A Novel Surrogate-guided Jaya Algorithm for the Continuous Numerical Optimization Problems

Fuqing Zhao[1], Ru Ma[1], Jianxin Tang[1]
[1]School of Computer and Communication Technology, Lanzhou University of Technology Lanzhou 730050, China
Fzhao2000@hotmail.com (Fuqing Zhao);
312783672@qq.com (Ru Ma);
jxtang2011@hotmail.com (Jianxin Tang)

Yi Zhang[2], Weimin Ma[3]
[2]School of Mechanical Engineering, Xijin University Xi'an710123, China
1049440546@qq.com
[3]School of Economics and Management, Tongji University, Shanghai 200092, China
mawm@tongji.edu.cn

***Abstract*—A new metaheuristic algorithm, named surrogate-guided algorithm(S-Jaya), is proposed to solve the single objective continuous optimization problems in this paper. A novel mutation strategy for the non-separable single objective continuous optimization problems is introduced to alter the search engine of the Jaya algorithm. The surrogate is embedded to accelerate the convergence of the population and avoid the proposed algorithm falling into the local optimal during the evolutionary process. The suggested S-Jaya algorithm to address the CEC 2017 benchmark problems is effective and validated. On the quality of solution and execution time, the experimental results reveal that the effectiveness of the S-Jaya algorithm is superior compare with the Jaya algorithm and its variants.**

Keywords—Jaya algorithm; Mutation strategy; Surrogate; Radial basis function

I. INTRODUCTION

In recent years, the evolution algorithms (EAs) have been applied to solve enormous practical issues in various domains, for instance, aviation, industrial manufacturing [1], and scheduling [2]. For practical engineering problems, there are various complexities, including non-linear, non-quadratic, non-convex, and ruggedness. The traditional mathematical methods, such as the Newton-Raphson method [3], gradient descent [4], and simplex algorithm [5], have to search for the entire search space, and easily generate combinatorial explosion problems.

L. Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: wangling @ tsinghua.edu.cn). Inspired by human intelligence and the order of nature, various intelligent optimization algorithms have been proposed to solve optimization problems. Owing to the briefness, powerful searchability, and outstanding generality, metaheuristics have been widely researched by scholars. A differential evolutionary algorithm [6] is proposed for simulating biological evolution. A genetic algorithm [7] is a method for searching for the optimal solution by simulating the natural evolution process. The particle swarm optimization algorithm [8] is raised to imitate the foraging mode of birds. Inspired by the principles of biogeography named biogeography-based optimization [9]. The water wave optimization algorithm is based on shallow water wave theory [10], which mainly simulates the propagation, refraction, and breaking phenomenon of water waves. For the above algorithms, except for tuning the common parameters, it is significant to tune algorithm-specific control parameters. Different combinations of algorithm-specific parameters in balancing exploitation and exploration are widely divergent during the evolutionary process. Owing to the sensitivity of algorithm-specific control parameters, applications of metaheuristic optimization algorithms are limited to solve disparate and complex practical problems. Therefore, it is necessary to propose an algorithm with few algorithm-specific control parameters to improve the universality of the metaheuristic algorithm.

An algorithm-specific parameter-less algorithm named Jaya was proposed by Rao [11] for solving constrained and unconstrained optimization problems. Any algorithm-specific control parameters except population size, number of maximum iteration, and number of design variables before the experiments are not adjusted in the Jaya algorithm. The core concept of the Jaya algorithm is that each solution acquired for an optimization problem gets close to the best solution and remote from the worst one. Besides, improved versions of the Jaya algorithm, named quasi-oppositional Jaya, are proposed to enhance the diversify of the population and increase the convergence rate of the Jaya algorithm [12]. The key feature is that the population size is voluntarily determined in Self-adaptive Jaya [13]. The diversity of the population is adjusted by altering the number of a subpopulation in self-adaptive multi-population Jaya [14]. The self-adaptive multi-population elitist Jaya algorithm adaptively changes the number of subpopulations based on the changing strength of the solutions to effectively monitor the problem landscape [15].

There are quality evaluations including computationally expensive numerical simulations or costly experiments, it takes thousands of fitness evaluations to find the optimal solution. With the increase of fitness evaluations, the cost is increased stupendously. Surrogate-assisted evolutionary algorithms (SAEAs) have been introduced. Part of the expensive fitness assessments is taken the place of computationally inexpensive approximate models [16]. SAEAs are referred to as surrogates or meta-models. In the past decade, evolutionary algorithms based on surrogates have been attracted various researchers to study. In the work of evolutionary algorithm based on surrogates, the regression model is established by the majority of researchers to approach the objective function and applied to replace the real fitness evaluation function to reduce the use of true fitness assessments. Two strategies are proposed to combine the authentic fitness function with the surrogate fitness function. One strategy is named individual-based evolution control. The

This study was financially supported by the National Natural Science Foundation of China under grant number 62063021. It was also supported by the Lanzhou Science Bureau project (2018-rc-98), Public Welfare Project of Zhejiang Natural Science Foundation (LGJ19E050001), respectively.

number of individuals each generation in the individual-based evolution control is evaluated by the surrogate fitness function. The other strategy is called generation-based evolution control. All the individuals in the population are evaluated with the surrogate fitness function every M generations. M is a real number and less than the maximum number of iterations [17]. The high usage surrogates include radial basis function (RBF) [18], kriging [19], and polynomial response surface (PRS) [20].

Despite the robustness of the Jaya algorithm, it takes thousands of fitness evaluations to find the optimal solution. For CEC 2017 benchmark problems [21], the optimal solutions of majority functions are still not found by the basic Jaya algorithm, especially for the large-scale instances of multi-mode and multi-extremum. Hence, the focus of this paper is how to build an accurate surrogate to reduce the real fitness evaluations in the optimization process of the Jaya algorithm. In consideration of the operation mechanism of the Jaya algorithm. The Surrogate-guided Jaya algorithm, combined RBF Surrogate model, is proposed to smooth out the local optima, help the algorithm avoid falling into the local optimum, and accelerate the convergence speed. The following modifications are summarized to the basic Jaya algorithm.

- A novel mutation strategy is proposed to reduce the randomness of the Jaya algorithm. The dimensional information of the outstanding solution is retained and the inferior one is discarded during the process of evolution by the novel mutation strategy.
- A strategy based on the preselection of the surrogate is introduced to save evaluation times and improve the search performance in the framework of the Jaya algorithm.

The remainder of this work is organized as follows. In Section 2, the description of the Jaya algorithm and radial basis function is presented. The Surrogate-guided Jaya algorithm is presented in Section 3. The experimental results are presented in Section 4. The discussions are presented in Section 5.

II. Jaya Algorithm And Radial Basis Function

A. Jaya algorithm

Let $O(x)$ is an objective function to be optimized (minimized or maximized). P initial solutions are randomly generated between the maximum and the minimum of each design variable. At any iteration i, population size is 'j' (i.e. $j = 1,2,\ldots,p$), the number of design variables is 'k' (i.e. $k = 1,2,\ldots,d$) and $O(x_{i,j,k})$ is the value of the kth variable for the jth candidate during the ith iteration, then this value is modified as per the following Eq.

$$x_{i+1,j,k} = x_{i,j,k} + r_1(x_{i,best,k} - |x_{i,j,k}|) - r_2(x_{i,worst,k} - |x_{i,j,k}|) \quad (1)$$

where $x_{i,best,k}$ and $x_{i,worst,k}$ represent the best and worst solutions in the current population. $x_{i+1,j,k}$ is the new value produced by $x_{i,j,k}$. r_1, r_2 are random numbers in the range of 0 and 1. The term "$r_1(x_{i,j,best} - |x_{i,j,k}|)$" indicates that the solution approach the best solution, the term "$-r_2(x_{i,j,worst} - |x_{i,j,k}|)$" shows that the solution escapes from the worst solution. $x_{i+1,j,k}$ is accepted when the function value of $x_{i+1,j,k}$ is better than the function value of $x_{i,j,k}$. An outstanding solution in the competition becomes the input to the next iteration.

B. Radial basis function

In this study, the surrogate-based on radial basis function (RBF) is used in the optimization process of the proposed algorithm. The advantage of RBF is that the speed of metamodeling is rapid compared with the Gaussian model [22].

Let $x_1, x_2, \ldots, x_n$ are distinct point, and the values of optimized problems are $f(x_1), f(x_2), \ldots, f(x_n)$. The interpolating form of RBF is described as follows.

$$f'(x) = \sum_{i=1}^{n} \lambda_i \phi(\| x - x_i \|) + p(x) \quad (2)$$

Where λ_i is the weight of the ith basis function, and $\|\cdot\|$ is the Euclidean norm. $r = x - x_i$ indicate the Euclidean distance between the predicted point x and the sample points x_i. $p(x)$ is a linear polynomial function $p(x) = a^T x + b$. The common forms of radial basis functions are described as follows.

$$\phi(r) = r \quad \text{(linear)} \quad (3)$$

$$\phi(r) = r^3 \quad \text{(cubic)} \quad (4)$$

$$\phi(r) = r^2 logr \quad \text{(thin-plate spline)} \quad (5)$$

$$\phi(r) = \sqrt{r^2 + \gamma^2} \quad \text{(multiquadric)} \quad (6)$$

$$\phi(r) = e^{-\gamma r^2} \quad \text{(Gaussian)} \quad (7)$$

The parameters $\lambda_1, \lambda_2, \ldots, \lambda_n$ of the radial basis, the function is acquired by the solution of the following linear equations.

$$\begin{pmatrix} \phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \quad (8)$$

Where ϕ is the $n \times n$ matrix wit $\phi_{ij} = \phi(\| x_i - x_j \|_2)$ and

$$P = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_n^T & 1 \end{pmatrix}, \qquad \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} \quad (9)$$

$$c = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \\ a \end{pmatrix}, \qquad F = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix} \quad (10)$$

If rank $(p) = D + 1$, the matrix $\begin{pmatrix} \phi & P \\ P^T & 0 \end{pmatrix}$ is nonsingular and the system has a unique solution. A unique model of radial basis function is acquired [22].

III. Surrogate-guided Jaya Algorithm

In this section, two mechanisms for ameliorating the performance of the basal Jaya algorithm are described. One method is the novel mutation strategy, which is applied to combine the excellent value of each dimension. The other method is that the surrogate-guided strategy for expensive problems is presented to save evaluation times. The framework of the proposed algorithm is depicted in Fig.1.

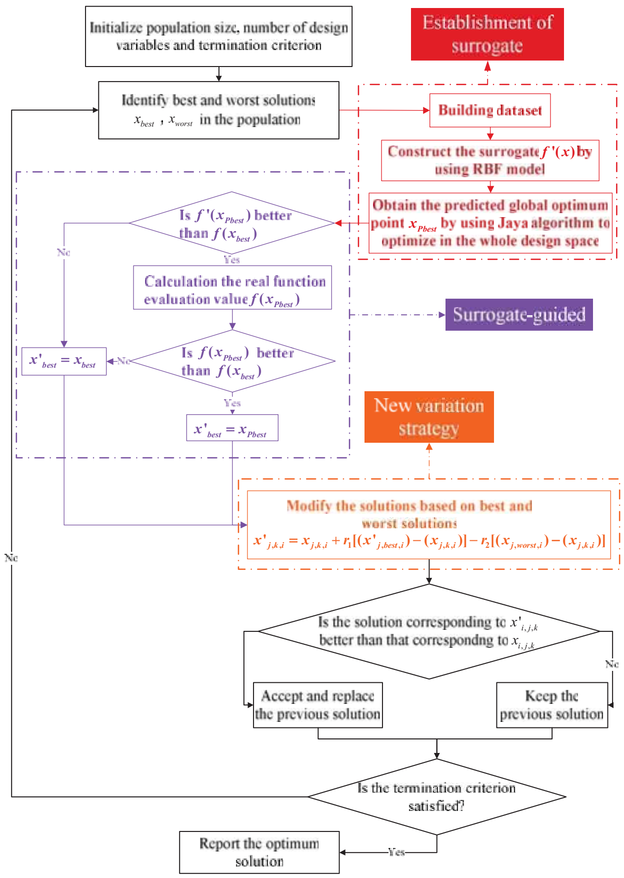


Figure.1 framework of surrogate-guided Jaya

A. A novel mutation strategy

The evolutionary information of the population is broken on account of taking the absolute value of each dimension. The randomness of the solution produced by the mutation strategy of the basic Jaya algorithm is uncontrollable. Thus, a novel efficient mutation strategy is proposed. The mutation formula is shown as follows.

$$x_{i+1,j,k} = x_{i,j,k} + r_1(x_{i,best,k} - x_{i,j,k}) - r_2(x_{i,worst,k} - x_{i,j,k}) \quad (11)$$

According to Eq.11, the satisfactory value of the historical solution in every dimension is retained and integrated for solving CEC 2017 benchmark problems. Not only the convergence speed of the algorithm is guaranteed, but also the diversity of the population is increased.

B. A surrogate-guided strategy

A surrogate-guided strategy based on RBF is introduced to the Jaya algorithm to smooth out local optima and reduce the number of evaluations. More than 5D initial sampling points are generated by Latin hypercube sampling to ensure the accuracy of modeling. As shown in Figure.1, a surrogate is built in every generation by the data in the dataset which is near to the solution of the current generation. The maximum number of modeling points is Mps, which is tuned according to the scales of different problems. Misleading of the surrogate is avoided, owing to that every optimal solution given by surrogate is evaluated by real function in the process of evolution, and the current optimal population is replaced unless true fitness of the optimal solution given by surrogate is smaller than the current optimal population. The formula of replacement is as follows.

$$x' = \begin{cases} x_{Pbest}, & if\ f'(x_{Pbest}) < f(x_{best})\ and \\ & f(x_{Pbest}) < f(x_{best}) \\ x_{best}, & else \end{cases} \quad (12)$$

The promising area is excavated by the surrogate-guided strategy. As shown in Figure.2, the local optimums are smooth out by the response surface. Under the guidance of yellow points predicted by surrogate, real fitness evaluations are saved from the region of a local optimal area, and the speed of convergence is accelerated.

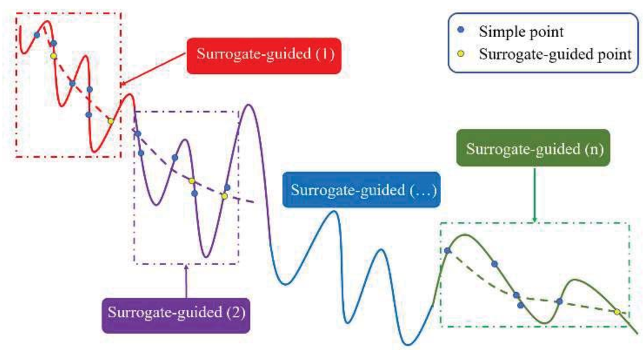


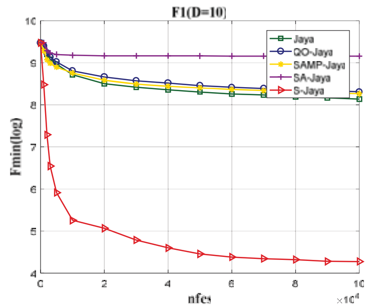
Figure.2 surrogate-guided evolution

A promising evolutionary direction is provided combined with the novel mutation strategy and the surrogate-guided strategy. The algorithm converges rapidly and keeps the algorithm from falling into a locally optimal solution. Based on the above description, the step of S-Jaya is given as follows.

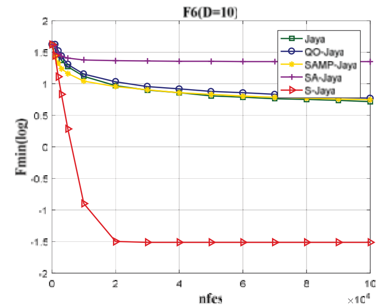
S-Jaya algorithm:	
Input:	N: population size T: maximum number of generation D: dimension Mps: maximum number of sample points
Step 1:	More than 5D initial sampling points are generated using the LHS, and these points are stored in the dataset. The top 25 individuals are select as the initial solution of the population.
Step 2:	In the dataset, Mps individuals close to the current population are selected for modeling. The Jaya algorithm with a novel mutation strategy is used to optimize the surrogate. The optimal solution of the current population is replaced by Eq.12.
Step 3:	Optimization of a real function using Jaya algorithm by Eq.11.
Step 4:	Reach the maximum number of iterations, output the optimal solution. Else, the algorithm jumps to step 2.

IV. Experimental Study and Discussion

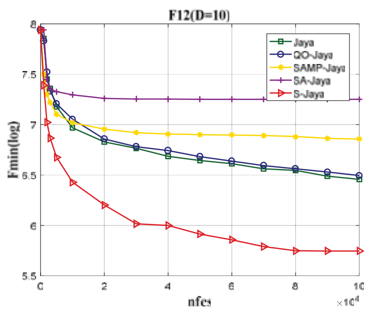
To assess the performance of the S-Jaya algorithm, unimodal and multimodal CEC 2017 benchmark problems are adopted. The dimension (D) is 10. The consequences are presented in Table.1. It can be seen that S-Jaya is more rapid than basal Jaya and its variants in convergence speed. Boxplots show that S-Jaya is stable. In conclusion, S-Jaya is superior to other comparison algorithms in global performance.



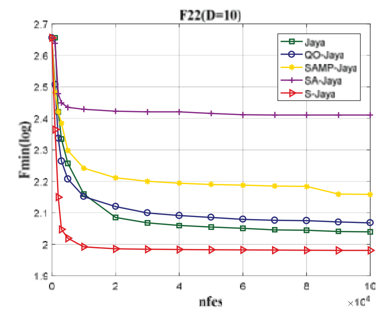
(a) f_1



(b) f_6

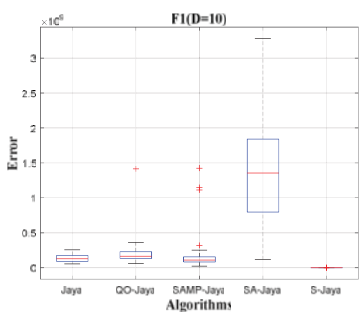


(c) f_{12}

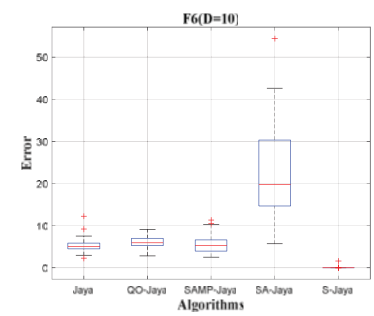


(d) f_{22}

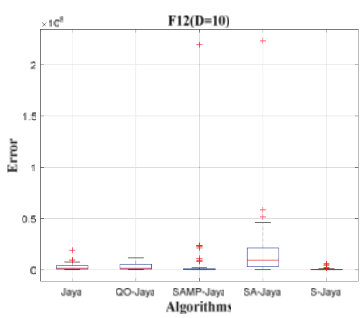
Figure.3 Convergence curves of Jaya, QO-Jaya, SAMP-Jaya, SA-Jaya, and S-Jaya for different CEC 2017 benchmark functions(10D)



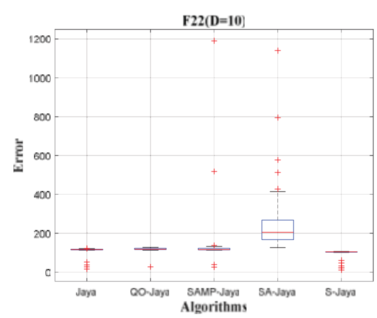
(a) f_1



(b) f_6



(c) f_{12}



(d) f_{22}

Figure.4 Boxplots of Jaya, QO-Jaya, SAMP-Jaya, SA-Jaya, and S-Jaya for different CEC 2017 benchmark functions(10D)

TABLE I. OPTIMIZING RESULTS TESTED BY BASIC JAYA, JAYA VARIANTS, AND S-JAYA

Fun	Jaya		QO-Jaya		SA-Jaya		SAMP-Jaya	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	1.54E+08	1.59E+08	1.87E+08	6.78E+07	1.57E+09	1.03E+09	1.12E+08	4.75E+07
F3	1.31E+03	3.89E+02	1.21E+03	4.72E+02	1.78E+04	1.25E+04	2.29E+03	3.42E+03
F4	8.24E+00	4.24E-01	8.10E+00	1.45E+00	1.04E+02	9.10E+01	1.11E+01	1.73E+01
F5	3.64E+01	5.96E+00	4.11E+01	5.39E+00	5.11E+01	1.32E+01	3.60E+01	6.57E+00
F6	5.48E+00	1.44E+00	5.81E+00	1.18E+00	2.05E+01	8.38E+00	5.77E+00	2.55E+00
F7	4.88E+01	6.48E+00	5.44E+01	5.61E+00	9.87E+01	4.56E+01	4.94E+01	7.86E+00
F8	3.64E+01	4.61E+00	3.54E+01	6.22E+00	4.99E+01	1.45E+01	3.23E+01	6.34E+00
F9	1.03E+01	5.17E+00	1.62E+01	2.39E+01	4.31E+02	3.13E+02	1.36E+01	1.58E+01
F10	1.04E+03	2.34E+02	1.21E+03	2.11E+02	1.19E+03	3.14E+02	**6.73E+02**	3.10E+02
F11	3.89E+01	1.15E+01	5.33E+01	1.74E+01	5.61E+02	5.20E+02	6.74E+01	5.37E+01
F12	2.11E+06	2.52E+06	3.28E+06	3.99E+06	4.11E+07	7.78E+07	8.72E+05	1.80E+06
F13	1.08E+04	1.25E+04	1.12E+04	9.45E+03	9.00E+04	3.86E+05	1.16E+04	1.20E+04
F14	4.77E+01	1.24E+01	1.12E+02	2.75E+02	4.95E+03	8.14E+03	1.89E+03	6.46E+03
F15	3.76E+02	1.96E+02	5.68E+02	2.63E+02	6.00E+03	9.97E+03	3.35E+03	9.20E+03
F16	7.50E+01	2.99E+01	8.66E+01	3.85E+01	2.80E+02	1.60E+02	7.40E+01	6.66E+01
F17	7.16E+01	9.99E+00	7.53E+01	1.54E+01	1.49E+02	9.18E+01	8.82E+01	3.85E+01
F18	3.69E+04	8.76E+03	4.48E+04	2.23E+04	4.39E+05	1.09E+06	3.93E+04	9.06E+03
F19	2.76E+02	2.36E+02	1.65E+03	5.54E+03	3.43E+04	6.19E+04	5.52E+03	2.95E+04
F20	5.20E+01	1.22E+01	9.11E+01	4.99E+01	1.40E+02	7.01E+01	4.82E+01	2.02E+01
F21	2.27E+02	3.07E+01	**1.12E+02**	9.93E+00	2.28E+02	4.20E+01	2.22E+02	2.86E+01
F22	1.12E+02	2.10E+01	1.19E+02	2.99E+00	2.69E+02	1.66E+02	1.66E+02	2.07E+02
F23	3.41E+02	6.34E+00	3.44E+02	5.20E+00	3.46E+02	9.57E+00	3.43E+02	9.45E+00
F24	3.58E+02	5.21E+01	**3.06E+02**	1.01E+02	3.75E+02	2.38E+01	3.62E+02	3.77E+01
F25	4.46E+02	1.47E+01	4.46E+02	1.86E+01	5.22E+02	7.15E+01	4.46E+02	2.30E+01
F26	4.66E+02	2.66E+02	**4.28E+02**	3.14E+01	7.22E+02	3.58E+02	5.46E+02	3.50E+02
F27	3.97E+02	2.74E+00	4.01E+02	1.29E+01	4.03E+02	1.04E+01	4.10E+02	2.59E+01
F28	5.90E+02	1.25E+02	**5.42E+02**	8.58E+01	5.79E+02	1.11E+02	5.76E+02	1.27E+02
F29	**2.83E+02**	3.28E+01	3.07E+02	3.79E+01	3.79E+02	8.29E+01	3.05E+02	5.47E+01
F30	**2.03E+05**	2.90E+05	3.46E+05	5.01E+05	1.03E+06	1.11E+06	5.34E+05	5.14E+05

TABLE II. p-VALUE OF WILCOXON'S RANK-SUM TEST FOR D = 10

S-Jaya vs	R+	R-	+	≈	−	Z	p-value	α=0.05	α=0.01
Jaya	388.50	46.50	26	0	3	-3.698^{b}	2.17E-04	Yes	Yes
QO-Jaya	338.00	97.00	25	0	5	-2.606^{b}	9.17E-03	Yes	Yes
SA-Jaya	435.00	0.00	29	0	0	-4.703^{b}	2.56E-06	Yes	Yes
SAMP-Jaya	415.00	20.00	28	0	1	-4.271^{b}	1.95E-05	Yes	Yes

V. Conclusion

According to the experimental results, the proposed algorithm named surrogate-guided Jaya algorithm shows obvious advantages compared with the basal Jaya algorithm and its variants. The accuracy of the solution, the convergence speed, and the stability of the algorithm are improved. The costs are saved by the surrogate-guided Jaya algorithm solving expensive optimization problems. The results confirm that the efficiency and quality of the proposed algorithm are superior to the basal Jaya algorithm and its variants. In conclusion, the proposed surrogate-guided Jaya algorithm is a fast, robust, and trustworthy algorithm for solving single-objective continuous optimization problems.

Acknowledgment

This study was financially supported by the National Natural Science Foundation of China under grant number 62063021. It was also supported by the Lanzhou Science Bureau project (2018-rc-98), Public Welfare Project of Zhejiang Natural Science Foundation (LGJ19E050001), respectively.

References

1. Tayarani-N, M.-H., X. Yao, and H. Xu, Meta-Heuristic Algorithms in Car Engine Design: A Literature Survey. Ieee Transactions on Evolutionary Computation, 2015. 19(5): p. 609-629.
2. Pan, Q.K., M.F. Tasgetiren, and Y.C. Liang, A discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem. Computers & Operations Research, 2008. 35(9): p. 2807-2839.
3. Fuerte-Esquivel, C.R., E. Acha, and H. Ambriz-Perez, A comprehensive Newton-Raphson UPFC model for the quadratic power flow solution of practical power networks. IEEE Transactions on Power Systems, 2000. 15(1): p. 102-109.
4. Bordes, A., L. Bottou, and P. Gallinari, SGD-QN: Careful Quasi-Newton Stochastic Gradient Descent. Journal of Machine Learning Research, 2009. 10: p. 1737-1754.
5. Chang, C.I., et al., A new growing method for simplex-based endmember extraction algorithm. Ieee Transactions on Geoscience and Remote Sensing, 2006. 44(10): p. 2804-2819.
6. Das, S. and P.N. Suganthan, Differential Evolution: A Survey of the State-of-the-Art. Ieee Transactions on Evolutionary Computation, 2011. 15(1): p. 4-31.
7. Grefenstette, J.J., Optimization of Control Parameters for Genetic Algorithms. IEEE Transactions on Systems, Man, and Cybernetics, 1986. 16(1): p. 122-128.
8. Zhan, Z.H., et al., Adaptive Particle Swarm Optimization. Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics, 2009. 39(6): p. 1362-1381.
9. Zhao, F., et al., A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem. Expert Systems With Applications, 2019. 126: p. 321-339.
10. Zhao, F., et al., An improved water wave optimisation algorithm enhanced by CMA-ES and opposition-based learning. Connection Science, 2019.p. 1-30
11. Venkata Rao, R., Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations, 2016. 7(1): p. 19-34.
12. Rao, R.V. and D.P. Rai, Optimisation of welding processes using quasi-oppositional-based Jaya algorithm. Journal of Experimental & Theoretical Artificial Intelligence, 2017. 29(5): p. 1099-1117.
13. Rao, R.V. and K.C. More, Design optimization and analysis of selected thermal devices using self-adaptive Jaya algorithm. Energy Conversion and Management, 2017. 140: p. 24-35.
14. Venkata Rao, R. and A. Saroj, A self-adaptive multi-population based Jaya algorithm for engineering optimization. Swarm and Evolutionary Computation, 2017. 37: p. 1-26.
15. Rao, R.V. and A. Saroj, An elitism-based self-adaptive multi-population Jaya algorithm and its applications. Soft Computing, 2019. 23(12): p. 4383-4406.
16. Wang, H., Y. Jin, and J. Doherty, Committee-Based Active Learning for Surrogate-Assisted Particle Swarm Optimization of Expensive Problems. Ieee Transactions on Cybernetics, 2017. 47(9): p. 2664-2677.
17. Loshchilov, I., M. Schoenauer, and M. Sebag, Comparison-Based Optimizers Need Comparison-Based Surrogates, in Parallel Problems Solving from Nature - Ppsn Xi, Pt I, R. Schaefer, et al., Editors. 2010. p. 364-373.
18. Gutmann, H.M., A Radial Basis Function Method for Global Optimization. Journal of Global Optimization, 2001. 19(3): p. 201-227.
19. Chugh, T., et al., A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization. IEEE Transactions on Evolutionary Computation, 2018. 22(1): p. 129-142.
20. Zongzhao, Z., et al. A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. in 2005 IEEE Congress on Evolutionary Computation. 2005.
21. Wu, G., R. Mallipeddi, and P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization. 2016.
22. Cai, X., et al., Surrogate-guided differential evolution algorithm for high dimensional expensive problems. Swarm and Evolutionary Computation, 2019. 48: p. 288-311.