Evolutionary Computation for Privacy-Preserving Optimization

Zhi-Hui Zhan (Corresponding Author), *Senior Member, IEEE*, Sheng-Hao Wu, *Student Member, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—Evolutionary computation (EC) is a kind of advanced computational intelligence (CI) algorithm and advanced artificial intelligence (AI) algorithm. EC algorithms have been widely studied for solving optimization and scheduling problems in various real world applications, which act as one the Big Three in CI and AI, together with fuzzy system and neural networks. Even though EC has been fast developed in recent years, there is an assumption that the algorithm designer can obtain the objective function of the optimization problem so that they can calculate the fitness values of the individuals to follow the "survival of the fittest" principle in natural selection. However, in real-world application scenario, there is a kind of problem that the objective function is privacy so that the algorithm designer can not obtain the fitness values of the individuals directly. This is the privacy preserving optimization problem (PPOP) where the assumption of available of objective function does not check out. How to solve the PPOP is a new emerging with seldom study but is also a challenging research topic in EC community. This paper proposes a rank-based cryptographic function (RCF) to protect the fitness value information. Specially, the RCF is adopted by the algorithm user to encrypt the fitness values of all the individuals as rank so that the algorithm designer does not know the exact fitness information but only the rank information. Nevertheless, the RCF can protect the privacy of the algorithm user but still can provide sufficient information to the algorithm designer to drive the EC algorithm. We have applied the RCF privacy preserving method to two typical EC algorithms including particle swarm optimization (PSO) and differential evolution (DE). Experimental results show that the RCF-based privacy preserving PSO and DE can solve the PPOP without performance loss.**

Keywords*—Evolutionary computation (EC); computational intelligence (CI); artificial intelligence (AI); privacy preserving optimization problem (PPOP); rank-based cryptographic function (RCF)

I. Introduction

Evolutionary computation (EC), together with fuzzy system and neural network, are the Big Three of computational intelligence (CI) and also the artificial intelligence (AI) [1]. Generally speaking, as shown in Fig. 1, EC family includes evolutionary algorithm (EA) and swarm intelligence (SI). The EAs are inspired by the biological evolution phenomenon, using selection, crossover, and mutation to produce better and better solutions during the evolution to approach the global optimum. The EAs mainly include genetic algorithm (GA), differential evolution (DE) [2][3], and estimation of distribution algorithm (EDA) [4]. The SI algorithms mainly include ant colony optimization (ACO), particle swarm optimization (PSO) [5], and artificial bee colony (ABC). The SI algorithms simulate the swarm behaviors of foraging, search for the global optimum by the guidance of personal experience and swarm experience. In recent years, the DE as typical EA and the PSO as typical SI, have fast developed and widely applied in many optimization problems. Therefore, this paper also focuses on these two typical algorithms.

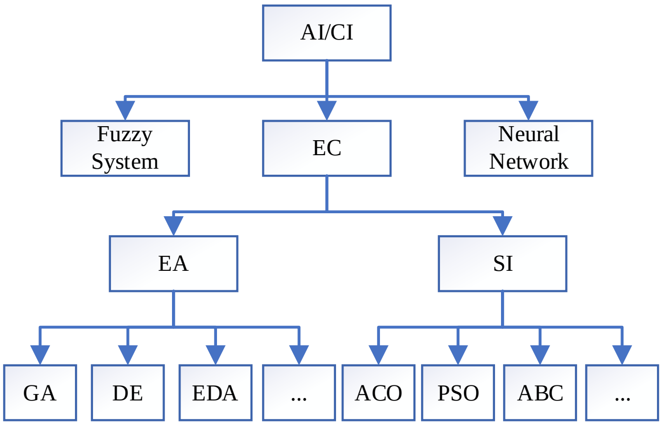

Fig. 1. The family of CI/AI.

The generic framework of an EC algorithm is illustrated as Fig. 2. Firstly, a population of individuals are randomly initialized, each individual represents for a solution to the problem. Then, the EC algorithm goes into a loop to approach the global optimum generation by generation. In every generation, the population reproduction and fitness evaluation are carried out. For example, in the DE algorithm, a new population is reproduced by using mutation and crossover operation. Then all the individuals in the new population are evaluated and each one compares with its parental individual, the one with better fitness value will

Z.-H. Zhan and S.-H. Wu are with the School of Computer Science and Engineering, South China University of Technology, 510006 Guangzhou, China, and also with the Pazhou Laboratory, 510330 Guangzhou, China. (Corresponding author: Zhi-Hui Zhan, Email: zhanapollo@163.com).

J. Zhang is with the Hanyang University, Ansan 15588, South Korea, and also with Victoria University, Melbourne, VIC 8001, Australia.

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102, in part by the Outstanding Youth Science Foundation under Grant 61822602, in part by the National Natural Science Foundations of China (NSFC) under Grant 61772207 and Grant 61873097, in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002, in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003, and in part by the Guangdong–Hong Kong Joint Innovation Platform under Grant 2018B050502006.

survive into the next generation. This is the simulation of the "survival of the fittest" in natural selection phenomenon. Another example is the PSO algorithm. It carries out the velocity update operation by using the guidance information of personally best position (*pBest*) and globally best position (*gBest*, i.e., the best one of all the *pBest*). Then it carries out the position update operation to reproduce a new population. Similar to DE, the PSO also follows the "survival of the fittest" in natural selection principle to update the *pBest* if the new position of the individual has better fitness value than its *pBest* position. Moreover, the *gBest* is updated if a new global better position has been found. Therefore, in EC algorithms, the evolution process is driven by the fitness values of the individuals. Better individuals will survive and will be used to reproduce new individuals. Therefore, the new individuals are expected to have general better survival ability than old individuals. This way, the EC algorithms can drive the individuals to be better and better, so as to approach the global optimum generation by generation.

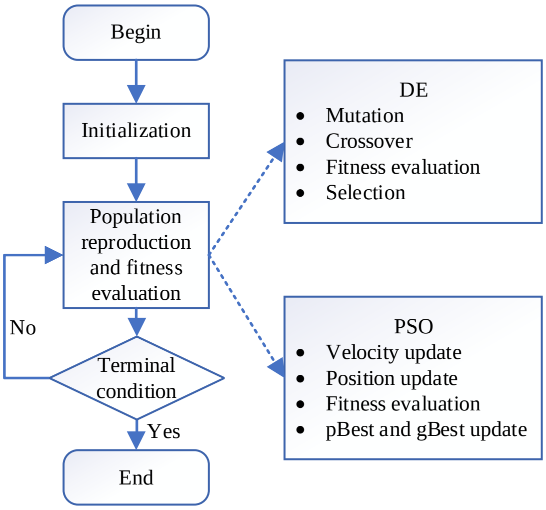

Fig. 2. The generic framework of EC algorithm.

Therefore, the fitness value evaluated by the objective function plays a vital role in the EC algorithm when solving optimization problem and it is always a default assumption in EC. However, some real-world application scenarios do not hold this assumption because the objective function is privacy-preserved. That is to say, the algorithm user does not want to disclose the objective function to the algorithm designer. We name this kind of optimization problem as privacy-preserving optimization problem (PPOP) in this paper. The PPOP is a new emerging with seldom study but is also a challenging research topic in EC community.

Even though EC algorithms have achieved great successes in complex optimization fields like large-scale optimization [6][7], dynamic optimization [8], multi-modal optimization [9][9]-[12], multi-/many-objective optimization [13]-[15], and expensive optimization [16]-[18], the research into PPOP progresses very slowly. Indeed, it is really difficult to make the EC algorithms still work if we do not know the objective function or the fitness value information. More difficult, as will be discussed Section II-B, there are some kinds of PPOP even needs to protect the variables of the problem, or both the variables and fitness of the problem. That is, if the algorithm user does not want to tell the algorithm designer the exact what to optimize, how can the algorithm designer do? This is still an open problem in using EC algorithm for solving PPOP. In this paper, we focus on the PPOP that is with fitness preserved.

For solving the fitness-preserving PPOP, a key issue is how to balance the privacy-preserving in the algorithm user and the information required by the algorithm designer. The user does not want to disclose the objective function or the fitness information due to the privacy preserving, but the designer needs this information to drive the evolution of the EC algorithm. A trade-off strategy may be the rank. For the algorithm designer, as most of the EC algorithms only use the fitness values to compare the individuals to determine which ones are better, the fitness values themselves are in fact not necessary, but the rank information of the fitness values is necessary and sufficient. On the other hand, if we encrypt the fitness values as rank and send such non-private information to the algorithm designer, the privacy of the algorithm user can be protected.

Therefore, this paper proposes a rank-based cryptographic function (RCF) to protect the fitness value information. Specially, the RCF is adopted by the algorithm user to encrypt the fitness values of all the individuals as rank so that the algorithm designer does not know the exact objective function nor the fitness information but only the rank information. Nevertheless, the RCF can protect the privacy of the algorithm user but still can provide sufficient information to the algorithm designer to drive the EC algorithm. Therefore, the RCF can make a balance between the privacy-preserving and the algorithm efficiency.

The contributions of our work include:

(1) As far as we know, it is the first time or one of the few works to try to extend the EC algorithms to solve the PPOP. This is a new emerging with seldom study but is also a challenging research topic in EC community.

(2) The definition of the PPOP and its variants has been discussed. This can bring a set of new challenging optimization problems into the EC community and can inspire lots of following works on using EC algorithms to solving various PPOP.

(3) The paper proposes a novel and innovative RCF to protect the objective function and fitness value information of the PPOP, which can protect the privacy of the algorithm user but still can provide sufficient information to the algorithm designer to drive the EC algorithm.

(4) The RCF privacy-preserving method has been applied to both PSO and DE, with the experimental results showing that they can solve the PPOP without performance loss. This will inspire more efficient future research works on this topic.

The rest of the paper is organized as follows. Some related works on privacy preserving optimization and EC, and the definition of PPOP are discussed in Section II. Section III describes the EC algorithms for PPOP. Section IV presents the experimental studies. In Section V, conclusions of this paper and future work are given.

II. Related Works and Problem Definition

A. EC and Privacy Preservation

In the literature, there are some researches on the combination of EC algorithms and the privacy preserving. These researches can be generally classified into two categories. One is using EC algorithms to solve special problems for better privacy preservation (i.e., solve the optimization problem in privacy preservation). The other is adopting privacy-preserving techniques to guarantee the information safety during the optimization process of EC algorithms.

In the first category, the researches consider to use EC to solve optimization problems in privacy preserving. The most representative work is the privacy preserving issue in the association rule mining problem. In specific, the association rule mining needs to discover frequent patterns and associations among sets of items in the databases and to protect the privacy of the rules. Dehkordi *et al.* [19] introduced a new multi-objective method for hiding sensitive association rules based on GA. The main purpose of this method was to fully support security of database and keep the utility of mined rules. Keshavamurthy *et al.* [20] proposed a local GA-based frequent pattern mining in the proposed system for privacy preservation in distributed databases. Krishnamoorthy *et al.* [21] proposed a system for association rule mining to hide a group of interesting patterns which contains sensitive knowledge such that modifications have minimum side effects like the number of modifications. Then, the PSO was adopted to minimize the number of modifications. Wu *et al.* [22] presented a multi-objective algorithm using a grid-based method PSO to find optimal solutions as candidates for preserving the privacy in data mining. To hide the association rules, Telikani *et al.* [23] proposed the improved binary ABC algorithm to enhance exploitation by designing a new neighborhood generation mechanism to balance between exploration and exploitation. In the e-healthcare area, Swathi *et al.* [24] proposed a privacy-preserving collaborative e-healthcare system that connected and integrated patients or caretakers into different groups. The recommended system used PSO to cluster e-profiles based on their similarities. In the Internet of Things area, Lin *et al.* [25] proposed a multi-objective ACO approach with transaction deletion to secure confidential and sensitive information. Nevertheless, there are also some works by using other optimization techniques rather than EC to solve the optimization problem in privacy preservation [26].

In the second category, the researches consider privacy preservation issues in the EC algorithm. That is, the intermediate data such as fitness value and decision variables in the optimization process may contain sensitive information of the user. For GA, the optimization process requires large storage and computing resources, it is well motivated to outsource the solving process of GA to the cloud server. However, the algorithm user would never want his data to be disclosed to cloud server. Thus, it is necessary for the user to encrypt the data before transmitting them to the server. But the user will encounter a new problem. The arithmetic operations we are familiar with cannot work directly in the ciphertext domain. To address this issue, Jiang *et al.* [27] proposed a privacy-preserving outsourced GA based on homomorphic encryption. Han *et al.* [28] proposed a protocol for secure GA in the scenario that two parties, each holding an arbitrarily partitioned data set, seek to perform GAs to discover a better set of rules without leaking their own private data. Sakuma *et al.* [29] proposed a protocol for local search and GA for the distributed traveling salesman problem.

Apart from the EC algorithms, the privacy preserving techniques also have been used in the traditional distributed optimization, which is usually based on gradient-based method. To enable privacy preserving requirement in decentralized optimization, Zhang *et al.* [30] proposed a privacy-preserving decentralized optimization approach based on alternating direction method of multipliers and partially homomorphic cryptography. For the distributed gradient-based optimization problem, Lu *et al.* [31] considered how to securely compute given functions and which functions should be computed in the first place. The authors proposed a multiparty secure computation algorithm based the homomorphic encryption to address the first issue and identified a class of functions which can be securely computed to address the second issue. For privacy preservation, Mao *et al.* [32] proposed a privacy preserving distributed optimization algorithm over time-varying directed communication networks by adding conditional noises to the exchanged states to solve the economic dispatch problem.

To summarize, the existing researches on the combination of EC algorithms and the privacy preserving mainly focus on the protection of the decision variables' data. However, there is another concern on the protection of the fitness value, which haven't received enough research attention. In the next subsection, the details of the different types of PPOP will be discussed.

B. PPOP

This section defines the PPOP and its variants. As shown in Fig. 3, according to the private content to be protected, the PPOP can be categorized as variable-preserving PPOP, fitness-preserving PPOP, and variable-fitness-preserving PPOP. In the variable-preserving PPOP, the algorithm user does not want to tell the algorithm designer what they really want to optimize (i.e., the variables of the optimization problem), but only tell the designer some information about the variables and of course have been encrypted. For example, to protect the privacy of variables in distributed optimization process, homomorphic encryption techniques are often adopted [30]-[32]. On the other hand, in the fitness-preserving PPOP, the user can tell the designer what they want to optimize, but does not want to tell the designer the objective function nor the fitness values of the individuals. For the variable-fitness-preserving PPOP, both the variables and fitness are privacy protected. In this paper, we focus on the fitness-preserving PPOP. That is, if the user does not give the objective function and wants to protect the fitness value, how can the designer use the EC algorithm to solve such a PPOP?

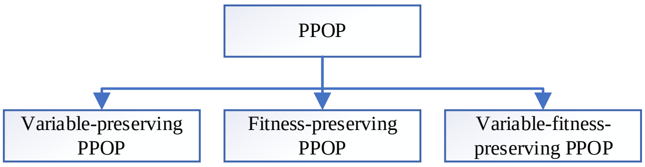

Fig. 3. The illustration of the PPOP variants.

III. EC for PPOP

A. General EC Framework with RCF

In order to drive the EC algorithm to search for the global optimum, the EC user has to provide the fitness information to the EC designer. In the case of PPOP, the fitness information of individual is encrypted via the RCF method by a cryptographic function (i.e., $c(x)$). This way, the objective function (i.e., $f(x)$) can be protected.

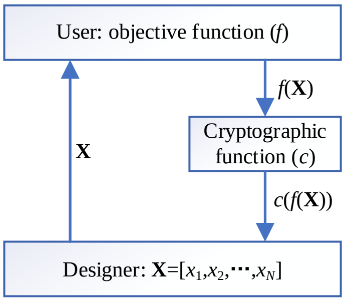

Fig. 4. The interaction between the user and the designer in RCF.

As shown in Fig. 4, in every generation (i.e., every interaction round between the user and the designer), the designer sends the population of individuals (i.e., **X**) to the user. Then, the user calculates the fitness values of each individual in **X** via the objective function. However, before sending the fitness information to the designer, cryptography is needed. Nevertheless, the cryptography can not be too strong that all the information is masked. Otherwise, the EC algorithm can not work without any useful information. Therefore, the paper proposes the RCF to protect the fitness value information but holds the rank information. That is, the fitness information is encrypted by the RCF as:

$$c(f(x)) \tag{1}$$

where x is the individual (i.e., the solution), the f is the objective function, and the c is the cryptographic function that returns the rank of the individual according to the fitness $f(x)$.

The RCF process may lead to duplicate fitness evaluation when the input individual have been evaluated before. To release the computational burden of user, we introduce a local database to store the evaluated individual. When the user receives individual from the designer, the user check if the individual is in the local database. If the individual is in the local database, the fitness value will be directly retrieved and used. Otherwise, the fitness of the new individual will be calculated to put forward the further RCF process.

Algorithm 1: PPPSO

```
Process of Designer in PPPSO
Begin
    Randomly initialize the population $\boldsymbol{X}$;
    $FEs=0$;
    $\boldsymbol{pBest}=\boldsymbol{X}$;
    Send the $\boldsymbol{pBest}$ to User;
    Receive the RCF and the updated $FEs$ from User;
    Determine the $\boldsymbol{gBest}$ as the $\boldsymbol{x}$ with smallest rank $c(f(\boldsymbol{pBest}))$;
    While $FEs \leqslant$ MaxFEs
        Update velocity and position using (2) and (3);
        For each particle $i$
            Send the current $\boldsymbol{x}_i$ and $\boldsymbol{pBest}_i$ to User;
            Receive the RCF and the updated $FEs$ from User;
            If $c(f(\boldsymbol{x}_i))<c(f(\boldsymbol{pBest}_i))$
                $\boldsymbol{pBest}_i=\boldsymbol{x}_i$;
                Send the current $\boldsymbol{pBest}_i$ and $\boldsymbol{gBest}$ to User;
                Receive the RCF and the updated $FEs$ from User;
                If $c(f(\boldsymbol{gBest}))<c(f(\boldsymbol{pBest}_i))$
                    $\boldsymbol{gBest}=\boldsymbol{pBest}_i$;
                End If
            End If
        End For
    End While
End

Process of User in PPPSO
Begin
    While not stop
        Receive a list of particles and current $FEs$ from Designer;
        For each particle $\boldsymbol{x}$ in the list
            If the particle $\boldsymbol{x}$ is in the local database
                Directly use its fitness $f(\boldsymbol{x})$ from local database;
            Else
                Evaluate the fitness and get $f(\boldsymbol{x})$;
                Insert the data $(\boldsymbol{x}, f(\boldsymbol{x}))$ into local database;
                $FEs=FEs+1$;
            End If
        End For
        Get $c(f(\boldsymbol{x}))$ of the particles in the list according to $f(\boldsymbol{x})$;
        Send the RCF and the updated $FEs$ to Designer;
    End While
End
```

B. PSO for PPOP

The pseudocode of PSO in solving the PPOP (namely PPPSO) is presented as **Algorithm 1** which includes the process of the algorithm Designer and the process of the algorithm User.

It should be noted that the PPPSO acts similar to that when it solves traditional optimization problem except that the fitness evaluation process is completed by the user, and only some rank information is sent back to the designer. Especially, after the initialization, the *pBest* of all individuals are set the same as their corresponding positions. Then all the *pBest* are sent to the User. The User calculates the fitness values of all the *pBest* and stores the *pBest* with the index and fitness value in the local database. However, the User does not tell the Designer the exact fitness values of the *pBest*, but only tells the Designer which *pBest* is the *gBest*. Then the PPPSO goes into the loop. In every generation, the Designer uses *pBest*'s and *gBest*'s positions to update the velocity and position of each individual x_i as:

$$v_i^d = \omega v_i^d + c_1 rand_1^d (pBest_i^d - x_i^d) + c_2 rand_2^d (gBest^d - x_i^d) \quad (2)$$

$$x_i^d = x_i^d + v_i^d \quad (3)$$

where d denotes the dimension of the position, ω is the self-learning coefficient, $rand_1$ and $rand_2$ are two uniformly generated number in [0,1], c_1 and c_2 are two coefficients for exemplar learning.

After all the new positions have been produced, the Designer sends all the new $\boldsymbol{x}$ positions to the User. The User calculates the fitness values of all the x and compares them with the *pBest* that are stored in the local database of the User according to the index pair by pair. After the comparison, the User updates the *pBest* in the local database and sends all the *pBest* to the Designer. Moreover, the *gBest* among all the *pBest* is also told to the Designer.

C. DE for PPOP

The pseudocode of DE in solving the PPOP (namely PPDE) is presented as **Algorithm 2** which also includes the process of the algorithm Designer and the process of the algorithm
User.

Algorithm 2: PPDE

Process of Designer in PPDE

```
Begin
1.  Randomly initialize the population $\boldsymbol{X}$;
2.  $FEs$=0;
3.  Send the population $\boldsymbol{X}$ to User;
4.  Receive the RCF and the updated $FEs$ from User;
5.  Determine the $\boldsymbol{gBest}$ as the $\boldsymbol{x}$ with smallest rank $c(f(\boldsymbol{pbest}))$;
6.  While $FEs \leqslant$ MaxFEs
7.      For each individual $i$
8.          Mutation and crossover using (4) and (5);
9.          Send the current $\boldsymbol{x_i}$ and $\boldsymbol{u_i}$ to User;
10.         Receive the RCF and the updated $FEs$ from User;
11.         If $c(f(\boldsymbol{u_i}))<c(f(\boldsymbol{x_i}))$
12.             $\boldsymbol{x_i}=\boldsymbol{u_i}$;
13.             Send the current $\boldsymbol{x_i}$ and $\boldsymbol{gBest}$ to User;
14.             Receive the RCF and the updated $FEs$ from User;
15.             If $c(f(\boldsymbol{x_i}))<c(f(\boldsymbol{gBest}))$
16.                 $\boldsymbol{gBest}=\boldsymbol{x_i}$;
17.             End If
18.         End If
19.     End For
20. End While
End
```

Process of User in PPDE

```
Begin
1.  While not stop
2.      Receive a list of individuals and current $FEs$ from Designer;
3.      For each individual $\boldsymbol{x}$ in the list
4.          If the individual $\boldsymbol{x}$ is in the local database
5.              Directly use its fitness $f(\boldsymbol{x})$ from local database;
6.          Else
7.              Evaluate the fitness and get $f(\boldsymbol{x})$;
8.              Insert the data ($\boldsymbol{x}$, $f(\boldsymbol{x})$) into local database;
9.              $FEs$=$FEs$+1;
10.         End If
11.     End For
12.     Get $c(f(\boldsymbol{x}))$ of the individuals in the list according to $f(\boldsymbol{x})$;
13.     Send the RCF and the updated $FEs$ to Designer;
14. End While
End
```

In the beginning, a population of individuals is randomly initialized. Then, all the individuals x are sent to the User for fitness evaluation. The User will also store all the x together with its index and fitness value in the local database, but will only tell the Designer which x is the best one. Then, the DE goes into the loop. In every generation, each individual produces its offspring v_i via the mutation operation as

$$v_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) \quad (4)$$

where r_1, r_2, and r_3 are randomly selected indexes in the population and F is the scaling factor, and the crossover operation as

$$u_{i,d} = \begin{cases} v_{i,d} & \text{if } rand_{i,d} \le Cr \text{ or } d = d_{rand} \\ x_{i,d} & \text{otherwise} \end{cases} \quad (5)$$

where u_i is the trial individual, $rand_{i,d}$ is uniformly sampled from [0,1], Cr is the crossover parameter, and d_{rand} is a randomly selected dimension.

After all the individuals have produced their offspring u, all the u will be sent to the User. The User evaluates all these u and compares them with those x according to the index pair by pair. The user will tell the designer for each individual i, whether the x_i or the u_i is better. Then the Designer can carry out the selection operation as lines 11 to 12 in **Algorithm 2**. Moreover, the User also stores the new better x positions and tells the Designer which one is the best position.

TABLE I THE 13 TEST FUNCTIONS

Function name	Test function	Search Space
Sphere	$f_1(x)=\sum_{d=1}^{D} x_d^2$	$[-100,100]^D$
Schwefel's P2.22	$f_2(x)=\sum_{d=1}^{D}\lvert x_d\rvert+\prod_{d=1}^{D}\lvert x_d\rvert$	$[-500,500]^D$
Quadric	$f_3(x)=\sum_{d_1=1}^{D}(\sum_{d_2=1}^{d_1} x_{d2})^2$	$[-100,100]^D$
Rosenbrock	$f_4(x)=\sum_{d=1}^{D-1}[100(x_{d+1}-x_d^2)^2+(x_d-1)^2]$	$[-10,10]^D$
Step	$f_5(x)=\sum_{d=1}^{D}(\lfloor x_d+0.5\rfloor)^2$	$[-10,10]^D$
Quadric Noise	$f_6(x)=\sum_{d=1}^{D} dx_d^4+random[0,1)$	$[-1.28,1.28]^D$
Schwefel	$f_7(x)=\sum_{d=1}^{D} -x_d \sin(\sqrt{\lvert x_d\rvert})+418.9829\cdot D$	$[-500,500]^D$
Rastrigin	$f_8(x)=\sum_{d=1}^{D}[x_d^2-10\cos(2\pi x_d)+10]$	$[-500,500]^D$
Noncontinuous Rastrigin	$f_9(x)=\sum_{d=1}^{D}[y_d^2-10\cos(2\pi y_d)+10]$ where $y_d=\begin{cases} x_d & \lvert x_d\rvert<0.5 \\ \frac{round(2x_d)}{2} & \lvert x_d\rvert\ge 0.5 \end{cases}$	$[-5.12,5.12]^D$
Ackley	$f_{10}(x)=-20\exp(-0.2\sqrt{1/D\sum_{d=1}^{D} x_d^2})$ $-\exp(1/D\sum_{d=1}^{D}\cos 2\pi x_d)+20+e$	$[-50,50]^D$
Griewank	$f_{11}(x)=1/4000\sum_{d=1}^{D} x_d^2-\prod_{d=1}^{D}\cos(x_d/\sqrt{d})+1$	$[-600,600]^D$
Weierstrass	$f_{12}(x)=\sum_{d=1}^{D}(\sum_{k=0}^{k_{\max}}[a^k\cos(2\pi b^k(x_d+0.5))])$ $-D\sum_{k=0}^{k_{\max}}[a^k\cos(2\pi b^k\cdot 0.5)]$ where $a=0.5, b=3, k_{\max}=20$	$[-0.5,0.5]^D$
Generalized Penalized	$f_{13}(x)=\frac{\pi}{D}\{10\sin^2(\pi y_1)+\sum_{d=1}^{D-1}(y_d-1)^2[1+10\sin^2(\pi y_{d+1})]$ $+(y_D-1)^2\}+\sum_{d=1}^{D}u(x_d,10,100,4)$ where $y_d=1+\frac{1}{4}(x_d+1), u(x_d,a,k,m)=\begin{cases} k(x_d-a)^m & x_d>a \\ 0 & -a\le x_d\le a \\ k(-x_d-a)^m & x_d<-a \end{cases}$	$[-50,50]^D$

TABLE II
THE EXPERIMENTAL RESULTS OF THE EC ALGORITHMS AND THE PRIVACY-PRESERVING EC ALGORITHMS

Function index	DE	PPDE	PSO	PPPSO
1	6.24e-23(5.17e-23)	6.24e-23(5.17e-23)=	9.79e-52(4.11e-51)	1.13e-52(4.18e-52)=
2	3.58e-11(2.36e-11)	3.58e-11(2.36e-11)=	3.00e+00(5.26e+00)	3.00e+00(5.26e+00)=
3	1.63e-02(1.15e-02)	1.63e-02(1.15e-02)=	1.22e+03(2.88e+03)	1.22e+03(2.88e+03)=
4	5.93e+00(1.04e+00)	5.93e+00(1.04e+00)=	4.00e+02(1.80e+03)	4.00e+02(1.80e+03)=
5	0.00e+00(0.00e+00)	0.00e+00(0.00e+00)=	0.00e+00(0.00e+00)	0.00e+00(0.00e+00)=
6	5.12e-01(2.79e-01)	8.13e-01(3.42e-01)=	5.50e-01(2.74e-01)	1.38e+00(6.49e-01)=
7	5.51e+03(7.18e+02)	5.63e+03(6.03e+02)=	2.65e+03(5.24e+02)	2.72e+03(4.65e+02)=
8	1.71e+02(9.75e+00)	1.61e+02(1.43e+01)=	3.48e+01(1.41e+01)	3.67e+01(1.82e+01)=
9	1.40e+02(1.57e+01)	1.45e+02(1.52e+01)=	2.43e+01(1.61e+01)	1.17e+01(1.14e+01)=
10	1.79e+01(5.49e+00)	1.19e+01(9.71e+00)=	1.28e-14(4.83e-15)	7.55e-15(0.00e+00)=
11	0.00e+00(0.00e+00)	3.29e-04(1.77e-03)=	1.80e-02(1.85e-02)	1.20e-02(1.60e-02)=
12	2.74e-07(2.97e-07)	2.64e-07(4.04e-07)=	3.45e-01(8.53e-01)	3.89e-01(8.74e-01)=
13	8.88e-24(7.85e-24)	1.43e-23(1.46e-23)=	6.91e-03(2.59e-02)	1.38e-02(4.43e-02)=

IV. EXPERIMENTAL STUDIES

A. *Experimental Settings*

In order to test the performance of the proposed PPPSO and PPDE for solving PPOP, we carry out the numerical experiment on 13 single-objective optimization functions. The 13 functions are commonly used in the literature for testing the performance of EC algorithms [2]. The details of the tested functions are shown in Table I. The 13 tested functions vary in multiple aspects that include unimodal and multimodal functions, noiseless and noisy functions, continuous and noncontinuous functions. The dimensionality *D* of the functions is 30. It should be noted that the proposed PPPSO and PPDE have no access to the original fitness value of the solution. Instead, PPPSO and PPDE get the RCF value from the user to drive the evolution. The PPPSO used in the paper is a standard PSO which adopts global communication topology, and the linearly decreasing self-learning weight from 0.9 to 0.4. The DE used in the paper is the DE/rand/1, which adopts the mutation strategy as Eq. (4). The scaling factor *F* is 0.5 and the crossover rate *Cr* is 0.9. The population size is set as 20 for the PPPSO and 100 for the PPDE. The terminal condition is set as the FEs reaching the maximum FEs allowed, which is set to 2e5 for all algorithms in all runs. The optimization process will run 30 times for each function independently to reduce the statistical error caused by the randomness of the EC algorithms. The final *gBest* fitness value averaged over 30 independent runs and the corresponding standard deviation are considered as the performance metric for the algorithm.

B. *Experimental Results of PPPSO and PPDE*

The experimental results are shown in Table II. More specifically, the average and the standard deviation of the final *gBest* fitness values are shown. Moreover, the Wilcoxon's rank sum test is carried out to validate if the original EC algorithms (e.g. PSO) perform significantly different fromr the privacy-preserving ones (e.g. PPPSO). The results of the Wilcoxon's rank sum test are shown in the columns of the proposed PPPSO and PPDE, given by a symbol that is '=' if there is significant difference. Otherwise, the symbol is '≠'. From Table II, it can be observed that the optimization performance of the original EC algorithms have no significant difference with the proposed privacy-preserving EC algorithms.

Therefore, the experimental results emphasize our contribution that the proposed EC algorithm in solving PPOP can work without performance loss.

V. CONCLUSION

This paper makes the first study on using EC algorithms to solve the PPOP with fitness information protection. Specially, the PPOP is difficult that the algorithm user wants to protect the objective function or the fitness values of the individuals while the algorithm designer needs the fitness information to drive the EC algorithms. To solve this contradictory issue, the RCF is proposed.

The RCF is very simple but efficient in both protecting the fitness privacy information of the algorithm user and providing sufficient information for the algorithm designer. Specially, the fitness values of the individuals have been encrypted into rank information, so that it is not necessary to tell the algorithm designer the objective function or the fitness value. Nevertheless, the algorithm designer can use the rank information to determine which solutions are better and can survive into next generation. Therefore, the RCF-based EC algorithm is suitable for solving PPOP.

We have applied the RCF to both PSO and DE and designed the two algorithms for solving PPOP. Experimental results shown that RCF-based privacy preserving PSO and DE can solve the PPOP without performance loss.

In summary, this paper opens a new research idea to bridge the EC algorithms with the PPOP, and has gained some promising results on the studies of RCF-based PSO and RCF-based DE. In the future work, other kinds of EC algorithms based on the RCF will be designed to solve the PPOP. Moreover, other types of PPOP, e.g., the PPOP with variables-preserving, the PPOP with both variables and fitness preserving, and the PPOP with stronger privacy-preserving level (i.e.., what if the algorithm user even does not want to the algorithm designer the rank information?), will be studied. Also, as the different difficulties of various PPOP variants, more cryptographic methods will be explored to better enable the EC algorithms to work and work well in these kinds of PPOPs.

References

[1] Z. -H. Zhan *et al.*, "Matrix-Based Evolutionary Computation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, to be published, doi: 10.1109/TETCI.2020.3047410.

[2] Z. H. Zhan, *et al.*, "Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Trans. Parallel and Dist. Syst*., vol. 28, no. 3, pp. 704–716, March. 2017.

[3] Z. H. Zhan, Z. J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Trans. Cybern*., to be published, doi: 10.1109/TCYB. 2019.2944873.

[4] Z. G. Chen, Y. Lin, Y. J. Gong, Z. H. Zhan, and J. Zhang, "Maximizing lifetime of range-adjustable wireless-sensor networks: a neighborhood-based estimation of distribution algorithm," *IEEE Trans. Cybern*., to be published, doi: 10.1109/TCYB.2020.2977858.

[5] X. Xia *et al.*, "Triple Archives Particle Swarm Optimization," *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 4862-4875, Dec. 2020, doi: 10.1109/TCYB.2019.2943928.

[6] Z. J. Wang, Z. H. Zhan, S. Kwong, H. Jin, and J. Zhang, "Adaptive granularity learning distributed particle swarm optimization for large-scale optimization," *IEEE Trans. Cybern*., to be published, doi: 10.1109/TCYB.2020. 2977956.

[7] X. Zhang, K. Du, Z. Zhan, S. Kwong, T. Gu, and J. Zhang, "Cooperative coevolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2019.2937565.

[8] X. -F. Liu *et al.*, "Neural network-based information transfer for dynamic optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1557-1570, May 2020, doi: 10.1109/TNNLS.2019.2920887.

[9] Z. J. Wang, et al., "Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems," *IEEE Trans. Evol. Comput*., vol. 22, no. 6, pp. 894–908, Dec. 2018.

[10] Z. J. Wang, Z. H. Zhan, Y. Lin, W. J. Yu, H. Wang, S. Kwong, and J. Zhang, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Trans. Evol. Comput*., vol. 24, no. 1, pp. 114-128, Feb. 2020.

[11] H. Zhao, Z. H. Zhan, Y. Lin, X. F. Chen, X. N. Luo, J. Zhang, S. Kwong, and J. Zhang, "Local binary pattern based adaptive differential evolution for multimodal optimization problems", *IEEE Trans. Cybern*., to be published, doi: 10.1109/TCYB.2019.2927780.

[12] Z. G. Chen, Z. H. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Trans. Evol. Comput*., to be published, doi: 10.1109/TEVC. 2019.2944180.

[13] Z. H. Zhan, J. Li, J. Cao, J. Zhang, H. S. Chung, and Y. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, April 2013.

[14] X. Liu, Z. H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, Aug. 2019.

[15] Z. G. Chen *et al.*, "Multiobjective cloud workflow scheduling: a multiple populations ant colony system approach," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912–2926, 2019.

[16] J. -Y. Li, Z. -H. Zhan, C. Wang, H. Jin, and J. Zhang, "Boosting data-driven evolutionary algorithm with localized data generation," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 923-937, Oct. 2020, doi: 10.1109/TEVC.2020.2979740.

[17] S. -H. Wu, Z. -H. Zhan, and J. Zhang, "SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, to be published, doi: 10.1109/TEVC.2021.3051608.

[18] J. -Y. Li, Z. -H. Zhan, H. Wang, and J. Zhang, "Data-driven evolutionary algorithm with perturbation-based ensemble surrogates," *IEEE Transactions on Cybernetics*, to be published, doi: 10.1109/TCYB.2020.3008280.

[19] M. N. Dehkordi, K. Badie, and A. K. Zadeh, "A novel method for privacy preserving in association rule mining based on genetic algorithms," *Journal of Software*, vol. 4, no. 6, pp. 555-562, 2009.

[20] B. N. Keshavamurthy, A. M. Khan, D. Toshniwal, "Privacy preserving association rule mining over distributed databases using genetic algorithm," *Neural Comput. & Applic.*, vol. 22, no. 1, pp. 351–364, March. 2013.

[21] S. Krishnamoorthy, G. S. Sadasivam, M. Rajalakshmi, K. Kowsalyaa, and M. Dhivya, "Privacy preserving fuzzy association rule mining in data clusters using particle swarm optimization," *International Journal of Intelligent Information Technologies (IJIIT)*, vol. 13, no. 2, pp. 1-20, 2017.

[22] T.-Y. Wu, J. C.-W. Lin, Y. Zhang, and C.-H. Chen, "A grid-based swarm intelligence algorithm for privacy-preserving data mining," *Applied Sciences*, vol. 9, no. 4, p. 774, 2019.

[23] A. Telikani, A. H. Gandomi, A. Shahbahrami, and M. N. Dehkordi, "Privacy-preserving in association rule mining using an improved discrete binary artificial bee colony," *Expert Systems with Application*s, vol. 144, p. 113097, 2020.

[24] M. Swathi and K. Sreedhar, "A cloud-based privacy-preserving e-healthcare system using particle swarm optimization," in *Proceedings of the Third International Conference on Computational Intelligence and Informatics*, 2020: Springer, pp. 133-143.

[25] J. C.-W. Lin, G. Srivastava, Y. Zhang, Y. Djenouri, and M. Aloqaily, "Privacy preserving multi-objective sanitization model in 6G IoT environments," *IEEE Internet of Things Journa*l, 2020, doi: 10.1109/JIOT.2020.3032896.

[26] F. Chen, B. Bruhadeshwar, and A. X. Liu, "Cross-domain privacy-preserving cooperative firewall optimization," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 857-868, June 2013, doi: 10.1109/TNET.2012.2217985.

[27] L. Jiang and Z. Fu, "Privacy-preserving genetic algorithm outsourcing in cloud computing," *Journal of Cybersecurity*, vol. 2, no. 1, p. 49, 2020.

[28] S. Han and W. K. Ng, "Privacy-preserving genetic algorithms for rule discovery," in *International conference on data warehousing and knowledge discovery*, 2007: Springer, pp. 407-417.

[29] J. Sakuma and S. Kobayashi, "A genetic algorithm for privacy preserving combinatorial optimization," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 1372-1379.

[30] C. Zhang, M. Ahmad, and Y. Wang, "ADMM based privacy-preserving decentralized optimization," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 565-580, 2018.

[31] Y. Lu and M. Zhu, "Privacy preserving distributed optimization using homomorphic encryption," *Automatica*, vol. 96, pp. 314-325, 2018.

[32] S. Mao, Y. Tang, Z. Dong, K. Meng, Z. Y. Dong, and F. Qian, "A privacy preserving distributed optimization algorithm for economic dispatch over time-varying directed networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1689-1701, 2020.