Personalized Search Inspired Fast Interactive Estimation of Distribution Algorithm and Its Application

Yang Chen, XiaoYan Sun*, *Member, IEEE,* DunWei Gong, *Member, IEEE,* Yong Zhang, *Member, IEEE,* Jong Choi, and Scott Klasky

***Abstract*—Interactive evolutionary algorithms have been applied to personalized search, in which less user fatigue and efficient search are pursued. Motivated by this, we present a fast interactive estimation of distribution algorithm by using the domain knowledge of personalized search. We first induce a Bayesian model to describe the distribution of the new user's preference on the variables from the social knowledge of personalized search. Then we employ the model to enhance the performance of interactive estimation of distribution algorithm in two aspects, i.e., (1) dramatically reducing the initial huge space to a preferred subspace and (2) generating the individuals of estimation of distribution algorithm by using it as a probabilistic model. The Bayesian model is updated along with the implementation of the estimation of distribution algorithm. To effectively evaluate individuals, we further present a method to quantitatively express the preference of the user based on the human-computer interactions and train a Radial Basis Function neural network as the fitness surrogate. The proposed algorithm is applied to a laptop search, and its superiorities in alleviating user fatigue and speeding up the search procedure are empirically demonstrated.**

***Index Terms*—Personalized Search, Interactive Estimation of Distribution Algorithm, Domain Knowledge, Naive Bayesian Model.**

I. INTRODUCTION

EVOLUTIONARY computation (EC), mimicking the natural principles of evolution [1], is powerful for solving complex optimization problems [2], [3]. A fitness function associated with the optimization problem forms the basis of EC due to "survival of the fittest", and the candidates with higher fitness are selected for further evolution. However, it is impossible to describe some problems with precise mathematical models. In the personalized optimization, e.g., product design, house layout design, and personalized search, solutions have to be evaluated and selected by users according to their own preference. In such scenarios, the fitness assignments on those solutions are subjective and relative, i.e., evaluations on the same solution can be greatly distinct under different conditions. Accordingly, traditional EC is no longer applicable, and it must be adapted by incorporating user evaluations.

Y. Chen, X.Y. Sun (*Corresponding author, e-mail: xysun78@hotmail.com), D.W. Gong, and Y. Zhang are with the School of Information and Electrical Engineering, China University of Mining and Technology, 221116 P.R. China.

J. Choi and S. Klasky are with the Computer Science and Mathematics Division, Oak Ridge National Laboratory, 1 Bethel Valley Road, Oak Ridge, TN 37831 America.

Interactive evolutionary computation (IEC), by involving user's evaluations in the evolutionary process, has been successfully developed and applied to various personalized optimization problems, i.e., product design, web page layout, and anti-collision design of vehicles [4], [5]. Compared with traditional EC on optimizing explicitly defined mathematical functions, IEC requires a user to evaluate solutions, which will inevitably bring heavy assessment burdens to the user when evaluating all individuals. Usually, the population size and evolutionary generations of IEC are restricted to small numbers due to user fatigue, which will greatly reduce the exploratory power of IEC in solving complex problems. Therefore, more attention has been paid to alleviate user fatigue and improve the exploration of IEC.

Three main kinds of strategies have been developed in improving IEC [6]: (1) designing more friendly user-computer interface or evaluation modes, e.g., using a fuzzy or interval number to assign fitness and ease the user's evaluation burden [7], (2) constructing a surrogate model with machine learning to evaluate individuals instead of users [8], and (3) modifying evolutionary operators to accelerate convergence [9], [10]. The performance of IEC has been remarkably enhanced with these improvements.

In these existing studies, three main drawbacks of IEC have not been adequately concerned. First, scoring-based user evaluations are still required in the evolutionary process, which runs the high risk of forcing the user to give up evaluations due to fatigue and aversion. Second, the most popular evolutionary mechanism in IEC is the genetic algorithm (GA), which may impede IEC in a fast search. Last, domain knowledge of the optimization problem has not been further extracted and used to improve IEC.

To solve the first problem, inspired by the personalized search or recommendation, Sun et al. proposed an interactive genetic algorithm by using an implicit evaluation mode based on user interactions and surrogates [11]. This framework can alleviate user fatigue and speed up the search process since a user is unnecessarily required to directly rank or score solutions anymore. However, the other two issues, i.e., effective evolutionary operators and the usage of domain knowledge of personalized search have not been involved.

Other powerful evolutionary mechanisms, e.g., Estimation of Distribution Algorithms (EDAs) have proven to outperform GAs on many complex optimization problems by merging GAs and probabilistic learning models [12]. The domain

knowledge as reflecting a user's preferences in personalized search can be greatly valuable for helping the search focus on the preferred subspace so as to enhance the operators of IEC. If these issues are further studied, the performance of IEC with implicit evaluation modes will be greatly improved. Motivated by this, we here develop a personalized search assisted interactive estimation of distribution algorithm under the implicit evaluation mode with domain knowledge (PS-IEDA-DK) for fast finding satisfactory candidates with less user fatigue.

In the proposed algorithm, we first extract the probabilistic model of user preference on the optimized variables from historical personalized search using Naive Bayesian estimation. Then, we use the Bayesian model to reduce the initial search space of the optimization problem to a preferred subspace for speeding up the entire search process. Moreover, an interactive EDA by initializing individuals with the initial preference probabilistic model on variables is proposed. For the successive evolution, the fitness estimation is carried out by constructing a Radial Bases Function (RBF) network as a surrogate with few interactive evaluations. This algorithm is expected to find satisfactory solutions with less user fatigue.

The main contributions of this algorithm are as follows. (1) Domain knowledge about the involved user's preference on the optimized variables is extracted from the statistical information using the Naive Bayesian model; (2) The initial exploration space of the designed IEDA is finely adapted to the user preferred region, which is beneficial to accelerating the exploration; (3) By considering the preference probability on variables, IEDA with a well-trained RBF surrogate is designed and improved.

The remainder of this paper is organized as follows. Section II describes the related work of EDA and the evolutionary optimization-assisted personalized search. The framework of the proposed algorithm is presented in Section III. Also, this section presents the details of the improved IEDA, especially the search space reduction strategy based on Naive Bayesian estimation, population initialization, and the preference surrogate. Section IV provides the application of the proposed algorithm together with the experimental results and analysis. Conclusions are given in Section V.

II. RELATED WORK

A. Estimation of Distribution Algorithm

EDA first proposed by Mühlenbein and Paaß [13] is a new evolutionary optimization mode by merging genetic algorithms with statistical learning. It macroscopically reveals a large amount of information about the population by building an explicit probabilistic model of those promising solutions. Then, the offspring is generated by sampling the probabilistic model [14]. EDAs have been widely applied to various problems [14], such as the no-idle permutation flow-shop scheduling problem (NIPFSP) [15], the bi-criteria stochastic job-shop scheduling problem with the uncertainty of processing time [16], the multi-objective reservoir flood control operation problem (MO-RFCO) [17], and Steiner tree problems existing in transportation, communication networks, biological engineering, and QoS multicast routing problems [18]. However, EDAs have not been applied to the IEC framework and the personalized search.

According to the variable dependencies, EDAs are classified into three types: dependency-free, bivariate dependencies, and multivariate dependencies [12], among which the dependency-free EDA is related to our algorithm and will be further addressed here.

The dependency-free EDA consists of three main parts: population initialization, probabilistic model update, and offspring generation. Specifically, the initial population B_0 is generated by sampling the initial probability vector $\mathbf{p}_0$ coming from a probabilistic model. The fitness of the initial individuals is calculated with the fitness function. Then, we use the truncation selection to select a set of promising candidates from the current population, thus updating the probabilistic model to get the probability vector $\mathbf{p}_{t+1}$. New population B_{t+1} is generated by sampling the updated probability vector $\mathbf{p}_{t+1}$.

Obviously, the definition of the probabilistic model is crucial. By using probabilistic models, EDAs are capable of introducing a priori information into optimization for the possibility of using Bayesian statistic [19], [20], [21]. The use of a priori information has been studied and used in optimization [14]. Practitioners can incorporate two sources of bias into EDAs: (1) a priori knowledge and (2) information obtained from prior EDA which runs on other similar problems; these two can also be combined with Bayesian statistics or other methods [19], [22]. Our algorithm treats the domain knowledge extracted from a user's preference as the prior knowledge and uses it to enhance EDA.

B. Evolutionary Algorithms Assisted Personalized Search

In the personalized search, modeling or tracking a user's preference based on the interactions performed by the users has been a hot topic [23], [24]. Chang et al. [25] constructed conditional preference nets (CP-nets) to describe the user's preference based on the historical search and then used the CP-net to filter searched results. Kassak et al. [26] expressed a user preference model using explicit and implicit feedbacks, e.g., rating and time spent on items. For dynamic recommendation under spare data circumstances, Tang et al. [27] proposed a novel dynamic personalized recommendation algorithm, in which the information contained in both ratings and profile contents was used by exploring latent relationships among ratings. All these researches have emphasized on the preference modeling process without concerning any optimization.

For obtaining reliable preference models, some researches have introduced EAs to optimize the structure or parameters of the preference models. Shaker et al. [28] proposed a novel approach for pairwise preference learning by combining an evolutionary method with random forest. They further presented an approach by articulating the Multivariate Adaptive Regression Spline (MARS) into an evolutionary optimization for pairwise preference learning [29]. Kuzma et al. [30] studied the possibilities of neural networks to predict the user's preference by incorporating an IEC. These researches have mainly focused on applying evolutionary algorithms to optimize the parameters of the user preference models rather than the search itself.

Combining the IGA with the content-based filtering technique, Kim et al. [31] presented an innovative recommender system to dynamically track a user's preference on music. Ahn [32] used an agent-based model to imitate user rational behavior and then adopted an evolution strategy with the agent-based model to seek the rational behavior of a user. To deal with content-based recommendations, Kant et al. [33] utilized Reclusive Methods (RMs) to handle uncertainty and the IGA was employed to the information retrieval.

These algorithms have used EC in an interactive manner to optimize the search process, but users are forced to be involved in the evolutionary process to give ratings. That is to say, the preference model associated with the domain knowledge of personalized search has not been integrated into IEC to accelerate the search process.

III. NAIVE BAYESIAN MODEL BASED INTERACTION ESTIMATION OF DISTRIBUTION ALGORITHM

A. Definitions

From the viewpoint of optimization, we give the definition of the personalized search in E-commerce. An item (also a solution) to be searched with n attributes (variables) is denoted as $X = \{x_1, x_2, \ldots, x_n\}$, and the attribute x_i has m_i values in the discrete and integer domain $S_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m_i}\}$ ($S_i = [a_i, b_i]$ if x_i varies continuously). Then, the personalized search is a combinatorial optimization problem and can be expressed as follows:

$$\begin{cases} \max \quad f(X) \\ s.t. \;\; X \in G \subset \prod_{i=1}^{n} S_i \end{cases} \tag{1}$$

where the value of $f(X)$ represents the user's preference or evaluation on a solution X, which cannot be explicitly defined. The symbol G is the feasible search space, i.e., all available items.

Supposing a user exactly prefers p attributes, i.e., the values of these p variables are specific. These attributes are denoted as $X^{\mathrm{K}} = \{x_1{}' = x_{1,i_1}, \ldots, x_j{}' = x_{j,i_j}, \ldots, x_p{}' = x_{p,i_p} | i_j \in \{1, 2, \ldots, m_j\}\}$. Accordingly, those $q = n - p$ attributes, denoted as $X^{\mathrm{NK}} = \{x_{p+1}, x_{p+2}, \ldots, x_n\}$, are variables to be optimized. Now, the personalized search is to find a solution in the subspace X^{NK} under the condition of the specified set X^{K}.

It is clear that the search efficiency is determined by the size of X^{NK}. The total number of solutions to be searched will be $S^{\mathrm{NK}} = \prod_{i=p+1}^{n} m_i$ if all variables in X^{NK} are explored. Usually, the value of p is smaller than three since the user often knows little about the search, and therefore, the value of S^{NK} will be very large. In such scenario, a fast and successful search is difficult to reach. On the contrary, if we can approximately obtain the user's preference probabilities on those variables in X^{NK} , then we will emphasize searching those preferred variables to speed up the exploration. Such preference probability distribution is a kind of domain knowledge and can be obtained from the associated historical search information.

Accordingly, we will concern on deriving the user's preference probability distribution based on the domain knowledge in the personalized search, and then reducing the large initial variable space to the subspace most preferred by the user. Also, with the preference probabilistic model, an enhanced IEDA is designed to effectively search the satisfactory solution.

B. Main Framework

The general framework of domain knowledge assisted IEDA in personalized search is demonstrated in Fig. 1, and the shaded parts as domain knowledge extraction, domain knowledge application, interactions-based surrogate modeling, and the domain knowledge-assisted IEDA are our main work.

(1) The domain knowledge extraction and expression form the basis of the algorithm, and some methods can be borrowed from the researches of user interest modeling in the personalized recommendation. In our algorithm, from the viewpoint of easily combining the domain knowledge with IEDA, we employ a Naive Bayesian model to obtain the preference probability distribution based on the historical adoption information of the searched items.

(2) The applications of the domain knowledge in enhancing the exploration of IEDA are also various, e.g., reducing the search space, improving the operators, or refining the distribution model of IEDA, and here, we expect to accelerate the search by effectively reducing the exploration space with the domain knowledge, i.e., the preference probability distribution, and serving as the probabilistic model of our IEDA.

(3) The pivotal issue of interactions-based surrogate fitness modeling is how to quantitatively define the associated user preference under different interactions, e.g., browsing, clicking, and saving. Only when we properly get the numerical preference, can we perform the machine learning-based surrogate modeling. In our algorithm, interactive time together with evaluation uncertainty is defined to represent the user's numerical assignments. With the evaluated individuals and their corresponding assignments, an RBF network is trained as a preference surrogate.

(4) An improved IEDA in using the Naive Bayesian model is finally developed. In the proposed IEDA, the Naive Bayesian model is employed as the initial probabilistic model of EDA, which guarantees the initial population to cover the most preferred variables.

The first two issues will be stated in detail in Section III-C, and the others will be addressed in sections III-D and III-E.

C. Space Reduction with Domain Knowledge

For p known attributes preferred by the user with their specific values as $X^{\mathrm{K}} = \{x_1{}' = x_{1,i_1}, \ldots, x_j{}' = x_{j,i_j}, \ldots, x_p{}' = x_{p,i_p} | i_j \in \{1, 2, \ldots, m_j\}\}$, the conditional probability of the user's preference on those variables in X^{NK} can be calculated with the Naive Bayesian model shown in Equation 2:

$$\begin{aligned} P\left(x_i \left| X^{\mathrm{K}}\right.\right) &= \frac{P(x_i) P\left(X^{\mathrm{K}} | x_i\right)}{P(X^{\mathrm{K}})} \\ &= \frac{P(x_i) P\left(x_1{}' | x_i\right) P\left(x_2{}' | x_i\right) \cdots P\left(x_p{}' | x_i\right)}{P(X^{\mathrm{K}})} \end{aligned} \tag{2}$$

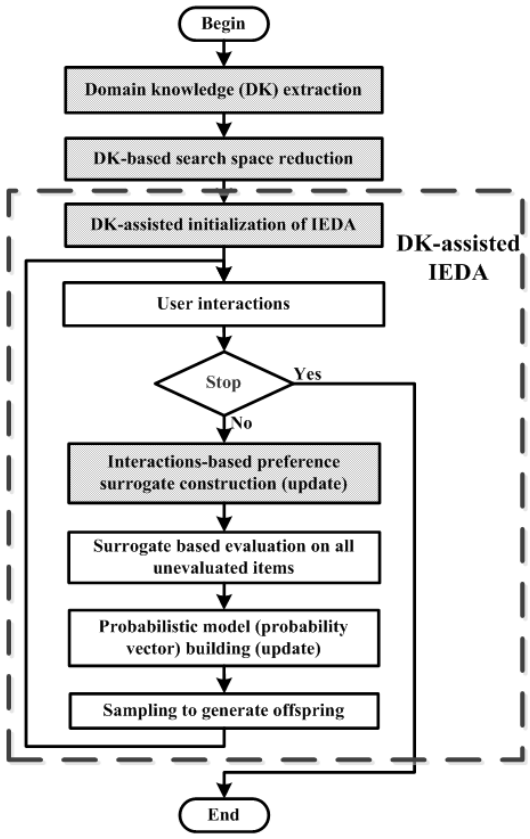


Fig. 1. Schematic of PS-IEDA-DK.

where $x_i \in X^{\mathrm{NK}}$, $i = 1, 2, \ldots, q$. Similarly, for the variable x_i, the conditional preference probability of its h-th value $x_{i,h}$, $h = 1, 2, \ldots, m_i$ is calculated from Equation 3:

$$\begin{aligned} P\left(x_i = x_{i,h} \left| X^{\mathrm{K}}\right.\right) &= \frac{P(x_{i,h}) P\left(X^{\mathrm{K}} | x_{i,h}\right)}{P(X^{\mathrm{K}})} \\ &= \frac{P(x_{i,h}) P\left(x_1{}' | x_{i,h}\right) P\left(x_2{}' | x_{i,h}\right) \cdots P\left(x_p{}' | x_{i,h}\right)}{P(X^{\mathrm{K}})} \end{aligned} \quad (3)$$

From equations 2 and 3, the specific value of $P\left(x_i = x_{i,h} \left| X^{\mathrm{K}}\right.\right)$ depends on the values of $P\left(x_i = x_{i,h}\right)$, $P\left(x_l{}' | x_{i,h}\right)$ ($l = 1, 2, \ldots, p$), and $P\left(X^{\mathrm{K}}\right)$, and we will present the specific calculation method by employing the historical information of searched items.

The set of the sold items belonging to the same category is Ω; the number of the items having the attribute as $(x_i = x_{i,h})$ is $M_i(h)$. Similarly, the number of the items simultaneously having the attributes as $(x_i = x_{i,h}, x_j = x_{j,k})$ is recorded as $M_{ij}(h, k)$, and that of the items including all known attributes is denoted as $M_{\sum p}$. Then, the values of $P\left(x_i = x_{i,h}\right)$, $P\left(x_l{}' | x_{i,h}\right)$, and $P\left(X^{\mathrm{K}}\right)$ can be calculated as follows:

$$P\left(x_i = x_{i,h}\right) = \frac{M_i(h)}{|\Omega|} \quad (4)$$

$$P\left(x_l{}' | x_{i,h}\right) = \frac{M_{li}(i_l, h)}{M_i(h)} \quad (5)$$

$$\begin{aligned} P\left(X^{\mathrm{K}}\right) &= P\left(x_1{}' = x_{1,i_1}, x_2{}' = x_{2,i_2}, \ldots x_p{}' = x_{p,i_p}\right) \\ &= \frac{M_{\sum p}}{|\Omega|} \end{aligned} \quad (6)$$

where $|\cdot|$ is the cardinal number of set $\cdot$.

We then calculate the value of $P\left(x_i = x_{i,h} \left| X^{\mathrm{K}}\right.\right)$ in Equation 3 with equations 4, 5, and 6. The larger the value of $P\left(x_i = x_{i,h} \left| X^{\mathrm{K}}\right.\right)$ is, the variable with the value as $(x_i = x_{i,h})$ is more preferred by the user. Hence, those values of the i-th variable with larger $P\left(x_i = x_{i,h} \left| X^{\mathrm{K}}\right.\right)$ must be involved in the initial search space. All the values of the i-th variable are sorted according to the descending order of $P\left(x_i = x_{i,h} \left| X^{\mathrm{K}}\right.\right)$ and saved in the set $\bar{S}_i$. Then, the set $\bar{S}_i$ is further reduced to $\bar{\bar{S}}_i$ by ignoring those elements with smaller preference probabilities. Using a parameter ε to control the reduction, we have

$$\left|\bar{\bar{S}}_i\right| = \varepsilon \times \left|\bar{S}_i\right| \quad (7)$$

On condition of the known set X^{K}, the reduced search space $\bar{S}^{\mathrm{NK}}$ is obtained as follows:

$$\bar{S}^{\mathrm{NK}} = \prod_{i=p+1}^{n} \bar{\bar{S}}_i \quad (8)$$

The size of $\bar{S}^{\mathrm{NK}}$ is restricted by the value of the parameter ε, which will greatly influence the exploration efficiency. Hence, the value of ε should be finely tuned in practice.

In the reduced exploration space, individuals will be initialized based on domain knowledge, and operators like selection and reproduction will then be further performed to the evolution. In this process, all individuals must be evaluated and compared for selecting elitists. However, in personalized search of E-commerce, the user only browses individuals instead of assigning any orders. In such scenario, we must build a user preference model to quantitatively evaluate individuals according to the user's interactions.

D. Interactions-based Preference Surrogate Model

The common issues in developing preference model in personalized search are sample collection, model selection, model training, and model updating. Here, we use an RBF network to build the preference model due to its merits in training and regression. Then, the hinge for obtaining a reliable preference model is the training samples. The set of training samples is usually written as $T = \{(X_i, f_i), i = 1, 2, \cdots, N_T\}$, where X_i is the individual (item) has been evaluated by the user, and f_i is the corresponding preference (assignment), which cannot be explicitly expressed.

Accordingly, the browsing time associated with user interactions is delivered to quantify the user's preference and to get the value of preference f_i for the individual X_i. We concern on three typical interactions, i.e., "click-browse-close", "click-browse-save", and "non-click". We denote the browsing time of the "click-browse-close" and the "click-browse-save" as $t_{\mathrm{CLO}}(X_i)$ and $t_{\mathrm{CLCT}}(X_j)$, respectively. For the individuals only browsed but not clicked by the user, we assign the same browsing time to them, and define the time as follows:

$$t_{\mathrm{NC}}(x_m) = \frac{t_{\mathrm{Pg}} - \sum t_{\mathrm{CLO}}(X_i) - \sum t_{\mathrm{CLCT}}(X_j)}{N_{\mathrm{NC}}} \quad (9)$$

$$f(X_i) = \begin{cases} rand\left[1-\alpha\cdot e^{-\frac{t_{\mathrm{CLCT}}(X_i)}{\delta_s}}, 1\right], \text{ if } X_i \text{ gets the "click} - \text{browse} - \text{save"} \\ rand\left[\chi-\beta\cdot e^{-2\cdot\frac{t_{\mathrm{CLO}}(X_i)}{\delta_s}}, \chi+\beta\cdot(1-e^{-\frac{t_{\mathrm{CLO}}(X_i)}{\delta_s}})\cdot e^{-\frac{t_{\mathrm{CLO}}(X_i)}{\delta_s}}\right), \text{ if } X_i \text{ gets the "click} - \text{browse} - \text{close"} \\ rand\left[0, \alpha\cdot e^{-\frac{t_{\mathrm{NC}}(X_i)}{\delta_s}}\right), \text{ if } X_i \text{ gets the "non} - \text{click"} \end{cases} \quad (10)$$

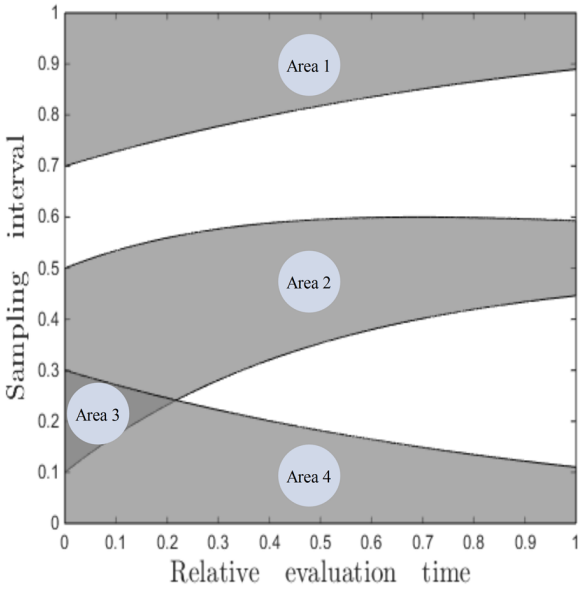


Fig. 2. Distribution of Fitness Along with Relative Evaluation Time.

where t_{Pg} is the total interactive time on the current search page, and N_{NC} is the number of individuals that are not clicked.

With the associated time of those different interactions on an individual X_i, the fitness $f(X_i)$ is defined in Equation 10 by considering the evaluation uncertainties. The fitness varies in the range $[0, 1]$.

In Equation 10, δ_s is a time scale parameter, and $rand[a, b]$ represents a uniform sampling in the interval $[a, b]$. The parameters α, β, and χ determines the interval length.

We set values of parameters as $\alpha = 0.3$, $\beta = 0.4$, and $\chi = 0.5$, label the x-axis with the evaluation time $\frac{t(X_i)}{\delta_s}$ and the y-axis with the sampling interval, and can get the distributions of $f(X_i)$ as shown in Fig. 2.

In Fig. 2, from the top to the bottom, the fitness distribution of the "click-browse-save" is in area 1, that of "click-browse-close" is in areas 2 and 3, and that of the "non-click" is in areas 3 and 4. It is clear that the interval area gradually narrows down as the interaction time increases. The area 3 indicates that sampling intervals of "click-browse-close" and "non-click" interaction overlap if the evaluation time is short.

With the training set T, the structure of the RBF can be determined. The input of the RBF is the individual X and the output is the fitness $f(X)$; accordingly, the RBF has n input nodes and one output. The Gaussian function based RBF is obtained as follows by using the training set T.

$$\hat{f}(X) = \sum_{i=1}^{N_{\mathrm{h}}} \omega_i e^{-\left(\frac{X-C_i}{\sqrt{2}\sigma_i}\right)^2} + b \quad (11)$$

where ω_i, C_i, and σ_i represent weight, average, and standard deviation of the i-th Gaussian function, respectively; parameter b represents the threshold, and N_{h} is the size of nodes in the hidden layer. The gradient-descent method is adopted to update the parameters (ω_i, σ_i) for training the neural network [34].

Along with the evolution, the RBF network will be updated to track the varied user preference. Training samples are first updated using newly searched items together with corresponding user interactions, and then they are employed to retrain the network to timely reflect the user's preference.

E. Interactive Estimation of Distribution Algorithm

A modified IEDA is developed here based on the probability distribution of the preference and the preference model. First, the initial probability vector of IEDA is obtained based on the preference probabilistic model of $P\left(x_i = x_{i,j} \left| X^{\mathrm{k}}\right.\right)$ in the preferred space $\bar{S}^{\mathrm{NK}}$ and shown in Equation 12:

$$\begin{aligned} \mathbf{p}_0 &= (p_1, \cdots, p_i, \cdots, p_q) \\ &= \left[P\left(x_1|X^{\mathrm{K}}\right), \cdots, P\left(x_i|X^{\mathrm{K}}\right), \cdots, P\left(x_q|X^{\mathrm{K}}\right)\right] \end{aligned} \quad (12)$$

The population is initialized by using $P\left(x_i|X^{\mathrm{K}}\right)$ $(x_i \in X^{\mathrm{NK}})$ in the framework of EDA.

In the evolutionary process, the probability vector $\mathbf{p}_t$ is updated with $p_i = P(x_i|X_S)$, where X_S are the selected elitists that come from the promising area. Here, the non-promising areas are assigned a very low probability, i.e. 10%, and individuals from these areas share the same probability. This strategy benefits to avoid being premature.

Then, the RBF preference model is constructed to evaluate all individuals' fitness. The probability for reproduction of EDA is updated with those promising individuals and then is used to generate N_{c} individuals. These individuals are displayed to the user for further interactions. Repeat this process until the user finds the optimal solution.

IV. EXPERIMENTS AND ANALYSIS

A. Experimental Setting

The proposed algorithm is applied to the personalized search on laptops and compared with other IECs to experimentally demonstrate its merits. The reason we select searching laptops mainly lies in that such a personalized search has more objective attributes than other things, e.g., fashions or sunglasses, which will be beneficial for the clear and objective illustration

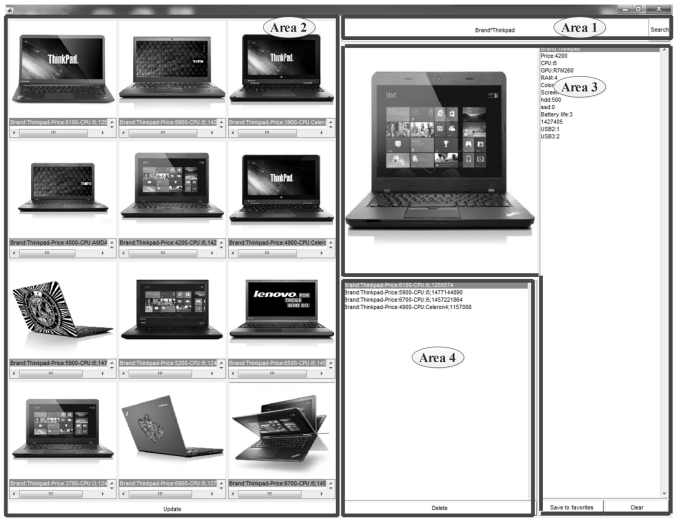


Fig. 3. Interface of platform.

of algorithms' performances. The laptop data (updated in April 2015) is from the JD.com, one of the largest B2C online retailers in China.

The platform is first developed with MATLAB 2015a, Python 3.3, and MongoDB 3. MATLAB deals with the algorithm and GUI, and Python connects MATLAB with the commodity database saved in MongoDB.

The interface of the platform, which consists of four main parts as marked with areas 1, 2, 3, and 4 in Fig. 3, is designed by considering attributes and sales record. The user first inputs keywords of the searched object in area 1 and then gets all corresponding results in area 2. The detail of each item will be displayed in area 3 for interacting, and the user's favorites will be saved and shown in area 4. By clicking items, the user can implement interactions as "click-browse-close" and "click-browse-save". Specifically, in each iteration, 12 items are displayed in area 1. If the user clicks an item, he/she can get the specific information shown in area 3. Then, the user chooses to save the item into the favorites in area 4 ("click-browse-save") or to close it ("click-browse-close"). Those items without any clicks are naturally treated as "non-click" ones.

We conduct two groups of experiments to adequately demonstrate the effectiveness of the proposed algorithm. The first group is designed to objectively demonstrate the merits of the algorithm by using a specified preference function instead of user interactions. The second group mimics the interaction environment of IEC by involving a user in the search process to show the practical performance of IEDA for personalized search in E-commerce.

Three other algorithms are compared here, i.e., the traditional interactive genetic algorithm (IGA), the support vector machine classification (SVMC) and the logic regression classification (LRC) based personalized search. The IGA is used to show the efficiency of our algorithm with EDA evolutionary mechanism. The purpose of selecting the other two classification-based personalized search methods is to demonstrate the entire performance of our algorithm in fast and successfully finding the satisfactory item.

As for the termination criterion, an expected item (can be randomly selected) is first designated as a reference, and the algorithm is terminated if the expected item is obtained or the user feels too fatigued to continue the search.

We use three indicators, i.e., the search time, the number of evaluated items, and the Discounted Cumulative Cost (DCC), to measure the proposed algorithm and the compared ones. The search time reflects the efficiency of the algorithms, and the shortest time indicates the fastest search. The number of evaluated items shows the total evaluation burden of the user, and smaller value means less user fatigue. Besides, it can also indicate the search efficiency since fewer evaluated items will cost less time. As is known, the user fatigue or evaluation burden is not only related with the number of evaluated items but also greatly influenced by other factors as items quality and ranking orders. Therefore, DCC is here used to further evaluate the user fatigue. We define the indicator DCC based on the Discounted Cumulative Gain (DCG) [35] in information retrieval. Specifically, all individuals presented to the user are classified into three groups according to the three different interactions. The individuals in the same group are viewed as relevant ones since the user has similar preferences on them. Correspondingly, the relevant degree of "click-browse-save" is set as 1, and that of "click-browse-close" and "non-click" are 2 and 3, respectively. All the interactively evaluated individuals are stored in a list, and its DCC is defined as $\mathrm{DCC}_p = \sum_{i=1}^{p-1} \frac{rel_i}{\log_2(1+p-i)} + rel_p$, where rel_i is the relevant degree of the item at position i in the list, and p is the length of the list.

We conduct 30 times of experiments for all algorithms and analyze the average of the experimental results.

B. Parameters Setting

The variables of the laptop search, the encoding strategy, and the involved parameters of all compared algorithms will be presented in detail.

1) Optimization Variables: In the personalized search, a solution is a combination of item's attributes, i.e., distinctive solutions consist of different attribute values. In our application, we select 12 common attributes of the laptop together with the number of their values shown in Table I as optimization variables. Taking the attribute RAM as an example, its seven different values are 4G, 8G, 16G, 32G, 2G, 6G, and 64G. The total size of our search is 1.5×10^{12}, so it is too difficult for using a basic local search heuristic algorithm to find a satisfactory item in the search space.

2) Encoding Strategy: Encoding is critical because the search space is very large. Specifically, only 60000 laptops are available, so it is a sparse search. We develop and compare three encoding schemes, i.e., the decimal encoding and other two binary ones. The preference probability distribution of variables will be combined into the decimal encoding and a binary one to improve the evolutionary performance. Specifically, a decimal encoding method with domain knowledge is termed as the attribute based ordinal decimal encoding (AODE). For the binary encoding method using domain knowledge is called the

TABLE I
ATTRIBUTES

Attribute name	Number of values
Brand	32
Color	16
Price	163
CPU	21
GPU	59
RAM	7
HDD	9
SSD	17
Screen size	9
Battery life	16
USB2.0	6
USB3.0	8

TABLE II
DECIMAL ENCODING OF CPU

Values of the attribute CPU	Decimal encoding
i5	1
i3	2
i7	3
Celeron4	4
Others	5
AMDFX	6
AMDA8	7
Celeron2	8
AMDA6	9
CoreM	10

attribute based ordinal Gray encoding (AOGE), and the other is the selling-amount based ordinal binary encoding (SOBE).

The decimal encoding scheme, AODE, uses decimal strings to represent all variables. For a laptop with q attributes, q decimal numbers are used to represent all these variables. The decimal number associated with an attribute value is determined by the preference probability, i.e., a higher probability corresponding to a smaller number. For example, with ten values of the attribute CPU, the decimal encoding is given in Table II. The combinations of these decimal numbers corresponding to q attributes are individual chromosomes and will be evolved. An example of the decimal-encoded individual of a laptop with ten attributes ($q = 10$) is shown in Table III on condition that the attributes "color" and "brand" are specified as "black" and "Thinkpad".

For the Gray encoding method, AOGE, if there are q attributes, then the encoded chromosome consists of q independent modules. Each module is a Gray code string, and its length is enough to represent all available values of the corresponding attribute, e.g., a 3-bit string can express the attribute with eight values. For the attribute with nine values, a 4-bit one is required. For the same examples given in Table II and Table III, the representations of the AOGE are given in Table IV and Table V.

As for SOBE, it is similar to AOGE except for different module encoding orders due to different encoding rules, which will not be explained anymore.

TABLE IV
GRAY ENCODING OF CPU

Values of the attribute CPU	Gray encoding
i3	0000
i5	0001
i7	0011
Celeron2	0010
CoreM	0110
Celeron4	0111
AMDA8	0101
AMDFX	0100
Others	1100
AMDA6	1101
AMDA10	1111

For IEC, with these encoding methods, after several evolutions, some individuals may not exactly match a real item after decoding. To locate a proper candidate for these individuals, the most similar item corresponding to the decoded individual is selected to be evaluated. For machine learning based algorithms, i.e., SVMC based interactive personalized search algorithm (SVMC-IPSA), and LRC based one (LRC-IPSA), all unevaluated items are classified by the classifier SVMC or LRC based on the interacted items, and those belonging to the class most preferred by the user will be selected as candidates and shown to the user.

3) Parameters: The operation parameters of IGA with different encoding strategies are finely tuned and set as follows: IGA with AODE encoding, we use Roulette Wheel Selection, Intermediate Crossover with crossover probability being 0.99, and Gaussian Mutation with mutation probability as 0.10; the average and the variation of the Gaussian function are 0 and 0.6, respectively. For the Gray and binary schemes, they use the same evolution operators and parameters. Specifically, Roulette Wheel Selection, Five-point Crossover with crossover probability being 0.99, and Binary Mutation with mutation probability as 0.20 are applied.

For LRC-IPSA and SVMC-IPSA, the K-fold cross validation with K being 3 is used. For the LRC-IPSA, the solver with coordinate descent algorithm is from LIBLINEAR, and the norm used in the penalization is L^2. For the SVMC-IPSA, the maximal iteration is 100, Gaussian function is used as the kernel function, and the penalty parameter C of the error term is 1.

C. Experiments on Parameter ε

We here first conduct three experiments to determine the value of the parameter ε since it is critical for controlling the efficiency of the reduction and search.

1) Relationship Between ε and Reduction Ratio: We take four attributes: brand, color, battery life, and RAM, as an example, plot the changes of the reduction ratio along with the values of ε, and present it in Fig. 4. Here, the reduction ratio is defined as the ratio of the number of solutions in the reduced space to that in the original one. The following conclusions can be drawn: (1) The reduction ratio increases along with the increase of ε from 0 to 1, which indicates that the shrinkage

TABLE III
EXAMPLE OF A LAPTOP DECIMAL ENCODING

Attribute	HDD	CPU	RAM	USB2	USB3	Price	GPU	Battery	Screen size	SSD
Phenotype	500	i3	4	1	2	3100	R7M265	3	15	0
Decimal encoding string	1	2	1	1	1	33	7	1	3	1

TABLE V
EXAMPLE OF A LAPTOP GRAY ENCODING

Attribute	HDD	CPU	RAM	USB2	USB3	Price	GPU	Battery	Screen size	SSD
Phenotype	500	i3	4	1	2	3100	R7M265	3	15	0
Gray encoding string	11	0000	00	000	00	1100000	0000	0001	00	0100

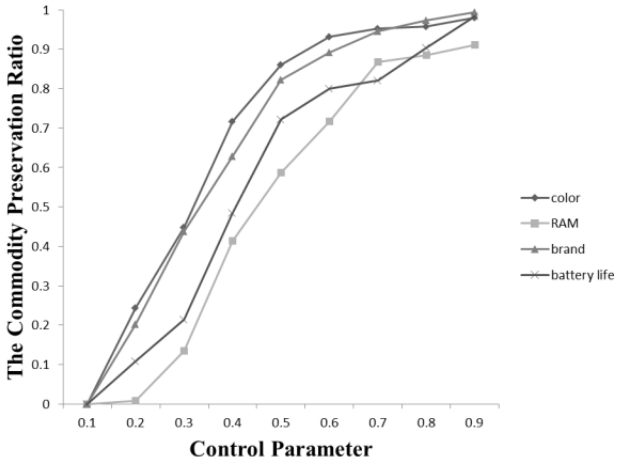


Fig. 4. Reduction ratios vs. different values of ε.

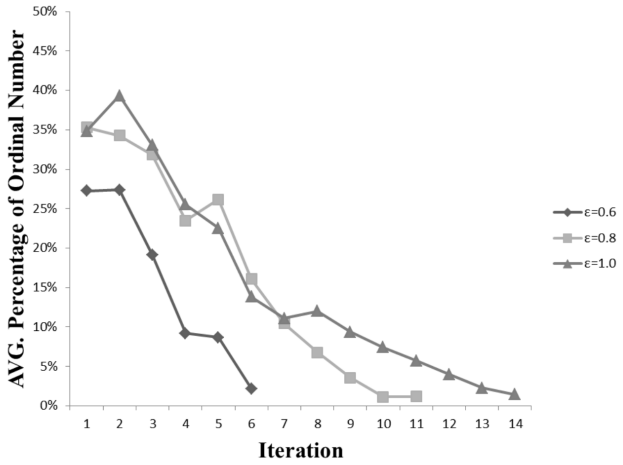


Fig. 5. Estimated fitness vs. values of ε.

TABLE VI
NUMBER OF SOLUTIONS EVALUATED AND DCC UNDER DIFFERENT VALUES OF ε

ε	Solutions evaluated		DCC	
	Av.	Std.	Av.	Std.
0.6	59.8667	24.9036	19.5742	10.3803
0.7	69.6	35.9757	23.4253	12.5696
0.8	77.7	40.8278	26.9544	11.4448
0.9	77.3667	35.5183	26.5281	8.75103
1.0	89.3333	48.3588	31.2891	12.0325

of search space decreases. When the value of ε approaches 1, the reduced search space is just equal to the original one. (2) When $\varepsilon \in [0, 0.6]$, the ratio changes sharply; when $\varepsilon \in [0.6, 1]$, it changes slightly. (3) About 80% of the items are preserved when the value of ε is close to 0.60, which indicates that not only the diversity of the search space is guaranteed but also the size of it is greatly reduced.

2) Influence of ε on User's Search Burden: On the condition of finding a specified solution, the number of solutions interacted by the user is counted when the value of ε varies. Here, the laptop with the ranking as 316 in the order of selling amount is set as the searched target, and the results of 30 times of experiments are listed in Table VI.

From Table VI, we can observe that both the number of evaluated individuals and the value of DCC increase along with the increment of ε (the decrements of space reduction). Both the evaluation time and burden of the user are the least when ε=0.6, indicating that the algorithm is the most efficient.

It is possible we may not find the expected candidate in the reduced space if one attribute of the item was excluded, e.g. for the attribute x_i, its j-th value $x_{i,j}$ is excluded when $P\left(x_{i,j}|X^{\mathrm{K}}\right) < \left[1 - \sum_{x_{i,h} \in \bar{\bar{S}}_i} P\left(x_{i,h}|X^{\mathrm{K}}p\right)\right]$. Since most preferred and determined attributes are guaranteed in the subspace with the highest possibility by the historical knowledge and the Bayesian model, such a case seldom appears. If the low-probability event occurred, we will restart a new search by asking the user to provide more accurate retrieval words.

3) Estimated Fitness vs. Value of ε: On the condition of finding the same solution, laptop ranking 316, the fitness values of all individuals are estimated and the ranking of the expected solution is recorded in each generation. For clarity, the ordinals of the expected solution in different reduced spaces are normalized by dividing the total number of the items in that space. The average percentages of the ordinal number in 30 runs are presented in Fig. 5.

Conclusions can be drawn from Fig. 5: (1) When the search space is not reduced, the estimated fitness has a severe oscillation and a slow convergence, which demonstrates that the estimated fitness has difficulties in guiding effective evolution. (2) As the value of ε decreases, the oscillation is diminished,

and the convergence is accelerated.

In our experiment, ten variables are optimized, so the exploration space is reduced to $0.6^{10} \approx 0.006$ of the initial one when ε=0.6. In the reduced space, only those preferred variables are emphasized by evolution; therefore, the search is accelerated, and estimation accuracy is also improved, which is the main reason that the performance of evolution is enhanced when the control parameter ε decreases. In summary, in our experiment, the optimal value of ε is 0.6.

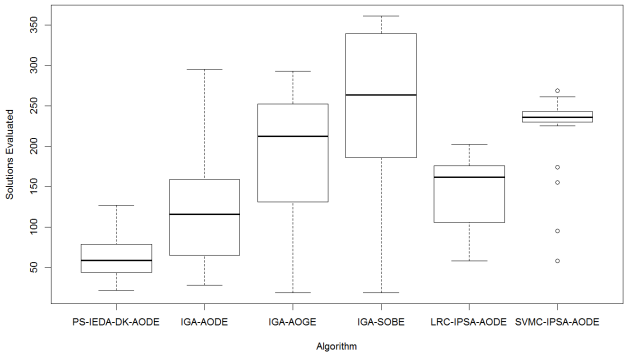
D. Objective Experiments with Precisely Defined Preference Function

A relative objective experiment is first conducted here by using a preference function to evaluate individuals instead of a real user. Supposing the expected solution is known, and the distance between a solution to the expected one is calculated as the preference function, i.e., solutions with smaller distances are more preferred ones.

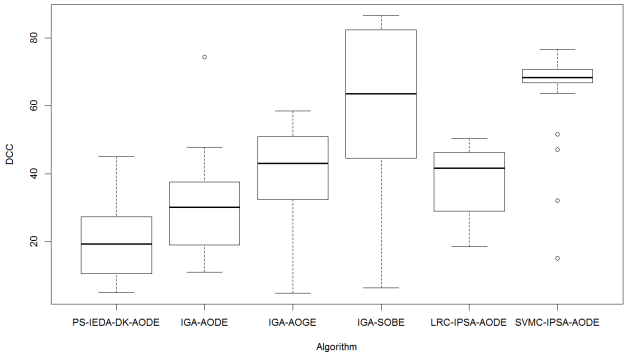
The proposed algorithm is compared with the IGA under three different encoding methods and two machine learning based interactive personalized search algorithms, i.e., our algorithm with attribute based ordinal decimal encoding (PS-IEDA-DK-AODE), IGA with attribute based ordinal decimal encoding (IGA-AODE), IGA with attribute based ordinal Gray encoding (IGA-AOGE), IGA with selling-amount based ordinal binary encoding (IGA-SOBE), LRC-IPSA with attribute based ordinal decimal encoding (LRC-IPSA-AODE), and SVMC-IPSA with attribute based ordinal decimal encoding (SVMC-IPSA-AODE). Three sets of experiments are delivered: (1) On the condition of finding a specified laptop, the numbers of items evaluated and DCC values of all algorithms are compared. (2) For finding 10 different designated laptops, the average numbers (standard deviation) of evaluated items and DCC values are compared. (3) The estimated fitness is recorded to demonstrate the effectiveness of the surrogate based evaluation.

1) Experiment 1: Supposing the expected laptop to be searched is the same as above, i.e., the item with the ranking as 316 in the order of selling amount with query words Color*black*Brand*Thinkpad. The average values and standard deviations of the numbers of evaluated items and DCC are shown in the eighth row in Table VII and Table VIII, respectively.

The box plots are further given in Fig. 6, and conclusions can be drawn as follows: (1) Among these 30 experiments, the proposed algorithm outperforms competitors, i.e., both the number of individuals evaluated by the fitness function and DCC are minimum, indicating that our algorithm can effectively alleviate the evaluation burden and accelerate the search. (2) The results of IGA-AODE are better than the other IGAs and classifier based algorithms; LRC-IPSA-AODE outperforms SVMC-IPSA-AODE. (3) The results of the compared algorithms are far worse than those of the proposed algorithm but better than that of the selling orders based search. (4) Both the average and the deviation of the result of our algorithm are smaller than those of the other algorithms, demonstrating that our algorithm outperforms the compared ones in better convergence, effectiveness, and stability.



(a) Number of evaluated items.



(b) Values of DCC.

Fig. 6. Results for a specified solution.

2) Experiment 2: For adequately addressing the performance of these compared algorithms, we randomly select ten solutions with different rankings as our targets and further analyze the experimental results. The average values and standard deviations on the indicators of 30 times of experiments are listed in Table VII and Table VIII, in which, the first column gives the selling orders of those expected items. The Mann-Whitney U-test with confidence level 0.95 is used to show the significance of our algorithm, and those results marked with the label † are significantly worse than the proposed algorithm.

According to the values in these two tables, we can conclude that our algorithm significantly outperforms the compared algorithms for most targets, and also the evaluation burden of our algorithm is greatly reduced and much less than those of the compared ones. Furthermore, the domain knowledge based decimal encoding scheme outperforms the binary ones in avoiding redundant and useless information introduced from the crossover and mutation.

3) Experiment 3: This experiment is designed to verify the approximation performance of the surrogate preference model. The laptop ranked with 243 is selected as the expected solution. Similar to the setting of Section IV-C3, the ordinal of the expected solution is normalized by dividing the total number of the items in the reduced space. The average percentages of the ordinal number in 30 runs are presented in Fig. 7.

Clearly, the ranking has a downward trend, i.e., along with the evolution, the fitness of the expected solution increases. Furthermore, there are a few fluctuations in the early evolutionary stage, which is mainly because training samples are

TABLE VII
AVERAGE AND STANDARD DEVIATION OF NUMBERS OF EVALUATED ITEMS FOR 10 EXPECTED SOLUTIONS

Ranking	PS-IEDA-DK-AODE	IGA-AODE	IGA-AOGE	IGA-SOBE	LRC-IPSA-AODE	SVMC-IPSA-AODE
153	**45.97(27.32)**	101.63(55.38)†	126.00(63.42)†	166.53(83.85)†	85.17(23.99)†	144.80(12.98)†
197	**56.17(30.72)**	128.90(59.58)†	96.60(56.04)†	197.73(108.80)†	173.93(57.03)†	214.13(52.96)†
200	**39.57(17.89)**	165.17(76.59)†	133.43(73.57)†	155.40(102.20)†	132.03(22.72)†	192.23(26.61)†
242	**76.87(17.88)**	100.73(72.03)	164.83(69.17)†	214.63(88.02)†	81.77(38.74)	150.07(9.74)†
243	**117.10(24.01)**	146.27(69.83)	173.97(110.76)	270.47(124.40)†	132.67(57.69)†	192.90(74.23)†
275	104.63(23.96)	138.10(64.37)	135.67(58.94)	249.33(114.43)	**89.47(33.73)**	166.13(14.75)
299	**81.33(30.01)**	163.27(83.68)†	208.40(73.14)†	264.73(125.94)†	99.27(45.32)	150.70(25.44)†
316	**59.87(24.90)**	117.33(60.53)†	192.77(74.07)†	245.83(97.78)†	145.43(41.31)†	224.53(46.08)†
340	**26.50(8.80)**	129.30(73.65)†	195.87(92.03)†	259.70(121.32)†	160.23(54.23)†	239.50(57.98)†
392	**121.23(33.15)**	130.60(81.07)	200.23(84.85)†	311.13(120.81)†	233.37(97.77)†	230.10(60.52)†

TABLE VIII
AVERAGE AND STANDARD DEVIATION OF DCC FOR 10 EXPECTED SOLUTIONS

Ranking	PS-IEDA-DK-AODE	IGA-AODE	IGA-AOGE	IGA-SOBE	LRC-IPSA-AODE	SVMC-IPSA-AODE
153	**20.28(12.04)**	29.96(14.38)†	38.75(16.82)†	59.39(24.59)†	24.13(5.60)†	48.70(3.83)†
197	**21.60(10.74)**	33.37(13.85)†	23.89(11.35)	47.02(23.29)†	50.19(14.52)†	61.76(14.61)†
200	**18.32(9.49)**	45.83(20.35)†	42.07(19.51)†	56.35(30.97)†	33.57(5.65)†	59.70(8.87)†
242	33.91(6.79)	29.93(19.55)	37.44(13.24)	51.84(19.80)	**22.09(9.52)**	49.99(3.18)
243	49.32(8.43)	35.41(15.49)	39.98(24.17)	69.55(29.00)	**32.46(13.44)**	55.19(19.71)
275	44.99(8.71)	41.39(18.05)	45.45(17.84)	84.86(33.01)	**24.60(9.20)**	56.42(5.74)
299	**31.58(8.76)**	43.20(19.90)	44.08(13.93)†	60.07(25.26)†	32.03(16.04)†	49.83(8.65)†
316	**26.53(8.75)**	30.78(13.29)†	40.33(13.61)†	59.63(22.02)†	38.30(9.67)†	64.99(12.85)†
340	**12.81(4.99)**	31.93(15.47)†	40.06(15.57)†	58.86(25.98)†	39.91(12.30)†	65.17(14.02)†
392	42.53(9.67)	**32.67(17.07)**	40.29(14.20)	68.87(24.80)	68.51(26.10)	60.21(15.15)

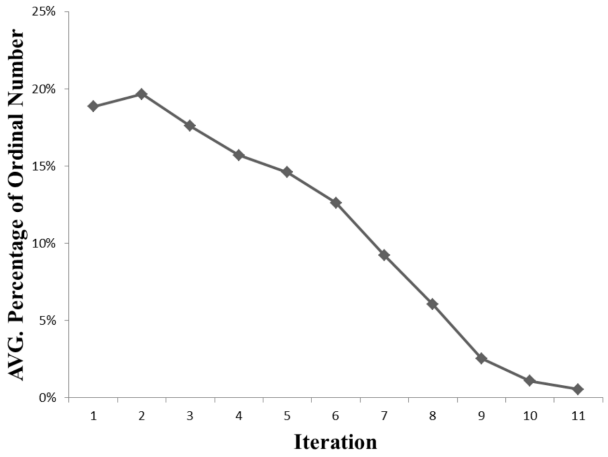


Fig. 7. AVG. percentage of ordinal number.

not enough to make a good approximation to the preference. As the evolution continues, the approximation performance of surrogate improves since the number of the samples gradually increases. Therefore, curves become smoother and smoother.

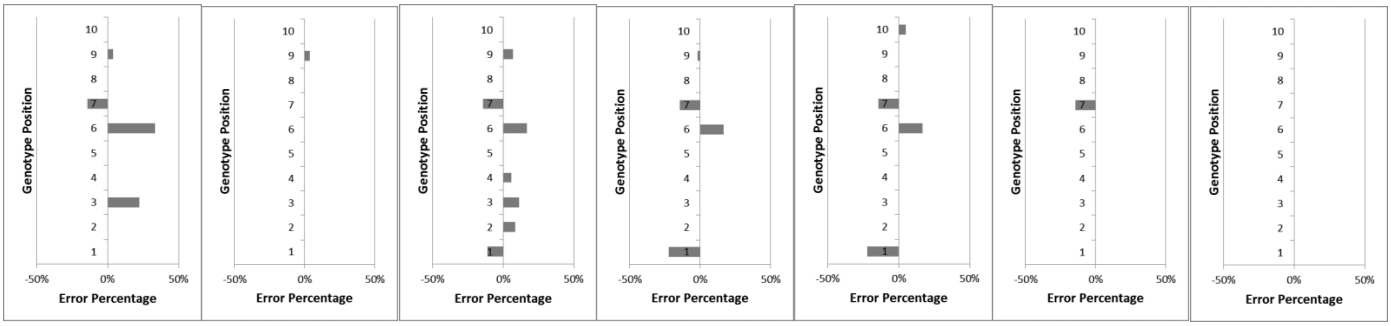
4) Experiment 4: The analysis of the genotypes is studied to further show the superiority of our algorithm. The varies of the distance on each gene between the expected solution and the current best individual in every generation can illustrate the dynamics of an EC. Therefore, such merit is used here. The laptop ranked with 243 is set as an expected solution. Our algorithm with decimal encoding, IGA-AODE, and LRC-IPSA-AODE are compared since they outperform other algorithms. For the decimal encoding, the varied range of each gene is quite different, and the absolute distance may cause confusion. Accordingly, the relative distance error, i.e., the percentage of the distance on a gene and the largest value of that gene, is calculated.

For clarity, the error percentages of seven iterations are recorded in figures 8, 9, and 10. In these figures, the ordinate indicates the genotypes positions and the grey bars are the errors. Shorter bar indicates a smaller error on the corresponding genotype. When the error gets zero, i.e., the genes of the current best individual and the expected one are the same, the grey bar will disappear, as can be seen in Fig. 8 (g).

From these figures, conclusions are drawn as follows: (1) With the same evolutionary generations, only our algorithm gets the expected solution. (2) The errors of the other two algorithms are larger than that of our method in total. (3) Our algorithm may have an ability to escape from local optimum as shown in Fig. 8 (b). All these indicate that our algorithm has a faster search, less user fatigue, and higher success.

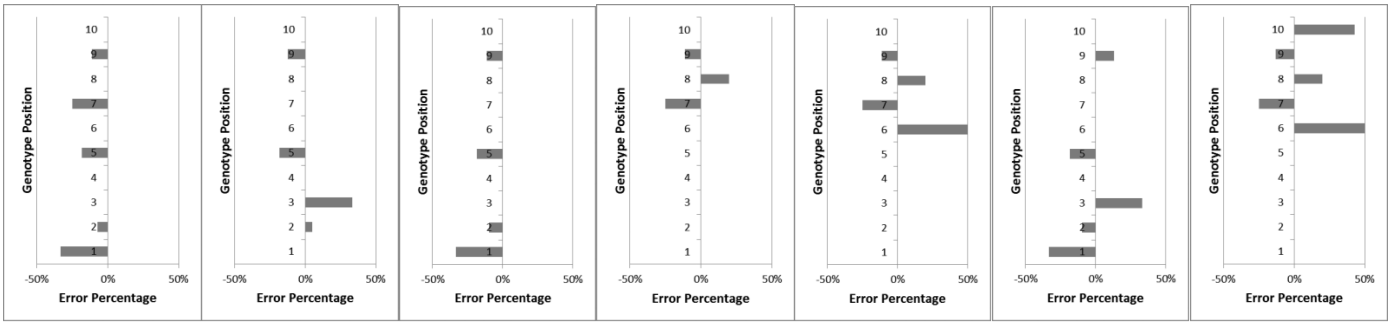
E. Comparative Experiments Involving a Real User

A real user is involved in this experiment to demonstrate the veritable performance of the proposed algorithm in alleviating user fatigue and accelerating search. The experimental setting here is similar to those of Section IV-D, and comparisons are conducted between PS-IEDA-DK-AODE, IGA-AODE, and LRC-IPSA-AODE.



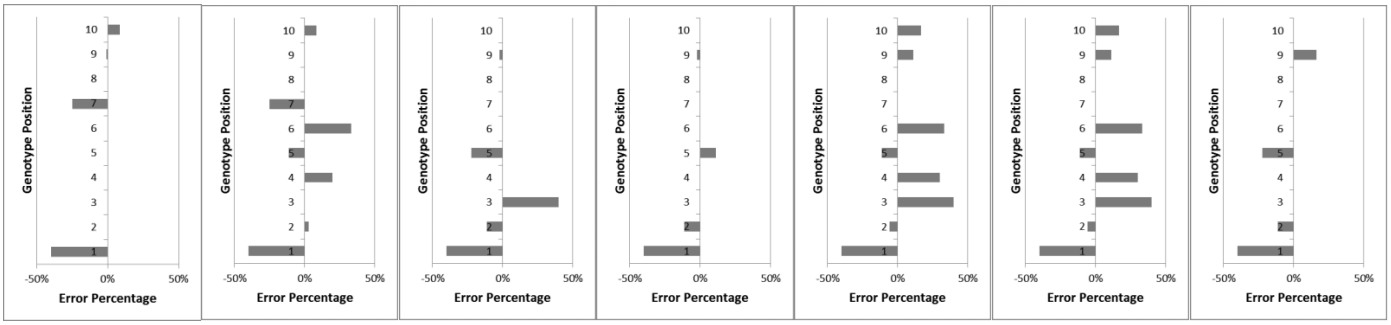
(a) Iteration=1 (b) Iteration=2 (c) Iteration=3 (d) Iteration=4 (e) Iteration=5 (f) Iteration=6 (g) Iteration=7

Fig. 8. Genotype analysis of PS-IEDA-DK-AODE.



(a) Iteration=1 (b) Iteration=2 (c) Iteration=3 (d) Iteration=4 (e) Iteration=5 (f) Iteration=6 (g) Iteration=7

Fig. 9. Genotype analysis of IGA-AODE.



(a) Iteration=1 (b) Iteration=2 (c) Iteration=3 (d) Iteration=4 (e) Iteration=5 (f) Iteration=6 (g) Iteration=7

Fig. 10. Genotype analysis of LRC-IPSA-AODE.

From the eighth row of Table IX and Table X, it is clear that the performance (search time and DCC) of the proposed algorithm is better than that of IGA-AOBE and LRC-IPSA-AODE. Specifically, the average search time of our algorithm is about 43.34% ($\frac{508.27}{1172.84} = 43.34\%$) of that of IGA-AOBE and 44.37% ($\frac{508.27}{1145.60} = 44.37\%$) of that of LRC-IPSA-AODE, and the DCC of our algorithm is about 36.31% ($\frac{12.79}{35.22} = 36.31\%$) of that of IGA-AODE and 36.72% ($\frac{12.79}{34.83} = 36.72\%$) of that of LRC-IPSA-AODE, which indicate that the proposed PS-IEDA-DK-AODE outperforms IGA-AOBE and LRC-IPSA-AODE in fast obtaining the expected solution and alleviating user fatigue.

Furthermore, ten different specified items are set as expected solutions, and the search time and DCC are compared among our algorithm, IGA-AODE, and LRC-IPSA-AODE. Table IX and Table X show the corresponding average values and standard deviations, in which the results marked with the label † represent that they are significantly worse than the proposed algorithm with the confidence level 0.95. Conclusions similar to those of the above experiments can also be drawn: the proposed algorithm has a higher search efficiency, less user fatigue, and better stability.

For the search efficiency and stability of the compared algorithms, results of subjective experiments (Table X) outperform those of the objective ones (Table VIII), which demonstrates that using a simple preference function to substitute a real user may not be a good choice for assessing the performance of IEC and machine learning based personalized search algorithms.

To summarize, (1) the proposed search space reduction strategy can not only improve the exploration efficiency but

TABLE IX
SEARCH TIME IN SUBJECTIVE COMPARATIVE EXPERIMENTS WITH TEN EXPECTED SOLUTIONS

Ranking	PS-IEDA-DK-AODE	IGA-AODE	LRC-IPSA-AODE
153	**311.49(116.71)**	1021.75(418.67)†	683.65(310.76)†
197	**565.28(285.05)**	1056.98(555.22)†	1530.42(652.00)†
200	**308.06(134.42)**	1432.31(723.64)†	1078.96(167.80)†
242	**748.92(150.28)**	1128.91(666.47)	820.85(297.67)
243	**777.60(193.56)**	1276.63(692.96)†	1469.38(137.00)†
275	**421.22(199.47)**	864.27(560.45)	846.12(309.06)†
299	**630.62(276.95)**	1534.43(593.05)†	783.53(368.58)
316	**508.27(171.58)**	1172.84(602.42)†	1145.60(391.83)†
340	**221.71(81.10)**	1275.50(727.51)†	1184.93(547.41)†
392	1087.18(259.65)	**953.72(478.87)**	2207.34(530.92)

TABLE X
DCC IN SUBJECTIVE COMPARATIVE EXPERIMENTS WITH TEN EXPECTED SOLUTIONS

Ranking	PS-IEDA-DK-AODE	IGA-AODE	LRC-IPSA-AODE
153	**16.48(6.72)**	33.86(14.19)†	22.05(8.04)†
197	**25.85(10.22)**	31.95(14.56)	46.37(18.51)†
200	**16.45(8.35)**	45.49(22.31)†	32.50(4.21)†
242	36.23(5.41)	**37.67(21.29)**	24.58(7.86)
243	39.41(7.30)	**34.16(15.98)**	40.88(3.83)
275	**23.33(8.43)**	31.06(18.48)	26.80(7.76)
299	31.59(11.93)	44.46(14.77)	**22.74(10.22)**
316	**12.79(8.30)**	35.22(17.03)†	34.83(9.34)†
340	**11.76(4.67)**	34.92(17.13)†	34.73(14.41)†
392	43.55(9.26)	**28.05(13.11)**	74.79(15.14)

also ensure the evolution diversity; (2) the approximation performance of the preference surrogate is improved in the reduced search space, which enhances the guidance of the evolutionary optimization search; (3) the proposed IEDA obviously outperforms the traditional IGA and machine learning based personalized search algorithms in improving the search efficiency, user's evaluation burden, and stability; (4)personalized search assisted with IEC will greatly improve the performance of the current E-commerce search in higher speed and less user fatigue; therefore, it is valuable to apply IEC to enhance the personalized search in E-commerce.

V. CONCLUSIONS

The personalized search is essentially an optimization problem but cannot be precisely defined with a mathematical model. Inspired by the merits of interest modeling, the mechanism of the interactive evolutionary algorithm, and the power of EDA, we present a personalized search oriented interactive estimation of distribution algorithm based on domain knowledge (PS-IEDA-DK). We propose a Naive Bayesian model based domain knowledge extraction, which is used not only to reduce the entire search space to a more preferred one but also to generate the initial population of IEDA. Moreover, according to the interactions performed in personalized search, a preference surrogate is designed to achieve the fitness estimation of EDA. The performance of IEDA in alleviating user fatigue and speeding up the search is experimentally demonstrated in the laptop search. Thus, PS-IEDA-DK is an alternative for improving the personalized search in E-commerce.

In the future, we will further focus on dynamically employing group intelligence to IEC for getting the more accurate preference model and reducing evaluation uncertainties in personalized search.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China with Grant No.61473298 and 61473299, the Fundamental Research Funds for the Central Universities with Grant No.2012QNA58, and the Innovation Project for College Graduates of Jiangsu Province with Grant No.KYLX16_0532.

REFERENCES

[1] I. Zelinka, "A survey on evolutionary algorithms dynamics and its complexity-mutual relations, past, present and future," *Swarm and Evolutionary Computation*, vol. 25, pp. 2–14, 2015.
[2] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Dec 2007.
[3] A. Arab and A. Alfi, "An adaptive gradient descent-based local search in memetic algorithm applied to optimal controller design," *Information Sciences*, vol. 299, pp. 117–142, 2015.
[4] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, Sep 2001.
[5] M. Fukumoto and S. Koga, *A Proposal for User's Intervention in Interactive Evolutionary Computation for Optimizing Fragrance Composition*. Cham: Springer International Publishing, 2014.
[6] H. Takagi, *Interactive Evolutionary Computation for Analyzing Human Characteristics*. Cham: Springer International Publishing, 2015.
[7] X. Sun and D. Gong, "Interactive genetic algorithms with individual's fuzzy and stochastic fitness," *Chinese Journal of Electronics*, vol. 18, no. 4, pp. 619–624, 2009.
[8] Y. Li, "Adaptive learning evaluation model for evolutionary art," in *2012 IEEE Congress on Evolutionary Computation*, June 2012, pp. 1–8.
[9] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, *An Interactive Simple Indicator-Based Evolutionary Algorithm (I-SIBEA) for Multiobjective Optimization Problems*. Cham: Springer International Publishing, 2015.
[10] A. B. Ruiz, M. Luque, K. Miettinen, and R. Saborido, *An Interactive Evolutionary Multiobjective Optimization Method: Interactive WASF-GA*. Cham: Springer International Publishing, 2015.
[11] X. Sun, Y. Lu, D. Gong, and K. Zhang, "Interactive genetic algorithm with CP-nets preference surrogate and application in personalized search," *Control and Decision*, vol. 30, no. 7, 2015.
[12] S. Zhou and Z. Sun, "A survey on estimation of distribution algorithms," *Acta Automatica Sinica*, vol. 33, no. 2, pp. 113–124, 2007.
[13] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions i. binary parameters," in *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, ser. PPSN IV. London, UK, UK: Springer-Verlag, Sep. 1996, pp. 178–187.
[14] M. Pelikan, M. W. Hauschild, and F. G. Lobo, *Estimation of Distribution Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
[15] J. Shen, L. Wang, and S. Wang, "A bi-population EDA for solving the no-idle permutation flow-shop scheduling problem with the total tardiness criterion," *Knowledge-Based Systems*, vol. 74, pp. 167–175, Jan. 2015.
[16] X. Hao, M. Gen, L. Lin, and G. A. Suer, "Effective multiobjective EDA for bi-criteria stochastic job-shop scheduling problem," *Journal of Intelligent Manufacturing*, vol. 8, pp. 1–13, Jan. 2015.
[17] J. Luo, Y. Qi, J. Xie, and X. Zhang, "A hybrid multi-objective PSO-EDA algorithm for reservoir flood control operation," *Appl. Soft Comput.*, vol. 34, no. C, pp. 526–538, Sep. 2015.
[18] L. Liu, H. Wang, and G. Kong, "An improved EDA for solving steiner tree problem," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 13, pp. 3483–3496, 2015.

[19] M. Hauschild and M. Pelikan, *Enhancing Efficiency of Hierarchical BOA Via Distance-Based Model Restrictions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[20] S. Baluja, *Incorporating a priori Knowledge in Probabilistic-Model Based Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[21] M. Pelikan, M. W. Hauschild, and P. L. Lanzi, *Transfer Learning, Soft Distance-Based Bias, and the Hierarchical BOA*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[22] M. W. Hauschild, M. Pelikan, K. Sastry, and D. E. Goldberg, "Using previous models to bias structural learning in the hierarchical BOA," in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '08. New York, NY, USA: ACM, 2008, pp. 415–422.

[23] F. Radlinski and T. Joachims, "Query chains: Learning to rank from implicit feedback," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05. New York, NY, USA: ACM, 2005, pp. 239–248.

[24] G. Wang and H. Liu, "Survey of personalized recommendation system," *Computer Engineering and Applications*, vol. 7, pp. 66–76, 2012.

[25] J. Chang, W. Zhou, J. Song, and D. Lin, "Design and implementation of a CP-nets-based and user preferences-oriented distributed policy-based agent architecture and algorithm," in *2010 Second International Conference on Communication Systems, Networks and Applications*, vol. 1, June 2010, pp. 155–159.

[26] O. Kassak, M. Kompan, and M. Bielikova, "User preference modeling by global and individual weights for personalized recommendation," *Acta Polytechnica Hungarica*, vol. 12, no. 8, pp. 27–41, 2015.

[27] X. Tang and J. Zhou, "Dynamic personalized recommendation on sparse data," *IEEE Trans. on Knowl. and Data Eng.*, vol. 25, no. 12, pp. 2895–2899, Dec. 2013.

[28] M. Abou-Zleikha and N. Shaker, *Evolving Random Forest for Preference Learning*. Cham: Springer International Publishing, 2015.

[29] M. Abou-Zleikha, N. Shaker, and M. G. Christensen, "Preference learning with evolutionary multivariate adaptive regression spline model," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 2184–2191.

[30] M. Kuzma and G. Andrejková, *Interactive Evolutionary Computation in Modelling User Preferences*. Cham: Springer International Publishing, 2015.

[31] H.-T. Kim, E. Kim, J.-H. Lee, and C. W. Ahn, "A recommender system based on genetic algorithm for music data," in *2010 2nd International Conference on Computer Engineering and Technology*, vol. 6, April 2010, pp. 414–417.

[32] H. J. Ahn, "Evaluating customer aid functions of online stores with agent-based models of customer behavior and evolution strategy," *Inf. Sci.*, vol. 180, no. 9, pp. 1555–1570, May 2010.

[33] V. Kant and K. K. Bharadwaj, *A User-Oriented Content Based Recommender System Based on Reclusive Methods and Interactive Genetic Algorithm*. India: Springer India, 2013.

[34] G. Dunwei, S. Xiaoyan, and R. Jie, "Surrogate models based on individual's interval fitness in interactive genetic algorithms," *Chinese Journal of Electronics*, vol. 18, no. 4, pp. 689–694, 2009.

[35] K. Järvelin and J. Kekäläinen, "IR evaluation methods for retrieving highly relevant documents," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '00. New York, NY, USA: ACM, 2000, pp. 41–48.

Yang Chen is a Ph.D. candidate majoring in control theory and control engineering in the School of Information and Electrical Engineering of China University of Mining and Technology, P.R. China. He is an interactive evolutionary computation (IEC) researcher with an interest in multi-objective optimization.

XiaoYan Sun received the PhD degree in control theory and control engineering from the China University of Mining and Technology in 2009. She is professor in the School of Information and Electronic Engineering, China University of Mining and Technology. Her research interests include interactive evolutionary computation, big data, and intelligence optimization. She is a member of the IEEE.

DunWei Gong received the PhD degree in control theory and control engineering from the China University of Mining and Technology in 1999. He is professor in the School of Information and Electronic Engineering, China University of Mining and Technology. His research interests include intelligence optimization and control. He is a member of the IEEE.

Yong Zhang received the B.S. and PhD degrees in Control theory and control Engineering from China University of Mining and Technology in 2006 and 2009, respectively. He is currently with the School of Information and Electronic Engineering, China University of Mining and Technology. His research interests include intelligence optimization and data mining. He is a member of the IEEE.

Jong Choi received his Ph.D. degree in Computer Science at Indiana University Bloomington in 2012 and his MS degree in Computer Science from New York University in 2004. He is a researcher working in Scientific Data Group, Computer Science and Mathematics Division, Oak Ridge National Laboratory (ORNL), Oak Ridge, Tennessee, USA.

Scott Klasky holds a Ph.D. in Physics from the University of Texas at Austin (1994), and has previously worked at the University of Texas at Austin, Syracuse University, and the Princeton Plasma Physics Laboratory. Dr. Klasky is a co-author on over 190 papers, and is the team leader of the Adaptable I/O System (ADIOS), which won an R&D 100 Award in 2013.