Auto-generated Test Paper Based on Knowledge Embedding

Duan Zeng-Hui, Hu Xing-Liu, Hao Guo-Sheng, Luo Fang, He Xiao-Dan, and He Yi-Yang

***Abstract*—Auto-Generated Test Paper (AGTP) has been deeply studied for many years, however, it is still a difficult problem and the certainty to access the best test paper (TP) is not guaranteed yet. In this paper, we put forward a method for AGTP based on knowledge embedding, which makes AGTP easier and faster. The knowledge to be embedded is studied and the mechanism behind it is analyzed. The embedded knowledge in this paper is from both the constraints of TP and the information of question repository (QR). The experiments validated the proposed method and found it is not only faster but also costs less computational resources to access the best TP than other method, such as evolutionary algorithm. What impressed is that the cost time to access the optimum does not rapidly increase with the size of QR. The knowledge plays the important role in AGTP, especially to efficiently improve the performance of the algorithms.**

***Index Terms*—Auto-generated test paper, evolutionary algorithm, knowledge, population initiation.**

I. Introduction

Test Paper (TP) has been the most widely used in the evaluation of learners for hundreds of years. With the development of information technology, the electric version TP becomes popular. And except for the electric TP created by experts, those auto generated are also widely adopted. With the Auto-Generated Test Paper (AGTP), not only the cost decreases for the time-saving of experts, but also the cheat behaviors decrease for the avoiding emission of the questions in TP. To avoid cheating, the system can even generate different TP for different students.

However, AGTP belongs to composition optimization problem and it is not an easy task. Therefore, many methods for AGTP were deeply studied. The first kind of methods are based on random selection. WANG made use of random algorithm for AGTP [1], in which, the candidate questions are divide into different group according to their properties. Those questions in the same group are considered homogeneous, therefore, for the optimization and satisfaction of the constraints of AGTP, they are equally considered and can be randomly selected. However this method needs to change the selected possibility for questions during the generation of TP and the knowledge is not used. The second kind of methods are based on Evolutionary Algorithms (EAs), such as the utility of Genetic Algorithm (GA) [2], [3], particle swarm optimization [4] and so on. Generally speaking, in GA based AGTP, a TP is coded into a chromosome, in which each gene is corresponding to a candidate question. Allele 1 means that the question is selected in the candidate TP, and vice versa for allele 0. In order to improve the quality of AGTP, Ding [5] designed a GA with odd crossover and even bit mutation operation. Zhang combined ant colony optimization [6] and GA together [7] to improve the quality of AGTP [8]. However, the knowledge is seldom used during the optimization.

Incorporation of the prior knowledge in algorithms has received increasing interest in recent years [9]-[12]. The embedded knowledge includes expert knowledge [13], meta-heuristics [14], [15] and human preferences [16], as well as domain knowledge [17], [18] acquired during the search process. It has been shown from various motivations that knowledge incorporation into search process could significantly improve the performance. One of the reasons for us to study the knowledge embedded AGTP is that we observed an impressed phenomenon, which is given in section II. Then the knowledge embedded in AGTP is studied in section III, and also the probability distribution and probability to access the best TP in initial population are analyzed. In section IV, two methods to embed knowledge are proposed: a coarse one and a fine one. Then the validated experiments are given in section V. At last, this paper is concluded in section VI.

II. An Impressed Phenomena in Auto-generated Test Paper

A. Randomness in AGTP

Generally speaking, in EAs, the initial population of candidate TPs is randomly generated and by which it is expected that each class of TP can be coved with equal probability. This is generally implemented by allocating equal probability to each candidate question for TP. Therefore, the uniform distribution is adopted with the value of mathematical expectation as 0.5. It is expected that the fitness of optimum in initial population does not varies with target score.

However, during our application, an impressed

Manuscript received December 22, 2018; revised June 13, 2019. This work was supported in part by Key Projects of National Innovation and Entrepreneurship Training Program for College Students in 2017 (No. 201710320027, 201710320024Z, 201710320120X), Jiangsu Province's Natural Science Foundation (BK20171114), Qing Lan Project, Jinling Institute of Technology's Talent Introduction Project (No.Jit-rcyj-201604), the National Natural Science Foundation of China (No.61673196).

Duan Zhen-Hui, Hao Guo-Sheng, He Xiao-Dan, and He Yi-Yang are with the School of Computer Science & Technology, Jiangsu Normal University, Xuzhou, 221000, China (e-mail: hgskd@jsnu.edu.cn).

Hu Xing-Liu is with the College of Intelligent Science and Control Engineering, Jinling Institute of Technology, Nanjing, Jiangsu, 211169, China (e-mail: xinghu8@163.com).

Luo Fang is with the School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China (e-mail: 18796328906@163.com).

doi: 10.18178/ijiet.2019.9.9.1286

phenomenon happened and it run out of our expectation. Our task is to generate TPs with both EAs and experts, i.e., the experts will specify some questions, denoted as Q1, and the algorithm will generate the other questions, denoted as Q2. During the process, we found that the time cost for EA to generate TP varies with the specified target score of Q2. This is intuitively out of our expectation that the fitness of best TP in initial population should not vary with target score. Then we formally model a simplified AGTP as following.

B. A Simplified Model of AGTP

If we can clearly get the rule behind the phenomenon, then we can use it to make AGTP quickly and easily. In order to study the rule and explain it clearly, here we formally studied a simplified AGTP, labeled as SAGTP, which excludes some constraints of TP and only try to generate TP to satisfy the constraints of the target score. In fact, for another constraint of coverage rate of knowledge, because of the randomness of initiation of population of questions in EA, it can be naturally satisfied.

Label the candidate questions repository as $QR=\{q_1,q_2,\ldots,q_{|QR|}\}$, then the TP generated by EA is a subset of QR, i.e. $TP \subseteq QR$. For any question in QR, we can define a function as follows:

$$b(q_i)=\begin{cases}1, q_i \in TP \\ 0, q_i \notin TP\end{cases} \tag{1}$$

This function means that if the candidate question q_i is selected in TP, then $b(q_i)=1$, vice versa for $b(q_i)=0$.

The model of AGTP can be represented as:

$$\begin{cases}\min f(X)=\left|st-S(X)\right| \\ s.t. H(\mathrm{X}) \le H_0\end{cases} \tag{2}$$

where X represents the candidate TP, and st is the target score specified for purpose, and $S(X)$ is the sum score of questions in the candidate TP. $H(X) \le H_0$ is the constraints. $X \in [0,1]^{|QR|}$ is the composition of $b(q_i)$:

$$X=\{b(q_1), b(q_2),\ldots, b(q_{|QR|})\} \tag{3}$$

Label the score of q_i as $s(q_i)$, then the optimization of SAGTP can be formulated as:

$$\min f(X)=\left|\sum s(q_i)-st\right|, q_i \in TP \tag{4}$$

And it also can be formulated as:

$$\min f(X)=\left|\sum s(q_i)b(q_i)-st\right|, q_i \in QR \tag{5}$$

C. The Impressed Phenomenon

The algorithm for the optimization of SAGTP in experiment was canonic GA. The evolutionary operators include roulette wheel selection, single point crossover and single point mutation. Also the elite retention strategy was adopted. The parameters for GA is as follows: crossover probability with 0.9 and mutation probability with 0.1. The initial population of TP is randomly generated with probability value 0.5.

In order to study the phenomenon, in the experiments, we varied st from 0 to 195 with step 5. We first collect the best TP in initial population P(0) and calculate the best TP $X_o(0)=\arg_X \min f(X), X \in \mathrm{P}(0)$. Then we depict the relationship between the fitness of $X_o(0)$ and the different problems, which is characterized with different value of st, in Fig. 1, which is the result of 100 runs of EAs for each st. The lower the fitness, the better the generated TP is, and when fitness is 0 means that the generated TP is the optimum.

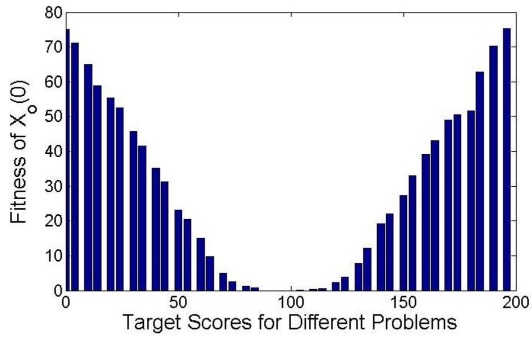

Fig. 1. Relationship between the fitness of Xo(0) in initial population and the target scores representing different optimization problems.

Fig. 1 shows that the fitness of the best TP in initial population set of AGTPs is related with the target score. For the extreme case st=0, the average fitness of the optimum $f_0(Xo(0))=75.077$ and when the target score st=5, $f_1(Xo(0))=71.23$. However, when $st \in \{90,95,100\}$, all the fitness of the best solution in initial population is 0, which means that the global optimum is accessed in initial population without exception! The average fitness of $f_i(Xo(0)$ is a monotonous increasing function of the distance defined as below:

$$d(st)=\min|st-s_0|, s_0 \in \{90,95,100\}$$

What does the above phenomena shown in Fig. 1 means? (1) The method to initiate the population is very suitable for the optimization problems with target score of {90, 95, 100} and not suitable for those far from {90, 95, 100}. Why does it happen? (2) There could be other methods that are suitable for other target score of AGTP. Then what are the methods? (3) Knowledge about the specified TP can be used to guide the method to initiate the population. What is the knowledge and how can we make use of it in EAs based on AGTP? We will answer these questions in the following sections.

III. Analysis of the Phenomenon Based on Probability Theory

A. Knowledge in AGTP

In order to study the phenomenon with probability theory, we introduce random variable here.

Let S_s denote the random variable of the sum score of candidate TP X. Different questions may be specified with different score in QR. Suppose the set of the type of score is

Then for the initial population P(0) with size of |P|, it will be an n-Bernoulli test with probability of $\Pr\{S_S=st\} \geq \max_{1\leq i\leq |W|} \Pr(W_i)$. Label N as the number of solution that belongs to the best TP, then the probability to access them in initial population is:

$$
\begin{aligned}
&\Pr(N>0)\\
&=1-\Pr(N=0)\\
&=1-C_{|P|}^{0}\Pr(S_s(X_i)=st)^0(1-\Pr(S_s(X_i)=st))^{|P|}\\
&=1-(1-\Pr(S_s(X_i)=st))^{|P|}\\
&\geq 1-\prod(1-\max\Pr(W_i))\\
&=1-(1-\max\Pr(W_i))^{|P|}
\end{aligned}
$$

From another view, there is:

$$
\begin{aligned}
&\Pr(N>0)\\
&=1-\Pr(S_s(X_1)\neq st, S_s(X_2)\neq st,\cdots,S_s(X_{|P|})\neq st)\\
&=1-\prod \Pr(S_s(X_i)\neq st)\\
&=1-\prod(1-\Pr(S_s(X_i)=st))\\
&\geq 1-\prod(1-\max\Pr(W_i))\\
&=1-(1-\max\Pr(W_i))^{|P|}
\end{aligned}
$$

Therefore we get the conclusion:

$$\Pr(N>0)\geq 1-(1-\max\Pr(W_i))^{|P|} \tag{16}$$

The higher $\Pr(N>0)$ is, the more best-TPs were generated in initial population. In order to make $\Pr(N>0)$ higher, we can make $\max\Pr(W_i)$ higher.

IV. Knowledge Embedded AGTP based on EAs

Based on the above analysis, this section will present two theorems for the knowledge embedding for AGTP in the population initialization.

A. A Coarse Knowledge Embedded Method

Label S_a as the sum score of candidate questions in *QR*. It is easy to know that $S_a=\sum(|S_i|s_i)$.

Theorem 1: When the probability for any question to be selected is set as $p=st/s_a$, the best TP will be accessed in initial population.

Proof: According to Bernoulli distribution, there is $E(w_i)=p|S_i|$. With $S_s=\sum(w_i s_i)$, we can get:

$$
\begin{aligned}
&E(|S_s-st|)\\
&=|\sum E(w_i)s_i-st|\\
&=|\sum p|S_i|s_i-st|\\
&=|p\sum|S_i|s_i-st|\\
&=|pS_a-st|
\end{aligned}
$$

Because $E(|S_s-st|)=|pS_a-st|$, in order to generate the best TP in initial population, what we can adjust is the sample distribution.

Let $E(|S_s-st|)=0$, i.e., the best TP is accessed, we can get $|pS_a-st|=0$, therefore, there is:

$$p=st/S_a \tag{17}$$

Then the coarse knowledge embedded method for best TP generation is to specify p value in (17), which is related with *st*, and it is the knowledge of the problem to be optimized.

B. A Fine Knowledge Embedded Method

In the coarse method, we assign the same value of p for different type score of questions. Furtherly, we can assign different p value for them to get a best TP in initial population.

Label the feasible number for the questions with score s_i in best TP as w_i, then we can get the theorem as below.

Theorem 2: When the probability for the question with score s_i to be selected is set as $p_i=(w_is_i)/s_{ai}$, the best TP will be accessed in initial population.

Proof: Taking the sampling from different group of questions with the same score as the Bernoulli distribution with different probability, then there is $E(w_i)=p|S_i|$. Therefore, the equation can be written in a more fine as following:

$$
\begin{aligned}
E(|S_s-st|)&=|\sum E(w_i)s_i-st|=|\sum p_i|S_i|s_i-st|\\
&=|\sum p_iS_{ai}-st|.
\end{aligned}
$$

Let $E(|S_s-st|)=0$, i.e., the best TP is accessed, we can get:

$$|\sum p_iS_{ai}-st|=0 \tag{18}$$

Because the feasible number for the question with score s_i in best TP is w_i, therefore, there is:

$$|\sum w_is_i-st|=0 \tag{19}$$

Combined with (18), we can get: $p_iS_{ai}=w_is_i$. Label $st_i=w_is_i$, Then there is:

$$p_i=st_i/S_{ai},\ i=1,2,\cdots,|S_p| \tag{20}$$

According to this theorem, all the feasible value of w_i can be used as knowledge to be embedded.

It is easy to see that Theorem 1 is just a case of Theorem 2. Here pi is the fine embedded knowledge, it is not only related with *st*, but also related with Sai which is the knowledge of QR.

Eq. (19) is an indeterminate equation, it provides the interface for user to control/optimize the TP flexibly. For example, providing there are 5 type of scores for all the questions in QR and the set of the scores is {2, 3, 5, 10, 15}. Then when *st* is set to 100. A set of number as {20, 10, 0, 0, 2} is one of the feasible candidate for the best TP, i.e., 100=2 20+3 10+5 0+10 0+15 2. Then according to *Theorem* 2, we can get the corresponding probability as $(p_1,p_2,p_3\ p_4,p_5)=((20\cdot2)/s_{a2},(10\cdot3)/s_{a3},0,0,(2\cdot15)/s_{a15})$.

For utility of Theorem 2, two steps are necessary for the optimization. The first step is the optimization of $W_i=\{w_1,w_2,\cdots,w_{|Sp|}\}$ and the second step is the optimization

Then for the initial population P(0) with size of |P|, it will be an n-Bernoulli test with probability of $\Pr\{S_S=st\} \geq \max_{1\leq i\leq |W|} \Pr(W_i)$. Label N as the number of solution that belongs to the best TP, then the probability to access them in initial population is:

$$
\begin{aligned}
&\Pr(N>0)\\
&=1-\Pr(N=0)\\
&=1-C_{|P|}^{0}\Pr(S_s(X_i)=st)^0(1-\Pr(S_s(X_i)=st))^{|P|}\\
&=1-(1-\Pr(S_s(X_i)=st))^{|P|}\\
&\geq 1-\prod(1-\max\Pr(W_i))\\
&=1-(1-\max\Pr(W_i))^{|P|}
\end{aligned}
$$

From another view, there is:

$$
\begin{aligned}
&\Pr(N>0)\\
&=1-\Pr(S_s(X_1)\neq st, S_s(X_2)\neq st, \cdots, S_s(X_{|P|})\neq st)\\
&=1-\prod\Pr(S_s(X_i)\neq st)\\
&=1-\prod(1-\Pr(S_s(X_i)=st))\\
&\geq 1-\prod(1-\max\Pr(W_i))\\
&=1-(1-\max\Pr(W_i))^{|P|}
\end{aligned}
$$

Therefore we get the conclusion:

$$\Pr(N>0)\geq 1-(1-\max\Pr(W_i))^{|P|} \tag{16}$$

The higher Pr(N>0) is, the more best-TPs were generated in initial population. In order to make Pr(N>0) higher, we can make $\max\Pr(W_i)$ higher.

IV. Knowledge Embedded AGTP based on EAs

Based on the above analysis, this section will present two theorems for the knowledge embedding for AGTP in the population initialization.

A. A Coarse Knowledge Embedded Method

Label S_a as the sum score of candidate questions in *QR*. It is easy to know that $S_a=\sum(|S_i|s_i)$.

Theorem 1: When the probability for any question to be selected is set as $p=st/s_a$, the best TP will be accessed in initial population.

Proof: According to Bernoulli distribution, there is $E(w_i)=p|S_i|$. With $S_s=\sum(w_i s_i)$, we can get:

$$
\begin{aligned}
&E(|S_s-st|)\\
&=|\sum E(w_i)s_i-st|\\
&=|\sum p|S_i|s_i-st|\\
&=|p\sum|S_i|s_i-st|\\
&=|pS_a-st|
\end{aligned}
$$

Because $E(|S_s-st|)=|pS_a-st|$, in order to generate the best TP in initial population, what we can adjust is the sample distribution.

Let $E(|S_s-st|)=0$, i.e., the best TP is accessed, we can get $|pS_a-st|=0$, therefore, there is:

$$p=st/S_a \tag{17}$$

Then the coarse knowledge embedded method for best TP generation is to specify *p* value in (17), which is related with *st*, and it is the knowledge of the problem to be optimized.

B. A Fine Knowledge Embedded Method

In the coarse method, we assign the same value of *p* for different type score of questions. Furtherly, we can assign different *p* value for them to get a best TP in initial population.

Label the feasible number for the questions with score s_i in best TP as w_i, then we can get the theorem as below.

Theorem 2: When the probability for the question with score s_i to be selected is set as $p_i=(w_i s_i)/s_{ai}$, the best TP will be accessed in initial population.

Proof: Taking the sampling from different group of questions with the same score as the Bernoulli distribution with different probability, then there is $E(w_i)=p|S_i|$. Therefore, the equation can be written in a more fine as following:

$$
\begin{aligned}
E(|S_s-st|)&=|\sum E(w_i)s_i-st|=|\sum p_i|S_i|s_i-st|\\
&=|\sum p_i S_{ai}-st|.
\end{aligned}
$$

Let $E(|S_s-st|)=0$, i.e., the best TP is accessed, we can get:

$$|\sum p_i S_{ai}-st|=0 \tag{18}$$

Because the feasible number for the question with score s_i in best TP is w_i, therefore, there is:

$$|\sum w_i s_i-st|=0 \tag{19}$$

Combined with (18), we can get: $p_i S_{ai}=w_i s_i$. Label $st_i=w_i s_i$, Then there is:

$$p_i=st_i/S_{ai},\ i=1, 2, \cdots, |S_p| \tag{20}$$

□

According to this theorem, all the feasible value of w_i can be used as knowledge to be embedded.

It is easy to see that Theorem 1 is just a case of Theorem 2. Here pi is the fine embedded knowledge, it is not only related with *st*, but also related with Sai which is the knowledge of QR.

Eq. (19) is an indeterminate equation, it provides the interface for user to control/optimize the TP flexibly. For example, providing there are 5 type of scores for all the questions in QR and the set of the scores is {2, 3, 5, 10, 15}. Then when *st* is set to 100. A set of number as {20, 10, 0, 0, 2} is one of the feasible candidate for the best TP, i.e., 100=2·20+3·10+5·0+10·0+15·2. Then according to *Theorem* 2, we can get the corresponding probability as $(p_1,p_2,p_3\ p_4,p_5)=((20\cdot 2)/s_{a2},(10\cdot 3)/s_{a3},0,0,(2\cdot 15)/s_{a15})$.

For utility of Theorem 2, two steps are necessary for the optimization. The first step is the optimization of $W_i=\{w_1,w_2,\cdots,w_{|Sp|}\}$ and the second step is the optimization with the initial population which is generated according to (20).

V. Experiments

After the observation of the mentioned impressed phenomenon, we carried out experiments to compare the knowledge embedded EAs (kmEAs) with canonical EAs (cEAs) and found that kmEAs always perform faster than cEAs. Therefore, it is unnecessary to continue the comparison between kmEAs and cEAs. But for the Coarse and Fine Knowledge Embedded Methods (CKEMs and FKEMs), we don't know which one performs better and how better it is. This is the focus of the experiments.

A. Experiments Setup

The setup of the experiments mainly includes the objects to be compared, the conditions of the experiments, and the indexes to be compared.

The objects to be compared include two methods, CKEM and FKEM.

The conditions of the experiments include: (1) population size, which ranges from 10 to 50 with step 10; (2) the size of QR, which ranges in the following set {198, 352, 606, 819, 1373}, which are the numbers of questions of certain subjects in our QR. We expect that the bigger the population size, the more effective the method is and the bigger the size of QR, the more difficult the problem is.

The indexes to be compared include: 1) the cost time to access the best TP. The less time cost, the more efficient the method is. 2) The number of the best TP accessed. The bigger the number of the best TP accessed, the more efficient the method is.

The experiments were carried out on personal computer. The CPU is Intel Core i7-6700HQ 2.6GHz and the memory size is 24G, and the disk is solid state disk 512G.

B. Algorithms

The experiments include three basic algorithms. The first algorithm is named as AGTP, which outputs a TP represented by a binary string. And its inputs include a probability value $p=st/S_a$ and a candidate question repository *QR*. The pseudo code of the algorithm AGTP is shown in Table I.

TABLE I: Pseudo Code of Algorithm AGTP

Name	AGTP
Input	*p*, *QR*
Output	A *TP* represented by binary strings

```
String X=""                      //X represents the i-th TP
 for q in QR                     //For each question in QR
   if random()<p                 //Compare the random and p value
     X +="1"                     //This question is selected in TP
   else
     X +="0"                     //This question is not selected in TP
   endif
 endfor
return X                         //Return the binary string
```

The second algorithm is named as "CKEM", which outputs a list of TP based on that from Algorithm I. And its inputs include *QR* and the target score *st*. Both *QR* and *st* are the source of knowledge to be embedded in the algorithm. The pseudo code of the algorithm CKEM is shown in TABLE II.

TABLE II: Pseudo Code of the Algorithm CKEM

Name	CKEM
Input	*st*, *QR*
Output	*TP*s represented by binary strings

```
int sa=0                             //Define sum score sa
for q in QR
  sa+=s(q)                           //Add all score to sa
endfor
List<String> result
for i=1 to |P|
 result.add(AGTP(st/sa, QR))         //Call algorithm AGTP
return result
```

The third algorithm is named as "FKEM", which also outputs a list of TP based on that from Algorithm AGTP. It has the same inputs as Algorithm CKEM. But it include two main steps to produce the knowledge. The first step is to generate $p_i=st_i/S_{ai}$ and then the second step is to call Algorithm AGTP similar to Algorithm CKEM. In FKEM, the solutions for $|\sum w_i s_i$ -*st*|=0 are attained by another GA, which also cost time in the experiments. The pseudo code of the algorithm FKEM is shown in TABLE III.

TABLE III: Pseudo Code of the Algorithm FKEM

Name	FKEM
Input	st, QR
Output	TPs represented by binary strings

```
Map<int, List<question>> map1 //Define a map for each S_i
for q in QR
  map1.get(s(q)).put(s(q), q)
   //Classify questions according to s(q)
endfor
store solutions for |∑w_i s_i -st|=0 into W
//Solving this equation with EAs or traversal method
List<String> result
for i=1 to |P|
  X=""
  for w in W
    QR_i= map1.get(s(q))
    X+= AGTP(st_i,QR_i)        //Call algorithm AGTP
  endfor
  result.add(X)                //Add the TP into result
endfor
return result
```

C. Results of the Experiments

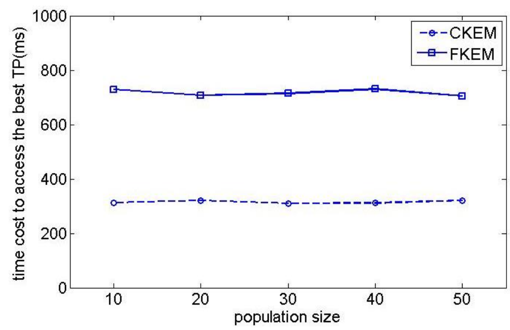

Fig. 2. Comparison of the time cost to access the best TP between CKEM and FKEM with the different population size.

First, with different population size, the time cost to access the best TP is compared between CKEM and FKEM and the result is shown in Fig. 2. It shows that the time cost to access the best TP does not increases or decrease with the population size. This means that the time cost is almost unrelated with population size. Then we studied the number of the order of individual that the first hit of the best TP in the population and found that the average value is 1.246, which means that if only the population size is greater than 2, the best TP can be accessed.

Secondly, with the different size of QR, the time cost to access the best TP is compared between CKEM and FKEM and the result is shown in Fig. 3. It shows that under both of the two cases, the time cost to access the best TP increases with the size of QR. But the time cost is acceptable in real application and the time most cost is 982ms.

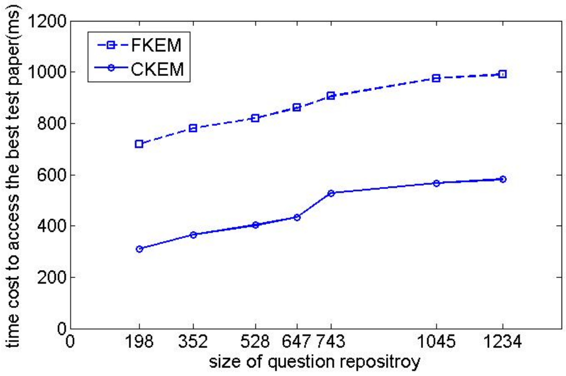

Fig. 3. Comparison of the time cost to access the best TP between CKEM and FKEM with the different size of QR.

Although from above two comparison, FKEM does not dominate CKEM. But when we compare the number of best TP hit by the algorithms, we found that FKEM is better than CKEM. This is carried out with different population size, ranging from 5 to 50 with step 5, then the number of best TP that the algorithms access are compared. The results is show in Fig. 4, in which the real number of best TP is 15. We found that FKEM can access all the 15 best TP when population size is greater than 30, while for CKEM, the number increases with the population size slowly.

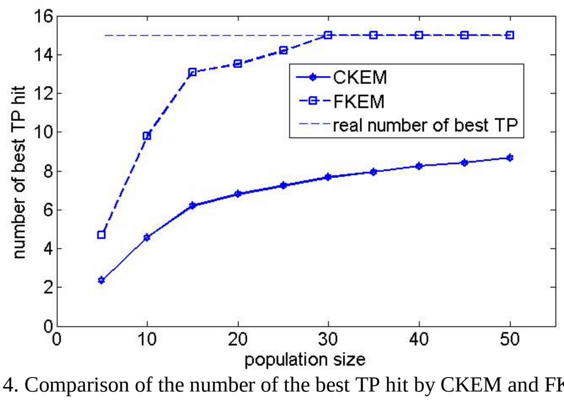

Fig. 4. Comparison of the number of the best TP hit by CKEM and FKEM with the different population size.

From the experiments, we found that in the index of time cost to access the best TP, CKEM is better than FKEM, but for the index of the number of best TP hit, FKEM is better than CKEM. What's more, both CKEM and FKEM are faster than canonical EAs and the increasing of the time cost to access the best TP with the size of *QR* can almost be omitted.

VI. Conclusion

In order to accelerate AGTP, an impressed phenomenon is presented and the rule behind it is studied. At the same time, we study the knowledge in AGTP and propose two methods, named as the coarse one and the fine one, to make use of the knowledge. The experiments show that CKEM is faster than FKEM, but FKEM can find more best TP than CKEM. The study of knowledge embedded methods provide a way for the acceleration of AGTP.

Acknowledgment

All the authors will thank XIE Chun-Li, an associate professor of School of Computer Science & Technology, Jiangsu Normal University. She gave us many academic and constructive advices, and helps us to correct this paper.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

G.-S. Hao conducted the research and wrote the paper; Z. H. Duan, F. Luo, X.-D. He and Y.-Y He carried out the experiments; X. L. Hu and Z. H. Duan analyzed the data and polished the paper; all authors had approved the final version.

References

[1] M. Wang, H.-J. Jin, and X.-R. Wang, "Research on set at random algorithm in intelligent generating test paper (in Chinese)," *Computer Engineering and Design*, vol. 27, no. 19, pp. 3583-3585, May 2006.

[2] R. Nan and J. Li, "Research on intelligent test paper based on improved genetic algorithm," *Social Informatics and Telecommunications Engineering*, pp. 173-176, 2018

[3] Y. C. Zhou, Y. Y. Li, and C. Feng, "Research on intelligent test paper auto-generating algorithm based on improved GA," in *Proc. 2009 Chinese Control and Decision Conference*, pp. 3938-3942, Guiling, China, June17-19, 2009.

[4] C. Zhang and J. Zhang, "Research on test paper auto-generating based on improved particle swarm optimization," *Seventh International Symposium on Parallel Architectures*, pp. 92-96, January 21, 2016.

[5] D. Ding, "Design and implementation of heuristic test paper composing algorithm and online exam system," Department of software, HuNan University, Changsha, 2013.

[6] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, Dec 2006.

[7] R Nowotniak and J. Kucharski, "GPU-based tuning of quantum-inspired genetic algorithm for a combinatorial optimization problem," *Bulletin of the Polish Academy of Sciences Technical Sciences*, vol. 60, no. 2, pp. 323-330, 2015.

[8] Y. K. Zhang, "Application of improved genetic algorithm in auto-generating paper," Master dissertation, School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou, 2017.

[9] D. Lim, Y. S. Ong, A. Gupta, C. K. Goh, and P. S. Dutta, "Towards a new Praxis in optinformatics targeting knowledge re-use in evolutionary computation: Simultaneous problem learning and optimization," *Evolutionary Intelligence*, vol. 9, no. 4, pp. 203-220, Dec 2016.

[10] J. Xidong and R. G. Reynolds, "Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: A cultural algorithm approach," in *Proc. the 1999 Congress on Evolutionary Computation*, pp. 1672-1678, vol. 3, 1999

[11] J. Johnson and S. J. Louis, "Case-initialized genetic algorithms for knowledge extraction and incorporation," *Knowledge Incorporation in Evolutionary Computation*, New York: Springer-Verlag Berlin Heidelberg, 2005, pp. 57-80.

[12] M. Khajah, R. Wing, R. Lindsey, and M. Mozer, "Integrating latent-factor and knowledge-tracing models to predict individual differences in learning," in *Proc. the 7th International Conference on Educational Data Mining*, pp. 99-106, 2014

[13] R. F. Coelho and P. Bouillard, "A multicriteria evolutionary algorithm for mechanical design optimization with expert rules," *International*

Journal for Numerical Methods in Engineering, vol. 62, no. 4, pp. 516-536, April 2005.
[14] Y. Zhang, X. Song, and D. Gong, "A return-cost-based binary firefly algorithm for feature selection," *Information Sciences*, vol. 418, pp. 561-574, Aug 2017.
[15] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, pp. 1-21, Jan 2013.
[16] G.-S. Hao, D.-W. Gong, J. Yuan, Y.-R. Yan, and J.-r. Yan, "User's attention knowledge learning in interactive evolutionary computation," presented at the Control and Decision Conference (CCDC), Guilin, China, June 17-19, 2009.
[17] K. Kim and S. Cho, "Systematically incorporating domain-specific knowledge into evolutionary speciated checkers players," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 615-627, May 2005.
[18] G. S. Hao, M. H. Lim, Y. S. Ong, H. Huang, and G. G. Wang. "Domination landscape in evolutionary algorithms and its applications," *Soft Computing*, vol. 23, no. 11, pp 3563–3570, June 2019.

Zeng-Hui Duan was born in Shenqiu, Henan province, China, 1995. He is an undergraduate student and majors in software engineering, with School of Computer Science & Technology in Jiangsu Normal University, Xuzhou, China, from 2015.

His current research interests include programing and evolutionary computation. He won the funding from the Key Projects of National Innovation and Entrepreneurship Training Program for College Students in 2017.

Xingliu Hu was born in Lingbi, Anhui Province, China 1974. She received a PhD in intelligent control from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2009, a MA in automation from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2004. Her current research includes algorithm design, intelligent control, fiber sensing.

Guo-Sheng Hao was born in Wanquan, Hebei province, China, 1972. He received the B.Sc. in exploration engineering, the M.Sc., and Ph.D. degrees, both in control theory and application, all from China University of Mining and Technology, Xuzhou, China, in 1997, 2005, and 2009, respectively.

In 1999, he joined the School of Computer Science & Technology, Jiangsu Normal University, where he became an associate professor in 2009 and a professor in 2017, he has been the director of the intelligent computing lab in his school. His current research interests include evolutionary computation, big data and software design.

Prof. Hao is the author of book Theory and Application of Interactive Evolutionary Computation (in Chinese, CUMT Press, 2016) and coauthor of books Interactive Evolutionary Theory and Methods Solving Optimization Problem with Implicit Indices (in Chinese, Zhengzhou University Press) and Principle and Application of Interactive Genetic Algorithm (in Chinese, 2007). He is also coauthored more than 40 journal and conference papers.

Fang Luo was born in Ganzhou, Jiangxi province, China, 1998. She is an undergraduate student and majors in software engineering, with School of Computer Science & Technology in Jiangsu Normal University, Xuzhou, China, from 2015. Her current research interests include programing and evolutionary computation.

Xiao-Dan He was born in Bijie, Guizhou province, China, 1995. She is an undergraduate student and majors in computer science & technology, with School of Computer Science & Technology in Jiangsu Normal University, Xuzhou, China, from 2015. Her current research interests include programing and big data.

Yi-Yang He was born in Xiangyang, Hubei province, China, 1997. She is an undergraduate student and majors in Software Engineering, with School of Computer Science & Technology in Jiangsu Normal University, Xuzhou, China, from 2015. Her current research interests include programing and evolutionary computation.