



---

# OPTIMIZED TRUST-AWARE RECOMMENDER SYSTEM USING GENETIC ALGORITHM

W. Yuan, D. Guan

---

**Abstract:** Trust-aware recommender system (TARS) recommends ratings based on user trust. It greatly improves the conventional collaborative filtering by providing reliable recommendations when dealing with the data sparseness problem. One basic research issue of TARS is to improve the recommending efficiency, in which the key point is to find sufficient number of recommenders efficiently for active users. Existing works searched recommenders via a skeleton, which consists of a number of hub nodes. The hub nodes are those who have superior degrees based on the scale-freeness of the trust network. However, existing works did not consider the skeleton maintenance cost and the coverage overlap between nodes of the skeleton. They also failed to suggest the proper size of the skeleton. This paper proposes an optimized TARS model to solve the problems of existing works. By using the genetic algorithm, our model chooses the most suitable nodes for the skeleton of recommender searching. It can achieve the maximum prediction coverage with the minimum skeleton maintenance cost. Simulation results show that compared with existing works, our model can reduce more than 90% of the skeleton maintenance cost with reasonable prediction coverage.

Key words: *trust-aware recommender system, genetic algorithm, searching efficiency, skeleton selection*

Received: May 8, 2016

DOI: 10.14311/NNW.2017.27.004

Revised and accepted: October 11, 2016

## 1. Introduction

The trust-aware recommender system (TARS) [1, 7, 8, 11–16] has recently been proposed to solve the data sparseness -problem of the conventional collaborative filtering (CF) method. The data sparseness problem is one of the main bottlenecks of CF. TARS predicts ratings based on user trust instead of user similarity. Trust is the measure of willingness to believe in a user based on the user's competence and behavior within a specific time [10]. By utilizing the transitivity of user trust,

---

Weiwei Yuan; Donghai Guan – Corresponding author; College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Bldg. of Computer Science, Nanjing Univ. of Aeronautics and Astronautic, Jiangning District, Nanjing City, Jiangsu Province, China; Collaborative Innovation Center of Novel Software Technology and Industrialization, China, E-mail: [yuanweiwei@nuaa.edu.cn](mailto:yuanweiwei@nuaa.edu.cn), [dhguan@nuaa.edu.cn](mailto:dhguan@nuaa.edu.cn)

TARS can provide reliable recommendations when dealing with the data sparseness problem [1, 7, 8, 10–16].

Two main research issues of TARS are improving the prediction accuracy and improving the recommending efficiency [14, 15]. The prediction accuracy of most TARS models is similar to that of the conventional CF, while the recommending efficiency is quite different in various TARS models. This paper mainly focuses on the problem of improving recommending efficiency. The key factor influences the recommending efficiency is how to find sufficient number of recommenders efficiently for active users. Recommenders are searched via trust propagations in the trust network from active users. The recommender searching should be efficient and sufficient. The complexity of the recommender searching should be minimized to reduce the system response time, and recommenders should be found as many as possible for the active users to predict the ratings.

Existing TARS models [7, 14–16] searched recommenders based on the scale-freeness of the trust network. A scale-free network is a network whose degree distribution follows the power law [2, 3]. The most notable characteristic of the scale-free network is that there exist some nodes (also known as the hub nodes) whose degrees greatly exceed the average, dominating the connectivity of the network [2, 3]. Existing TARS models built a skeleton that consists of a number of the hub nodes. They first propagate the user trust from the active user to the skeleton, and then utilizes the superior degrees of the hub nodes to find recommenders efficiently and sufficiently.

However, the recommending efficiency of existing TARS models needs further improvement. Existing TARS models select nodes with the highest degrees in the trust network to construct the skeleton, while they did not mention what is the proper size of the skeleton. The size of the skeleton is closely related to the recommending efficiency of TARS. By increasing the size of the skeleton, it is easier for TARS to find recommenders since more hub nodes with superior connections are involved in the skeleton. However, this also increases the cost to maintain the skeleton. The larger the skeleton size is, the higher the skeleton maintenance cost is. Furthermore, it is not always necessary to involve all the top  $k$  nodes with the highest degrees in the skeleton: there may exist some overlap between the coverage of these nodes, i.e., the nodes in the skeleton have some co-trusted nodes in the trust network. This unnecessary redundancy increases the skeleton maintenance cost.

To solve the problems of existing TARS model, this work proposes an optimized TARS model by using the classical optimization algorithm, the genetic algorithm (GA). We use GA to optimize the skeleton selection strategy of existing TARS model, selecting the proper hub nodes for the skeleton construction. This is achieved by considering both the influence of skeleton maintenance cost to the recommending efficiency and the influence of coverage redundancy of nodes in the skeleton to the recommending efficiency. By achieving the maximum prediction coverage with the minimum skeleton maintenance cost, our proposed TARS model can greatly improve the recommending efficiency. Simulation results show that it is possible to reduce more than 90% of the skeleton maintenance cost while achieving 90% of the original prediction coverage (with three hops of trust propagations from the skeleton).

The following of this paper is organized as follows: Section 2 introduces related works. Section 3 presents our proposed our GA based TARS model in detail. Section 4 gives simulation results using real application data. Finally, the last section concludes this paper and discusses future work.

## 2. Related works

It is a fundamental requirement of TARS to efficiently search as many recommenders for the active users as possible. However, most studies [1, 8, 11–13] did not provide any information on how they find recommenders. A few other works [7, 16] briefly mentioned that they search the entire trust network to find the recommenders. Since it does not miss any node reachable by trust propagations, TARS can achieve high prediction coverage. However, its computational complexity is high:  $o(k^d)$ , where  $k$  is the average degree of the trust network and  $d$  is the maximum allowable trust propagation distance, especially when TARS has large-scale trust networks. It has been verified that the proper maximum allowable trust propagation distance [15] can be set as  $d \approx \ln n / \ln k$ , where  $n$  is the number of nodes in the trust network, and  $k$  is the average degree of each node. In this work, we set all maximum allowable trust propagation distances as  $d \approx \ln n / \ln k$ .

Some TARS models [14, 15] improve the recommending efficiency by setting the recommender searching mechanism based on the scale-freeness of the trust network. It has been verified that these models have reasonable prediction coverage and is computationally much less expensive, which provide a much better recommending efficiency. These models choose the hub nodes of the trust network to construct a skeleton for trust propagation. The skeleton is regarded as one super node: the node trusted by any node of the skeleton is regarded as the node trusted by the skeleton, and if a node trusts any node of the skeleton, it is regarded as trusting the skeleton. The details of recommender searching mechanism of these TARS models are given in Algorithm 1. The skeleton acts as the trust propagation bridge for recommender searching of each active user: the active user is first connected to the skeleton by the shortest path of trust propagation, and the recommenders are searched by the trust propagated from the skeleton.

The skeleton selection mechanism of existing TARS models is given in Algorithm 2. They first set a threshold value of the degree, and then selects all the nodes whose degrees exceed this threshold.

The nodes of the skeleton fulfill:

$$k_S \gg k \quad (1)$$

where  $k$  is the average degree of the trust network, and  $k_S$  is the average degree of the skeleton.

The computational complexity of existing TARS models is

$$O(k^{d_{AS}} + k^{d_{SR}}) = O(k^{\max(d_{AS}, d_{SR})}), \quad (2)$$

where  $d_{AS}$  is the trust propagation distance from the active user to the skeleton, and  $d_{SR}$  is the trust propagation distance from the skeleton to the recommenders.

**Algorithm 1** Recommender searching mechanism of existing TARS models.

---

Input:  $G = (V, E)$ : the trust network of TARS;  $a$ : the active user;  $r$ : the target recommender.

Output: recommenderFlag: the flag describing whether  $r$  is found.

Parameters:  $V_S$ : the skeleton; skeletonFlag: the flag describing whether  $V_S$  is found;  $V_1$  and  $V_{\text{temp}}$ : temporary vertex set.

Recommender Searching Mechanism:

```

skeletonFlag = false;
recommenderFlag = false;
 $V_1 = \{a\}$ ;
while (skeletonFlag = false)
  for  $\exists v_i \in V_1$ 
    for  $j = 1, 2, \dots, |V|$ 
      if  $v_j \in V$  and  $v_j \notin V_1$  and  $e_{ij} \in E$ 
        if  $v_j \in V_S$ 
          skeletonFlag = true;
break;
if (skeletonFlag = true)
  break;
 $V_{\text{temp}} = \{v_j : v_j \in V, v_j \notin V_1, e_{ij} \in E\}$ ;
 $V_1 = V_{\text{temp}}$ ;
 $V_1 = V_S$ ;
while (recommenderFlag = false)
  for  $\exists v_i \in V_1$ 
    for  $j = 1, 2, \dots, |V|$ 
      if  $v_j \in V$  and  $v_j \notin V_1$  and  $e_{ij} \in E$ 
        if  $v_j = r$ 
          recommenderFlag = true;
break;
if (recommenderFlag = true)
  break;
 $V_{\text{temp}} = \{v_j : v_j \in V, v_j \notin V_1, e_{ij} \in E\}$ ;
 $V_1 = V_{\text{temp}}$ ;

```

---

**Algorithm 2** The skeleton selection strategy of existing TARS models.

---

Input:  $G = (V, E)$ : the trust network of TARS.

Output:  $V_S$ : the skeleton.

Parameters: degreeThres: the threshold of the degree for skeleton selection; degree(\*): the degree of the selected node.

Skeleton selection strategy of existing TARS model:

```

 $V_S = \emptyset$ ;
for  $i = 1, 2, \dots, |V|$ 
  if  $\text{degree}(v_i) > \text{degreeThres}$ 
     $V_S = V_S + v_i$ ;

```

---

### 3. Optimized GA based TARS model

#### 3.1 Problem representation

Let  $G = (V, E)$  represents the trust network in TARS, where  $V = \{v_i, i = 1, 2, \dots, n\}$  represents the nodes and  $E = \{e_{ij}, i \in [1, n], j \in [1, n], i \neq j\}$  represents the edges in the trust network:

$$e_{ij} = \begin{cases} 1, & i \text{ trusts } j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Let  $a$  be the target active user that TARS gives the recommendation,  $a \in V$ . Let  $t$  be the target item that TARS predicts  $a$ 's rating on. Let  $R = \{r_i, i = 1, 2, \dots, m\}$  be all the users rated  $R \subseteq V$ . Searching the recommenders for the active user means finding the trust propagation paths  $P_{aR'}$ , where  $a$  is the source of trust propagation and  $R' = \{r_i, i = 1, 2, \dots, m', m' \leq m\}$  refers to all the recommenders that can be found by TARS,  $R' \subseteq R$ .

$$P_{aR'} = \{e_{av_1} e_{v_1 v_2} e_{v_2 v_3} \dots e_{v_{k-1} v_k} e_{v_k r_i}, e_{**} \in E, v_* \in V, r_i \in R' \subseteq R\} \quad (4)$$

To achieve high recommending efficiency, TARS model should maximize  $m'$  while minimizing  $P_{aR'}$ .

Existing TARS models, as introduced in Section 2, search the recommenders via the skeleton, i.e.,  $a$  is first connected to the skeleton by the shortest length of the trust propagation, and the recommenders are searched via trust propagation from the skeleton. Let  $G_S = (V_S, E_S)$  be the skeleton of existing TARS model  $G_S \subset G$ ,  $V_S \subset V$ ,  $E_S \subset E$ ,  $V_S = \{v_i^S, i = 1, 2, \dots, n_S\}$ ,  $E_S = \{e_{ij}^S, i \in [1, n_S], j \in [1, n_S], i \neq j\}$ . For any  $v_i^S \in V_S$ ,  $\text{Degree}(v_i^S) \gg \text{Degree}(v_i)$  and  $|E_S|/|V_S| \gg |E|/|V|$ . In this work, we consider the undirected trust network, i.e.,  $e_{ij} = e_{ji}$ . Where  $i \neq j$ ,  $\text{Degree}(v_i) = \text{num}(e_{i*}) = \text{num}(e_{*i})$ . The value  $\text{num}(e_{i*})$  means the number of edges with  $v_i$  as one node. By using existing TARS models:

$$P_{aR'} = P_{aV_S} + P_{V_S} + P_{V_S R'} \quad (5)$$

where  $P_{aV_S}$  refers to the trust propagation path from  $a$  to any  $v_i^S \in V_S$ ,  $P_{V_S}$  refers to the trust propagation path between nodes of the skeleton, and  $P_{V_S R'}$  refers to the trust propagation path from any  $v_i^S \in V_S$  to  $R'$ . Since the skeleton fulfills  $|E_S|/|V_S| \gg |E|/|V|$ , (5) can be approximately written as

$$P_{aR'} \approx P_{aV_S} + P_{V_S R'} \quad (6)$$

Let  $\text{Cost}(V_S)$  be the cost to maintain  $V_S$ . It is reasonable to assume that  $\text{Cost}(V_S)$  is positively related to  $|V_S|$ , i.e., it requires more system effort to maintain the skeleton at a larger scale. In this work, we set:

$$\text{Cost}(V_S) = C \cdot |V_S| \quad (7)$$

where  $C$  is a constant.

The coverage of the skeleton with  $l$  hops of trust propagations is represented as

$$Coverage_l(V_S) = Coverage_{l-1}(V_S) + num(e_{v_i^{l-1}*}) \quad (8)$$

where  $num(e_{v_i^{l-1}*})$  refers to the number of edges with  $v_i^{l-1}$  as one node, and  $v_i^{l-1}$  refers to any node that is  $l - 1$  hops away from the skeleton.

### 3.2 Our proposed model

Existing TARS models only focus on  $Coverage_l(V_S)$ , not considering  $Cost(V_S)$ . Though increasing  $|V_S|$  increases  $Coverage_l(V_S)$ ,  $Cost(V_S)$  also increases. Furthermore, there will probably be some overlap between the coverage of the nodes in the skeleton. This unnecessary redundancy increases  $|V_S|$ , which further increases  $Cost(V_S)$ .

Our proposed TARS model improves existing TARS models by optimizing the skeleton selection strategy to maximize  $Coverage_l(V_S)$  while minimizing  $Cost(V_S)$ . The basic searching mechanism of our proposed TARS model is similar to that of existing TARS models, as shown in Algorithm 1. The difference is that our new proposed model improves the skeleton selection strategy, which is the key part of the searching mechanism of existing TARS model (the skeleton selection strategy of existing TARS models is given in Algorithm 2). This is achieved by utilizing the genetic algorithm (GA), which is a classical optimization algorithm. GA is inspired by natural evolution, where populations of individuals compete to survive and reproduce themselves. A population of solutions is maintained and a reproductive process allows parent solutions to be selected from the population. Offspring solutions are produced and they exhibit some of the characteristics of each parent. The fitness of each solution can be related to the objective function values. Analogous to biological processes, offspring with relatively good fitness levels are more likely to survive and reproduce, with the expectation that fitness levels throughout the population will improve as it evolves. The flow diagram of genetic algorithm is shown in Fig. 1. The starting point of GA is the initialization of populations, typically in the form of a string or chromosome. The fitness of each individual in the population is then calculated, in which the fitness function interprets the chromosome and a higher fitness value of the chromosome means it is better than others. Using some satisfying constraints, the survivors are selected based on their fitness. The selection intends to improve the average quality of the population by giving the high-quality chromosomes a better chance to get copied into the next generation. The selected individuals act as the parents to reproduce the offspring, using the crossover and mutation operators. This process continues until the constraints are satisfied.

Based on GA, our new proposed TARS model has an optimized skeleton selection strategy, as listed in Algorithm 3. In our model, the chromosomes of the population are represented as follows: let  $P$  be the set of all populations, for any  $p \in P$ ,  $len(p) = |V_S|$ , where  $len(p)$  represents the length of  $p$ , and  $V_S$  is the skeleton candidate, which fulfills  $|E_S|/|V_S| \gg |E|/|V|$  for each chromosome  $p = g_1, g_2, g_3, \dots, g_{n_S}$ , in which  $g_i$  is the gene of the chromosome:

$$g_i = \begin{cases} 1, & \text{if } v_i^S \text{ is selected for the final skeleton} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

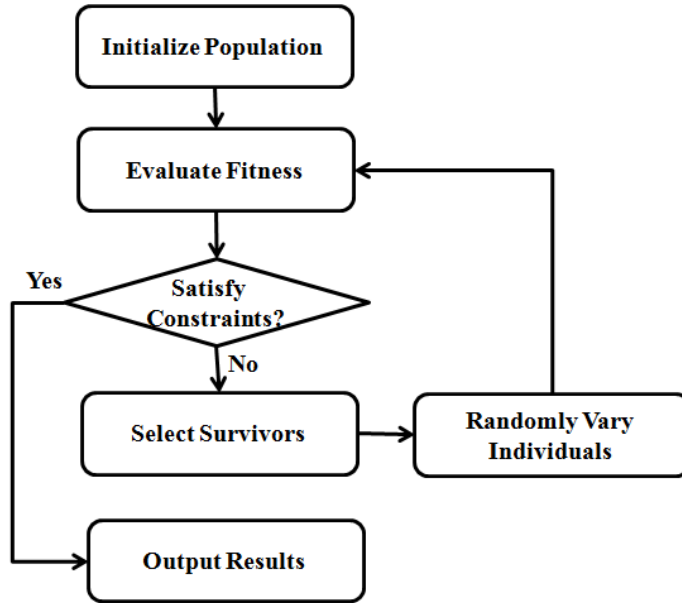


Fig. 1 The flow diagram of the genetic algorithm.

We assign a unique number to each node in the skeleton candidate, which corresponds to one gene of the chromosome. If the node in the skeleton candidate is selected, the corresponding gene is set to be 1; otherwise, the corresponding gene is set to be 0. An example is shown in Fig. 2 to describe how we set the chromosome:  $V_S = \{v_i, i = 1, 2, \dots, 8\}$ , so  $len(p) = |V_S| = 8$ , if  $v_1, v_4, v_5$  and  $v_8$  are selected as the skeleton nodes, the chromosome is represented as 10011001.

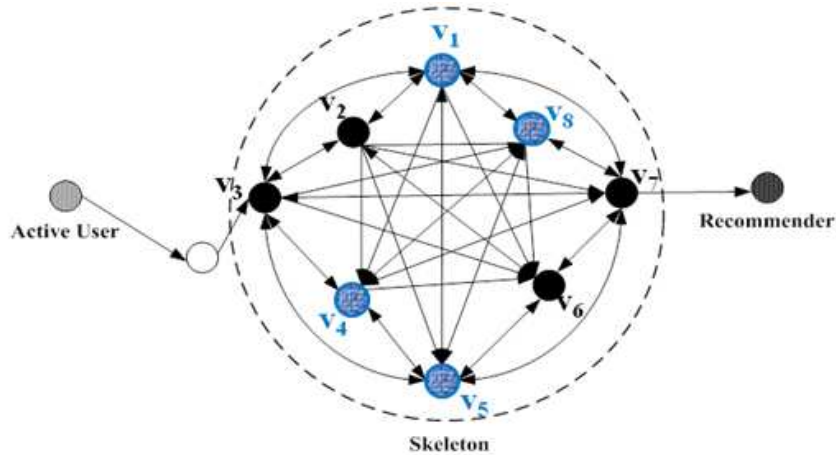


Fig. 2 An example of the chromosome 10011001.

---

**Algorithm 3** The skeleton selection strategy of our proposed TARS model.

---

Input:  $G = (V, E)$ : the trust network of TARS;  $V_S$ : the skeleton candidate

Output:  $V'_S$ : the optimized skeleton

Parameters:  $k$ : the size of the population;  $P$ : population;  $p_*$ : individual in the population;  $o_*$ : generated offspring;  $fit(*)$ : fitness of \*;  $Coverage_1(*)$ : coverage of TARS with one hop of trust propagation from the skeleton;  $Thres_{Coverage}$ : threshold of coverage;  $|V_S|$ : size of  $V_S$ ;  $p_i(g_j)$ : the  $g_j$ -th gene of  $p_i$ 's chromosome; generationStep: GA generation steps; maxGenerationStep: maximum GA generation steps.

Optimized Skeleton Selection Strategy:

for  $i = 1, 2, \dots, k$

    randomly generate  $p_i$  from  $V_S$  ;

while generationStep  $\leq$  maxGenerationStep

    randomly select 2 individual  $p_i, p_j \in P$ ;

$p_i, p_j$  crossover, generate two offspring  $o_1$  and  $o_2$ ;

$o_1, o_2$  mutation;

    delete  $p_i$  and  $p_j$  from  $P$ ;

    calculate  $fit(p_i), fit(p_j), fit(o_1)$  and  $fit(o_2)$ ;

    select two individual with the highest fitness;

    insert into  $P$ ;

for  $i = 1, 2, \dots, k$

    calculate  $fit(p_i)$ ;

select  $p_i$  with the highest fitness;

if  $Coverage_1(p_i) \leq Thres_{Coverage}$

    Break;

generationStep++;

for  $i = 1, 2, \dots, k$

    calculate  $fit(p_i)$ ;

select  $p_i$  with the highest fitness;

for  $j = 1, 2, \dots, |V_S|$

    if  $p_i(g_j) = 1$

        Select  $g_j$  in  $V_S$  to construct  $V'_S$ ;

---

The fitness function of our proposed model is set as

$$fit(V'_S) = \frac{Coverage_l(V'_S)/Coverage_l(V_S)}{Cost(V'_S)/Cost(V_S)} \quad (10)$$

where  $V'_S$  is optimized skeleton,  $V_S$  is the original skeleton candidate, and  $l$  is the number of hops that are propagated from the skeleton.

For each generation, we randomly select two individuals from the population to act as the parents. This randomness helps to overcome the local optimum in each generation. Physically, the random initialization chooses genes (nodes) from the nodes with the highest degree in the trust network in a random manner during the encoding process. The generated offspring compete with their parents, and the two with the highest fitness survive to update the population. We set two constraints for our proposed method: the generation is terminated if its coverage (coverage

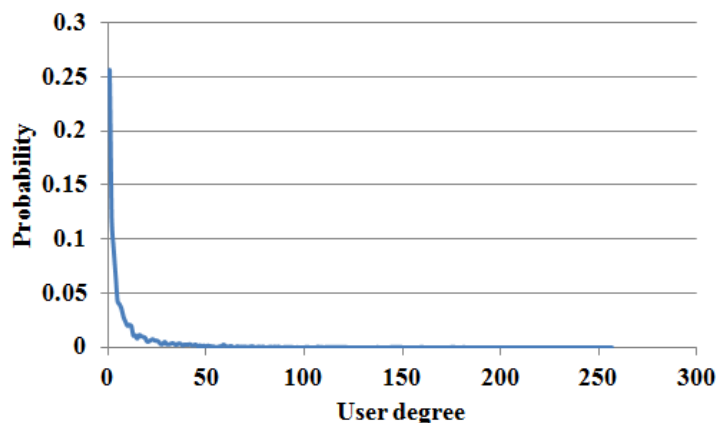


with one hop of trust propagation from the skeleton) is too low or if it reaches the maximum number of generation steps.

## 4. Experimental results

### 4.1 Experiments setup

The experiments were held on the publicly released TARS dataset Epinions which is available at (<http://www.trustlet.org/wiki/Datasets>). Epinions consists of 40163 users' 664824 ratings on 139738 items. In this work, we consider the undirected trust network, i.e., if user A trusts users B, user B also trusts user A. The user trust is extracted using the method given in [14-15], where  $Thres_s = 0.75$  and  $Thres_I = 5$ . We get 37616 undirected trust statements between 6182 users. Setting  $d \approx \ln n / \ln k$  [15], we further let the maximum trust propagation distance in this experiment be 3 hops. The degree distribution of our selected trust network is given in Fig. 3, and clearly shows that the trust network is a scale-free network, which means that the hub nodes exist and are capable of constructing the skeletons for our proposed TARS model and existing TARS models.



**Fig. 3** The degree distribution of our experimental data.

To initiate our new proposed TARS model, we first need to set a proper size for skeleton candidate  $V_S$ . If  $|V_S|$  is too large, the length of the chromosome in our new proposed TARS model is too long. This increases the computational complexity. If  $|V_S|$  is too small, the coverage of our new proposed TARS model is limited. Fig. 4 shows the simulation results of the relationship between the recommender coverage and the size of the skeleton candidate, in which the recommender coverage refers to the ratio of recommenders that could be covered by TARS. This work mainly focuses on the recommender coverage, and the prediction coverage is used alternately with the recommender coverage in this paper. The simulation results show that the recommender coverage increases with (1) increasing skeleton size and (2) increasing trust propagation distance from the skeleton to the recommender. Based on the analysis in Fig. 4, we set the 100 as  $|V_S|$ , i.e., choose 100 users (1.62%

of the total users) with the highest degrees in the trust network as the skeleton candidate. For our selected 100 skeleton candidate nodes, we randomly select one node and examine its coverage with other nodes in the skeleton candidate. The simulation result is shown in Fig. 5. It shows that for some nodes, the ratio of their recommender coverage overlap is high, around 20%. So it is necessary to select proper nodes for the skeleton to reduce the redundancy.

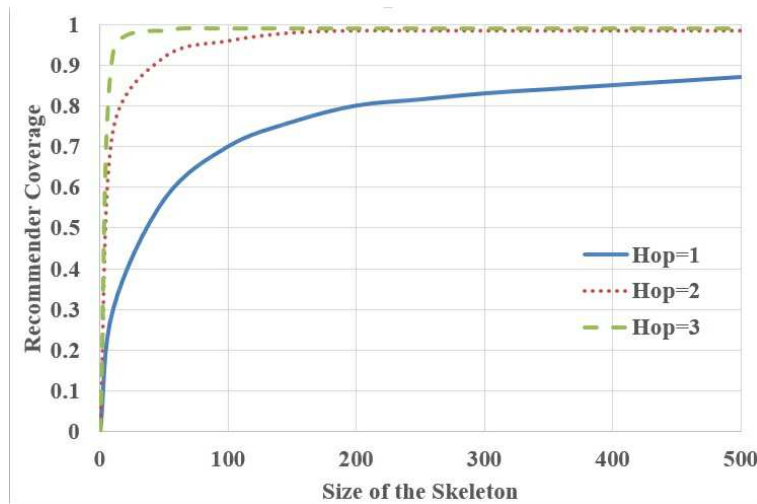


Fig. 4 The recommender coverage with difference sizes of the skeleton.

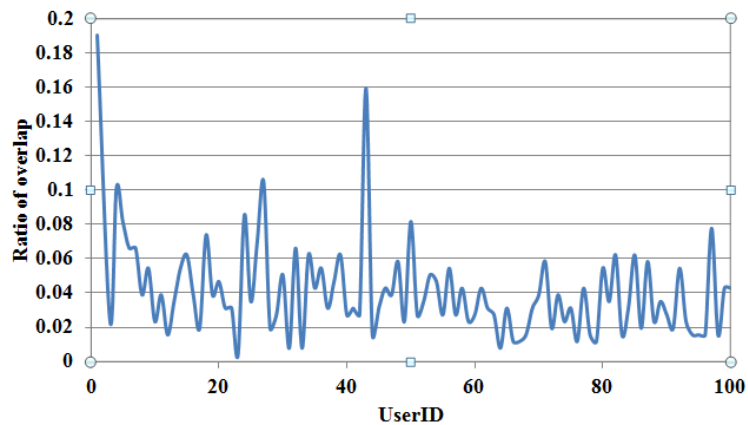
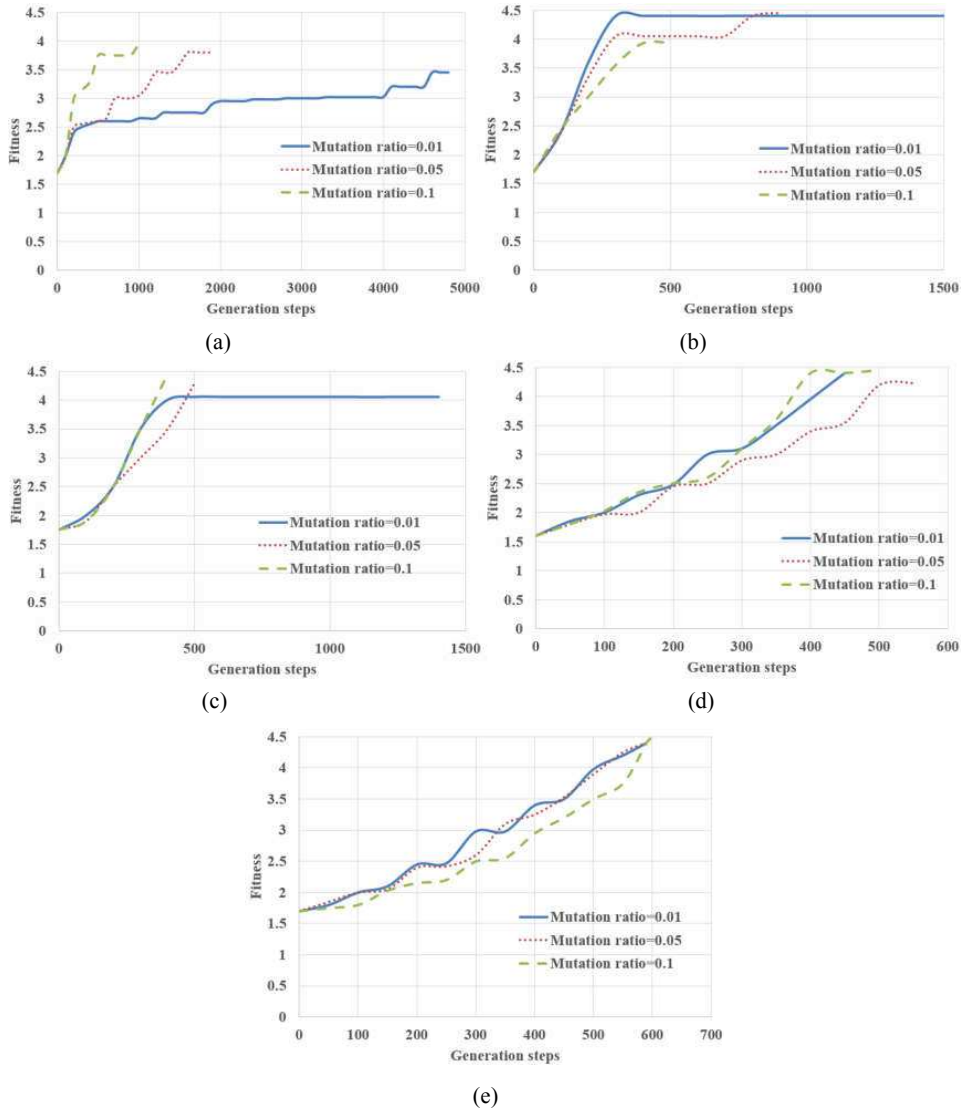


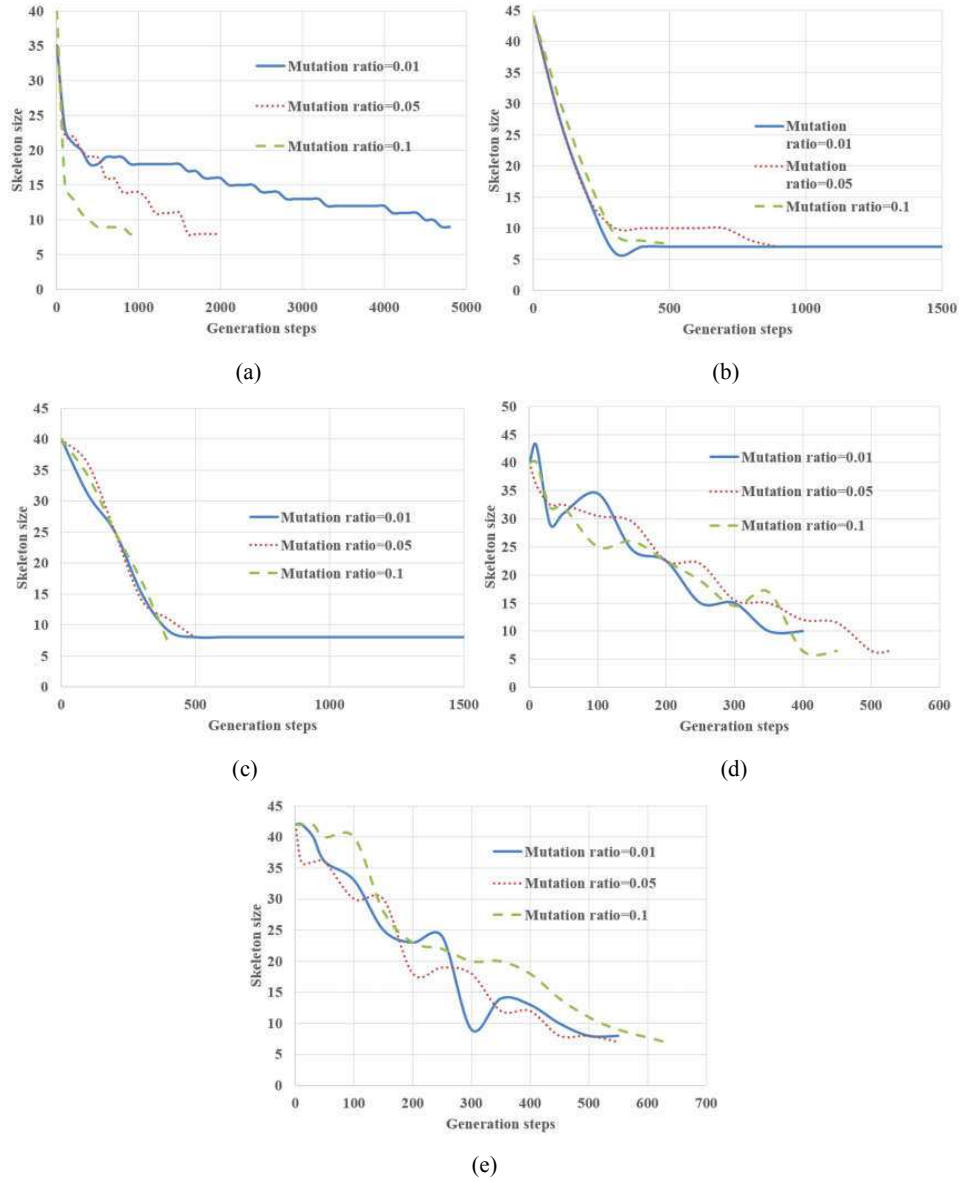
Fig. 5 The recommender coverage overlap between a randomly selected node and all other nodes in the skeleton candidates.

## 4.2 Experimental results and analysis

Fig. 6 gives the simulation results of the fitness values by using different population sizes. The fitness value is calculated by (10). Fig. 7 gives the simulation results of the final skeleton size chosen by our new proposed TARS model. The population sizes in both simulations are set to be 10, 20, 30, 40 and 50. By increasing the population size, the proposed skeleton selection strategy can select solutions over a wider range. We use uniform crossover, which evaluates each gene in the parent



**Fig. 6** The fitness value given (a) population = 10, (b) population = 20, (c) population = 30, (d) population = 40 and (e) population = 50.



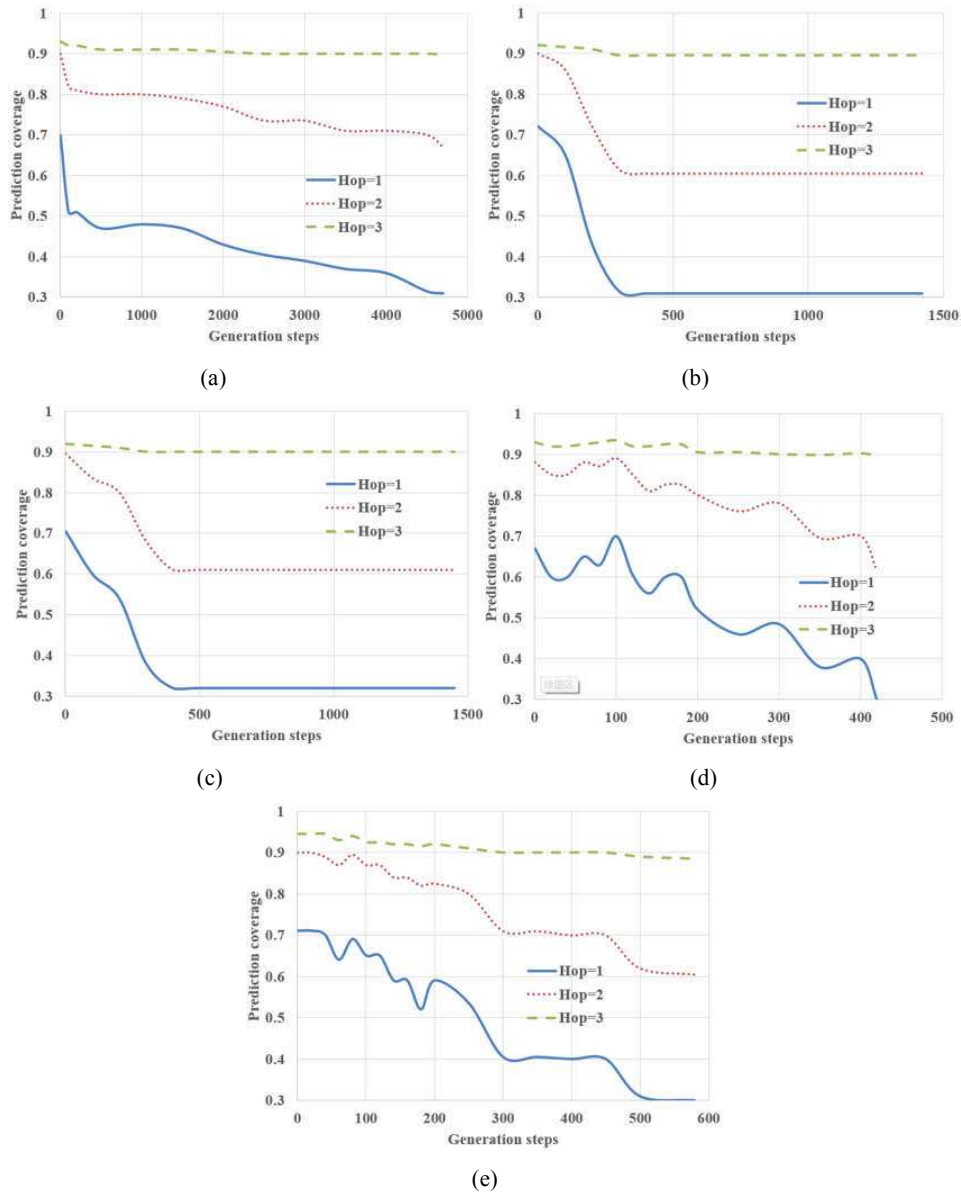
**Fig. 7** The skeleton size given (a) population = 10, (b) population = 20, (c) population = 30, (d) population = 40 and (e) population = 50.

chromosomes for exchange with a probability of 0.5, as the crossover operator in the proposed our new proposed TARS model. The mutation ratios for the mutation operator are set to be 0.01, 0.05 and 0.1. We set the *maxGenerationStep* in our new proposed TARS model as 50,000, and *ThresCoverage* equals 0.3. Fig. 6 and Fig. 7 show that: (1) before our new proposed TARS model converges, the

fitness value increases as the number of generation steps increases, while the selected skeleton size decreases as the number of generation steps increases. This means the skeleton selected by our new proposed TARS model can achieve higher recommender coverage with lower skeleton maintenance cost as the number of generation steps increases, and our new proposed TARS model continuously reduces the skeleton size to minimize the skeleton maintenance cost. (2) Our new proposed TARS model converges with a small skeleton size.

For all the simulations, the final selected skeleton size is between 5 nodes and 10 nodes, which is only 5% to 10% of the original skeleton candidate size. (1) When the population size equals 10, 20, 30 and 40, it takes our new proposed TARS model more generation steps to converge as the population size increases. When the population size equals 50, it takes more generation steps to converge than when the population size equals 40. (2) When the population size equals 10, 20 and 30, our new proposed TARS model converges with fewer generation steps and a higher mutation ratio. When the population size equals 40 or 50, our new proposed TARS model converges with similar generation steps for different mutation ratios. (3) With the decreasing mutation ratio, the convergence procedure for our new proposed TARS model is more sensitive to the population size. Comparing the convergence procedures for mutation ratios of 0.01 and 0.1 shows that our new proposed TARS model converges faster with a larger population size (when the population size is no larger than 40) for the former case.

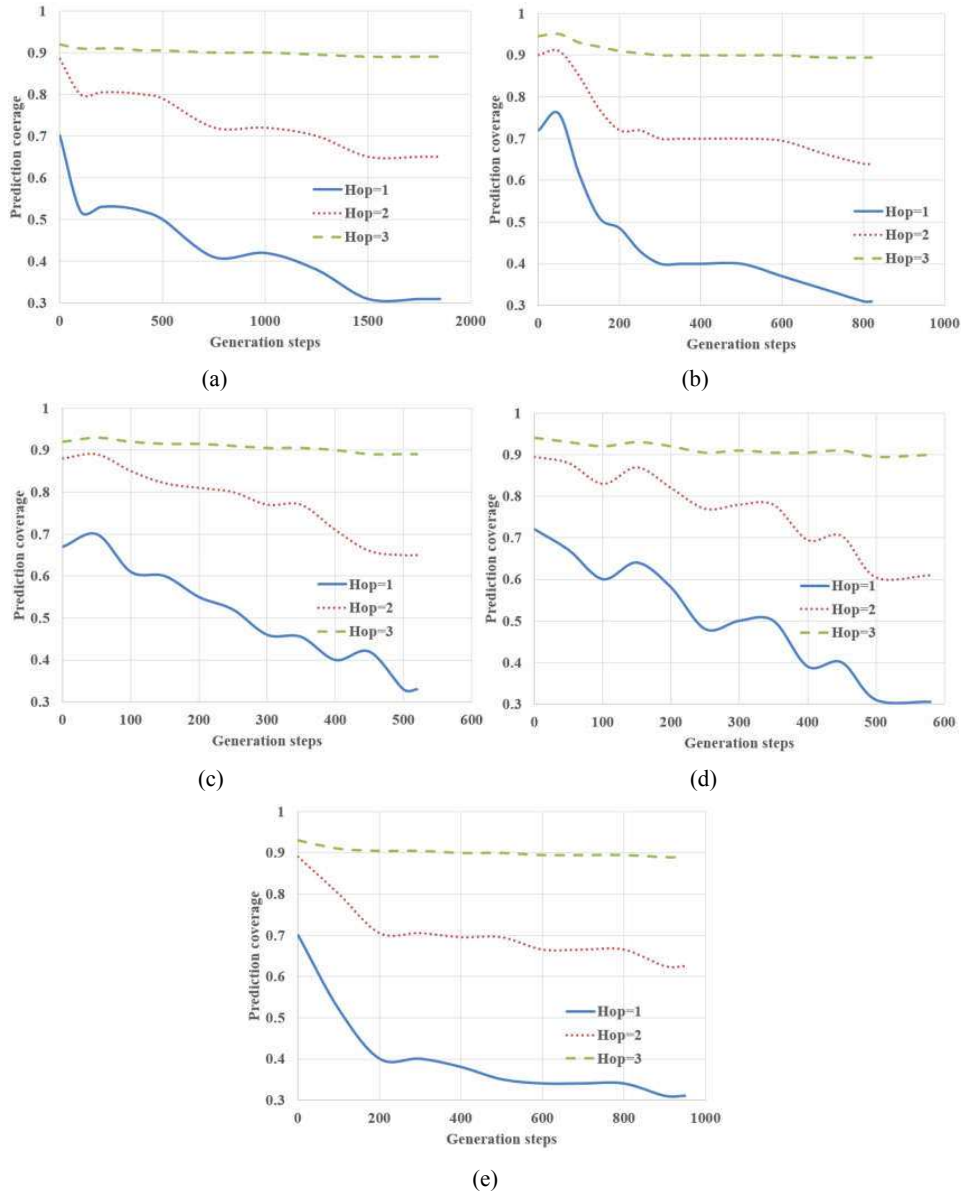
Fig. 8, Fig. 9 and Fig. 10 show the relationship between the prediction coverage and the number of generation steps using our new proposed TARS model when the population size is equal to 10, 20, 30, 40 and 50. Specifically, Fig. 8, Fig. 9 and Fig. 10 give the simulation results for mutation ratios of 0.01, 0.05 and 0.1. For the simulation results shown in Fig. 8, Fig. 9 and Fig. 10, we verify the prediction coverage with one hop, two hops and three hops for trust propagation from the skeleton. Fig. 8, Fig. 9 and Fig. 10 show that: (1) the recommender coverage decreases as the number of generation steps increases in our new proposed TARS model. This is because the selected skeleton size is decreasing, as shown in Fig. 7. However, since the fitness value of our new proposed TARS model is increasing, the overall performance of our new proposed TARS model is still increasing, i.e., though the prediction coverage is decreasing, the skeleton maintenance cost decreases more. (2) When our new proposed TARS model converges, the recommender coverage is more than 60% with two hops of trust propagation from the skeleton, and the recommender coverage is around 90% with three hops of trust propagation from the skeleton. This performance is reasonable since the skeleton maintenance cost is only around 5% to 10% of the original cost when our new proposed TARS model converges. (3) When the mutation ratio equals 0.01, our new proposed TARS model converges fastest when the population size equals 40. When the mutation ratio equals 0.05 or 0.1, our new proposed TARS model converges fastest when the population size equals 30. For example, when the mutation ratio equals 0.05, if the population size is smaller than 30, our new proposed TARS model needs more steps to converge if the population size decreases; if the population size is larger than 30, our new proposed TARS model also needs more steps to converge if the population size increases. (4) When the population size equals 10, 20 or 30, our new proposed TARS model converges faster if the mutation ratio increases. When



**Fig. 8** The prediction coverage given mutation ratio = 0.01 and (a) population = 10, (b) population = 20, (c) population = 30, (d) population = 40 and (e) population = 50.

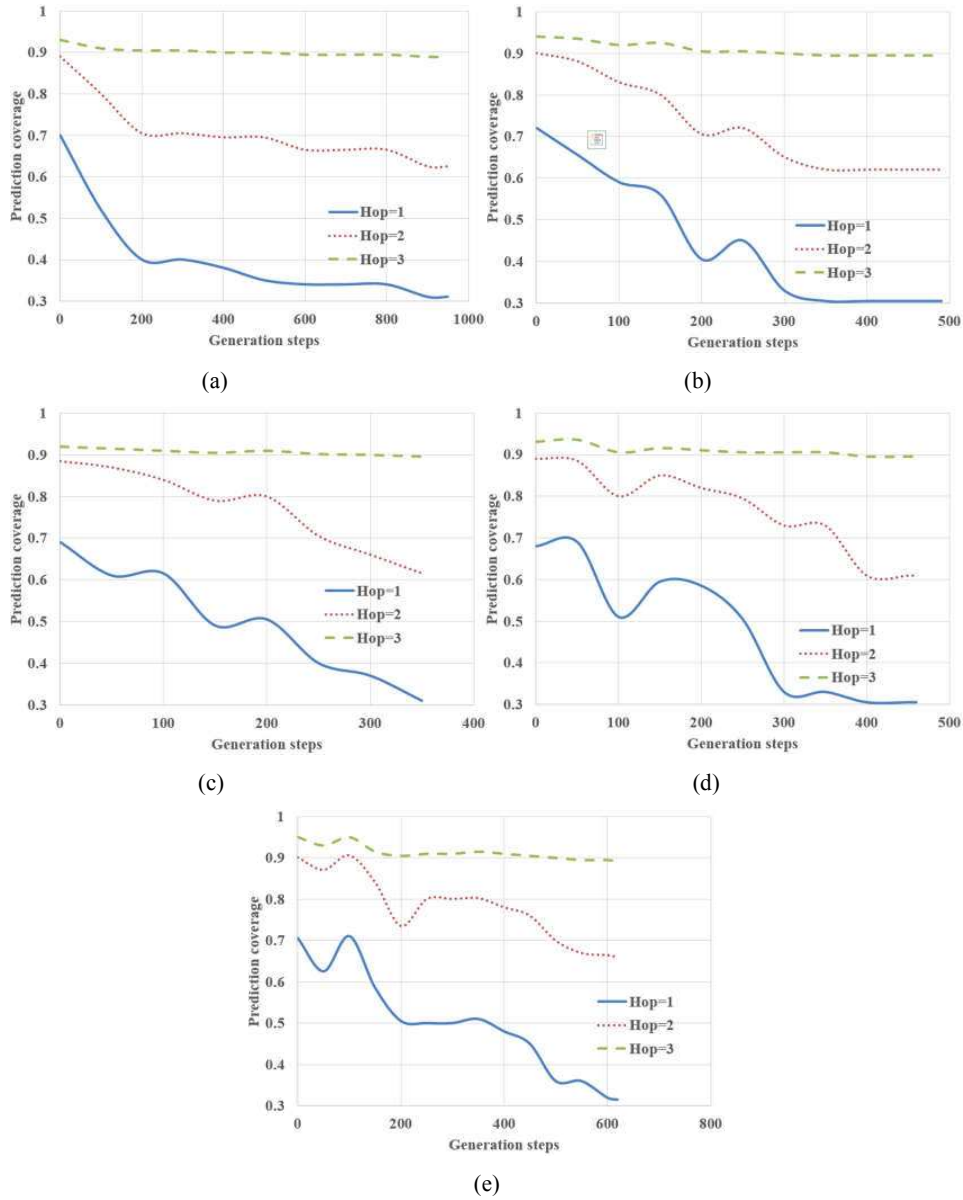
the population size equals 40 or 50, there is no significant difference between the coverage generation steps due to changing the mutation ratio.

By considering the overall performance of the recommender coverage and the skeleton maintenance cost, it is proper to set the population size to 30 and the



**Fig. 9** The prediction coverage given mutation ratio = 0.05 and (a) population = 10, (b) population = 20, (c) population = 30, (d) population = 40 and (e) population=50.

mutation ratio equals 0.05 or 0.1 in our experiments. This will result in the fastest convergence of our new proposed TARS model. Compared with existing TARS model, our new proposed model can greatly reduce the skeleton maintenance cost while achieving reasonable recommender coverage. With two hops of trust propa-



**Fig. 10** The prediction coverage given mutation ratio = 0.1 and (a) population = 10, (b) population = 20, (c) population = 30, (d) population = 40 and (e) population = 50.

gation from the skeleton, the skeleton maintenance cost can be reduced more than 90% while the recommender coverage is more than 60% by our proposed model. With three hops of trust propagation from the skeleton, our new proposed TARS model can even reach around 90% of the original recommender coverage. This can greatly improve the recommending efficiency of existing TARS models.



## 5. Conclusions and future work

This paper proposes a TARS model which greatly improves the recommending efficiency of existing TARS models. The recommender searching mechanism of our new proposed TARS model utilizes an optimized skeleton selection strategy. Considering the skeleton maintenance cost and the prediction coverage, our new proposed TARS models uses the classical optimizing algorithm GA to choose the most optimized skeleton. The chosen skeleton can achieve the maximum prediction coverage with the minimum skeleton maintenance cost for TARS. Simulations were held on the real application data to verify the effectiveness of our proposed TARS model. It shows that compared with existing TARS model, our new proposed TARS model can reduce the skeleton maintenance costs by more than 90% with reasonable prediction coverage. This greatly improves the recommending efficiency of existing TARS models. We will focus on further research details of TARS in our future work. We will improve the rating prediction mechanism of TARS to achieve better prediction accuracy. In the present work, the trust propagation distance is regarded as the main attribute influencing the prediction accuracy, but we plan to incorporate other possible attributes, such as the betweenness and the closeness of the nodes. Secondly, most of the works mainly focus on using the trust propagation in TARS, while the research on using distrust is still in its beginning stages. It has been shown that though the number of distrust statements is very limited in real applications, they give very valuable information. We will try to use TARS in more related applications to achieve better performances [13–16].

### Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61672284). It was also supported by China Postdoctoral Science Foundation (No. 2016M591841), Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (No. CAAC-ITRB-201501, CAAC-ITRB-201602) and Civil Aviation Administration of China (No. AS-SA 2015/21).

### References

- [1] ABDERRAHIM N., BENSLIMANE S.M. STRESS: A social trust-aware system for recommending web services. *International Journal of Information Systems in the Service Sector*, 2015, 7(3), pp. 40–58, doi: [10.4018/IJISSS.2015070103](https://doi.org/10.4018/IJISSS.2015070103).
- [2] BARABÁSI A.L., BONABEAU E. XIA. *Scientific American*, 2003, 288(5), pp. 60–69, doi: [10.1038/scientificamerican0503-60](https://doi.org/10.1038/scientificamerican0503-60).
- [3] BARABÁSI A.L., Scale-free networks: a decade and beyond, *Science*, 2009, 325, pp. 412–413, doi: [10.1126/science.1173299](https://doi.org/10.1126/science.1173299).
- [4] FU Z., SUN X., LIU Q., ZHOU L., SHU J. Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, *IEICE Transactions on Communications*, 2015, E98-B(1), pp. 190–200, doi: [10.1587/transcom.E98.B.190](https://doi.org/10.1587/transcom.E98.B.190).
- [5] GU B., SHENG V.S., WANG Z., HO D., OSMAN S., LI S. Incremental learning for v-Support Vector Regression, *Neural Networks*, 2015, 67, pp. 140–150, doi: [10.1016/j.neunet.2015.03.013](https://doi.org/10.1016/j.neunet.2015.03.013).

- [6] GUO P., WANG J., LI B., LEE S., A variable threshold-value authentication architecture for wireless mesh networks, *Journal of Internet Technology*, 2014, 15(6), pp. 929–936, doi: [10.6138/JIT.2014.15.6.05](https://doi.org/10.6138/JIT.2014.15.6.05).
- [7] GUPTA S., NAGPAL S. An empirical analysis of implicit trust metrics in recommender systems. In: *2015 International Conference on Proceedings of Advances in Computing, Communications and Informatics (ICACCI 2015)*, IEEE, 2015, pp. 636–639.
- [8] LI G., LIAO L., SONG D. ZHANG Z. WANG J. A novel approach to trust-aware service recommendation. In: *Proceedings of the 4th International Conference on Computer Engineering and Networks*, Springer International Publishing, 2015, pp. 787–795.
- [9] MA T., ZHOU J., TANG M., TIAN Y., AL-DHELAAN A., AL-RODHAAN M., LEE S., Social network and tag sources based augmenting collaborative recommender system, *IEICE transactions on Information and Systems*, 2015, (E98-D)4, pp. 902–910, doi: [10.1587/transinf.2014EDP7283](https://doi.org/10.1587/transinf.2014EDP7283).
- [10] MASSA P., AVESANI P. Trust metrics in recommender systems. In: *Proceedings of Computing With Social Trust*, 2015, pp. 259–285.
- [11] MORADI P., AHMADIAN S. A reliability-based recommendation method to improve trust-aware recommender systems. *Expert Systems with Applications*, 2015, 42(21), pp. 7386–7398, doi: [10.1016/j.eswa.2015.05.027](https://doi.org/10.1016/j.eswa.2015.05.027).
- [12] TAHTA U.E., SEN S., CAN A.B. GenTrust: A genetic trust management model for peer-to-peer systems. *Applied Soft Computing*, 2015, 34, pp. 693–704, doi: [10.1016/j.asoc.2015.04.053](https://doi.org/10.1016/j.asoc.2015.04.053).
- [13] WANG Y., LI L., LIU G. Social context-aware trust inference for trust enhancement in social network based recommendations on service providers. *World Wide Web*, 2015, 18(1), pp. 159–184, doi: [10.1007/s11280-013-0241-5](https://doi.org/10.1007/s11280-013-0241-5).
- [14] YUAN W.W., GUAN D.H., KIM T. OS\_Searching: optimized skeleton based recommender searching mechanism for the trust-aware recommender system. In: *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, ACM, 2016, pp. 98–105.
- [15] YUAN W.W., GUAN D.H., LEE S.Y. Skeleton searching strategy for recommender searching mechanism of trust-aware recommender systems. *The Computer Journal*, 2015, 58(9), pp. 1876–1883, doi: [10.1093/comjnl/bxu011](https://doi.org/10.1093/comjnl/bxu011).
- [16] ZOU H., GONG Z., ZHANG N., ZHAO W., GUO J. TrustRank: a cold-start tolerant recommender system. *Enterprise Information Systems*, 2015, 9(2), pp. 117–138, doi: [10.1080/17517575.2013.804587](https://doi.org/10.1080/17517575.2013.804587).