IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 29, 2019, accepted October 22, 2019, date of publication November 5, 2019,
date of current version November 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951700

A Survey on Swarm Intelligence Search Methods Dedicated to Detection of High-Order SNP Interactions

SHOUHENG TUO, HAO CHEN, AND HAIYAN LIU
School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Corresponding author: Shouheng Tuo (tuo_sh@126.com)

This work was supported in part by the Natural Science Foundation of China under Grant 61571341, in part by the Ministry of Education through the Humanities and Social Science Project of China under Grant 19YJCZH148, in part by the Special Scientific Research Program of Education Department of Shaanxi Province under Grant 19JK0806, and in part by the Science Foundation of the Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing.

ABSTRACT Detecting high-order single-nucleotide polymorphism (SNP) interactions is of great importance for the discovery of pathogenic causes of human complex diseases. However, a considerable computing challenge exists in analyzing each SNP combination at a genome-wide scale. Swarm intelligence search (SIS) is an effective and efficient method for solving NP-hard problems and has been extensively researched for detecting high-order SNP interactions. In this review, we first analyze the strengths and limitations of existing methods such as exhaustive search using cluster computing and parallel computing, stochastic search and high-performance computing. Then, SIS algorithms for the detection of high-order SNP interactions are introduced in detail. The algorithms discussed are the genetic algorithm (GA), ant colony optimization (ACO), harmony search (HS), particle swarm optimization (PSO), differential evolution (DE), cuckoo search (CS), fish swarm (FS) and artificial bee colony (ABC). Finally, we discuss the characteristics and limitations of the involved methods and provide several suggestions for improving SIS algorithms to detect high-order SNP interactions.

INDEX TERMS Swarm intelligence, single-nucleotide polymorphisms, high-order SNP interaction, detection.

I. INTRODUCTION

With the rapid progress of genome sequencing technology, the cost of genome-wide sequencing has been greatly reduced, and genome-wide data volumes have increased rapidly, making genome-wide association studies (GWAS) widely involved in detecting single-nucleotide polymorphisms (SNPs) associated with complex human diseases.

During the past decade, thousands of associated SNPs have been successfully identified since the first GWAS for age-related macular degeneration was presented by Klein *et al.* [1], with the main focus being on individual SNPs that are isolated based on their contribution to disease status. However, SNPs identified in GWAS can only explain a small fraction of the heritability of complex diseases [2]. More recently, it has become widely recognized that high-order SNP interactions may contribute to a given pathogenicity. High-order SNP interactions represent a combination of multiple SNPs jointly affecting complex diseases either linearly or nonlinearly, the detection of which is of great significance for the discovery of pathogenic causes of human complex diseases.

The central point in the detection is how to accurately discriminate the relationships between high-order SNP interactions and disease states and how to quickly explore the high-order SNP interactions on a genome-wide scale, in which the largest challenge is to develop an efficient search method to resolve the combinatorial explosion of SNPs. An exhaustive search that has been widely applied to find pairwise SNP interactions requires a considerable computational cost for evaluating the association of all k-order ($k>2$) SNP combinations, which is obviously impractical for discovering high-order SNP interactions on a genome-wide scale.

To address this problem, some search techniques, such as high-performance computing and stochastic searches, have

The associate editor coordinating the review of this manuscript and approving it for publication was Xiangtao Li.

been proposed to speed up the detection of high-order SNP interactions. High-performance computing generally adopts supercomputers or parallel processing techniques, such as cluster computing and parallel computing, to accelerate the computation. Guo *et al.* proposed a dynamic clustering algorithm to detect high-order epistatic interactions based on cloud computing [3]. Goudey *et al.* analyzed high-order SNP interactions using an exhaustive method via high-performance computing [4]. Wan *et al.*, Yung *et al.*, and Gyenesei *et al.* employed Boolean bitwise operations and multithreaded parallelization to achieve excellent computational efficiency [5]–[7].

A stochastic search employs random sampling procedures to find high-order SNP combinations that are associated with disease status. The method aims to decrease the time complexity by reducing the number of SNP combinations needed to calculate the association with a phenotype. Zhang and Liu proposed a Markov chain Monte Carlo (MCMC) method to iteratively detect epistatic interactions by calculating the posterior association probability of a locus and its interaction partners with the disease [8]. Wang *et al.* used MCMC to detect high-order SNP interactions [9]. Han *et al.* applied a fast branch-and-bound algorithm and MCMC to screen SNPs [10]. The SNPHarvester algorithm employs a stochastic search to discover significant SNP groups [11].

High-performance computing attempts to evaluate the association of all k-order SNP combinations using high-performance computer systems, such as parallel computing and cloud computing. However, when k is greater than 3, the detection of k-order SNP interactions from genomic data with hundreds of thousands of SNPs is still ineffective due to the enormous computational burden. Although traditional stochastic search algorithms can discover some k-order SNP interactions, these algorithms are still insufficient for the detection of high-order SNP interactions when using posterior association probabilities of individual loci, especially for the detection of complex disease models with minimal or no marginal effects.

Attention is now turning to the discovery of high-order SNP interactions using SIS techniques. SIS is a collective behavior that mimics natural or artificial decentralized, self-organizing systems. SIS rapidly achieves an overall understanding in a complex environment through mutual communication and learning among individuals in the group and can easily be used to solve high-dimensional complex optimization problems. In engineering fields such as the physical design of very large-scale integrated circuits and large-scale scheduling and planning optimization problems, SIS techniques have been widely applied.

Compared to traditional optimization algorithms (e.g., gradient descent), SIS has the following advantages:

(1) SIS is a global optimization method because it does not depend on the initial search point and can escape from a local optimum when the population is trapped in a local search.

(c2) The objective function of SIS is not restricted to continuity and differentiability but may be expressed in any form.

(3) SIS is powerful for exploring a complex search space and has the ability to discover the global optimal solution rapidly via learning and communicating between individuals in a population.

For the problem of detecting high-order SNP interactions from high-dimensional space, it is beneficial to employ the SIS method to accelerate the search process by discovering some candidate k-order SNP combinations that have an association with disease status. SIS methods have a strong advantage in reducing the time complexity and can find candidate solutions without evaluating all k-order SNP combinations. During the past ten years, SIS methods have attracted wide attention in the study of epistasis analysis.

II. SNP INTERACTION

Let a set of SNP variables $X = \{x_1, x_2, \ldots, x_N\}$ indicate N SNP loci for n samples, and $Y = \{y_1, y_2, \ldots, y_J\}$ denotes the phenotype variable ($J = 2$ for disease models). The homozygous major allele, heterozygous allele and homozygous minor allele in the sample dataset are defined as 0, 1 and 2, respectively. For a k-order*SNP* combination, there are $I = 3^k$ genotype combinations. n_i is the number of samples in the dataset with SNP loci having the value of the i-th genotype combination, and n_{ij} represents the number of samples with the i-th genotype combination that are actually associated with disease state y_j.

Definition (High-Order SNP Interaction): Let $S_k = \{x_{s_1}, x_{s_2}, \ldots, x_{s_k}\}(1 < k < N, x_{s_i} \in X)$ be a set with k SNP loci. $f(S_k, Y)$ is a score function for evaluating the association between S_k and disease state Y. A *k-order* SNP combination S_k is said to be jointly interacted with Y if and only if $\forall S' \subset S_k \wedge f(S_k, Y) \succ f(S', Y)$ ($\succ$ is a binocular operator for comparing the association strength with disease) and is said to be strongly associated with Y if $f(S_k, Y) > \theta$ (θ is a threshold value). A k-order SNP combination S_k is called a k-order SNP interaction if and only if it is truly a disease-causing SNP combination with Y.

Mathematical Model:

To detect k-order SNP interactions using the SIS algorithm, the mathematical model can be expressed as follows:

$$\underset{X}{\textbf{maximum}} f(X, Y) \tag{1}$$

where $X = (x_{s_1}, x_{s_2}, \ldots x_{s_k})$ is a k-order SNP combination, $x_{s_i} \in \{x_1, x_2, \ldots x_N\}(1 \leq s_i \leq N; x_i \in \{1, 2, \ldots, N\}$, $i = 1, 2, \ldots, k)$. Y denotes the disease status (1 for case and 0 for control). $f(X, Y)$ is the scoring function (objective function) for evaluating the strength of association between X and Y.

This mathematical model is a combinatorial optimization model; it is very difficult from the whole genome to find a disease-causing SNP combination. The main reason is the enormous computational burden. The number of k-order SNP

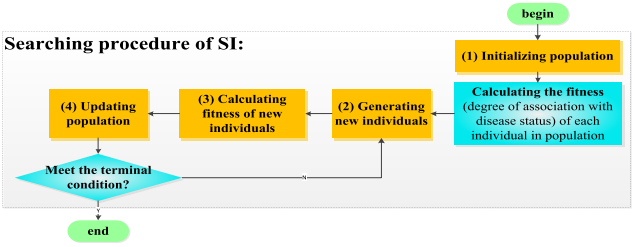

FIGURE 1. The search procedure of the SIS approach.

combinations for data with n SNPs is equal to $C_n^k \propto n^k$. Clearly, it is impractical to examine and evaluate the associations of all feasible k-order SNP combinations from a dataset with hundreds of thousands of SNPs using an exhaustive search algorithm (the time complexity is $O(N^k)$).

Therefore, the SIS algorithm is used to reduce the number of evaluations of the associations between k-order SNP combinations and disease status.

III. SIS METHODS

A. BASIC FRAMEWORKS AND DESIGN IDEAS OF SI

In SIS methods, one or multiple populations of candidate individual solutions to a search problem (an optimization problem) evolve toward better individuals. As Figure 1 shows, the mathematical model (see equation 1) should first be established for the optimization problem. The objective function (fitness function or scoring function) is designed to calculate the fitness of the solutions. Designing suitable encoding for individual solutions is also important. Generally, the mathematical model can be expressed as follows:

$$\begin{aligned} &\underset{X}{\textbf{Minimize}}\ f(X) \\ &\textbf{Subject to}: \begin{cases} g_j(X) \le 0, & j = 1, 2, \ldots, s \\ h_j(X) = 0, & j = 1, 2, \ldots, w \end{cases} \end{aligned}$$

where $f(X)$ is the objective function that is used to evaluate the candidate solution $X = (x_1, x_2, \ldots, x_n) \in S$ as the decision vector. $S = \prod_i^n \left[x_i^L, x_i^U\right]$, where n is the number of variables. x_i^L and x_i^U denote the lower and upper bounds of the decision variable $x_i (i \in \{1, 2, \ldots, n\})$. $g_j(X)$ is the j^{th} inequality constraint function, s is the number of inequality constraint functions. $h_j(X)$ represents the equality constraint function, and w is the number of equality constraint functions.

As shown in Figure 1, the process of SIS has four key steps:

1) Initializing a population of candidate solutions randomly in a search space and then calculating the fitness value of candidate solutions via an objective function $f(X)$.
2) Generating new individuals.
3) Calculating the fitness values of new individuals.
4) Updating the individuals in the population according to the rules of "survival of the fittest".
5) Repeating steps (2)-(4) until the terminal condition (i.e., the maximum number of objective evaluations) is reached.

B. GENETIC ALGORITHM (GA)

The genetic algorithm (GA), which imitates the biological evolutionary process of natural selection, is one of the most classic SIS methods and involves three key operators (mutation, crossover and selection) for evolution [12]. Mutation and crossover are used to generate new individual solutions called offspring individuals. A selection operator, such as tournament selection or roulette selection, chooses individual solutions from the offspring population and parent population through a fitness-based process. The GA is easily applied to the discovery of genetic models and epistasis detection. Moore *et al.* introduced a GA to discover complex genetic models in which a k-order genetic model was represented as a candidate solution. The objective function was defined by maximizing the variance in penetrance values and minimizing the variance in marginal penetrance values of genetic models [13], [14]. Shah and Kusiak employed a GA for gene/SNP selection [15]. Yang *et al.* proposed a multi-objective genetic ensemble algorithm (named the **GE** algorithm) to search for gene-gene interactions on a genome-wide scale [16]. In GE, the GA is used to find SNP subsets that are considered to be potential gene-gene and gene-environment interactions, and three evaluating methods (blocking integration strategy [17], majority voting [18], and double fault statistic [19]) are employed to calculate the fitness of individual solutions. Figure 2 presents a simple example of a GE for searching high-order SNP interactions. Mooney *et al.* proposed a GA algorithm, which is guided by the structure of a gene interaction network, to discover groups of SNP pairs that are jointly associated with bipolar disorder, and the chi-squared test was utilized as an objective function to evaluate the association between SNP pairs and disease status [20].

The traditional GA is a binary-coded algorithm that appears to be quite capable of determining the model of k-order SNP combinations as follows:

$$\begin{aligned} &\underset{X}{\text{minimum}}\ f(X, Y), X = (x_1, x_2, \ldots, x_n) \\ &\text{Subject to}: \sum_{i=1}^{n} x_i = k, \quad x_i \in \{0, 1\} \end{aligned}$$

where $f(X, Y)$ is the objective function for evaluating the association between SNP combination X and disease status Y. x_i denotes the i^{th} SNP, which means that if $x_i = 1$, the SNP was put into the k-order SNP combination; otherwise, $x_i = 0$.

To solve this model, the GA can be used to find the best SNP combinations that are associated with diseases. The crossover and mutation operators based on binary coding are employed to generate new SNP combinations. However, in practice, the total number (n) of SNPs in a genome is far greater than k, which makes the search for k-order disease-causing SNP combinations very slow for hundreds of thousands of SNPs when employing the binary-coded

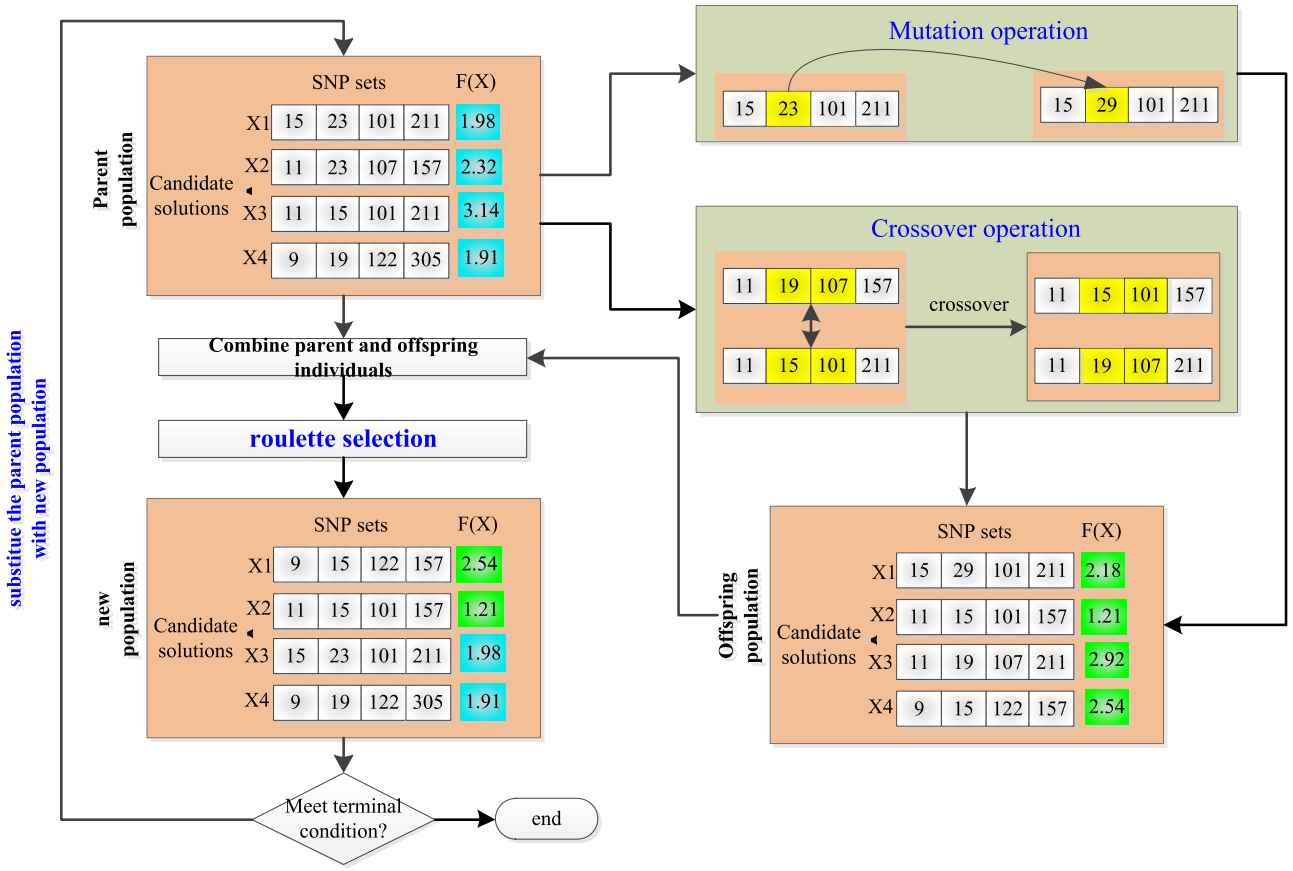

FIGURE 2. An example of a GE for detecting high-order SNP interactions. $X_i (i = 1,2,3,4)$ represents a candidate solution (4th-order SNP combination), and $F(X_i)$ is the fitness value (score representing association with disease status) of X_i. Each candidate solution of the parent population mutates with mutation probability *MR* and undergoes crossover with crossover probability *CR*.

GA algorithm. In addition, for two different order SNP combinations, k-order and s-order ($k \neq s$), their fitness values are incommensurable due to different dimensions. As an example, the chi-square test, Bayesian network and mutual information are utilized as objective functions. The score in machine learning (e.g., multifactor dimensionality reduction (MDR), logistic regression) used as the objective function to evaluate the association is comparable but very time consuming to generate.

Therefore, the real-value-coded GA algorithm may be more efficient than the binary-coded algorithm for detecting high-order SNP interactions; the goal is to detect k-order SNP interactions. The mathematical model can be expressed as equation (1) accordingly. In this case, the time complexity of real-coded GA is $O(k \times NP)$ (population initialization) + $O(T \times k \times NP)$ (crossover and mutation) + $O(T \times NP \log NP)$ (selection), where NP is the population size, and T is the maximum number of generations.

C. ANT COLONY OPTIMIZATION (ACO)

The ant colony optimization (ACO) algorithm was inspired by the foraging behavior of ants searching for an optimal path from their colony to a source of food [21]. The ants discover the shortest paths by cooperating and communicating with the pheromones they release; each ant perceives pheromones along the path and releases new pheromone trails. When facing multiple paths, the ants tend to choose a path that is marked with strong pheromones [22].

Over the past ten years, the ACO algorithm has received considerable attention and has become the most popular SIS algorithm for detecting high-order SNP interactions because it can easily detect high-order SNP interactions and is very efficient for exploring the shortest path from a network consisting of SNPs.

When using ACO to detect SNP interactions, a k-order SNP combination $\{SNP_1, SNP_2, \ldots, SNP_k\}$ denotes an "ant walking path" in one iteration. After the t^{th} iteration, the pheromone τ_{ij} of edge ($SNP_i \rightarrow SNP_j$) is updated as follows:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \#s \cdot \Delta\tau_{ij}(t), (i \neq j)$$

where $\tau_{ij}(t)$ denotes the pheromone of edge ($SNP_i \rightarrow SNP_j$) at iteration t. $\Delta\tau_{ij}(t)$ is the new additional pheromones of edge ($SNP_i \rightarrow SNP_j$), which is contributed by ants passing the edge ($SNP_i \rightarrow SNP_j$) at iteration t. ρ represents the evaporation coefficient between 0 and 1. $\#s$ is the number of ants passing the edge ($SNP_i \rightarrow SNP_j$) at iteration t.

Each ant chooses a walking direction (next SNP) according to the pheromone of the adjacent edge. For example,

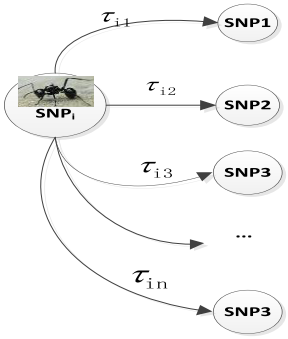

FIGURE 3. Ant path selection for detecting SNP interactions.

in Figure 3, an ant has reached SNP_i; next; it selects an unreached SNP_j with probability $\tau_{ij} \Big/ \sum_{j=1}^{n} \tau_{ij}$.

A detailed ACO algorithm for detecting SNP interactions was introduced by Shang *et al.* [23] and Jing and Shen [24]. Shang *et al.* reviewed the ACO algorithms that are applied to detect epistatic interactions and analyzed the strengths and limitations of the involved ACO methods in detail.

To analyze epistasis in human disease, Greene *et al.* combined expert knowledge and multifactor dimensionality reduction (MDR) with the ACO algorithm to quickly explore the epistatic interactions, and the expert knowledge was obtained from tuned relief (TuRF) [25]. Rekaya and Robbins employed an ACO algorithm (ACA) to analyze gene interactions; they used two-layer pheromones for ants to choose a path and logistic regression to evaluate the association between genotype and haplotype [26]. Wang *et al.* proposed a two-stage ACO algorithm (AntEpiSeeker) that aims to discover a highly suspected and reduced SNP set quickly in the first stage. The chi-squared test was used as an objective function; in the second stage, they conducted an exhaustive search of epistatic interactions within the SNP set [27]. Sulovari *et al.* integrated ACO into the MDR package and selected the SNPs that were associated with disease status; they utilized pathway studio scores as biological expert knowledge [28]. Shang *et al.* incorporated heuristic information into ACO to direct ants in the search process for improving the computational efficiency and solution accuracy [29]. Sun *et al.* adopted the fitness function Svalue, path selection and memory-based strategy to enhance the power [30], and then, introduced heuristic information in ACO for identifying epistasis [31]. Guan and Zhao proposed a self-adjusting ACO-based information entropy to identify epistatic interactions [32]. Shang *et al.* systematically reviewed 25 ACO-based epistasis interaction approaches [23].

To enhance the performance of identifying the various disease models, multiple evaluation criteria have been utilized as objective functions in ACO. Jing and Shen presented a multi-objective ACO algorithm for SNP epistasis detection (MACOED) that employs both logistic regression and Bayesian network methods to evaluate the association between the SNP combination and disease status [24].

To improve the computational performance, a GPU-based ACO algorithm was introduced by Sinnott-Armstrong *et al.* for the analysis of genome-wide epistasis, in which MDR was utilized as an objective function [33]. Christmas *et al.* adopted ACO to identify the genetic variant association with type 2 diabetes [34]. Sapin *et al.* employed the ACO algorithm to discover small numbers of SNPs on a genome-wide scale and then evaluated the association of these SNPs with phenotype using a decision tree or contingency table model [35]. In addition, they incorporated the tournament path selection and tabu list into ACO for the analysis of GWAS [36]. Yuan *et al.* introduced a fast adoptive ACO algorithm (FAACOSE) to detect SNP interactions [37].

ACO has a natural advantage when it is used to detect high-order SNP interactions; however, it requires a large-size ant population and leads to a high computational cost. The time complexity of ACO for detecting k-order SNP interactions from a dataset with n individuals and N SNP markers is $O(k \times NP)$(initialization) $+O(T \times k \times NP \times N)$(updating pheromone of each edge).

D. HARMONY SEARCH (HS)

The harmony search (HS) algorithm, inspired by the improvisation process of jazz musicians, is a very simple evolutionary algorithm [38], [39]. The initial goal of HS was to solve discrete optimization problems. The HS algorithm has excellent performance and efficiency in solving combinatorial optimization problems such as structural design and traffic routing. In HS, each harmony corresponds to a vector consisting of k decision variables. Good harmonies constitute a harmony memory (HM) used to improvise better harmonies. The HM size (HMS) is defined as the number of harmonies in the HM.

Algorithm 1 introduces the steps of HS for the detection of high-order SNP interactions. In each iteration, the worst harmony, $X^{idworst}$, with the largest fitness is replaced by a newly generated harmony X^{new} if X^{new} has stronger association with disease status than the worst $X^{idworst}$ in HM.

The steps for generating a new harmony are as algorithm 2.

The HS algorithm is well suited for detecting k-order SNP interactions at the genome-wide scale, in which a harmony H_i ($i = 1,2,\ldots,$HMS)denotes a k-order SNP combination ($SNP_1,SNP_2,\ldots,SNP_k$) whose goal is to find the best harmonies (k-order SNP combination) that are associated with disease status.

The HS includes three operators: a combination operator, pitch adjustment and random selection. A combination operator is used to select SNPs from the HM, which records good SNP combinations. Pitch adjustment is adopted to locally improve the SNP combinations. Random selection can be used to discover new SNP combinations. To detect k-order SNP interactions from a dataset with n samples and N SNP markers using the HS algorithm, the time complexity is $O(k \times NP)$ (initialization) $+O(T \times k)$ (improvising new harmony). Obviously, the time complexity of HS has nothing to do with N.

Algorithm 1 HS for the Detection of k-Order SNP Interaction

Inputs

(1) dataset with M samples and N SNPs

(2) terminal condition (MaxFEs): maximum number of calculating the associations.

(3) HMCR: harmony memory consideration rate.

(4) PAR: pitch adjustment rate.

Output: a set of SNP combinations associated with disease status.

Step 1. Randomly initialize harmony memory HM $\left(X^1, X^2, \ldots, X^{HMS}\right)^{\mathrm{T}}$ and calculate the association of each harmony in HM.

$$\mathrm{HM} = \left[\begin{array}{c|c} X^1 & f(X^1) \\ X^2 & f(X^2) \\ \vdots & \vdots \\ X^{HMS} & f(X^{HMS}) \end{array}\right] = \left[\begin{array}{cccc|c} x_1^1 & x_2^1 & \cdots & x_N^1 & f(X^1) \\ x_1^2 & x_2^2 & \cdots & x_N^2 & f(X^2) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_N^{HMS} & f(X^{HMS}) \end{array}\right]$$

Step 2. Generate new harmony X^{new} as **algorithm 2.**

Step 3. Update HM using X^{new}.

If X^{new} is better than the worst $X^{idworst}$ of HM

$X^{idworst} = X^{new}$

End

Step 4. Check the terminal condition. If the terminal condition is met, output the HM. Otherwise, goto Steps 2.

Algorithm 2 Generation of New Harmony

For i = 1→k

If rand(0,1) < HMCR

// (1) combination operator

ra ← select from {1, 2, …, HMS} randomly.

$x_i^{new} = x_i^{ra}$

//(2) pitch adjust

If rand (0,1)<PAR

x_i^{new} is adjusted according to a local search strategy.

EndIf

Else *// (3) select randomly from search space*

x_i^{new} ← select a value from set {1,2,…,N} randomly.

EndIf

End For

HS has a strong ability to explore the search space. We developed two HS algorithms (**FHSA-SED 2016** and **NHSA-DHSC 2017**) [40], [41] to detect high-order SNP interactions. Both of these algorithms include two phases: searching and testing. In **FHSA-SED (2016)**, an HS is adopted to discover some candidate SNP pairs in the search phase. The Bayesian-network -based K2-score and Gini score are employed to evaluate the association between pairwise SNPs and disease status. A local search strategy with a two-dimensional tabu table is presented to avoid repeatedly evaluating some SNP combinations that have strong marginal effects. In the testing stage, the G-test statistic is used to verify the candidate SNP pairs. In the **NHSA-DHSC (2017)** algorithm **,** a niching strategy is incorporated into the HS algorithm (NHSA), which serves as a tabu search region to prevent HS from becoming trapped in local optima. Three computationally lightweight and complementary evaluation criteria (Bayesian-network-based K2-score, Gini score and joint entropy) serve as objective functions of NHSA, where the joint entropy is utilized as a heuristic factor to guide the search for detecting SNP interactions with minimal or no marginal effect. Figure 4 presents an example explaining the process of the NHSA-DHSC algorithm for detecting a 3-way SNP combination model with a total of 10 SNPs. The process is divided into two stages: the first stage involves searching candidate solutions, and the second stage involves testing the authenticity of each candidate solution.

E. DIFFERENTIAL EVOLUTION (DE) ALGORITHM

The differential evolution (DE) algorithm proposed by Storn and Price is a very efficient SIS method [42]. Given its the advantages of having only a few parameters, strong search ability, and simplicity, DE has been widely used to address engineering optimization problems [43], [44]. The DE algorithm is analogous to the GA and involves three evolution operations:

1) **Mutation operation**. Mutation in the DE creates a donor solution $V^j(\mathrm{t}) = \left(v_1^j(t), v_2^j(t), \ldots, v_n^j(t)\right)$ corresponding to an individual solution $X^j(\mathrm{t}) = \left(x_1^j(t), x_2^j(t), \ldots, x_n^j(t)\right)$ in a population as follows:

$$V^j(t) = X^{r1}(t) + F \times \left(X^{r2}(t) - X^{r3}(t)\right)$$

where F is the scale factor that controls the learning rate. $V^j \in S^v = \prod_i^n \left[v_i^L, v_i^U\right]$. v_i^L vand v_i^U denote the lower and upper bounds, respectively, of $x_i^j (i \in \{1, 2, \ldots, n\}, j \in \{1, 2, \ldots, NP\})$. NP is the population size.

2) **Crossover operation**. Crossover creates a trail solution $U^j(\mathrm{t}) = \left(u_1^j(t), u_2^j(t), \ldots, u_n^j(t)\right)$ as follows:

$$u_i^j(t) = \begin{cases} x_i^j(t), & if\ rand\ (0, 1) \geq CR\ or\ i = I \\ v_i^j(t), & otherwise \end{cases}$$

where CR is the crossover rate for determining the learning of $X^j(t)$ from $V^j(t)$. In each iteration, the higher the CR is, the higher the learning rate from $V^j(t)$.

3) **Selection operation**. Selection determines whether the trail solution $U^j(\mathrm{t})$ can replace $X^j(t)$ in the t^{th} iteration. If the fitness of $U^j(\mathrm{t})$ is superior to that of $X^j(t)$, $X^j(t)$ will be replaced by $U^j(\mathrm{t})$.

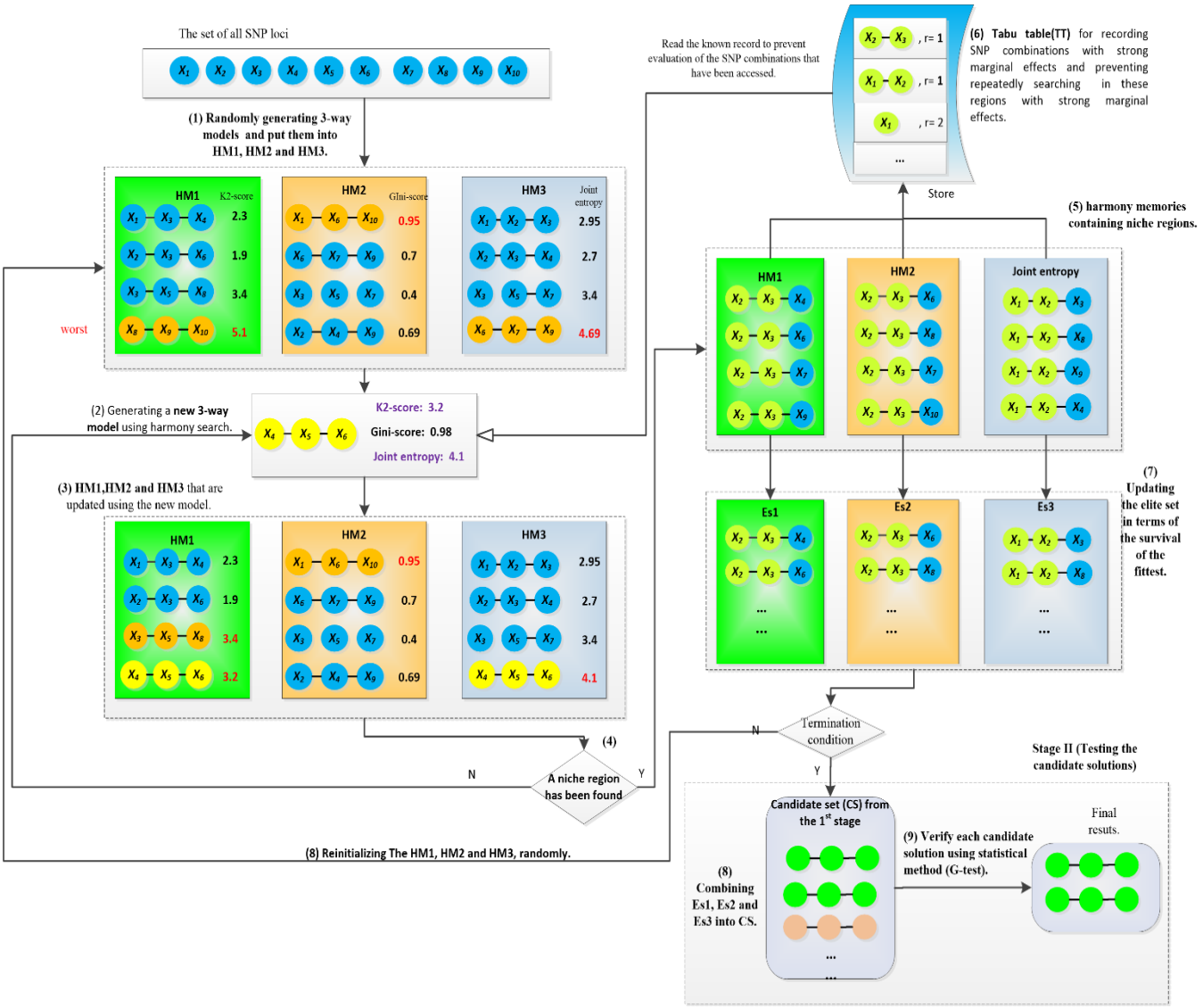

FIGURE 4. An example of NHSA-DHSC for detecting high-order SNP interactions.

Compared with the GA, the DE algorithm is a real-value-coded algorithm that solves continuous optimization problems very well; however, for discrete combinational optimization problems, a modification of the evolution strategy or bound constraints is required. Yang *et al.* proposed two DE algorithms (MODEMDR 2017 and CT-BDE 2018) [45], [46] for detecting gene-gene interactions. In **MODEMDR**, the n-order SNP combination is represented using an n-digit decimal code, and two MDR-based contingency table measures (**CCR** and **NMI**) are utilized as the fitness functions. The CCR is defined as the correct classification rate, and the NMI is the normalized mutual information. The **CT-BDE** combines a Taguchi catfish method and binary DE to identify SNP-SNP interactions; the Taguchi catfish method aims to prevent premature convergence, and MDR is used as the objective function for evaluating the degree of association between SNP combinations and disease status. The time complexity of DE is $O(k \times NP)$ (population initialization) $+O(T \times k \times NP)$ (crossover and mutation). Compared to GA, DE does not contain a selection operator, and it can directly decide a new solution regarding whether to stay or to leave.

The DE strategy is very efficient for discovering a continuous region that contains multiple SNPs associated with phenotype; however, it has difficulty exploring high-order SNP interactions in which the functional SNPs are far apart from each other.

F. PARTICLE SWARM OPTIMIZATION (PSO)

Particle swarm optimization(**PSO**) is another SIS technique developed by Eberhart and Kennedy. This method mimics the foraging behavior of a flock of birds or school of fish [47]. In PSO, each particle is denoted as a bird or fish and has two behaviors: social and individual. The movements of

particles are guided by their known historical best position (individual behavior) and the best-known position of the current swarm (social behavior), which can be expressed as follows:

$$V_i(t+1) = \omega V_i(t) + rand(0,1)\times c_1 \times \left(X_i^{old} - X_i^{best}\right) + rand(0,1) \times c_2 \left(X_i^{old} - X^{Gbest}(t)\right)$$
$$X_i(t+1) = X_i(t) + V_i(t+1)$$

where $\left(X_i^{old} - X^{Gbest}(t)\right)$ and $\left(X_i^{old} - X_i^{best}\right)$ are the social behavior and the individual behavior, respectively, of particle X_i, and c_1 and c_2 denote the learning factors of particle X_i. $\Omega \in (0, 1)$ is the inertia weight of the learning velocity. X_i^{best} is the historical best position of the i^{th} particle. X^{Gbest} denotes the globally best position of the entire swarm.

The PSO algorithm has a very high convergence rate for resolving certain complex optimization problems. Yang *et al.* first presented a double-bottom chaotic map PSO algorithm (DBM-PSO) to detect high-order genetic interactions [48]. In DBM-PSO, double-bottom maps are adopted to balance the exploration power and exploitation power, with the aim of preventing the PSO from becoming trapped in a local optimum. A particle denotes a candidate solution that represents a *k*-order SNP interaction. Shang *et al.* proposed an improved opposition-based learning PSO (named IOBLPSO) for the detection of SNP-SNP interactions [49]. IOBLPSO employs opposition-based learning to improve the global exploration power, uses dynamic inertia weight to cover a wide search and adopts post-processing to carry out a deep search in the suspected SNP sets.

The time complexity of PSO for the detection of k-order SNP interaction is $O(k \times NP)$ (initialization) $+O(T \times k \times NP)$ (velocity update) $+O(T \times k \times NP)$ (location update).

Similar to DE, PSO is a very effective method for solving continuous optimization problems and finding a small continuous gene region containing multiple disease-causing SNPs.

1) Three important details, which are not mentioned in the literature in designing the detection algorithm using DE and PSO, should be emphasized. For an SNP combination $X = (x_1, x_2, \ldots, x_n)$, the decision variables x_i should be an integer. However, it can be real-value coded for accelerating the search.
2) $x_i \neq x_j$ *if* $i < j$.
3) $x_i < x_j$ *if* $i < j$. This is not a requirement but is important for improving the search ability and avoiding becoming trapped during a local search.

In the later simulation experiments, all SIS algorithms are designed based on these rules, which can effectively improve the performance of the algorithms.

G. CUCKOO SEARCH (CS)

Cuckoo search (CS), inspired by the obligate brood parasitism of certain cuckoo species that lay their eggs in the nests of other species, is a new SIS algorithm in which a candidate solution is represented by cuckoo eggs [50].

CS is designed based on three rules:

(1) Each cuckoo lays its egg in a randomly chosen nest.

(2) The best candidate solution (nest) that has the best quality of eggs will survive as the next generation.

(3) The number of candidate solutions (nests) in the evolutionary process is fixed, and the host bird finds the egg that is laid by the cuckoo with a probability $p \in (0, 1)$ and dumps the found eggs.

A cuckoo X_i performs a Levy flight as follows:

$$X_i^{t+1} = X_i^t + \alpha \oplus Levy(\lambda)$$

where $\alpha > 0$ denotes the step size of the flight and represents entry-wise multiplication.

Algorithm 3 CS Algorithm

Initializing population with n host nests X_i (i = 1,2,...,n)
do
(1) Get a cuckoo X_i^{t+1} randomly by Levy flight.
$X_i^{t+1} = X_i^t + \alpha \otimes L(s, \lambda)$
$L(s, \lambda) = \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}$, s is the flight step.
α is the scaling factor of flight step.
(2) Calculate its fitness value f_i.
(3) Sort fitness values.
(4) Select a nest X_j randomly from population.
(5) If f_i is superior to f_j, replace X_j with X_i^{t+1}.
(6) Abandon some worse nests with probability p_a and generate new nests via Levy flight.
While (not satisfying terminal conditions).

The time complexity of CS for the detection of k-order SNP interaction is $O(k \times NP)$ (initialization) $+O(T \times NP \log NP)$ (sort) $+O(T \times NP)$ (abandon) $+O(T \times k^2 \times NP)$ (generating new solution with levy flight).

The CS algorithm has fewer parameters than certain other SIS algorithms, such as PSO and DE, and has been widely applied in the fields of computer science and engineering. Aflakparast *et al.* introduced a CS algorithm (named **CSE**) for detecting high-order genetic interactions [51]. In the CSE algorithm, the SNPs are partitioned into subgroups of SNPs according to the natural genomic order and associated genes, and the Bayesian-network- based score is utilized as an objective function to evaluate the association between the SNP combination and disease status. In the simulation experiments of CSE, the functional SNPs are divided into different groups; however, this is difficult for a real dataset.

H. FISH SWARM (FS)

The fish swarm (FS) search algorithm is inspired by the biological behavior of fish, in which each fish perceives a concentration of food and aims to find the best food source by applying three behaviors (preying, following and swarming) [52].

(1) Preying describes how a fish tends to reach a better food source from the current location as follows.

$$X_I = X_i + Random(Visual)$$
$$X_{i,next} = X_i + \begin{cases} Random(Step)\dfrac{X_I - X_i}{\|X_I - X_i\|}, & if\ f(X_i) < f(X_I) \\ Random(Step), & otherwise \end{cases}$$

where $Random(Visual)$ denotes that fish X_i searches randomly in its visible range. $Random(Step)$ represents random movement with each $Step$.

(2) The following behavior describes how a fish attempts to trail a better solution (food). Let $X_{i,neighbor}^{best}$ be the best neighbor of X_i. If $f_{\max}/n_f > \sigma f(X_i)$ (which means it is not crowded around $X_{i,neighbor}^{best}$), X_i will move toward $X_{i,neighbor}^{best}$; otherwise, it will prey as (1) in Step. where $f_{\max}$ represents the maximum fitness of solutions in this region.

(3) Swarming aims to assemble the fish and prevent fish from becoming trapped in a local search (too dense). At Generation t, the X_i will take a step forward to the fellow center X_c if $f(X_c)/n_f > \sigma f(X_i)$ (which means the fellow center is not a crowd); otherwise, it will execute the prey behavior.
where n_f is the number of fishes in the near region. σ denotes the crowd factor.

The time complexity of AFS is $O(k \times NP)$ (initialization) $+O(T \times k \times NP)$ (preying) $+O(T \times k \times NP)$ (following) $+O(T \times k \times NP)$ (calculating X_c).

Zhang *et al.* presented an artificial fish swarm (AFS) algorithm (fish swarm logic regression, FSLR) to identify interacting genetic variations. In the FSLR algorithm, the position of a fish is denoted as the SNP combination, and the association between the SNP combination and disease status denotes the food concentration around the fish. Logic regression was used to evaluate the association of SNP combinations with phenotype. The AFS algorithm aims to enhance the search speed and explore the interacting genetic variations [53] by facilitating communication among the fish group.

I. ARTIFICIAL BEE COLONY (ABC)

The artificial bee colony (ABC) algorithm, inspired by the foraging behavior of swarming honeybees, involves three operations: (1) discovering new food sources with the employed bees, (2) selecting good food sources with the onlooker bees, and (3) exploring new food sources with the scout bees.

In ABC, the employed bee x_i explores a new food source V_i as follows:

$$V_i = X_i + r_i(X_i - X_j), \quad (i \neq j, i = 1, 2, \ldots, n)$$

where $r_i \in [-1, 1]$ is a uniformly distributed random number.

If the fitness of V_i is better than that of X_i, X_i will be replaced by V_i.

The onlooker bee chooses the food source according to the fitness value of the food source, and the selection probability is calculated as follows:

$$P_i = \frac{a \cdot f_i}{\max(f)} + (1 - a), \quad a \in (0, 1)$$

The scout bee discovers a new food source X_{new} randomly within the search space as follows:

$$X_{new} = X^{\min} + r \cdot (X^{\max} - X^{\min})$$

where $X^{\max}$ and $X^{\min}$ are the upper and lower boundaries of the search space, respectively. r is a uniformly distributed random number in (0,1).

ABC has a powerful global exploration ability for solving complex optimization problems [54]. Accordingly, Li *et al.* proposed a two-objective ABC algorithm (named EIMOABC/D) to detect genetic interactions in a genome-wide manner. In EIMOABC/D, the Bayesian network score and Gini index are employed as the objective functions. The first objective aims to measure the relevance between the SNP combination and disease status, and the second objective is used to measure the SNP impurity of the SNP combination. In addition, a mutual-information-based local search algorithm is applied to avoid revisiting the solutions [55].

J. OTHER SIS EPISTASIS DETECTION ALGORITHMS

Moore *et al.* developed a grid-based stochastic search to detect hierarchical sets of interacting SNPs, named Crush-MDR [56], which employs expert biological knowledge as heuristic factors to guide probabilistic search and uses MDR as the objective function. Crush-MDR is a population-based stochastic search algorithm. It uses MDR as an objective function, conducts the search based on an opportunistic evolution strategy to maximize the efficiency, evolves candidate solutions on distributed computing nodes, and can adopt expert knowledge from any source to guide the search. Sun *et al.* presented a multi-objective evolution algorithm (SEE) to detect SNP epistasis [57]. In SEE, eight objective functions are integrated to measure the association between SNP combinations and phenotype.

In these proposed detection algorithms based on SIS, the ACO was the focus of greater attention compared to other algorithms. However, ACO usually has good performance when the population size is equal to or greater than the node number (SNP number of data sets), and it is memory and computationally intensive.

TABLE 1 summarizes the characteristics of SIS algorithms for detecting SNP interactions.

IV. EXPERIMENTAL ANALYSIS

A. PERFORMANCE COMPARISON OF SEVEN CLASSICAL SWARM INTELLIGENCE ALGORITHMS

To compare the detection power of seven classical swarm intelligence search algorithms (GA, ACO, HS, CS, DE, PSO, ABC and AFS), we investigate them on 12 DME (disease models with marginal effects) models (parameters of the 12 DME models are summarized in Table E-1 of NHSA-DHSC [41]). The 12 DME models involve 3 types of models:

TABLE 1. Overview of SI-based methods.

algorithm	Name	Objective function	Implementation and URL	Reference
Genetic algorithm (GA)	GE	Ensemble of classifiers	**JAVA**:http://www.cs.usyd.edu.au/~yangpy/software/Gesnpx.html	Yang, P., et al. (2010)
	\	Permutation test	/	Mooney, M., et al. (2011)
Ant colony optimization (ACO)	TuRF_ACO	MDR	**JAVA**: http://epistasis.org/	C. S. Greene, et al. (2008)
		MDR		Sulovari A, et al. (2013)
	ACA	Logistic regression	/	R. Rekaya et al. (2009)
	\	Multifactor dimensionality reduction (MDR)	/	K. Robbins et al. (2009)
	AntEpiSeeker	χ^2 test	**C++**:http://nce.ads.uga.edu/~romdhane/AntEpiSeeker/AntEpiSeeker1.0_sourcecode.zip	Wang Y, et al. (2010)
	MACOED	(1) Bayesian-network-based K2-score (2) Logistic regression	**MATLAB and C++** http://www.csbio.sjtu.edu.cn/bioinf/MACOED/	Jing and Shen (2015)
	AntMiner	χ^2 test	**MATLAB**: https://sourceforge.net/projects/antminer/files/	Shang et al. (2012)
	epiACO	(1) Mutual information (2) Bayesian network	**MATLAB:**https://sourceforge.net/projects/epiaco1/files/epiACO.rar/download	Sun et al. (2017)
	IEACO	Information entropy	/	*Guan and Zhao (2019)*
	ACO	MDR	**JAVA, C++** http://www.multifactordimensionalityreduction.org/	Sinnott-Armstrong N A, et al. (2010)
	ACO-Tabu	Pearson's chi-squared test	/	E. Sapin, et al. (2015)
	ACO	Pearson's chi-squared test	/	E. Sapin, et al. (2014)
	FAACOSE	(1) Logistic regression-based AIC score (2) Explanation score	/	Lin Yuan et al. (2017)
	\	Three self-defining functions.	/	Christmas J, et al. (2011)
Harmony search (HS)	FHSA-SED	(1) Bayesian network-based K2-score (2) Gini index	**MATLAB**:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4807955/bin/pone.0150669.s005.zip https://github.com/shouhengtuo/FHSA-SED	Tuo, S., et al. (2016).
	NHSA-DHSC	(1) Bayesian network-based K2-score. (2) Gini index (3) Joint entropy	**MATLAB** https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5599559/bin/41598_2017_11064_MOESM4_ESM.zip https://github.com/shouhengtuo/NHSA-DHSC	Tuo, S., et al. (2017).
Particle swarm optimization (PSO)	IOBLPSO	Mutual information	/	Shang et al. (2015)
	DBM-PSO	χ^2 test	/	Yang, CH et al. (2014)
Cuckoo search (CS)	CSE	Bayesian network	**MATLAB** http://lbb.ut.ac.ir/dynamic/uploads/soft/CSE.rar	M Aflakparast, et al. (2014)
Differential evolution (DE)	MODEMDR	MDR	/	Yang, CH et al. (2017)
	CT-BDE	MDR	/	Yang, CH et al. (2017)
Artificial bee colony (ABC)	EIMOABC/D	(1) Bayesian network (2) Gini index (3) Mutual information	/	Li XT, et al. (2018)
Fish swarm (FS)	FSLR	Logistic regression	http://www.engr.uconn.edu/~jiw09003/ (this url is invalid now.)	Zhang XP et al. (2013)
Evolutionary algorithm	SEE	Conditional entropy Gini index Bayesian network	https://github.com/sunliyan0000/SEE	Liyan Sun et al. (2019)

"/" denotes that the source code is not public. "\" represents that there is no name for the proposed algorithm.

multiplicative model, threshold model and concrete model. In the simulation experiments, we generate 100 datasets for each DME model using software GAMETES [79], which aims to test the robustness for various disease models.

To ensure a fair comparison, all algorithms are implemented using MATLAB (The source code of ACO is revised according to the source of MACOED [24], and CS is from CSE [51]. Other source codes are re-implemented in MATLAB). The Bayesian-network-based K2-score is employed as the objective function for the seven algorithms, and the same terminal condition, defined as the maximum number of evaluations of SNP combinations (Max_ES), is adopted for the seven algorithms. Max_ES is set to 2500 for datasets with 100 SNPs and to 50,000 for datasets with

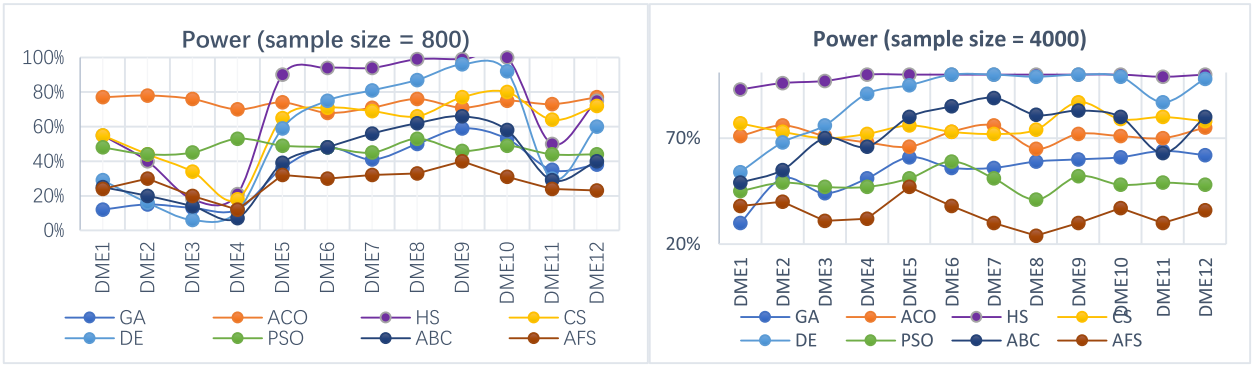

FIGURE 5. Power comparison of the seven algorithms on 12 DME models with 100 SNPs.

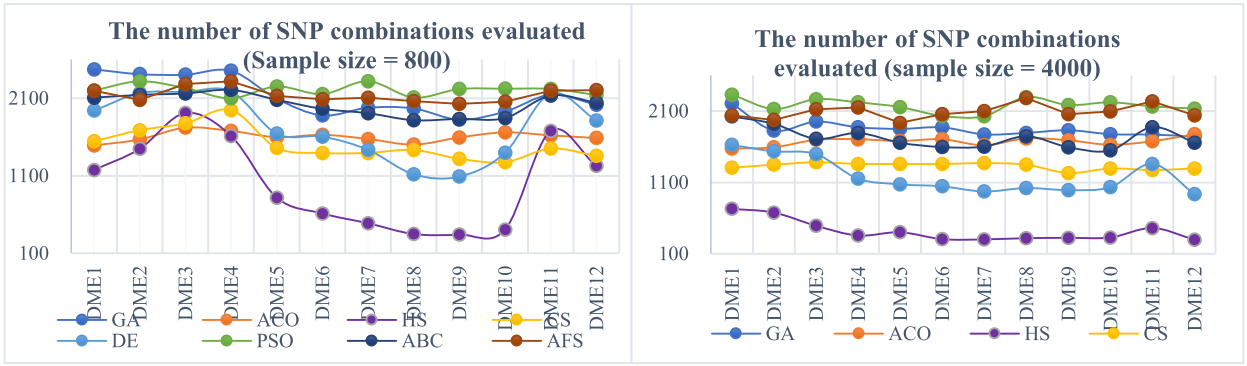

FIGURE 6. The number of SNP combinations that have been evaluated before the disease-causing SNP combination is found or the terminal condition is met.

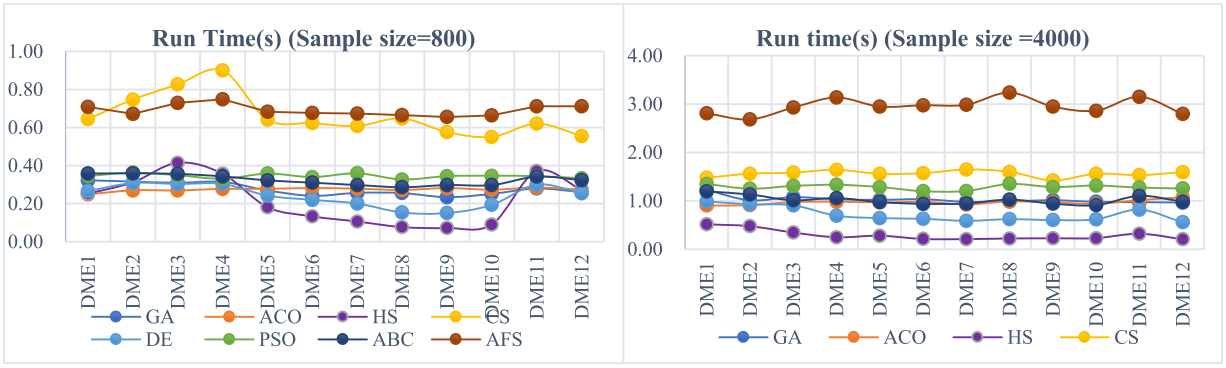

FIGURE 7. The average run time for finding the disease-causing SNP combinations from data with 100 SNPs.

1000 SNPs. Table 2 summarizes the parameters of the seven algorithms. All experiments were performed on a Windows 10 64-bit system with an Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz and 16 GB of RAM, and all the programs were written in MATLAB R2018a.

Three metrics (power, ME, and runtime) are adopted to compare the performance of the seven SIS algorithms for detecting high-order SNP interactions.

(1) $Power = \frac{\#S}{\#T}$

Power is a measure of the capability of detecting the disease-causing models from all datasets, where #S is the number of disease-causing models found out of #T datasets before the terminal conditions of search algorithm are met. In the experiments, #T = 100 for all disease models.

(2) **Run time** denotes the mean time that an algorithm takes to find a disease-causing model.

(3)**ME** denotes the mean number of SNP combinations that will be evaluated using the objective function until the disease-causing model is found.

B. (EXPERIMENT 1) 100 SNPS WITH SAMPLE SIZES OF 800 AND 4000

The simulation experiments are performed on datasets with 100 SNPs and sample sizes of 800 and 4000.

Figure 5 presents the detection power of eight algorithms used to solve 12 DME models with sample sizes of 800 (left figure) and 4000 (right figure). The results show that for 12 DME models with a sample size of 800, the ACO

FIGURE 8. Detection power of the seven algorithms (sample size = 4000 and 1000 SNPs).

maintains a steady and high detection power and is superior to other methods for the DME1-DME4 models. HS and DE are superior to other methods with respect to detection power for models DME5-DME10. The right figure indicates that HS and DE are more powerful than the other methods for most models with a sample size of 4000.

Figure 6 shows the mean number of SNP combinations that have been evaluated (ME) before the disease-causing SNP

TABLE 2. Parameter settings for the seven algorithms.

GA	ACO	HS	CS	DE	PSO	ABC
NP = 50 CR = 0.3 mutation rate:0.3	NP = 50 $\rho = 0.7$	HMS = 50 HMCR = 0.95 PAR = 0.35	NP = 50, $\alpha = 1/\sqrt{G}$ (G is the generation number) $\lambda = 1, s = 1$, $p_a = 0.25$	NP = 50 CR = 0.3 F=0.5	NP = 50 $c_1 = 1.5, c_2 = 1.5$ $\omega = 0.9$	NP = 50 $a = 0.9$
To make a fair comparison, the maximum number of SNP-combination evaluations (Max_ES) is set to 2500 for datasets with 100 SNPs and 50,000 for datasets with 1000 SNPs.						

combination is found or the terminal condition is satisfied (Max_ES = 2500). Generally, the smaller ME is, the better the algorithm. As shown, HS obtains lower ME in finding the functional SNP combination compared to other algorithms. PSO and AFS require evaluating a greater number of SNP combinations.

In Figure 7, the average run time is presented. The result indicates that the HS takes less time than the other methods on most DME models. DE is also competitive for all 12 DME models. CS and AFS require greater run times than the other algorithms to find the disease-causing models.

From Figures 5-7, it can be seen that HS and DE are the most powerful methods for detecting the disease-causing SNP combinations.

C. (EXPERIMENT 2) 1000 SNPS WITH A SAMPLE SIZE OF 4000

Figure 8 shows the detection powers, ME and runtime of the eight algorithms on 12 DME models with 1000 SNPs. The results indicate that ACO has a steadier and higher detection power (>80%) than CSE, PSO, GA, ABC and AFS on 12 DME models and is superior to HS and DE with respect to detection power for models DME1, DME2 and DME 11. HS and DE have obvious advantages over the other methods for DME 4-DME 10 and DME 12.

In terms of the ME and run time of the eight algorithms, HS and DE outperform the other algorithms: they require less run time and evaluate fewer SNP combinations than do the other six algorithms when detecting the disease-causing SNP combination. In contrast, the CS and AFS evaluate more SNP combinations and require longer run times than the other methods when discovering the disease-causing SNP combinations.

V. DISCUSSION

An SIS algorithm conducts a global search through the power of the group, therein aiming to enhance the perception of individual searchers in the search space through communication and learning between individuals in the group. SIS has received considerable attention for the detection of SNP interactions. However, this method is still not sufficient for the detection of high-order SNP interactions with minimal or no marginal effects because there are very few heuristic factors (clues) in exploring high-order SNP interactions with no marginal effects from the hundreds of thousands of SNPs in the genome. The key to SIS is to develop a good objective function (evaluation criterion) for calculating the association between genotypes and phenotypes. An effective objective function has the ability to guide the SIS algorithm to explore some clues (such as different distributions of genotypes) that can further lead the algorithm to find high-order SNP interactions on a genome-wide scale. In existing research, the most common evaluation criteria (objective functions) involve the Bayesian-network-based score [58]–[60], mutual information [61], [62], logistic-regression-based score [63], [64], MDR [65], [66], Gini index [67]–[70], statistical test methods (e.g., chi-square test, G-test, and t-test) etc. These criteria usually have a high precision in evaluating a pure k-order SNP interaction (in which the k SNPs jointly affect complex diseases, and the number of SNPs is the same); however, they are ineffective for determining the association difference of SNP combinations that contain only some of the disease-causing SNPs. For example, a 4-th-order SNP combination (SNP_1, $\mathbf{SNP_2}$, SNP_3, $\mathbf{SNP_4}$) contains two disease-causing SNPs ($\mathbf{SNP_2}$, $\mathbf{SNP_4}$), which are the components of a 4-th-order SNP interaction ($\underline{\mathbf{SNP_2}}$, $\underline{\mathbf{SNP_4}}$, SNP_7, SNP_9). This scenario is poorly distinguished from a 4-th-order SNP combination that does not contain any disease-causing SNPs by most existing evaluation criteria. This is the main reason why detecting pure high-order SNP interactions is so difficult; the SIS cannot find the clues for detecting the disease-causing SNP combinations.

Hence, developing an effective heuristic search factor is of great importance to SIS algorithms for the detection of high-order SNP interactions. In the NHSA-DHSC algorithm, we employed the joint entropy of k-order SNP combinations of case samples as one of the evaluation criteria, where the joint entropy aims to find the subtle differences in the genotype distributions between disease-causing SNP combinations and non-disease-causing SNP combinations in the case samples. The experimental results indicate that joint entropy is effective in guiding the search algorithm to find high-order SNP interactions for certain disease models with minimal or no marginal effects; however, it is also not sufficient to detect SNP interactions for disease models without marginal effects, and the simulation datasets do not follow the Hardy-Weinberg law. Therefore, combining multiple complementary evaluation criteria can be regarded as an option for identifying diverse disease models.

In recent years, multi-objective optimization algorithms that are used to enhance the identification power of the SNP interaction have been adopted to detect various SNP

interactions [24], [57]; however, for disease models without marginal effects, the performance of these algorithms is still not satisfactory. The goal of a multi-objective algorithm is to find a set of Pareto-optimal solutions (nondominated solutions), each of which satisfies the objectives at an acceptable level without being dominated by any other solution. However, we found that some SNP combinations (dominated solutions) that are eliminated during the search by the nondominated solutions are the true disease-causing SNP combinations, which results in an increase in the number of false negative errors. In FHSA-SED [40], NHSA-DHSC [41] and MCDA-MDR [73], the experimental results indicate that multicriteria algorithms are superior to multi-objective optimization algorithms. In future studies, lightweight and complementary evaluation criteria should also be considered to screen for suspected SNP interactions that have a strong association with disease status using any SIS in the first search phase. Multistage search, multiple populations and multiple criteria are important choices in developing SIS methods for improving the exploration power when detecting high-order SNP interactions.

To better understand detection in SIS, the joint entropy [40], relative entropy [71] and Kullback-Leibler (KL) divergence [72] should be considered in the design of heuristic factors. In addition, research should also focus on gaining insight from 2nd-order and 3rd-order SNP combinations, which should be explored with exhaustive search methods using high-performance computers such as HiSeeker [61].

In recent years, it has been proposed to focus efforts on the analysis of low-frequency and rare variants on GWAS [2], which represents a new area for SIS. In the era of biomedical big data, SIS will be more widely used in applications such as large-scale copy number variation analysis [74], [75], protein interaction [77], multi-omics data analysis and drug discovery [78].

REFERENCES

[1] R. J. Klein, C. Zeiss, E. Y. Chew, J. Y. Tsai, R. S. Sackler, C. Haynes, A. K. Henning, J. P. SanGiovanni, S. M. Mane, S. T. Mayne, M. B. Bracken, F. L. Ferris, J. Ott, C. Barnstable, and J. Hoh, "Complement factor H polymorphism in age-related macular degeneration," *Science*, vol. 308, pp. 385–389, Apr. 2005.

[2] V. Tam, N. Patel, M. Turcotte, Y. Bossé, G. Paré, and D. Meyre, "Benefits and limitations of genome-wide association studies," *Nature Rev. Genet.*, vol. 20, pp. 467–484, May 2019, doi: 10.1038/s41576-019-0127-1.

[3] X. Guo, Y. Meng, N. Yu, and Y. Pan, "Cloud computing for detecting high-order genome-wide epistatic interaction via dynamic clustering," *BMC Bioinf.*, vol. 15, no. 1, p. 102, 2014, doi: 10.1186/1471-2105-15-102.

[4] B. Goudey, M. Abedini, J. L. Hopper, M. Inouye, E. Makalic, D. F. Schmidt, J. Wagner, Z. Zhou, J. Zobel, and M. Reumann, "High performance computing enabling exhaustive analysis of higher order single nucleotide polymorphism interaction in genome wide association studies," *Health Inf. Sci. Syst.*, vol. 3, no. 1, p. S3, 2015, doi: 10.1186/2047-2501-3-S1-S3.

[5] X. Wan, C. Yang, Q. Yang, H. Xue, X. Fan, and N. L. Tang, "BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies," *Amer. J. Hum. Genet.*, vol. 87, no. 3, pp. 325–340, 2010, doi: 10.1016/j.ajhg.2010.07.021.

[6] L. S. Yung, C. Yang, X. Wan, and W. Yu, "GBOOST: A GPU-based tool for detecting gene–gene interactions in genome–wide case control studies," *Bioinformatics*, vol. 27, no. 9, pp. 1309–1310, 2011, doi: 10.1093/bioinformatics/btr114.

[7] A. Gyenesei, J. Moody, C. A. Semple, C. S. Haley, and W.-H. Wei, "High-throughput analysis of epistasis in genome-wide association studies with BiForce," *Bioinformatics*, vol. 28, no. 15, pp. 1957–1964, 2012, doi: 10.1093/bioinformatics/bts304.

[8] Y. Zhang and J. S. Liu, "Bayesian inference of epistatic interactions in case-control studies," *Nature Genet.*, vol. 39, no. 9, pp. 1167–1173, Sep. 2007, doi: 10.1038/ng2110.

[9] J. Wang, T. Joshi, B. Valliyodan, H. Shi, Y. Liang, H. T. Nguyen, J. Zhang, and D. Xu, "A Bayesian model for detection of high-order interactions among genetic variants in genome-wide association studies," *BMC Genomics*, vol. 16, no. 1, p. 1011, 2015, doi: 10.1186/s12864-015-2217-6.

[10] B. Han, X.-W. Chen, Z. Talebizadeh, and H. Xu, "Genetic studies of complex human diseases: Characterizing SNP-disease associations using Bayesian networks," *BMC Syst. Biol.*, vol. 6, no. 3, p. S14, 2012, doi: 10.1186/1752-0509-6-S3-S14.

[11] C. Yang, Z. He, X. Wan, Q. Yang, H. Xue, and W. Yu, "SNPHarvester: A filtering-based approach for detecting epistatic interactions in genome-wide association studies," *Bioinformatics*, vol. 25, no. 4, p. 504, 2009.

[12] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1996.

[13] J. H. Moore, L. W. Hahn, M. D. Ritchie, T. A. Thornton, and B. C. White, "Application of genetic algorithms to the discovery of complex models for simulation studies in human genetics," in *Proc. Genet. Evol. Comput. Conf.*, W. B. Langdon, Ed. San Mateo, CA, USA: Morgan Kaufmann, 2002.

[14] J. H. Moore, L. W. Hahn, M. D. Ritchie, T. A. Thornton, and B. C. White, "Routine discovery of complex genetic models using genetic algorithms," *Appl. Soft Comput.*, vol. 4, no. 1, pp. 79–86, 2004.

[15] S. C. Shah and A. Kusiak, "Data mining and genetic algorithm based gene/SNP selection," *Artif. Intell. Med.*, vol. 31, no. 3, pp. 183–196, 2004.

[16] P. Yang, J. W. Ho, A. Y. Zomaya, and B. B. Zhou, "A genetic ensemble approach for gene-gene interaction identification," *BMC Bioinf.*, vol. 11, no. 1, p. 524, 2010, doi: 10.1186/1471-2105-11-524.

[17] G. Bontempi, "A blocking strategy to improve gene selection for classification of gene expression data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 4, no. 2, pp. 293–300, Apr. 2007, doi: 10.1109/TCBB.2007.1014.

[18] L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 27, no. 5, pp. 553–568, Sep. 1997, doi: 10.1109/3468.618255.

[19] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Inf. Fusion*, vol. 6, no. 1, pp. 63–81, 2005, doi: 10.1016/j.inffus.2004.04.008.

[20] M. Mooney, B. Wilmot, The Bipolar Genome Study, and S. McWeeney, "'The GA and the GWAS: Using genetic algorithms to search for multilocus associations," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 9, no. 3, pp. 899–910, 2011, doi: 10.1109/TCBB.2011.145.

[21] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[22] C. Blum, "Ant colony optimization: Introduction and recent trends," *Phys. Life Rev.*, vol. 2, no. 4, pp. 353–373, 2005.

[23] J. Shang, X. Wang, X. Wu, Y. Sun, Q. Ding, J.-X. Liu, and H. Zhang, "A review of ant colony optimization based methods for detecting epistatic interactions," *IEEE Access*, vol. 7, pp. 13497–13509, 2019.

[24] P.-J. Jing and H.-B. Shen, "MACOED: A multi-objective ant colony optimization algorithm for SNP epistasis detection in genome-wide association studies," *Bioinformatics*, vol. 31, no. 5, pp. 634–641, Mar. 2015.

[25] C. S. Greene, J. M. Gilmore, J. Kiralis, P. C. Andrews, and J. H. Moore, "Optimal use of expert knowledge in ant colony optimization for the analysis of epistasis in human disease," in *Proc Evol. Comput., Mach. Learn. Data Mining Bioinf.* Berlin, Germany: Springer, 2009, pp. 92–103.

[26] R. Rekaya and K. Robbins, "Ant colony algorithm for analysis of gene interaction in high-dimensional association data," *Brazilian J. Animal Sci.*, vol. 38, pp. 93–97, Jul. 2009.

[27] Y. Wang, X. Liu, K. Robbins, and R. Rekaya, "AntEpiSeeker: Detecting epistatic interactions for case-control studies using a two-stage ant colony optimization algorithm," *BMC Res. Notes*, vol. 3, no. 1, p. 117, 2010.

[28] A. Sulovari, J. Kiralis, and J. H. Moore, "Optimal use of biological expert knowledge from literature mining in ant colony optimization for analysis of epistasis in human disease," in *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics* (Lecture Notes in Computer Science), vol. 7833, L. Vanneschi, W. S. Bush, and M. Giacobini, Eds. Berlin, Germany: Springer, 2013.

[29] J. Shang, J. Zhang, X. Lei, Y. Zhang, and B. Chen, "Incorporating heuristic information into ant colony optimization for epistasis detection," *Genes Genom.*, vol. 34, no. 3, pp. 321–327, Jun. 2012.

[30] Y. Sun, J. Shang, J.-X. Liu, S. Li, and C.-H. Zheng, "epiACO—A method for identifying epistasis based on ant Colony optimization algorithm," *BioData Mining*, vol. 10, p. 23, Jul. 2017, doi: 10.1186/s13040-017-0143-7.

[31] Y. Sun, X. Wang, J. Shang, J.-X. Liu, C.-H. Zheng, and X. Lei, "Introducing heuristic information into ant colony optimization algorithm for identifying epistasis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published, doi: 10.1109/TCBB.2018.2879673.

[32] B. Guan and Y. Zhao, "Self-adjusting ant colony optimization based on information entropy for detecting epistatic interactions," *Genes*, vol. 10, no. 2, p. 114, Feb. 2019, doi: 10.3390/genes10020114.

[33] N. A. Sinnott-Armstrong, C. S. Greene, and J. H. Moore, "Fast genome-wide epistasis analysis using ant colony optimization for multifactor dimensionality reduction analysis on graphics processing units," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Portland, OR, USA, Jul. 2010, pp. 215–216.

[34] J. Christmas, E. Keedwell, and T. M. Frayling, "Ant colony optimisation to identify genetic variant association with type 2 diabetes," *Inf. Sci.*, vol. 181, no. 9, pp. 1609–1622, 2011.

[35] E. Sapin, E. Keedwell, and T. Frayling, "Ant colony optimisation of decision trees for the detection of gene-gene interactions," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Belfast, U.K., Nov. 2014, pp. 57–61, doi: 10.1109/BIBM.2014.6999248.

[36] E. Sapin, E. Keedwell, and T. Frayling, "An ant colony optimization and tabu list approach to the detection of gene-gene interactions in genome-wide association studies [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 54–65, Nov. 2015, doi: 10.1109/MCI.2015.2471236.

[37] L. Yuan, C.-A. Yuan, and D.-S. Huang, "FAACOSE: A fast adaptive ant colony optimization algorithm for detecting SNP epistasis," *Complexity*, vol. 2017, Sep. 2017, Art. no. 5024867, doi: 10.1155/2017/5024867.

[38] Z. W. Geem, J. H. Kim and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search algorithm," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[39] Z. W. Geem, "Novel derivative of harmony search algorithm for discrete design variables," *Appl. Math. Comput.*, vol. 199, no. 1, pp. 223–230, 2008.

[40] S. Tuo, J. Zhang, X. Yuan, Y. Zhang, and Z. Liu, "FHSA-SED: Two-locus model detection for genome-wide association study with harmony search algorithm," *PLoS ONE*, vol. 11, no. 3, 2016, Art. no. e0150669, doi: 10.1371/journal.pone.0150669.

[41] S. Tuo, J. Zhang, X. Yuan, Z. He, Y. Liu, and Z. Liu, "Niche harmony search algorithm for detecting complex disease associated high-order SNP combinations," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 11529, doi: 10.1038/s41598-017-11064-9.

[42] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, Berkeley, CA, USA, Tech. Rep. TR-95-012, 1995, vol. 3.

[43] L. Wang, Q.-K. Pan, P. N. Suganthan, W.-H. Wang, and Y.-M. Wang, "A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems," *Comput. Oper. Res.*, vol. 37, no. 3, pp. 509–520, 2010.

[44] S. Tuo, J. Zhang, X. Yuan, and L. Yong, "A new differential evolution algorithm for solving multimodal optimization problems with high dimensionality," *Soft Comput.*, vol. 22, no. 13, pp. 4361–4388, 2018, doi: 10.1007/s00500-017-2632-5.

[45] C. H. Yang, L. Y. Chuang, and Y. D. Lin, "Multiobjective differential evolution-based multifactor dimensionality reduction for detecting gene–gene interactions," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 12869, doi: 10.1038/s41598-017-12773-x.

[46] C.-H. Yang, Y.-K. Kao, L.-Y. Chuang, and Y.-D. Lin, "Catfish Taguchi-based binary differential evolution algorithm for analyzing single nucleotide polymorphism interactions in chronic dialysis," *IEEE Trans. Nanobiosci.*, vol. 17, no. 3, pp. 291–299, Jun. 2018.

[47] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromach. Hum. Sci.*, Nagoya, Japan, Oct. 1995, pp. 39–43.

[48] C. H. Yang, Y. D. Lin, L. Y. Chuang, and H. W. Chang, "Double-bottom chaotic map particle swarm optimization based on chi-square test to determine gene-gene interactions," *BioMed Res. Int.*, vol. 2014, May 2014, Art. no. 172049, doi: 10.1155/2014/172049.

[49] J. Shang, Y. Sun, S. Li, J.-X. Liu, C.-H. Zheng, and J. Zhang, "An improved opposition-based learning particle swarm optimization for the detection of SNP-SNP interactions," *BioMed Res. Int.*, vol. 2015, Jan. 2015, Art. no. 524821, doi: 10.1155/2015/524821.

[50] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.

[51] M. Aflakparast, H. Salimi, A. Gerami, M. P. Dubé, S. Visweswaran, and A. Masoudi-Nejad, "Cuckoo search epistasis: A new method for exploring significant genetic interactions," *Heredity*, vol. 112, no. 6, pp. 666–674, 2014, doi: 10.1038/hdy.2014.4.

[52] X. Li, Z. Shao, and J. Qian, "Optimizing method based on autonomous animats: Fish-swarm algorithm," *Syst. Eng.*, vol. 22, no. 11, pp. 32–38, 2002.

[53] X. Zhang, J. Wang, A. Yang, C. Yan, F. Zhu, Z. Zhao, and Z. Cao, "Identifying interacting genetic variations by fish-swarm logic regression," *BioMed Res. Int.*, vol. 2013, Jul. 2013, Art. no. 574735, doi: 10.1155/2013/574735.

[54] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, 2007.

[55] X. Li, S. Zhang, and K.-C. Wong, "Nature-inspired multiobjective epistasis elucidation from genome-wide association studies," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published, doi: 10.1109/TCBB.2018.2849759.

[56] J. H. Moore, P. C. Andrews, R. S. Olson, S. E. Carlson, C. R. Larock, M. J. Bulhoes, J. P. O'Connor, E. M. Greytak, and S. L. Armentrout, "Grid-based stochastic search for hierarchical gene-gene interactions in population-based genetic studies of common human diseases," *BioData Mining*, vol. 10, p. 19, Dec. 2017, doi: 10.1186/s13040-017-0139-3.

[57] L. Sun, G. Liu, L. Su, and R. Wang, "SEE: A novel multi-objective evolutionary algorithm for identifying SNP epistasis in genome-wide association studies," *Biotechnol. Biotechnol. Equip.*, to be published, doi: 10.1080/13102818.2019.1593052.

[58] H. J. Cordell, "Epistasis: What it means, what it doesn't mean, and statistical methods to detect it in humans," *Hum. Mol. Genet.*, vol. 11, no. 20, pp. 2463–2468, 2002.

[59] H. J. Cordell, "Detecting gene–gene interactions that underlie human diseases," *Nature Rev. Genet.*, vol. 10, pp. 392–404, Jun. 2009.

[60] J. Zhao, L. Jin, and M. Xiong, "Test for interaction between two unlinked loci," *Amer. J. Hum. Genet.*, vol. 79, no. 5, pp. 831–845, 2006.

[61] J. Liu, G. Yu, Y. Jiang, and J. Wang, "HiSeeker: Detecting high-order SNP interactions based on pairwise SNP combinations," *Genes*, vol. 8, no. 6, p. 153, 2017.

[62] X. Cao, G. Yu, J. Liu, L. Jia, and J. Wang, "ClusterMI: Detecting high-order SNP interactions based on clustering and mutual information," *Int. J. Mol. Sci.*, vol. 19, no. 8, p. 2267, Aug. 2018.

[63] S. Purcell, B. Neale, K. Toddbrown, B. Neale, K. Todd-Brown, L. Thomas, M. A. R. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. W. De Bakker, M. J. Daly, and P. C. Sham, "PLINK: A tool set for whole-genome association and population-based linkage analyses," *Amer. J. Hum. Genet.*, vol. 81, no. 3, pp. 559–575, 2007.

[64] T. Schüpbach, I. Xenarios, S. Bergmann, and K. Kapur, "FastEpistasis: A high performance computing solution for quantitative trait epistasis," *Bioinformatics*, vol. 26, no. 11, pp. 1468–1469, 2010.

[65] R. L. Collins, T. Hu, C. Wejse, G. Sirugo, S. M. Williams, and J. H. Moore, "Multifactor dimensionality reduction reveals a three-locus epistatic interaction associated with susceptibility to pulmonary tuberculosis," *Biodata Mining*, vol. 6, no. 1, p. 4, 2013.

[66] D. R. Velez, B. C. White, A. A. Motsinger, W. S. Bush, M. D. Ritchie, S. M. Williams, and J. H. Moore, "A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction," *Genet. Epidemiol.*, vol. 31, no. 4, pp. 306–315, 2007.

[67] J. C. Kässens, L. Wienbrandt, J. González-Domínguez, B. Schmidt, and M. Schimmler, "High-speed exhaustive 3-locus interaction epistasis analysis on FPGAs," *J. Comput. Sci.*, vol. 9, pp. 131–136, Jul. 2015.

[68] T. Hu, Y. Chen, J. W. Kiralis, and J. H. Moore, "ViSEN: Methodology and software for visualization of statistical epistasis networks," *Genet. Epidemiol.*, vol. 37, no. 3, pp. 283–285, 2014.

[69] L. Ceriani and P. Verme, "The origins of the Gini index: Extracts from Variabilità e Mutabilità (1912) by Corrado Gini," *J. Econ. Inequal.*, vol. 10, no. 3, pp. 421–443, 2012.

[70] S. Yitzhaki and E. Schechtman, *The Gini Methodology: A Primer on a Statistical Methodology*. Springer, 2012.

[71] C.-I. Chang, Y. Du, J. Wang, S.-M. Guo, and P. D. Thouin, "Survey and comparative analysis of entropy and relative entropy thresholding techniques," *IEE Proc.—Vision, Image Signal Process.*, vol. 153, no. 6, pp. 837–850, Dec. 2007.
[72] S. Lin, "Kullback–Leibler divergence for detection of rare haplotype common disease association," *Eur. J. Hum. Genet.*, vol. 23, no. 11, p. 1558, 2015.
[73] C. H. Yang, Y. D. Lin, and L. Y. Chuang, "Multiple-criteria decision analysis-based multifactor dimensionality reduction for detecting gene–gene interactions," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 1, pp. 416–426, Jan. 2019, doi: 10.1109/JBHI.2018.2790951.
[74] X. Yuan, M. Gao, J. Bai, and J. Duan, "SVSR: A program to simulate structural variations and generate sequencing reads for multiple platforms," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published.
[75] X, "Yuan, J. Bai, J. Zhang, L. Yang, J. Duan, Y. Li, and M. Gao, "CONDEL: Detecting copy number variation and genotyping deletion zygosity from single tumor samples using sequence data," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, to be published.
[76] X. Yuan, J. Zhang, L. Yang, J. Bai, and P. Fan, "Detection of significant copy number variations from multiple samples in next-generation sequencing data," *IEEE Trans. Nanobiosci.*, vol. 17, no. 1, pp. 12–20, Dec. 2018.
[77] J. Z. Ji, L. Jiao, C. C. Yang, J. W. Lv, and A. D. Zhang, "MAE-FMD: Multi-agent evolutionary method for functional module detection in protein-protein interaction networks," *BMC Bioinf.*, vol. 15, no. 1, p. 325, 2014, doi: 10.1186/1471-2105-15-325.
[78] M. Shackelford, "Evolutionary algorithm for drug discovery interim design report," 2014, *arXiv:1403.4871*. [Online]. Available: https://arxiv.org/abs/1403.4871
[79] R. J. Urbanowicz, "GAMETES: A fast, direct algorithm for generating pure, strict, epistatic models with random architectures," *BioData Mining*, vol. 5, no. 1, pp. 1–16, 2012.

SHOUHENG TUO received the B.S. degree in computer application from the Lanzhou University of Finance and Economics, Lanzhou, China, in 2001, the M.S. degree in computer application technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2008, and the Ph.D. degree from Xidian University, Xi'an, China, in 2017. He is currently an Associate Professor with the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an. His current research interests include intelligent computing and bioinformatics.

HAO CHEN received the B.S. degree in computer science and the Ph.D. degree in power electronics and power transmission from the Xi'an University of Technology, Xi'an, China. He is currently an Associate Professor with the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications. He has authored more than 40 articles. His current research interests include evolutionary algorithms and their applications in the real world.

HAIYAN LIU received the B.S. degree from Shandong University, China, in 2007, and the Ph.D. degree from Xidian University, China, in 2018, all in computer science. She is currently a Lecturer with the School of Computer Science and Technology, Xi'an University of Posts and Telecommunications. Her current research interests include operational research and computational intelligence.