
	Num of seagulls(s)	10,20, 30, 40, 50
WOA	Num of whales(W)	10, 20, 30, 40, 50,60,70,80
GWO	Num of wolfs(G)	10, 20, 30, 40, 50,60,70,80
PSO	ω_ρ	0.2, 0.4, 0.6, 0.8
	c_1	1,1.5,2
	c_2	1,1.5,2
	Particle values(P)	10,20,30,40,50

TABLE 1. List of Parameters

4. MAIN RESULTS

In this section, we discuss and analyze the results of the experiments conducted on the three fixed-point problems. We begin by exploring the optimal parameter settings for both the original SOA and the modified NuM-SOA across different problem instances. Finally, we present a comparative analysis of the proposed algorithm against competitive fixed-point methods, including state-of-the-art approaches, the Bat Algorithm as a representative of swarm intelligence techniques, and the original SOA.

4.1. Fixed Points in Quasilinear Metric Spaces. The following definition of fixed points is focused on collections of convex, closed, and compact sets over vector spaces.

Definition 4.1: *Let $\mathscr{KC}(\mathbb{R}^n)$ denote the collection of nonempty, convex, compact, and closed subsets of the Euclidean space $\mathbb{R}^n$. Define a set-valued mapping $F : \mathbb{R}^n \to \mathscr{KC}(\mathbb{R}^n)$. A point $x \in \mathbb{R}^n$ is called a* fixed point *of F if*

$$x \in F(x).$$

The set of all fixed points of F is denoted by

$$Fix(F) = \{x \in \mathbb{R}^n \mid x \text{ is a fixed point of } F\}.$$

4.2. Properties of Fixed Points in Quasilinear Metric Spaces. To adapt to the space considered in this study, a new notion of contraction is introduced.

Definition 4.2: *Let $\mathscr{KC}(\mathbb{R}^n)$ be the collection of nonempty, convex, compact, and closed subsets of $\mathbb{R}^n$. A mapping $F : \mathbb{R}^n \to \mathscr{KC}(\mathbb{R}^n)$ is called a* set-valued Lipschitz mapping *if*

$$\mathscr{H}(F(x), F(y)) \leq \beta \|x - y\|, \quad \forall x, y \in \mathbb{R}^n.$$

The constant β is called the Lipschitz constant. *If $\beta < 1$, then F is called a* set-valued contraction mapping.

The following theorem provides a sufficient condition for the existence of a fixed point of a set-valued mapping in a quasilinear metric space. Since $\mathbb{R}^n$ is complete, it follows that $(\mathscr{KC}(\mathbb{R}^n), \mathscr{H})$ is also a complete metric space.

Theorem 4.3: *Let $(\mathscr{KC}(\mathbb{R}^n), \mathscr{H})$ be a quasilinear metric space and let $F : \mathbb{R}^n \to \mathscr{KC}(\mathbb{R}^n)$. If F is a contraction, then F has a fixed point.*

Proof. Take an arbitrary point $x_0 \in X$. Since $F(x_0)$ is bounded and nonempty, the quantity $d(x_0, F(x_0))$ is also bounded. Let $0 \leq k < 1$ be the contraction constant of F. Choose $r > 0$ such that

$$d(x_0, F(x_0)) < (1 - k)r.$$

Construct a sequence $(x_n) \subset \mathbb{R}^n$ with $x_n \in F(x_{n-1})$. We will show that this sequence lies in the closed ball $B(x_0, r)$. Choose $x_1 \in F(x_0)$ such that

$$\|x_1 - x_0\| < (1 - k)r.$$

By induction, select $x_n \in F(x_{n-1})$. Since F is a contraction,

$$\mathscr{H}(F(x_n),F(x_{n-1})) \leq k\|x_n - x_{n-1}\|,$$

there exists $x_{n+1} \in F(x_n)$ such that

$$\|x_{n+1} - x_n\| < k\|x_n - x_{n-1}\|.$$

From the pattern obtained starting with x_1, we have

$$\|x_{n+1} - x_n\| < k^n(1-k)r \quad \text{for all } n \geq 0.$$

Hence,

$$\begin{aligned}
\|x_n - x_0\| &\leq \|x_1 - x_0\| + \sum_{i=1}^{n-1} \|x_{i+1} - x_i\| \\
&< (1-k)r + \sum_{i=1}^{n-1} k^i(1-k)r \\
&= (1-k)r(1+k+\cdots+k^{n-1}) \\
&= (1-k)r\left(\frac{1-k^n}{1-k}\right) = r(1-k^n) < r.
\end{aligned}$$

By Definition, it follows that $x_n \in B(x_0,r)$, and thus the sequence (x_n) is bounded.

Next, we show that (x_n) is a Cauchy sequence. For $m > n$, we obtain

$$\|x_m - x_n\| \leq \sum_{i=n}^{m-1} \|x_{i+1} - x_i\| < \sum_{i=n}^{\infty} k^i(1-k)r = k^n(1-k)r\sum_{j=0}^{\infty} k^j = k^n r.$$

Since $0 \leq k < 1$, we have $k^n r \to 0$ as $n \to \infty$, and hence (x_n) is a Cauchy sequence.

Because X is complete, the sequence (x_n) converges to some $x \in X$. Since $x \in B(x_0,r)$, we obtain

$$\mathscr{H}(F(x_n),F(x)) \leq k\|x_n - x\| \to 0.$$

For each n, we have $x_{n+1} \in F(x_n)$, which implies

$$d(x_{n+1},F(x)) \leq \mathscr{H}(F(x_n),F(x)) \to 0.$$

Since $x_{n+1} \to x$ and $F(x)$ is closed, it follows that $x \in F(x)$, i.e.,

$$d(x,F(x)) = 0.$$

Therefore, x is a fixed point of F. □

5. Examples of Fixed Points in Quasilinear Metric Spaces

5.1. Example 1. Let $F : [-5,5] \to \mathscr{KC}(\mathbb{R})$ be defined by

$$F(x) = \left\{ \frac{x^2}{50} \right\} \cup \{\cos(x) + 1\} = \left\{ p, q | p = \frac{x^2}{50} \text{ and } q = \cos(x) + 1 \right\}.$$

for each $x \in [-5,5]$. We first show that the mapping F admits a fixed point. To prove the existence of a fixed point, we verify whether F is a contraction. Let $x, y \in [-5,5]$. Then we obtain

$$\begin{aligned}
\mathscr{H}_+(F(x),F(y)) &= \sup_{a\in F(y)} \inf_{b\in F(x)} \|a-b\| \\
&= \sup \left\{ \inf \left\{ \frac{|y^2-x^2|}{50}, \left| \frac{y^2}{50} - \cos(x) - 1 \right| \right\}, \right. \\
&\qquad \left. \inf \left\{ \left| \cos(y) + 1 - \frac{x^2}{50} \right|, |\cos(y) - \cos(x)| \right\} \right\} \\
&\le \frac{|y^2-x^2|}{50} = \frac{|x+y||x-y|}{50} \le \frac{10}{50}|x-y| = \frac{1}{5}|x-y|.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathscr{H}_-(F(x),F(y)) &= \sup_{a\in F(x)} \inf_{b\in F(y)} \|a-b\| \\
&\le \frac{|x^2-y^2|}{50} = \frac{|x+y||x-y|}{50} \le \frac{1}{5}|x-y|.
\end{aligned}$$

Hence,

$$\mathscr{H}(F(x),F(y)) \le \frac{1}{5}|x-y|.$$

By Definition 4.2, the mapping F is a contraction with Lipschitz constant $\frac{1}{5}$. Therefore, by Theorem 4.3, the mapping F has a fixed point.

After establishing the existence of a fixed point, we proceed to approximate the fixed point using swarm intelligence methods. As illustrated in Figure 2, the set-valued mapping $F(x)$ contains two elements, denoted by a and b.

According to the definition of a fixed point for a set-valued function, a point x is a fixed point if $x \in F(x)$. Therefore, a candidate solution must be sufficiently close to at least one element of the set $F(x)$. Based on this observation, the fitness function is defined as the minimum distance between x and the elements of $F(x)$:

$$f_{\text{fit}} = \min\{|x-a|,\ |x-b|\}. \tag{18}$$

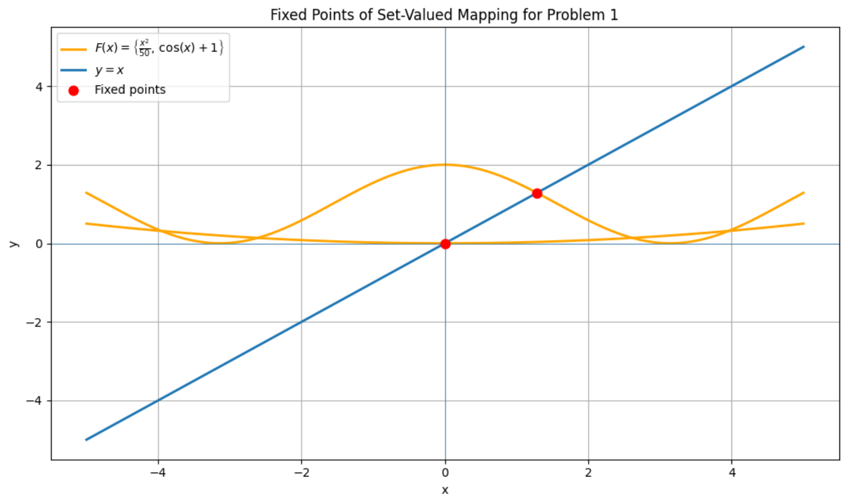


FIGURE 2. Problem 1

In this problem, all algorithms are evaluated using cross-validation. The corresponding results are summarized in Figure 5. Darker colors indicate better solution quality for the corresponding parameter settings, while white regions correspond to parameter combinations for which the algorithm achieves zero error. Based on the cross-validation results, the optimal parameter values are reported in Table 2.

From Figure 5, SOA, Num-SOA, and PSO show better performance when using a larger population size. This behavior is expected, as a larger number of particles improves population diversity and enhances the balance between exploration and exploitation, reducing the risk of premature convergence.

SOA shows better performance when the parameter ν is in the range of 0.4–0.7 and when ω_s is small. A moderate value of ν allows SOA to maintain sufficient exploration while still converging effectively, whereas a smaller ω_s reduces excessive oscillatory movements, leading to more stable convergence toward optimal solutions.

Num-SOA outperforms SOA and achieves better performance when ν is larger and ω_s is smaller. The numerical enhancement in Num-SOA improves search precision, enabling the algorithm to benefit from stronger exploration (larger ν) while maintaining convergence stability through a reduced step size controlled by ω_s.

PSO exhibits better performance when the inertia weight ω_ρ is small, with $c_1 = 1.5$ and $c_2 = 1$. A smaller inertia weight encourages exploitation by limiting particle velocity, while the chosen acceleration coefficients provide a balanced influence between personal experience and global best information.

WOA and GWO generally perform better with a larger number of particles due to increased search coverage of the solution space. However, this improvement is not strictly monotonic. WOA achieves better performance with 70 particles than with 80 particles, which may be attributed to excessive exploration causing slower convergence. In contrast, GWO continues to benefit from increased population size, as its hierarchical leadership mechanism can effectively manage larger populations and guide the search process more efficiently. From Table 3, all

Algorithm	Parameters	value
SOA	ν	0.5
	ω_s	0.1
	Num of seagulls(s)	50
Num-SOA	ν	0.3
	ω_s	0.4
	Num of seagulls(s)	50
WOA	Num of whales(W)	70
GWO	Num of wolfs(G)	80
PSO	ω_ρ	0.2
	c_1	1.5
	c_2	1
	Particle values(P)	50

TABLE 2. Chosen Parameter for Problem 1

algorithms successfully identify all fixed points of Problem 1, indicating that each method is capable of solving the problem effectively. As expected, Num-SOA demonstrates better performance than SOA, WOA, and GWO, achieving significantly smaller best error and MSE values. Among the latter three algorithms, WOA performs slightly better than GWO and SOA, while GWO and SOA exhibit comparable performance, indicating similar convergence accuracy.

The highest overall performance is achieved by PSO. Although its best error is only marginally better than that of Num-SOA, PSO exhibits a substantially lower MSE, reflecting a more stable and consistent convergence behavior across multiple runs. This stability can be attributed to the velocity update mechanism in PSO, which effectively balances exploration and exploitation, leading to reduced variability in the obtained solutions. In contrast, while Num-SOA attains very high accuracy in terms of best error, its slightly higher MSE suggests greater sensitivity to stochastic variations during the search process.

Algorithm	Best x	$F(x)$	Best Error	MSE
SOA	0.0000	$\{0.000, 2.000\}$	0.000	0.000
	1.283	$\{0.0329, 1.283\}$	$1.33 \cdot 10^{-6}$	$5.597 \cdot 10^{-10}$
Num-SOA	0.0000	$\{0.000, 2.000\}$	0.000	0.000
	1.283	$\{0.0329, 1.283\}$	$1.33 \cdot 10^{-15}$	$5.139 \cdot 10^{-27}$
WOA	0.0000	$\{0.000, 2.000\}$	0.000	0.000
	1.283	$\{0.0329, 1.283\}$	$2.75 \cdot 10^{-10}$	$1.983 \cdot 10^{-17}$
GWO	0.0000	$\{0.000, 2.000\}$	0.000	0.000
	1.283	$\{0.0329, 1.283\}$	$1.25 \cdot 10^{-9}$	$1.629 \cdot 10^{-12}$
PSO	0.0000	$\{0.000, 2.000\}$	0.000	0.000
	1.283	$\{0.0329, 1.283\}$	$2.22 \cdot 10^{-16}$	$4.930 \cdot 10^{-32}$

TABLE 3. Performance comparison of different algorithms on Problem 1

5.2. **Example 2.** Let $F : [\frac{1}{2}, \frac{3}{2}] \to \mathscr{K}\mathscr{C}(\mathbb{R})$ be defined by

$$F(x) = [\sin(x), \cos(x)].$$

for each $x \in [\frac{1}{2}, \frac{3}{2}]$. We first prove the existence of a fixed point by showing that F is a contraction. Let $x, y \in [\frac{1}{2}, \frac{3}{2}]$. Then

$$\begin{aligned}\mathscr{H}_+(F(x), F(y)) &= \sup_{a \in F(x)} \inf_{b \in F(y)} \|a - b\| \\ &\leq \sup\{|\sin(x) - \sin(y)|, |\cos(x) - \cos(y)|\} \\ &= \sup\left\{\left|2\sin\left(\tfrac{x-y}{2}\right)\cos\left(\tfrac{x+y}{2}\right)\right|, \left|2\sin\left(\tfrac{x-y}{2}\right)\sin\left(\tfrac{x+y}{2}\right)\right|\right\}.\end{aligned}$$

Since $\sin\left(\frac{x+y}{2}\right)$ and $\cos\left(\frac{x+y}{2}\right)$ are strictly less than 1 on the interval $x, y \in [\frac{1}{2}, \frac{3}{2}]$, there exists a constant $k < 1$ such that

$$\sin\left(\tfrac{x+y}{2}\right), \cos\left(\tfrac{x+y}{2}\right) \leq k.$$

Using the inequality $\sin t < t$ for $t > 0$, we obtain

$$\mathscr{H}_+(F(x), F(y)) < k|x - y|.$$

By a similar argument,

$$\mathscr{H}_-(F(x), F(y)) < k|x - y|.$$

Thus,

$$\mathscr{H}(F(x), F(y)) < k|x - y|, \quad k < 1,$$

which shows that F is a contraction. Therefore, by Theorem 4.3, the mapping F admits a fixed point.

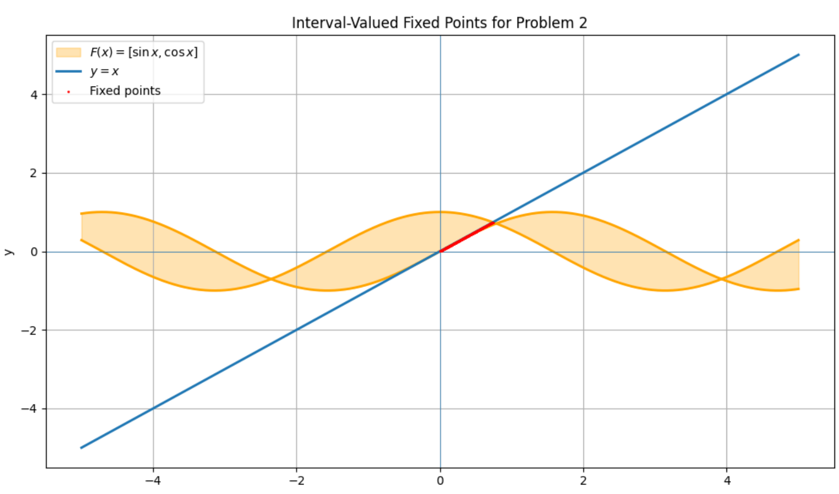


FIGURE 3. Problem 2

Based on Figure 3, the fixed point in this problem is very many. Since $F(x)$ is in interval, to acommodate the definition of fixed point of set-valued function, the fitness function is defined as

$$f_{fit} = \begin{cases} 0, & \text{if } x \leq b \text{ or } x \geq a \\ |x-b|, & \text{if } x \geq b \\ |x-a|, & \text{if } x \leq a \end{cases} \tag{19}$$

Since the set of all fixed points forms an interval, the objective in this problem is to determine the bounds of that interval. In this case, the error-based metric is no longer appropriate, as many candidate solutions immediately yield zero error. Therefore, a different evaluation criterion is introduced based on the number of iterations required for convergence. This metric allows us to assess and compare the convergence speed of the algorithms, identifying which method reaches the fixed-point interval more efficiently. As in the previous experiment, all

Algorithm	Parameters	value
SOA	v	0.4
	ω_s	0.1
	Num of seagulls(s)	50
Num-SOA	v	0.9
	ω_s	0.7
	Num of seagulls(s)	50
WOA	Num of whales(W)	80
GWO	Num of wolfs(G)	80
PSO	ω_p	0.2
	c_1	1
	c_2	1
	Particle values(P)	40

TABLE 4. Chosen Parameter for Problem 2

algorithms are evaluated using cross-validation. As shown in Figure 6, SOA achieves better performance when $\nu = 0.4$ and ω_s takes smaller values, whereas the opposite trend is observed for NuM-SOA, which performs better with larger values of both ν and ω_s. PSO favors smaller parameter values, while both WOA and GWO require larger population sizes to achieve optimal performance.

All algorithms are able to identify the boundary of the fixed-point interval, albeit with different convergence times. PSO exhibits the fastest convergence, achieving a very low average number of iterations. This behavior can be attributed to PSO's velocity–position update mechanism, which allows particles to move directly toward promising regions using both personal and global best information, resulting in rapid boundary detection.

In contrast, the remaining algorithms require more than 2000 iterations on average, with WOA converging the fastest among them, followed by SOA, GWO, and NuM-SOA. These algorithms rely more heavily on encircling, leader-following, or spiral-based movements, which tend to introduce oscillatory behavior near the boundary of the fixed-point interval. Consequently, additional iterations are required to stabilize the solution. Overall, these results demonstrate that PSO converges significantly faster than WOA, SOA, GWO, and NuM-SOA for boundary detection in interval-valued fixed-point problems.

Algorithm	Best x	$F(x)$	Minimum Iteration	Iteration Average
SOA	0.500	$[0.479, 0.878]$	2	2
	0.7390	$[0.674, 0.739]$	1006	2176.20
Num-SOA	0.500	$[0.479, 0.878]$	2	2
	0.7390	$[0.674, 0.739]$	2447	2495.55
WOA	0.500	$[0.479, 0.878]$	1	1
	0.7390	$[0.674, 0.739]$	1783	2211.25
GWO	0.500	$[0.479, 0.878]$	1	1
	0.7390	$[0.674, 0.739]$	176	2248.9
PSO	0.500	$[0.479, 0.878]$	1	2.1
	0.7390	$[0.674, 0.739]$	41	43.25

TABLE 5. Performance comparison of different algorithms on Problem 2

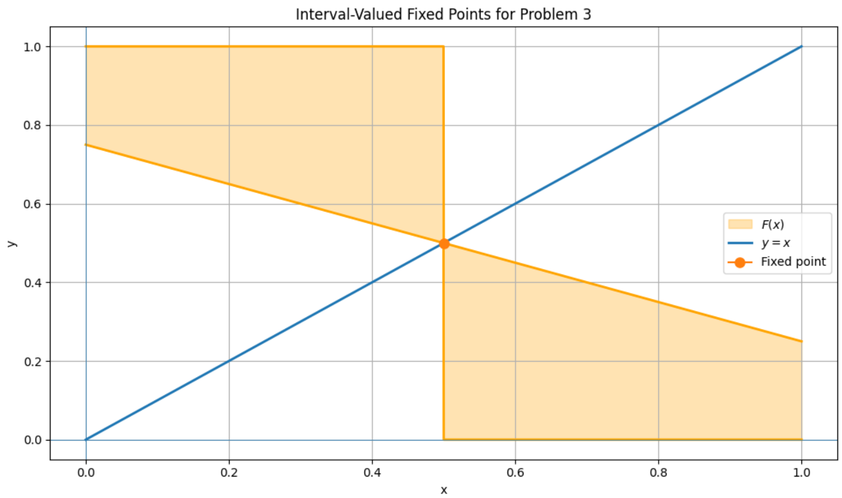


FIGURE 4. Problem 3

5.3. **Example 3.** Let $F:[0,1]\to\mathscr{K}\mathscr{C}(\mathbb{R})$ be defined by

$$F(x)=\begin{cases}\left[\frac{3}{4}-\frac{x}{2},\,1\right], & 0\le x\le\frac{1}{2},\\[2ex] \left[0,\,\frac{3}{4}-\frac{x}{2}\right], & \frac{1}{2}<x\le 1.\end{cases}$$

for each $x\in[0,1]$. We show that F has a fixed point by verifying that it is a contraction.

For both cases, let $x,y\in[0,1]$. Direct computation yields

$$\mathscr{H}_{+}(F(x),F(y))\le\frac{1}{2}|x-y|,\quad \mathscr{H}_{-}(F(x),F(y))\le\frac{1}{2}|x-y|.$$

Hence,

$$\mathscr{H}(F(x),F(y))\le\frac{1}{2}|x-y|.$$

By Definition 4.2, the mapping F is a contraction with Lipschitz constant $\frac{1}{2}<1$. Therefore, by Theorem 4.3, the mapping F has a fixed point.

Based on Figure 4, the mapping $F(x)$ is interval-valued. Therefore, a new fitness function must be constructed. Since the definition of a fixed point for a set-valued function is given by $x\in F(x)$, it is sufficient to verify that x lies within the corresponding interval. If we assume that $F(x)=[a,b]$, then the optimization problem can be formulated using Equation 20.

$$f_{fit} = \begin{cases} 0, & \text{if } x \leq b \text{ or } x \geq a \\ |x-b|, & \text{if } x \geq b \\ |x-a|, & \text{if } x \leq a \end{cases} \tag{20}$$

In this problem, all algorithms are evaluated using cross-validation. The corresponding results are summarized in Figure 7. From this we choose the best parameter listed on Table 6. SOA and Num-SOA both show better performance when v and ω_s is the smallest valued as possible with the biggest number of seagulls show best performance. PSO perform best when the particle number is 50 and with ω_p, r_1, r_2 as small as possible. Both WOA and GWO best perform when the number of population is 60 and 80.

Algorithm	Parameters	value
SOA	v	0.1
	ω_s	0.1
	Num of seagulls(s)	50
Num-SOA	v	0.1
	ω_s	0.1
	Num of seagulls(s)	50
WOA	Num of whales(W)	60
GWO	Num of wolfs(G)	80
PSO	ω_p	0.2
	c_1	1
	c_2	1
	Particle values(P)	50

TABLE 6. Chosen Parameter for Problem 3

All algorithms successfully converge to a single fixed point. Overall, NuM-SOA outperforms SOA, WOA, and GWO. SOA exhibits slightly better performance than WOA and GWO, while WOA and GWO show comparable results. PSO achieves the best performance, attaining zero error consistently. Although NuM-SOA is also able to reach zero error, PSO demonstrates higher consistency across runs. The superior performance of PSO can be attributed to its direct velocity–position update mechanism, which enables particles to rapidly converge toward the feasible region defined by the fixed-point condition $x \in F(x)$. In this problem, the objective landscape is simple and well-structured, allowing PSO to exploit global and personal best information efficiently, resulting in zero error with high consistency.

NuM-SOA outperforms the original SOA due to the introduction of the decay-based parameter ν, which improves the balance between exploration and exploitation. This mechanism allows NuM-SOA to avoid premature convergence and refine solutions more effectively near the fixed point, enabling it to also reach zero error, albeit with slightly lower consistency than PSO. SOA, WOA, and GWO rely more heavily on stochastic position-updating strategies and

Algorithm	Best x	$F(x)$	Best Error	MSE
SOA	0.500	$[0.500, 1.000]$	$7.13 \cdot 10^{-10}$	$5.436 \cdot 10^{-15}$
Num-SOA	0.500	$[0.500, 1.000]$	0.000	$5.547 \cdot 10^{-33}$
WOA	0.500	$[0.500, 1.000]$	$9.91 \cdot 10^{-9}$	$1.569 \cdot 10^{-12}$
GWO	0.500	$[0.500, 1.000]$	$1.64 \cdot 10^{-8}$	$1.471 \cdot 10^{-13}$
PSO	0.500	$[0.500, 1.000]$	0.000	0.000

TABLE 7. Performance comparison of different algorithms on Problem 3

collective behaviors, which can introduce small oscillations around the fixed point. While SOA benefits from directional movement and therefore performs better than WOA and GWO, the latter two exhibit similar performance due to their comparable encircling and leader-following dynamics, resulting in comparable error levels.

6. CONCLUSION

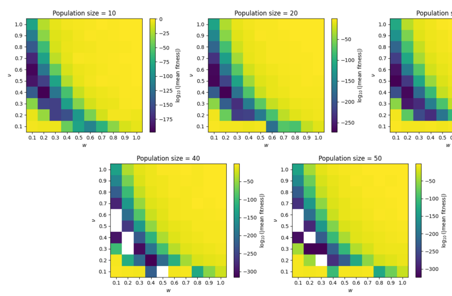
This study presented a unified analytical and computational framework for approximating fixed points of set-valued mappings in quasilinear metric spaces. From the analytical perspective, sufficient conditions for the existence of fixed points were established using set-valued contraction mappings under the Hausdorff metric. The completeness of the underlying metric space ensures the applicability of classical fixed-point arguments, providing a rigorous theoretical foundation for the numerical approximation procedures employed in this work.

On the computational side, swarm intelligence algorithms were systematically adapted to approximate fixed points by reformulating the fixed-point condition $x \in F(x)$ as an optimization problem. The design of the fitness functions was guided directly by analytical considerations. For mappings with isolated fixed points, error-based metrics derived from the distance to the set $F(x)$ were used, whereas for interval-valued fixed-point sets, convergence speed and boundary detection were adopted as evaluation criteria. This problem-dependent formulation ensures that the numerical objectives remain consistent with the underlying analytical structure of the fixed-point problem.

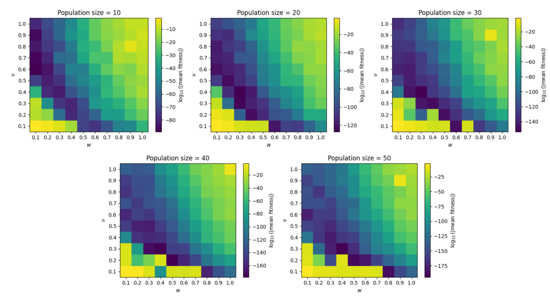
Three representative examples were investigated to illustrate the interaction between analytical properties and algorithmic behavior. In all cases, the contraction conditions were analytically verified, guaranteeing the existence of fixed points and explaining the observed convergence patterns. The numerical results demonstrate that all considered algorithms are capable of approximating the analytically guaranteed fixed points, while exhibiting different levels of accuracy, stability, and convergence speed depending on both algorithmic mechanisms and problem characteristics.

Among the tested methods, PSO consistently achieved the fastest convergence and the highest stability across different problem settings, particularly in interval-valued fixed-point problems. The proposed NuM-SOA showed clear improvements over the original SOA, indicating that the decay-based modification enhances the balance between exploration and exploitation, especially near analytically defined fixed-point regions. While SOA, WOA, and GWO also successfully converged, their reliance on collective stochastic movements resulted in slower convergence and higher variability in some cases.

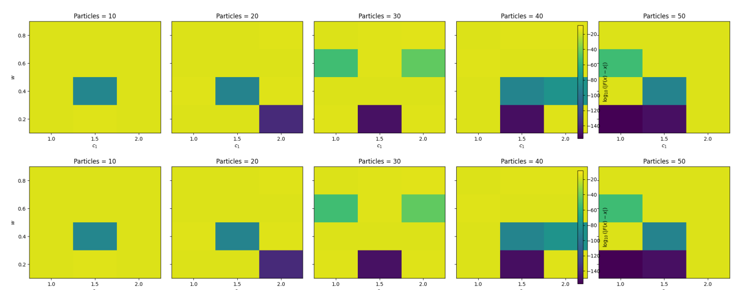
Overall, the results confirm that swarm intelligence algorithms, when supported by appropriate analytical assumptions, provide a robust and flexible approach for fixed-point approximation in quasilinear metric spaces. This work highlights the importance of integrating analytical fixed-point theory with computational optimization techniques, offering a practical methodology for problems where classical iterative schemes are difficult to apply. Future research may focus on extending this framework to non-contractive mappings, higher-dimensional set-valued operators, and adaptive swarm strategies, further strengthening the connection between theory and computation.



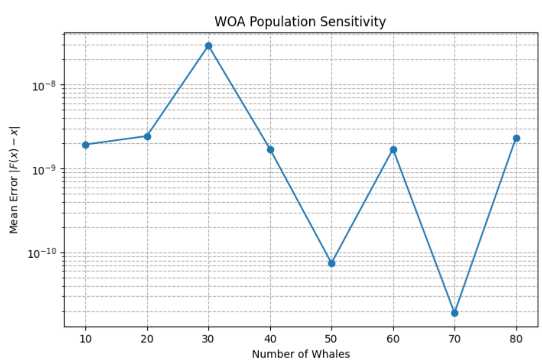
(A) SOA



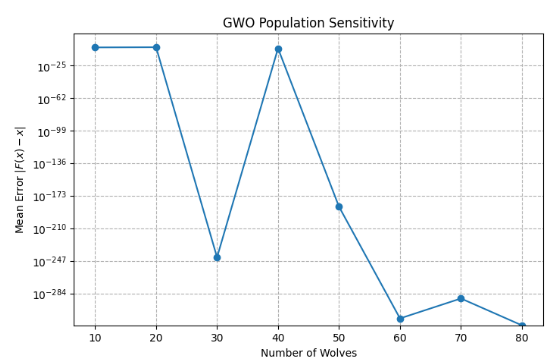
(B) NuM-SOA



(C) PSO

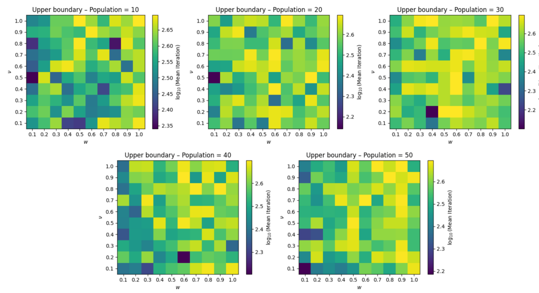


(D) WOA

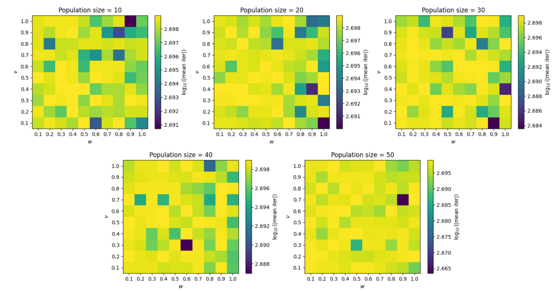


(E) GWO

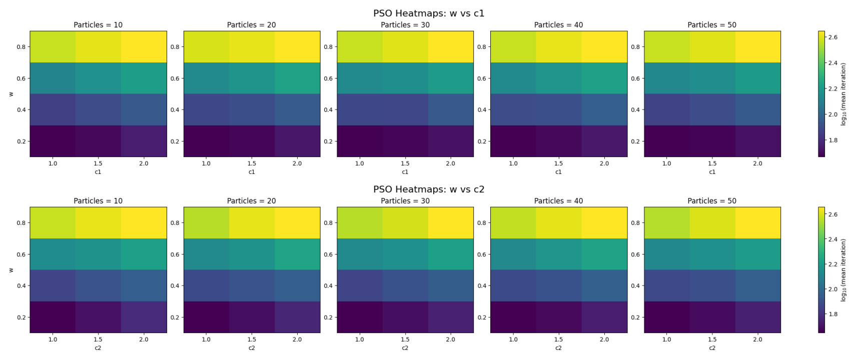
FIGURE 5. Cross-validation results for all algorithms on Problem 1.



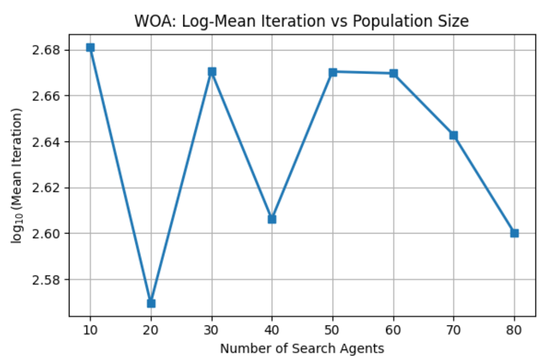
(A) SOA



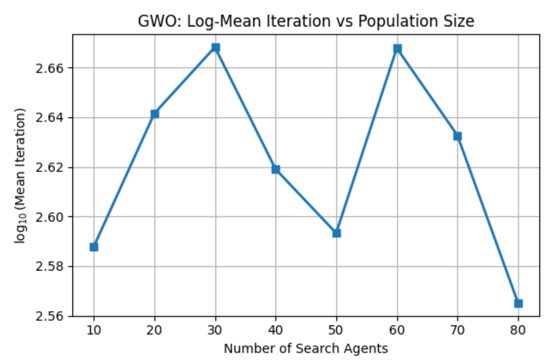
(B) NuM-SOA



(C) PSO

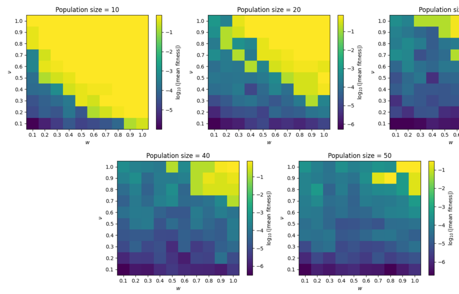


(D) WOA

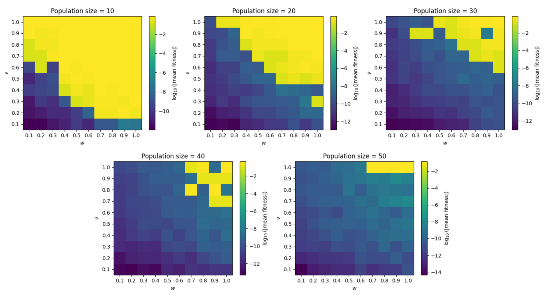


(E) GWO

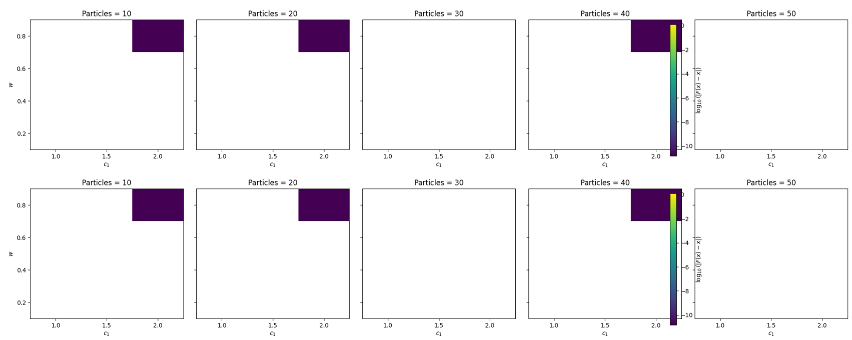
FIGURE 6. Cross-validation results for all algorithms on Problem 2.



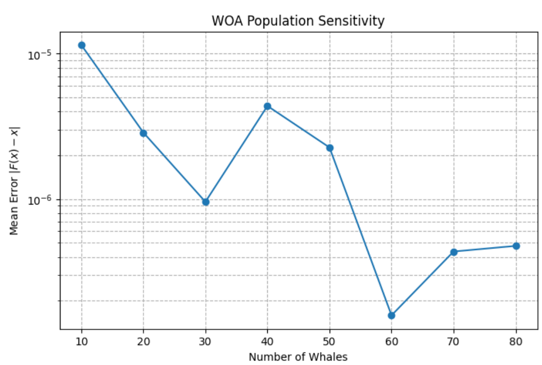
(A) SOA



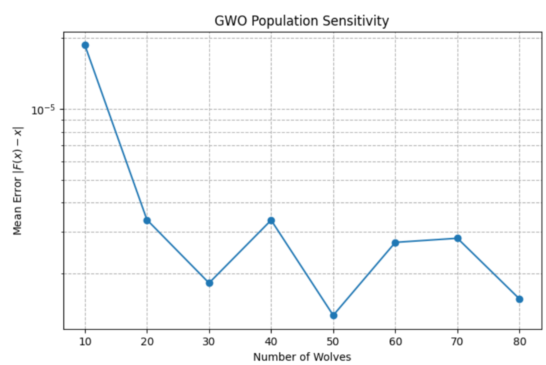
(B) NuM-SOA



(C) PSO



(D) WOA



(E) GWO

FIGURE 7. Cross-validation results for all algorithms on Problem 3.

Acknowledgment

This paper is funded by Professor Grant number 06871.1/UN10.F0901/B/PT/2025 of University of Brawijaya, Malang, East Java, Indonesia.

Conflict of Interests

The authors declare that there is no conflict of interests.

References

[1] S.M. Aseev, Quasilinear Operators and Their Application in the Theory of Multivalued Mappings, Trudy Mat. Inst. Steklov 167 (1985), 25–52.

[2] K.C. Border, Fixed Point Theorems with Applications to Economics and Game Theory, Cambridge University Press, 1985. https://doi.org/10.1017/CBO9780511625756.

[3] L.E.J. Brouwer, Über Abbildung Von Mannigfaltigkeiten, Math. Ann. 71 (1911), 97–115. https://doi.org/10.1007/BF01456931.

[4] P.L. Combettes, J.C. Pesquet, Fixed Point Strategies in Data Science, IEEE Trans. Signal Process. 69 (2021), 3878–3905. https://doi.org/10.1109/tsp.2021.3069677.

[5] Z. Denkowski, S. Migórski, N.S. Papageorgiou, Set-Valued Analysis, in: An Introduction to Nonlinear Analysis: Theory, Springer, Boston, 2003: pp. 405–516. https://doi.org/10.1007/978-1-4419-9158-4_4.

[6] M.M. Deza, E. Deza, Encyclopedia of Distances, in: Encyclopedia of Distances, Springer, Berlin, Heidelberg, 2009: pp. 1–583. https://doi.org/10.1007/978-3-642-00234-2_1.

[7] G. Dhiman, V. Kumar, Seagull Optimization Algorithm: Theory and Its Applications for Large-Scale Industrial Engineering Problems, Knowledge-Based Syst. 165 (2019), 169–196. https://doi.org/10.1016/j.knosys.2018.11.024.

[8] G. Dinca, J. Mawhin, History of the Brouwer Fixed Point Theorem, in: Progress in Nonlinear Differential Equations and Their Applications, Springer, Cham, 2020: pp. 391–412. https://doi.org/10.1007/978-3-030-63230-4_8.

[9] S. Nadler, Multi-Valued Contraction Mappings, Pac. J. Math. 30 (1969), 475–488. https://doi.org/10.2140/pjm.1969.30.475.

[10] S. Kakutani, A Generalization of Brouwer's Fixed Point Theorem, Duke Math. J. 8 (1941), 457–459. https://doi.org/10.1215/S0012-7094-41-00838-4.

[11] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: Proceedings of the IEEE International Conference on Neural Networks, IEEE, pp. 1942–1948, 1995.

[12] S. Kumar, A Short Survey of the Development of Fixed Point Theory, Surv. Math. Appl. 8 (2013), 91–101.

[13] N.C. Maharani, M. Muslikh, A.R. Alghofari, A Modified Seagull Optimization Algorithm for Fixed Point Approximation in Nonlinear Functions, in: MSCEIS IWALS 2025 Conference, 2025. https://confbeam.org/2025/msceis/kfz/abstract/VgBYAMR2G.

[14] M. Mavrovouniotis, C. Li, S. Yang, A Survey of Swarm Intelligence for Dynamic Optimization: Algorithms and Applications, Swarm Evol. Comput. 33 (2017), 1–17. https://doi.org/10.1016/j.swevo.2016.12.005.

[15] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Adv. Eng. Softw. 95 (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

[16] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Adv. Eng. Softw. 69 (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[17] H.K. Pathak, Applications of Fixed Point Theorems, in: An Introduction to Nonlinear Analysis and Fixed Point Theory, Springer, Singapore, 2018: pp. 659–763. https://doi.org/10.1007/978-981-10-8866-7_9.

[18] H.L. Royden, P.M. Fitzpatrick, Real Analysis, Pearson, 2010.

[19] A.H. Soliman, T. Nabil, A. Zidan, On Quasi-Partial Generalized Type of Metric Spaces and an Application to Complexity Analysis of Computer Algorithms, Alex. Eng. J. 59 (2020), 1233–1238. https://doi.org/10.1016/j.aej.2020.01.053.

[20] J. Tang, G. Liu, Q. Pan, A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends, IEEE/CAA J. Autom. Sin. 8 (2021), 1627–1643. https://doi.org/10.1109/jas.2021.1004129.