

PSOX: Leveraging Historical and Global Optima to Enhance Diversity and Speed in Genetic Algorithm Crossover

Xiaobo Jin¹, Jiashu Tu^{2*}

¹ Taizhou Vocational College of Science & Technology, Taizhou, 318000, China

² Taizhou First People's Hospital, Taizhou, 318000, China

* tujiashu1@126.com

<https://doi.org/10.70695/IAAI202503A1>

Abstract

This study introduces PSOX, a novel crossover operator for real-coded genetic algorithms, strategically integrating particle swarm optimization (PSO) principles to revolutionize search efficiency. Unlike conventional crossover methods limited to intra-generational information exchange, PSOX leverages guidance from both the global best solution and historical optima across generations, enabling adaptive exploration of promising search regions while preserving population diversity. Rigorous evaluations on 15 benchmark function (spanning unimodal, multimodal, and highly complex landscapes) demonstrate that PSOX outperforms five state-of-the-art operators in solution accuracy, stability, and convergence speed, particularly when paired with optimal mutation strategies. A detailed analysis of mutation rate impacts provides actionable parameter-tuning insights for diverse optimization scenarios. By bridging GA's evolutionary robustness with PSO's directional learning, PSOX offers a breakthrough for accelerating convergence in complex problem domains.

Keywords Genetic Algorithms; Optimization; Crossover Operators; Particle Swarm Optimization

1 Introduction

In the recent years, most of real-life problems such as Vehicle Routing Problem (VRP), Job Shop Scheduling (JSS) and Maximum Clique Problem(MCP) are NP-hard problems [1], which means those problems can not be solved in exact polynomial-time. To enhance solution efficiency, advanced metaheuristic frameworks such as particle swarm optimization (PSO) [2], Cover-Relax-Search (CRS) [3] and ant colony optimization (ACO) [4] have been rigorously developed and empirically validated across diverse application contexts.

Genetic algorithm (GA), a widely recognized metaheuristic approach derived from Darwinian principles of natural selection and biological evolutionary mechanisms [5], has demonstrated significant efficacy in solving complex optimization problems across diverse domains. In the genetic algorithm, a single chromosome is used to represent a feasible solution, and a specific number of chromosomes are utilized to model and conduct the exploration of the solution space of a given problem. The fitness value serves as a crucial metric for quantitatively assessing the quality of candidate solutions. By simulating the processes in nature, such as the selection of mates by organisms, crossover and exchange of chromosomes, genetic mutation, and natural selection, the algorithm continuously optimizes the feasible solutions. The earliest genetic algorithms (GAs) adopted binary string encoding for chromosomes, but this approach suffers from issues such as low precision and high computational complexity. In contrast, real-coding encoding is more suitable for most engineering problems. GAs that utilize real-coded chromosomes are termed Real Coded GAs [6].

The crossover operator with the objective of evolving superior chromosomes serves as a fundamental mechanism in GAs by facilitating information exchange between chromosomes to explore the search space [6-7]. Based on the building-block hypothesis [8], the crossover operator is the preeminent operator in GAs. According to this hypothesis, certain key genetic components contribute to the emergence of highly fit individuals within the population. The crossover operation effectively allows individuals to merge these essential genetic components, facilitating the creation of superior offspring. Consequently, substantial research efforts have focused on devising sophisticated real-coded crossover operators to improve the performance of Real-Coded GAs.

This study introduces PSOX, an innovative crossover operator inspired from the Particle Swarm Optimization(PSO) algorithm. We will compare the genetic algorithm using this operator with four other mainstream operators. For equitable algorithm comparison, we evaluate both our proposed methods and conventional generational GAs using 15 benchmark problems with diverse complexity levels. We comprehensively analyze and compare the proposed operator from perspectives such as the solution accuracy, stability and convergence speed of the problems. Moreover, through combinatorial experiments with different mutation operators and mutation rates, we investigate the interaction between our operator and different mutation operator.

The paper is structured as follows: Section 2 presents a literature review on real-coded operators, Section 3 details the proposed PSOX crossover, Section 4 describes the experimental setup for all GAs, Section 5 discusses the performance comparison of various GAs, and Section 6 provides concluding remarks.

2 Related Work

Numerous crossover operators for real-coded GAs have been proposed in the literature, which can be broadly categorized into two types according to the differences in the information content they incorporate. One is to generate offspring by constructing linear equations of parental information, which can be described by following equation:

$$O = \sum_i^n \alpha_i v_i \quad (1)$$

The relative crossover operator is presented in Table 1. These crossover operators solely employ the linear combination of parents to generate offspring. The exploration and convergence of the algorithm in the solution space are achieved by controlling the values of parameters. The differences among these crossover operators are mainly reflected in two aspects. One is the number of parents such as Simplex crossover (SX) [14] is multi-parents when others are two-parents, and the other is the way of determining the coefficient values like Laplace crossover uses Laplace distribution to generate coefficient and Logistic crossover uses Logistic distribution [9]. However, the offspring produced by these operators may not consistently evolve toward improved solutions due to the influence of low-fitness individuals in the population. This will lead to a slower convergence speed of the algorithm.

Table 1. Crossover operator by linear combination

Name of crossover operation	Formula	Parameter
LX[10]	$O = p^1 + (p^2 - p^1)L$	$L = \mu + s \cdot \ln\left(\frac{U}{1-U}\right)$
FX[11]	$O = \text{uniform}(\min(p^1, p^2), \max(p^1, p^2))$	No parameter
AX[12]	$O = \alpha p^1 + (1-\alpha)p^2$	$\alpha \in (0, 1)$
BLX- α [13]	$O = \text{uniform}(p^1 - \alpha(p^2 - p^1), p^2 + \alpha(p^2 - p^1))$	$\alpha \in (0, 1)$
SBX[9]	$O^1 = 0.5[(1+\beta)p^1 + (1-\beta)p^2]$ $O^2 = 0.5[(1-\beta)p^1 + (1+\beta)p^2]$	$\beta = \begin{cases} (2u)^{\frac{1}{n_c+1}}, & \text{if } u \leq 0.5, \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{n_c+1}}, & \text{if } u \geq 0.5 \end{cases}$
LX[6]	$O^1 = p^1 + \beta p^1 - p^2 $ $O^2 = p^2 + \beta p^1 - p^2 $	$\beta = \begin{cases} a-b \log_e(u), & \text{if } u \leq 0.5 \\ a+b \log_e(u), & \text{if } u > 0.5 \end{cases}$
SX[14]	$O = (1+\epsilon) \left(x^1 \cdot \frac{1}{n} \sum_{i=1}^n x^i \right)$	$\epsilon \geq 0$ $n \geq 3$

The other crossover operators incorporate additional spatial information to generate offspring, typical related operators include PCX and UNDX, whose crossover formulas are presented in Table 2. Although these operators perform outstandingly in some test problems, their high computational complexity leads to low algorithm operation efficiency, making them unsuitable for problems with high timeliness requirements.

Table 2. Crossover operator with additional spatial information

Name of crossover operation	Formula	Other spatial information
PCX[15]	$\vec{y} = \vec{x}^{(p)} + w_{\zeta} \vec{d}^{(p)} + \sum_{i=1, i \neq p}^{\mu} w_{\eta} \vec{D} e^{(i)}$	Both the perpendicular distance from a point to a line and the spatial orthonormal basis are utilized to guide the direction of offspring generation.
UNDX[16]	$\vec{y} = \vec{g} + \sum_{i=1}^{\mu-1} w_i \vec{d}^{(i)} e^{(i)} + \sum_{i=\mu}^n v_i D \vec{e}^{(i)}$	

3 The Proposed PSO-Inspired Crossover Operator

PSO is inspired by the collective behavior of biological swarms (e.g., bird flocks, fish schools) in adapting to environments, locating resources, and evading predators through information sharing. Hassan et al. [17] have demonstrated through experiments that the convergence speed and computational efficiency of the PSO algorithm are significantly higher than those of the genetic algorithm. However, the mutation operator in the GA algorithm enables the algorithm to exhibit superior performance in escaping from local optima [18].

Therefore, extensive research has focused on integrating PSO and GA algorithms. Most of these hybrid approaches are based on the combination at the algorithm framework level. For example, the hybrid PSO-GA algorithm proposed in [19-20] runs the PSO algorithm and the GA algorithm independently in two stages. The results of the one algorithm are used as the initial population for the another algorithm to perform further optimization. X.H. Shi et al. proposed a PSO-GA algorithm which allows the two algorithms run independently, and then, under the set number of iterations, partial high-quality populations are exchanged. R Huang et al. [21] use PSO as the main framework and the population is divided into three groups during the iteration process, and applying GA-based crossover and mutation operations to each group separately, and finally the update formula of PSO itself is applied. Traditional crossover operators only allow individuals in the current generation to perform crossover. The stochastic nature of selection may result in the inclusion of sub-optimal individuals. As a result, crossover may disrupt elite genes, causing offspring fitness divergence and superior gene loss, thereby impeding convergence. Inspired by the PSO algorithm, we proposes a novel PSO-Inspired crossover operator named PSOX. It mimics the learning method of the PSO algorithm and enables a more fine- grained integration of the PSO and GA algorithms, rather than a crude combination of frameworks. This crossover enables individuals to incorporate information from both the global best solution and other individuals' historical optima, so as to inherit excellent genes and accelerate the convergence process. The offspring are given by the equations:

$$O = w \cdot p_i + c_1 r_1 \vec{d}^{(j)} + c_2 r_2 \vec{d}^{(g)}, \quad (2)$$

where $r_1, r_2 \sim \frac{e^{-(x-\mu)/s}}{s(1+e^{-(x-\mu)/s})^2}$

$$\begin{aligned} \vec{d}^{(j)} &= p_{j_{best}} - p_i, \text{ where } j \neq i \\ \vec{d}^{(g)} &= p_{g_{best}} - p_i \end{aligned} \quad (3)$$

Where w, c_1, c_2 constant terms, which respectively determine the extent to which the offspring learn from themselves, the historical optimal solutions of other individuals, and the global historical optimal solution. $\vec{d}^{(j)}$ represents learning direction from other historical- best individual and in order to gather more diversity information, $p_{j_{best}}$ is the one that is most different from the current individual among the $n(n = 8)$ candidate resources, and cosine similarity is employed for measurement. Different from the setting in traditional crossover algorithms where both parents must come from the current generation, the PSOX algorithm allows individuals to perform inter- generational crossover to inherit the optimal genes of the global and other individuals. $p_{g_{best}}$ denotes the global best solution of the current population. Consequently, $\vec{d}^{(g)}$ represents the learning direction of an individual towards this optimal solution. Meanwhile, random number r_1, r_2 sampled from the Logistic probability density function with inare employed to prevent premature convergence to local optima.

4 Experimental Setup

Three experiments were designed. Firstly, Experiment 1 (Section 4.1) evaluated the proposed crossover operator against benchmark operators using 15 test functions to assess algorithmic accuracy and stability. Secondly, Experiment 2 (section 4.2) focused on convergence speed comparison: four test functions were selected from Experiment 1 results, on which PSOX performed relatively poorly while other operators showed good performance, to analyze convergence speed under the context of small population sizes. Thirdly, Experiment 3 (section 4.3) explored the effects of different mutation rates on GAs equipped with PSOX.

4.1 Accuracy and Stability Test

To systematically evaluate the proposed algorithm's performance, fifteen 30-dimensional benchmark functions were employed to compare the novel crossover operator against five existing operators. These problems exhibit varying levels of difficulty and multi-modality characteristics. Meanwhile, to examine the crossover algorithm's sensitivity to mutation operators, all crossover variants were paired with both Gaussian mutation (GM) and non-uniform mutation (NUM) operators. The real-coded GA incorporating the PSO-inspired crossover (PSOX) operator was simulated, with performance evaluated through mean and standard deviation metrics.

Table 3 presents the uniform testing environment for all algorithms, with fixed parameter values maintained throughout every run. The parameter configurations for comparative crossover operators follow Table 4 specifications. All implementations utilize Python on a Windows platform with 2.9GHz processor and 32GB RAM.

Table 3. Experimental parameter settings

No.of turns	No.of generations Population	No.of Population initial	Population	Select operator	Crossover rate	Mutation operator	Mutation rate
30	1000	300	Uniform random	Tournament with size=3	0.8	NUM	0.1
30	1000	300	Uniform random	Tournament with size=3	0.8	GM	0.1

Table 4. Parameter settings of the crossover operator

	FX	BLX- α	SBX	LX	PSOX
Parameter1	$\alpha = \text{uniform}(0,1)$	$\alpha = 0.5$	eta = 2	a = 0	b = 0.5
Parameter2	-	-	-	b = 0	$c_1 = 1.5, c_2 = 1.5$

4.2 Convergence Speed Test

In Experiment 1, we only focused on the accuracy and stability of the results after a fixed number of iterations, without examining the performance during testing process. In Experiment 2, we selected four test functions (Test cases 4, 5, 7, 11) on which PSOX performs relatively poorly while other crossover operators show good performance. Under the conditions of GM as the mutation operator, a population size of 100, and 500 iterations, with all other settings remaining the same as in Experiment 1, thirty independent trials were performed to assess both convergence speed and process stability of the proposed algorithm.

4.3 Effects of Mutation Rate Test

The experimental results from both Experiment 1 and Experiment 2 confirm PSOX's robust convergence characteristics and operational stability. However, considering that PSOX is inspired by the PSO algorithm, it may inherit the drawback of being prone to getting trapped in local optimal solutions. The mutation rate in genetic algorithms critically governs population diversity. By combining PSOX with different mutation rates, under the experimental conditions of GM as the mutation operator, a

population size of 100, and 100 iterations, we conducted 30 independent experiments for each mutation rate parameter to analyze the impact of the mutation rate on the genetic algorithm equipped with PSOX.

5 Results and Discussion

This section presents the analysis and summary of three experiments in Sections 6.1, 6.2, and 6.3, offering a comprehensive evaluation of the proposed PSOX operator from multiple perspectives.

5.1 Performance Comparison of Different Crossover Operators

The performance evaluation proceeds in two stages: (1) comparative analysis of means and standard deviations across all GAs to assess algorithmic performance and stability, followed by (2) application of the Kruskal-Wallis H-test ($\alpha = 0.05$) to detect significant inter-algorithm performance differences on identical test functions. A significant Kruskal-Wallis H-test result (+) indicates performance differences among algorithms, while a non-significant result (~) suggests no statistical difference.

Performance Comparison of Different Operators Combined with the NUM Mutation Operator

Table 5 presents the mean \pm SD of optimal objective values across generations for each test case, with corresponding Dunnett test p-values in Table 6. Among the 15 test functions, the optimal values found by PSOX are not significantly smaller only in 4 test cases (Cases 3, 4, 14, and 15). In the remaining test cases, the optimal values obtained by PSOX are significantly smaller than those of other crossover operators. Notably, in Test Cases 1, 8, 9, 10, 11 and 13, the average optimal values of the fitness function found by PSOX are several orders of magnitude higher than those of other algorithms. Meanwhile, by comparing the standard deviations, the results of PSOX are slightly larger in Cases 5, 14 and 15, while in other cases, this value is far smaller than that of other crossover operators.

Table 5. Mean and Std of objective function values and Kruskal-Wallis H-test for mean objective function values for all GAs with NUM

Problem number	FX-NUM		BLX- α -NUM		SBX-NUM		LX-NUM		PSOX-NUM		Kruskal-Wallis H-test
	Mean	std	Mean	std	Mean	std	Mean	std	Mean	std	
1	7.8E-01	9.2E-01	2.7E-03	2.0E-03	2.2E+00	7.5E-01	6.5E-01	9.4E-01	6.8E-16	9.0E-16	+
2	2.1E-05	1.5E-05	1.4E-06	1.4E-06	1.9E-06	1.5E-06	6.3E-07	7.1E-07	4.3E-07	3.1E-07	+
3	4.7E-01	2.9E-01	1.5E-01	2.4E-01	5.4E-01	2.6E-01	7.3E-02	1.5E-01	0.0E+00	0.0E+00	+
4	1.8E-01	7.2E-02	2.2E-01	1.7E-01	2.3E-01	2.0E-01	1.3E-01	1.1E-01	3.5E-01	8.3E-02	+
5	1.0E-01	5.3E-02	3.2E-02	3.1E-02	1.9E-01	9.9E-02	2.6E-02	3.6E-02	9.6E-01	4.9E-01	+
6	1.5E+00	4.0E+00	1.9E+00	3.7E+00	2.0E+00	4.1E+00	2.0E+00	4.7E+00	3.2E+00	8.8E+00	+
7	6.2E+00	3.3E+00	1.1E+00	1.4E+00	2.2E+00	3.4E+00	8.1E+00	1.3E+00	2.8E+00	1.5E-01	+
8	3.5E+00	1.8E+00	5.1E+00	1.4E+00	6.7E+00	1.4E+00	2.8E+00	8.4E+00	0.0E+00	0.0E+00	+
9	1.6E-04	1.0E-04	9.7E-07	8.6E-07	5.3E-04	3.5E-04	8.3E-07	5.4E-07	0.0E+00	0.0E+00	+
10	1.8E-03	1.2E-03	1.0E-05	8.7E-06	5.7E-03	2.5E-03	1.0E-05	7.4E-06	0.0E+00	0.0E+00	+
11	3.1E+00	7.9E-01	6.2E+00	2.0E+00	8.8E+00	2.5E+00	6.9E+00	2.0E+00	3.9E-179	0.0E+00	+
12	9.1E-01	3.5E-01	3.1E-01	1.2E-01	1.1E+00	4.0E-01	4.8E-01	1.8E-01	1.1E-03	6.5E-04	+
13	3.5E+00	2.1E+00	1.4E+00	1.1E+00	1.1E+00	7.4E+00	1.5E+00	1.1E+00	0.0E+00	0.0E+00	+
14	1.7E-01	7.4E-02	1.6E-01	1.8E-01	2.5E-01	1.8E-01	1.2E-01	1.0E-01	3.4E-01	1.7E-01	+
15	1.0E-01	5.0E-02	2.8E-02	3.7E-02	1.5E-01	7.1E-02	1.8E-02	3.7E-02	1.1E+00	4.4E-01	+

Table 6. P-value and result of Dunnett Test for all GAs with NUM

Problem number	FX-NUM		BLX- α -NUM		SBX-NUM		LX-NUM	
	Mean	std	Mean	std	Mean	std	Mean	std
1	1.0000	+	0.8378	+	1.0000	+	1.0000	+
2	1.0000	+	0.9296	+	0.9555	+	0.8572	+
3	1.0000	+	1.0000	+	1.0000	+	0.9962	+
4	0.0000	~	0.0009	~	0.0037	~	0.0000	~
5	0.0000	~	0.0000	~	0.0000	~	0.0000	~
6	1.0000	+	1.0000	+	1.0000	+	1.0000	+
7	0.9739	+	0.9993	+	1.0000	+	0.9938	+
8	1.0000	+	1.0000	+	1.0000	+	1.0000	+
9	0.9999	+	0.8376	+	1.0000	+	0.8370	+
10	1.0000	+	0.8383	+	1.0000	+	0.8383	+
11	1.0000	+	1.0000	+	1.0000	+	1.0000	+
12	1.0000	+	1.0000	+	1.0000	+	1.0000	+
13	1.0000	+	0.8364	+	1.0000	+	0.8365	+
14	0.0000	~	0.0000	~	0.0268	~	0.0000	~
15	0.0000	~	0.0000	~	0.0000	~	0.0000	~

Performance Comparison of Different Operators Combined with the GM Mutation Operator

Based on the observations from Tables 7 and 8, Test Case 2 exhibits consistent performance across all crossover operators. For Test Case 2, all GAs find same solution and Kruskal-Wallis H-test result shows negative. For Test Case 15, the PSOX operator performs worse than AX, FX and BLX- α in all remaining test.

Table 7. Mean and Std of objective function values and Kruskal-Wallis H-test for mean objective function values for all GAs with GM

Problem number	FX-GM		BLX- α -GM		SBX-GM		LX-GM		PSOX-GM		Kruskal-Wallis H-test
	Mean	std	Mean	std	Mean	std	Mean	std	Mean	std	
1	7.7E+00	1.2E+00	1.2E+01	2.3E+00	1.8E+01	4.1E-01	1.8E+01	1.0E+00	6.8E-16	9.0E-16	+
2	3.1E-07	0.0E+00	3.1E-07	0.0E+00	3.1E-07	0.0E+00	3.1E-07	0.0E+00	3.1E-07	0.0E+00	~
3	1.2E+01	4.6E+00	3.7E+01	2.7E+01	2.2E+02	7.3E+01	2.0E+00	3.2E+00	0.0E+00	0.0E+00	+
4	1.0E-02	3.2E-02	4.4E-01	7.7E-01	1.1E+00	8.9E-01	3.8E-01	5.0E-01	5.4E-05	2.0E-05	+
5	8.6E-04	2.8E-03	2.1E-02	7.8E-02	4.6E-01	8.2E-01	2.5E-01	5.6E-01	9.6E-02	1.3E-01	+
6	4.9E+01	1.5E+01	1.2E+02	3.6E+01	1.7E+02	2.8E+01	1.7E+02	2.8E+01	8.8E+00	1.6E+01	+
7	1.3E+02	1.5E+02	2.2E+04	9.3E+04	3.7E+05	4.1E+05	2.8E+02	4.2E+02	2.8E+01	1.8E-01	+
8	1.4E+00	1.8E+00	1.5E+00	1.1E+00	3.4E+00	2.3E+00	8.7E-01	6.5E-01	3.6E-23	0.0E+00	+
9	7.0E-05	2.2E-05	1.1E-04	3.7E-05	2.9E-04	8.6E-05	1.6E-04	5.0E-05	1.6E-23	0.0E+00	+
10	1.3E-03	5.7E-04	2.1E-03	7.6E-04	4.8E-03	1.5E-03	2.3E-03	1.0E-03	3.0E-23	0.0E+00	+
11	1.9E+01	4.0E+00	6.4E+01	7.7E+00	6.5E+01	5.8E+00	6.6E+01	6.1E+00	1.4E-11	7.8E-119	+
12	2.3E+00	8.1E-01	9.2E-01	4.8E-01	6.8E+00	2.2E+00	7.6E-01	2.4E-01	1.7E-03	9.4E-04	+
13	8.4E+02	3.3E+02	1.3E+03	4.1E+02	7.4E+03	3.3E+03	1.7E+03	5.0E+02	8.8E-22	0.0E+00	+
14	1.4E-02	4.5E-02	3.3E-01	4.4E-01	1.2E+00	9.4E-01	3.3E-01	4.5E-01	5.8E-05	4.7E-05	+
15	7.6E-04	2.8E-03	6.0E-02	2.0E-01	3.8E-01	5.2E-01	3.7E-01	7.3E-01	1.8E-01	3.6E-01	+

Table 8. p-value and result of Dunnett Test for all GAs with GM

Problem number	FX-GM		BLX-GM		SBX-GM		LX-GM	
	Mean	std	Mean	std	Mean	std	Mean	std
1	1.0000	+	1.0000	+	1.0000	+	1.0000	+
2	-	-	-	-	-	-	-	-
3	0.9959	+	1.0000	+	1.0000	+	0.8959	+
4	0.8551	+	1.0000	+	1.0000	+	1.0000	+
5	0.4643	+	0.5530	+	1.0000	+	0.9964	+
6	1.0000	+	1.0000	+	1.0000	+	1.0000	+
7	0.8340	+	0.9403	+	1.0000	+	0.8350	+
8	1.0000	+	1.0000	+	1.0000	+	1.0000	+
9	1.0000	+	1.0000	+	1.0000	+	1.0000	+
10	1.0000	+	1.0000	+	1.0000	+	1.0000	+
11	1.0000	+	1.0000	+	1.0000	+	1.0000	+
12	1.0000	+	1.0000	+	1.0000	+	1.0000	+
13	0.9999	+	1.0000	+	1.0000	+	1.0000	+
14	0.8655	+	1.0000	+	1.0000	+	1.0000	+
15	0.0152	~	0.0352	~	0.9995	+	0.9992	+

5.2 Analysis of the Convergence Speed

For the test, the Laplace crossover (LX) operator and the Arithmetic crossover (AX) operator were selected as reference objects for comparison with the proposed PSOX operator. The comparison results are shown in Figure 1. The PSOX operator exhibits superior convergence speed compared to LX across all four test functions. For functions 5 (Levy Montalvo 2) and 7 (Rosenbrock), PSOX demonstrates comparable performance to AX. In function 4 (Levy Montalvo 1), PSOX achieves marginally faster convergence and significantly enhanced stability relative to AX. Furthermore, PSOX shows both accelerated convergence and improved stability in function 11 (Schewefel 4). Overall, benefiting from its strategy of continuously learning from the globally optimal individual, the PSOX operator has remarkable convergence speed and convergence accuracy. Its characteristic of high stability ensures that the results obtained from a single run of the algorithm have a very high level of confidence, thus reducing the waste of resources caused by repeated testing.

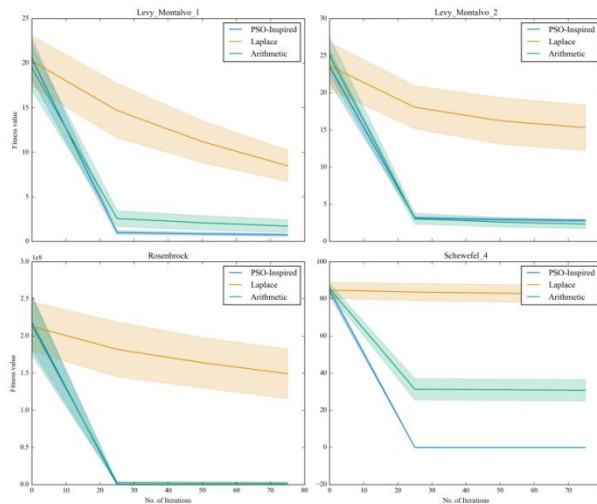


Fig. 1. Comparison of the convergence speeds of different operators in four test cases

5.3 The Relationship between PSOX and the Mutation Rate

As clearly shown in Figure 2, for Test Functions 4 and 5 (Levy Montalvo 1 and Levy Montalvo 2), the algorithm's performance improves with the increase in mutation rate, and its stability gradually enhances.

In contrast, for Test Functions 7 and 11 (Rosenbrock and Schewefel 4), the algorithm's performance is inversely proportional to the mutation rate, and the standard deviation of the optimal fitness function values gradually increases.

Test Functions 4 and 5 are characterized by dense multi-peaks, complex oscillations, and multi-scale closed contour lines. These features make the PSOX operator primarily focused on convergence prone to falling into local optima. Therefore, a higher mutation rate is required to maintain population diversity and escape local optima. According to the operator's formula, once an individual successfully escapes a local optimum, this information is rapidly disseminated to all individuals through crossover, enabling overall population evolution. In contrast, Test Functions 7 and 11 exhibit images with narrow valleys, separable chain structures, and strong regularity, which are more suitable for lower mutation rates. A higher mutation rate may disrupt important genes and lead to divergent results.

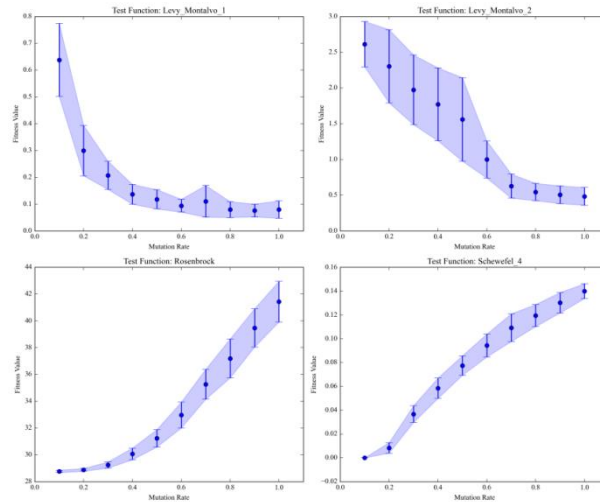


Fig. 2. The influence of the mutation rate on the GA with the PSOX operator

6 Conclusion

This paper proposes a crossover operator for real-coded representation, inspired by the PSO algorithm, called the PSO-Inspired Crossover operator (PSOX). PSOX is a multi-parent crossover method that allows cross-generational operation, simultaneously learning from the globally optimal individual and the historical optimal solutions of other individuals. Through three experiments, we comprehensively tested and analyzed the accuracy, stability, and convergence speed of the genetic algorithm configured with relevant crossover operators, and explored the relationship between the performance of PSOX and the mutation rate.

In the first experiment, we compared the PSOX operator with five other typical operators on 15 test functions. Regardless of whether the mutation operator was NUM or GM, in most test cases, PSOX outperformed other operators in terms of accuracy and stability. In a few test functions such as Function 4 and 5, due to the dense local optima and highly oscillatory terrain, PSOX showed poor performance. However, it was found in Experiment 3 that increasing the mutation rate could effectively solve such problems. Meanwhile, the performance of PSOX-GM was better than that of PSOX-NUM.

In the second experiment, four test functions were selected to compare PSOX with two other operators. It was found that the PSOX operator had a very fast convergence ability and stable convergence performance. The third experiment demonstrated the influence of different mutation operators on PSOX and provided suggestions for the application of PSOX: For problems with smooth, continuous, and highly regular solution spaces, even if there are many local optima, a lower mutation rate is more appropriate. For problems with high oscillations and multi-scale closed contours, a larger mutation rate should be used, and it can even be set to 1. Such a setting can significantly improve the solution results.

Acknowledgement

This work was funded by the Zhejiang Provincial Health Commission, China, project no. 2025ZL602.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Tkatek, S., Bahti, S., & Abouchabaka, J. (2020, October). Artificial intelligence for improving the optimization of NP-hard problems: A review. *International Journal of Advanced Trends in Computer Science and Engineering*, 9, 7411 – 7420. <https://doi.org/10.30534/ijatcse/2020/73952020>
2. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 — International Conference on Neural Networks* (Vol. 4, pp. 1942 – 1948). IEEE.
3. Huang, W., Isenberg, N. M., Drgona, J., Vrabie, D. L., & Dilkina, B. (2025). Cover-relax-search: A primal heuristic for binary quadratic programs. In *PODANI, J. (1994). Multivariate Data Analysis in Ecology and Systematics* (SPB Publishing, The Hague). <https://doi.org/10.1016/j.cemconcomp.2017.02.001>
4. Dorigo, M., Birattari, M., & Stutzle, T. (2006, November). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4), 28 – 39. <https://doi.org/10.1109/mci.2006.329691>
5. Michalewicz, Z., & Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1), 1 – 32. <https://doi.org/10.1162/evco.1996.4.1.1>
6. Deep, K., & Thakur, M. (2007, May). A new crossover operator for real coded genetic algorithms. *Applied Mathematics and Computation*, 188(1), 895 – 911. <https://doi.org/10.1016/j.amc.2006.10.047>
7. Sastry, K., & Goldberg, D. (2002, June). Analysis of mixing in genetic algorithms: A survey.
8. Holland, J. H. (1992, April). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.
9. Naqvi, F. B., Shad, M. Y., & Khan, S. (2020, October). A new logistic distribution based crossover operator for real-coded genetic algorithm. *Journal of Statistical Computation and Simulation*, 91(4), 817 – 835. <https://doi.org/10.1080/00949655.2020.1832093>
10. Radcliffe, N. (2003, March). Equivalence class analysis of genetic algorithms. *Complex Systems*, 5.
11. Rocke, D. M., & Michalewicz, Z. (2000, March). Genetic algorithms + data structures = evolution programs. *Journal of the American Statistical Association*, 95(449), 347. <https://doi.org/10.2307/2669583>
12. Eshelman, L. J., & Schaffer, J. D. (1993). Real-coded genetic algorithms and interval-schemata. In *Foundations of genetic algorithms 2* (pp. 187 – 202). Elsevier.
13. Agrawal, R., Deb, K., & Agrawal, R. (2000, June). Simulated binary crossover for continuous search space. *Complex Systems*, 9.
14. Tsutsui, S., Yamamura, M., & Higuchi, T. (1999). Multi-parent recombination with simplex crossover in real-coded genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*.
15. Deb, K., Anand, A., & Joshi, D. (2002, December). A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10(4), 371 – 395. <https://doi.org/10.1162/106365602760972767>
16. Ono, I., Kita, H., & Kobayashi, S. (2003). A real-coded genetic algorithm using the unimodal normal distribution crossover. In *Advances in natural computation* (pp. 213 – 237). Springer.
17. Boeringer, D. W., & Werner, D. H. (2003). A comparison of particle swarm optimization and genetic algorithms for a phased array synthesis problem. In *IEEE Antennas and Propagation Society International Symposium* (Vol. 1, pp. 181 – 184). IEEE.
18. Kachitvichyanukul, V. (2012, September). Comparison of three evolutionary algorithms: GA, PSO, and DE. *Industrial Engineering and Management Systems*, 11(3), 215 – 223. <https://doi.org/10.7232/iems.2012.11.3.215>
19. Garg, H. (2016, February). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation*, 274, 292 – 305. <https://doi.org/10.1016/j.amc.2015.11.001>
20. Khan, I., Pal, S., & Maiti, M. K. (2019, October). A hybrid PSO-GA algorithm for traveling salesman problems in different environments. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 27(05), 693 – 717. <https://doi.org/10.1142/s0218488519500314>
21. Huang, R., Fu, X., & Pu, Y. (2023, January). A novel fractional accumulative grey model with GA-PSO optimizer and its application. *Sensors*, 23(2), 636. <https://doi.org/10.3390/s23020636>

Biographies

1. **Xiaobo Jin** master of applied statistic degree from JiLin University of Finance and Economics. Currently, working as lecturer of big data technology in Taizhou Vocational College of Science & Technology, mainly engaged in data analysis, computer vision, optimization algorithm, graph neural network, etc.

2. **Jiashu Tu** received the master degree in traditional chinese medicine from Chengdu University of TCM. Currently, working as attending physician in Taizhou first people's Hospital. At present, the main academic research direction is the application of artificial intelligence in traditional Chinese Medicine.

PSOX：融合歷史與全局最優的遺傳交叉多樣性加速機制

金曉波¹，屠嘉樹²

¹台州科技職業學院，台州，中國，318000

²台州市第一人民醫院，台州，中國，318000

摘要：本研究提出一種面向實數編碼遺傳算法的新型交叉算子PSOX，通過策略性地融合粒子羣優化（PSO）機制，顯著提升搜索效率。與傳統僅依賴當代種羣信息交換的交叉方法不同，PSOX同時利用全局最優解與跨代歷史最優解進行引導，在保持種羣多樣性的同時自適應地探索高潛力搜索區域。基於15個基準函數（涵蓋單峯、多峯及高複雜度地貌）的系統評估表明，PSOX在解精度、穩定性與收斂速度方面均優於五種現有先進算子，尤其在與最優變異策略耦合時表現突出。進一步對變異率影響的深入剖析，為不同優化場景提供了可操作的參數調優指南。PSOX通過橋接遺傳算法的進化穩健性與粒子羣優化的方向學習優勢，為複雜問題域的快速收斂提供了新範式。

關鍵詞：遺傳算法；優化；交叉算子；粒子羣優化

1. 金曉波，吉林財經大學應用統計學碩士，現任台州科技職業學院大數據技術講師，主要研究方向為數據分析、計算機視覺、優化算法、圖神經網絡；
2. 屠嘉樹，成都中醫藥大學中醫學碩士，現任台州市第一人民醫院主治醫師，主要研究方向為人工智能在中醫領域的應用。