Liu *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:273
DOI 10.1186/s13638-016-0766-2

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Optimal virtual network embedding based on artificial bee colony

Xu Liu, Zhongbao Zhang*, Ximing Li and Sen Su

Abstract

As one of the key challenges in network virtualization, the problem of virtual network embedding has attracted significant attention from researchers. In this problem, it needs to embed virtual networks with both node and link demands into a shared physical network. The main goal of this problem is to embed more virtual networks to gain more revenue. However, the prior approaches still suffer from low performance and await to be further optimized in terms of this goal. In this paper, we design an artificial bee colony-based virtual network embedding algorithm, called VNE-ABC, to solve this problem. The core idea of this algorithm is to leverage the iterations and intelligence of artificial bee colony to achieve a more optimal solution for virtual network embedding. Through simulations, we show that our proposed algorithm gains about 35.4% more revenue than the existing algorithm.

Keywords: Network virtualization, Virtual network embedding, Artificial bee colony, Optimization

1 Introduction

Recently, network virtualization has been proposed as the key technology to reshape the future Internet. It allows multiple heterogeneous virtual networks (VNs), requested by service providers (SPs), to coexist on the same shared physical network (PN) managed by the infrastructure providers (InP). Each VN can run personalized protocol and designate its own topology with different node and link constraints in the PN. This paper concerns the problem of VN embedding, which is to embed or map the VNs to the PN while satisfying its node and link constraints.

The VN embedding problem has received significant attention [2, 3, 10, 15, 17, 21, 26, 28, 32]. In these prior studies, the authors either applied relaxation and rounding technique [7] or designed heuristic methods [3, 26] to perform the virtual network embedding. In our previous studies [2, 28], we proposed to employ one of the meta-heuristics, i.e., the particle swarm optimization, to optimize the VN embedding problem. However, due to the well-known reason that the particle swarm optimization is easy to trap into local optimum, the performance of this algorithm is still awaiting to be optimized. In this paper, we propose to leverage another meta-heuristic, i.e., artificial bee colony (ABC), to further optimize the VN embedding problem.

In particular, the main idea of ABC algorithm is to simulate the process of a group of bees gathering honey. They fully exploit individual intelligence to improve the efficiency and effect of the overall work greatly. Specifically, in ABC algorithm, three kinds of bees are involved, i.e., employed bee, onlooker bee, and scout bee. The employed bee and onlooker bee are used to update solutions to better ones in each iteration and the scout bee is applied to avoid local optimal solutions.

However, in the process of applying the artificial bee colony to the VN embedding problem, there are several challenges for studying such problem.

- Basic ABC only deals with continuous optimization problem. However, the VN embedding problem is a discrete optimization problem, and thus, it cannot be used to our problem directly.
- Naturally, the randomness of ABC may result in slow convergence in our context.
- The roulette method in basic ABC algorithm leads to low performance for solving our problem.

Toward these ends, we conquer the above challenges as follows:

*Correspondence: zhongbaozb@bupt.edu.cn
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, 10 Xitucheng Road, Haidian District, 100876 Beijing, China

- We redefine the parameters and operations of the parameters (such as position and updating operations etc.) according to our problem.
- We propose a preferential exchange strategy to update the food resource to achieve better convergence.
- Furthermore, we propose a pheromone and sensitivity model as an alternative to the roulette method.

Through extensive simulations, we show that our algorithm outperforms the state-of-the-art algorithm by generating more 35.4% long-term average revenue. The key contributions of this paper are twofold: (i) we propose to optimize the virtual network embedding with artificial bee colony technique called VNE-ABC and (ii) we conduct side-by-side comparisons between our algorithm and the state-of-the-art algorithm to demonstrate the resource efficiency of VNE-ABC for the PN.

We organize the remainder of the paper as follows. Section 2 presents the problem definition. Section 3 presents our proposed solution and section 4 evaluates this algorithm. Section 5 reviews the related work and section 6 concludes this paper.

2 Modeling and formulation

In this section, we describe the network model and present the definition of VN embedding and performance metrics.

2.1 Network modeling

A physical network (PN) can be represented by a weighted graph $G_\mathrm{p} = (N_\mathrm{p}, L_\mathrm{p}, A_\mathrm{p}^\mathrm{n}, A_\mathrm{p}^\mathrm{l})$, where N_p represents the set of physical nodes, L_p represents the set of physical links, and A_p^n and A_p^l are the attributes of the physical nodes and links, respectively.

Likewise, a virtual network (VN) can also be represented as a weighted graph $G_\mathrm{v} = (N_\mathrm{v}, L_\mathrm{v}, R_\mathrm{v}^\mathrm{n}, R_\mathrm{v}^\mathrm{l})$, where N_v represents the set of virtual nodes, L_v represents the set of virtual links, and R_v^n and R_v^l are the requirements of the nodes and links, respectively.

Typically, attributes of a node include processing capacity (i.e., CPU), storage, and location of the node. Attributes of a link include the bandwidth and delay. Similar to most of previous studies [2, 3, 7, 26, 30], in this paper, we consider the CPU attribute for nodes and the bandwidth attribute for links as our main focus.

A VN request can be defined by $VNR(G_\mathrm{v}, T_\mathrm{a}, T_\mathrm{d})$, where T_a is the arrival time and T_d is the duration time of the VN request staying in the PN. As shown in Fig. 1a, it presents a VN request with both CPU and bandwidth requirements, and Fig. 1b presents a PN with both CPU and bandwidth capacities. Note that the numbers in rectangles represent the CPU attribute and the numbers over the links represent the bandwidth attribute, respectively.

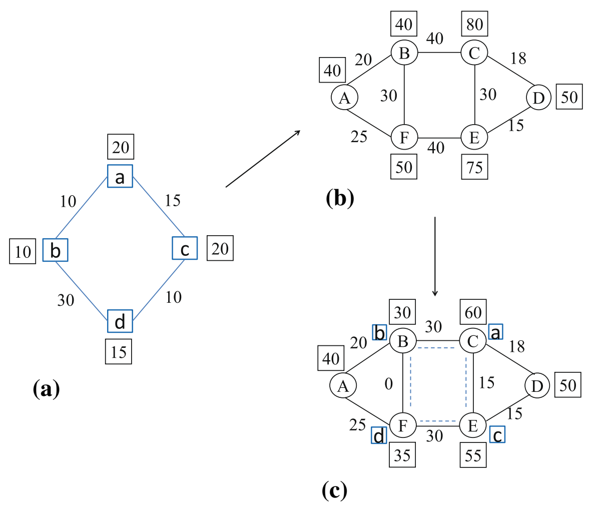

Fig. 1 Example of VN embedding. **a** VN request. **b** Physical network. **c** Physical network

2.2 Problem definition

Now, we formally present the definition of VN embedding as follows. The VN embedding problem can be defined by a mapping $M : G_\mathrm{v}(N_\mathrm{v}, L_\mathrm{v}) \rightarrow G_{\mathrm{p}(N'_\mathrm{p}, P'_\mathrm{p})}$ from G_v to a subset of G_p, where $N'_\mathrm{p} \subset N_\mathrm{p}$ and $P'_\mathrm{p} \subset P_\mathrm{p}$. Note that, P_p denotes all loop-free paths of the physical network. In particular, the VN embedding can be decomposed into two steps: node mapping and link mapping. Node mapping places the virtual nodes into different N_p while satisfying their node constraints; link mapping places the virtual links into the P_p while satisfying their link constraints. Fig. 1c shows a feasible VN embedding solution for this VN request. Specifically, the node mapping solution is $\{a \rightarrow C, b \rightarrow B, c \rightarrow E, d \rightarrow F\}$ and the link mapping solution is $\{(a, b) \rightarrow (C, B), (a, c) \rightarrow (C, E), (c, d) \rightarrow (E, F), (d, b) \rightarrow (F, B)\}$.

2.3 Performance metrics

We define the long-term average revenue as follows:

$$\lim_{T\to\infty} \frac{\sum_{i=1}^{N} R^i(G_\mathrm{v})}{T}, \tag{1}$$

where $R^i(G_\mathrm{v}) = (\sum_{n_\mathrm{v}\in N_\mathrm{v}} CPU(n_\mathrm{v}) + \sum_{l_\mathrm{v}\in L_\mathrm{v}} BW(l_\mathrm{v})) \cdot T_\mathrm{d}$, representing the revenue for accommodating the ith VN request[1].

Similarly, we define the long-term resource cost as:

$$\lim_{T\to\infty} \frac{\sum_{i=1}^{N} C^i(G_\mathrm{v})}{T}, \tag{2}$$

where $C^i(G_\mathrm{v}) = (\sum_{n_\mathrm{p}\in N_\mathrm{p}} CPU(n_\mathrm{p}) + \sum_{l_\mathrm{p}\in L_\mathrm{p}} BW(l_\mathrm{p})) \cdot T_\mathrm{d}$, representing the resource cost in physical network for accommodating the ith VN request.

We use the following formula as the RC ratio:

$$\sum\nolimits_{i=1}^{N} \frac{R^i(G_\mathrm{v})}{C^i(G_\mathrm{v})}. \qquad (3)$$

We use

$$\lim_{T\to\infty} \frac{\sum\limits_{t=0}^{T} V_S}{\sum\limits_{t=0}^{T} V} \qquad (4)$$

to represent the acceptance ratio, where $\sum\limits_{t=0}^{T} V_S$ represents the number of VN requests successfully embedded, and $\sum\limits_{t=0}^{T} V$ represents the total number of VN requests.

In this paper, like most of prior studies, we consider the long-term average revenue as the main performance metric.

3 Algorithm design

In this section, we will first introduce the idea of basic ABC algorithm and then present our proposed VNE-ABC algorithm.

3.1 Inspiration of ABC

Bee is a typical kind of social animal. They fully exploit individual intelligence to improve the efficiency and effect of the overall work greatly. In nature, this kind of animal, no matter where they are, can always find the best honey resource. In the process of gathering honey, three kinds of bees are involved: employed bees, onlooker bees, and scout bees. When an employed bee finds a food resource, it will go back to the beehive with the information of that food resource and lead other onlooker bees to that place. Then onlooker bees will begin to search potential better food resource around that food resource. Scout bees are responsible for finding potential new food resource in the whole searching space. Based on this model, Karaboga came up with a new meta-heuristic search algorithm [16], i.e., the artificial bee colony (ABC) algorithm.

3.2 Basics of ABC

The ABC algorithm simulates the intelligent behavior of bees when they are gathering honey. Due to its excellent characteristics such as simpleness, flexibility, and robustness, it has been widely applied in many fields. However, its model is immature. Sometimes, it is not suitable for dealing with NP-hard discrete optimization problems.

In basic ABC algorithm, it also divides bees into three types: employed bees, onlooker bees, and scout bees. Each food resource represents a solution to the corresponding problem. Assuming that each food resource is gathered by an employed bee, and each food resource and each employed bee are one-to-one mapping. Each employed bee searches for better food resource around the food resource which it attaches to. When an employed bee finds a better one, the bee will attach to the new one and discard the poorer one. In Fig. 2, the red square represents our current objective employed bee. When we want to update the food resource it attaches to, we first select another employed bee randomly (i.e., the green square in the figure). There lies a difference vector between these two squares. Then, the searching space of the objective employed bee is the circle, its center is the position of the objective employed bee, and the length of radius is the length of different vectors. The objective employed bee will search for a better food resource in the circle. If it finds a better one, then it will attach to the new one (i.e., the yellow square in the figure). Based on the information the employed bees provide, onlooker bees will select a certain food resource according to some methods, typically roulette method in basic ABC algorithm, and then search for better food resource around the food resource they choose. The searching space of the onlooker bee is determined in the same manner as employed bee searching. If onlooker bees find better food resources, they will inform the employed bees to update the food resource which they attach to. Scout bees deal with solutions which have been stopped updating, which avoids local optimum at some degree. Specifically, when an employed bee is trapped into local optimum, a scout bee will search for a feasible food resource randomly and the employed bee will give up the previous food resource and attach to the new random food resource. Hence, the scout bee search space is the whole

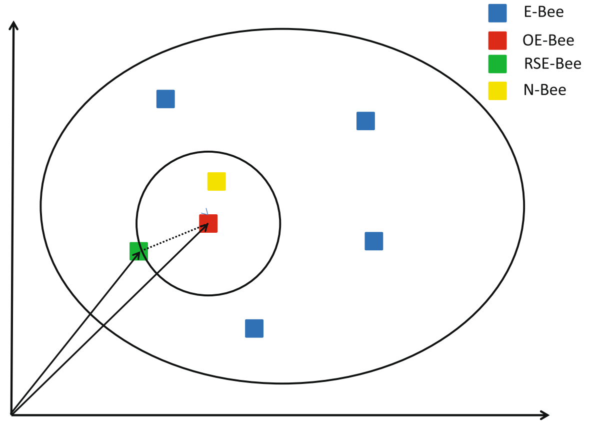

Fig. 2 Basic ABC algorithm illustration. A *blue square* refers to an employed bee. A *red square* refers to an objective employed bee. A *green square* refers to a randomly selected employed bee. A *yellow square* refers to a new selected employed bee

space, as shown in the ellipse in Fig. 2. Through the cooperation between these three types of bees, we can obtain the optimal or near-optimal solution to our problem.

3.3 VNE-ABC algorithm

In VNE-ABC algorithm, we first initialize k food resources (initial solutions) and bee colony. These bees will work together in three searching steps to update food resources. After several iterations, we can get a good solution.

Basic ABC algorithm has the following drawbacks so that we cannot apply it directly into our problem.

1. Basic ABC is not suitable for VN embedding problem since it only deals with continuous optimization problem. It cannot be used in our problem directly.
2. In order to measure the quality of different VN embedding solutions, it needs to define the fitness function in VN embedding context.
3. During the previous employed bee searching step and onlooker bees searching step, the update formula is as follows:

$$V(j) = X_i(j) + (\mathrm{rand}(-1,1))(X_i(j) - X_r(j)). \quad (5)$$

   The randomness of this equation will give rise to a slow convergence.
4. In the onlooker bee searching steps, roulette is a greedy method, which will cause solutions to be trapped into local optimum. In order to solve these problems, we introduce pheromone and sensitivity model to replace the roulette method.

We need to make some customizations to make this algorithm more suitable for virtual network embedding problem.

Discrete ABC for VN embedding. We first label each physical node with a unique index. Let the vector X represent one possible mapping method. Each dimension of this vector is the index of the corresponding physical node. Take the $X = (1, 3, 7, 9)$ for example. The four virtual nodes of a virtual request is mapped to the first, the third, the seventh, and the ninth physical node in the PN, respectively.

Fitness function customization. Since for different mapping solutions they consume the same resources on physical nodes, the difference lies in that they consume different bandwidth resources, i.e., mapping a virtual link to a physical link or a physical path where several physical links are involved. Thus, in order to measure the quality of different VN mapping methods, we consider minimizing the bandwidth resource consumption as the main objective function. Recall that X represents a possible mapping solution. We use $f(X)$ to represent the bandwidth consumption of this solution as shown in Eq. 6.

$$\sum_{(u,v)\in L_v} \sum_{(i,j)\in L_p} f_{ij}^{uv} \times \mathrm{BW}(l_{uv}), \quad (6)$$

where, f_{ij}^{uv} equals to 1 if virtual link l_{uv} is mapped to physical link l_{ij}. Otherwise, it equals to 0.

Similar to most of prior studies [2, 3, 26, 30], we apply the shortest path algorithm to complete link mapping and calculate the bandwidth consumption of each solution. The VNE-ABC algorithm attempts to find the best solution (i.e., the solution with the lowest fitness) for each request to minimize $f(X)$. We expect to obtain an embedding method with low fitness, i.e., low bandwidth consumption, since more requests can be embedded in *PN* if bandwidth resource can be saved. For example, as shown in Fig. 1, the fitness of this mapping solution $\{(a,b) \to (C,B), (a,c) \to (C,E), (c,d) \to (E,F), (d,b) \to (F,B)\}$ is the actual total bandwidth cost on the physical links $(C,B), (C,E), (E,F), (F,B)$, i.e., $10+15+10+30 = 65$.

Preferential exchange strategy. In the employed bee searching step and onlooker bees searching step, we propose a preferential exchange strategy. When we are updating the ith solution X_i, we first choose another solution X_j randomly where $i \neq j$. Then we will calculate the fitness of these two solutions and their selected probability. For example, the fitness of these two solutions is represented by f_i and f_j, respectively. Then the probability P_i for X_i and P_j for X_j are $P_i = \frac{f_j}{f_i+f_j}$ and $P_j = \frac{f_i}{f_i+f_j}$, respectively, which means the solution X_i will be selected in a probability of P_i and the solution X_j will be selected in a probability of P_j. Once a solution is selected, X_j for example, then the value of current dimension of the selected solution X_j^d will be used to replace the solution X_i^d. Each dimension of X_i will be updated through the above strategy.

$$X_i^d = \begin{cases} X_i^d \text{ if } \mathrm{rand}(0,1) \leq P_i \\ X_j^d \text{ if } \mathrm{rand}(0,1) > P_i \end{cases} \quad (7)$$

For example, if X_i is (1, 2, 3, 4) and X_j is (5, 6, 7, 8) and we get a random number 0.3 which is bigger than P_i, then X_i will become (5, 2, 3, 4). We also do the same operation for the rest three dimensions.

In this way, we make ABC algorithm more suitable for the discrete problem. However, this method will narrow the searching space, since the searching space just depends on the potential solutions generated from the beginning. To deal with this problem, we will discard the worst solution, whose $f(X)$ is the largest, and re-initialize it in scout bees searching step to let more potential solutions come into exchange.

Algorithm 1: The procedure of VNE-ABC algorithm

```
1: Initialization: Set the number of food resources to $k$,
   limit trial time to $t$, and iteration time to $i$.
2: Employed bees searching: For each food resource,
   employed bees search around them and using
   Formula 7 to update food resources. If a food
   resource does not change, $t$++.
3: Onlooker bees searching: Scout bees first uses
   Formula 8 to select proper food resource and then
   uses Formula 7 to update those selected food
   resource. If a food resource does not change, $t$++.
4: Scout bees searching: Scout bees first check those
   food resources in $l$ attempts. For those food
   resources whose attempt exceed $l$. Then those food
   resources will be re-initialized. And the worst food
   resource will be re-initialized in any case.
5: Iterate or end: Examine the iteration time. If it is less
   than $i$, go to step 2. Otherwise, go to step 6.
6: End: Output the best solution and stop.
```

Pheromone and sensitivity model. In the onlooker bees searching steps, we introduce a new method called pheromone and sensitivity model as an alternative of roulette method. In this new model, we use the following equation for solution selection:

$$\begin{cases} n(f_i) = \begin{cases} \frac{f_i - f_{\min}}{f_{\max} - f_{\min}}, \text{if} f_{\max} \neq f_{\min} \\ 0, \text{else} \end{cases} , \\ n(f_i) \leq S(k) \end{cases} \quad (8)$$

where f_i is the fitness of individual i; $f_{\max}$ and $f_{\min}$ are the maximum and minimum fitness among food resources; $n(f_i)$, the normalize fitness, is called the pheromone of the ith food resource; the kth onlooker bee sensitivity, i.e., $S(k)$ is randomly chosen from 0 to 1. For example, given five food resources, assume their pheromone are 0.1, 0.2, 0.3, 0.4, and 0.5, respectively. We get 0.35 for $S(k)$. Then the first three food resources will be selected. The traditional roulette model in basic ABC algorithm is a greedy method, which may cause it to trap into local optimum. The benefit of pheromone and sensitivity model lies in that it can avoid this problem to some extent.

From the above discussion, we finally come to the following steps about VNE-ABC algorithm which is shown in Algorithm 1 (Fig. 3).

3.4 Time complexity analysis

Note that, for once node mapping and link mapping, the time complexities are $O(|N_v| \cdot |N_p|)$ and $O(|L_v|^3 \cdot |N_p|)$, respectively. Thus, VNE-ABC is a polynomial-time algorithm in terms of $O(KTI)$ times of once node mapping and link mapping.

4 Experimental results

4.1 Experiment settings

The PN topology has 50 nodes and 283 links. The CPU and bandwidth resource of PN nodes and links are real numbers uniformly distributed between 50 and 100. The CPU and bandwidth requirements of virtual nodes and links are real number uniformly distributed between 50 and 100. We assume that in every 100 time units the arrival of virtual network requests obey Poisson process with average value 5. The duration time of each virtual network request follows exponential distribution; their average duration time is 500 time units. Each pair of virtual network nodes has a possibility of 0.5 to be connected. Every simulation experiment will run about 50,000 time units, which includes 2000 virtual requests. For VNE-ABC algorithm, we set the number of food resources to 5 and the maximum iteration time to 20.

4.2 Simulation results and analysis

From Fig. 5a, d, compared to VNE-PSO, VNE-ABC generates 35.4% more long-term average revenue and increases the virtual network requests acceptance by 11%. PSO algorithm has a high rate of convergence, but it traps into local optimum easily. In contrast, for ABC algorithm, it avoids this problem to some extent due to the following reason: the scout bees searching in basic ABC algorithm discards those solutions which are constant for several iterations. In our modified ABC algorithm, we go a step further. When using pheromone and sensitivity model to replace roulette model, it avoids local optimum to some extent as well. In the basic ABC algorithm, Formula 5 is not suitable for VN embedding problem. We propose an alternative (i.e., Formula 7), which not only makes ABC algorithm adapt into virtual network embedding problem, but also makes solutions evolve toward the optimal solution.

From Figs. 4, 5 and 6, we can see that the degree of long-term average revenue of VNE-ABC outperforming VNE-PSO arrives at peak (i.e., 35.4%) when the VNs are normal-sized. When the VNs are small-sized and large-sized, the long-term average revenue strength of VNE-ABC over VNE-PSO are 19.6 and 29.6%, respectively. This is expected, because when the VNs are small-sized, both two algorithms have a relatively high request acceptance rate; however, when the VNs are large-sized, the PN is overloaded. Thus, both two algorithms have a relatively low request acceptance rate. The advantage of VNE-ABC over VNE-PSO has not been fully exhibited.

5 Related work

The VN embedding problem has been proven to be NP-hard. To solve this problem, two branches of techniques are proposed. The first one is formulating it to MLP or ILP and applying standard techniques such as relaxation and rounding [1, 6]. The other one is based on heuristics. For

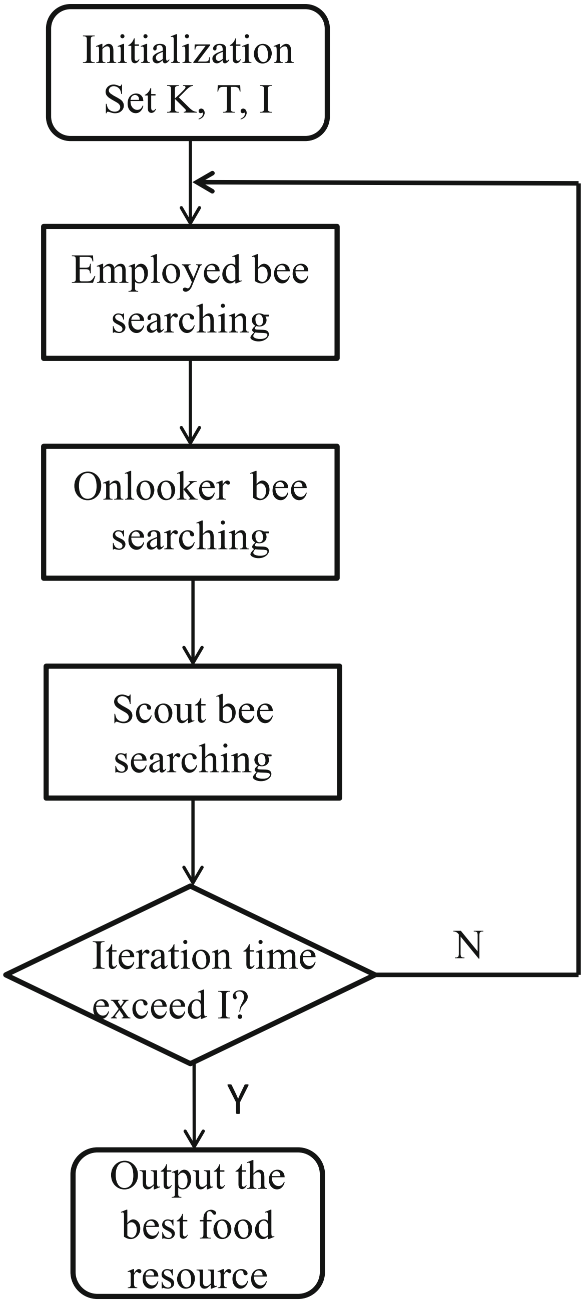

Fig. 3 The procedure of VNE-ABC algorithm

example, Yu et al. [26] proposed to use the CPU and bandwidth information as a measurement for the node mapping. Cheng et al. [3] proposed to leverage the topology information of the nodes to design two VN embedding algorithms RW-Maxmatch and RW-BFS to solve the VN embedding problem.

The first branch of techniques can get a better solution while incurring a very large time complexity. The second branch of techniques have a very low time complexity while the quality of solutions is awaiting to be optimized. To make a good balance between the quality of solution and time complexity, meta-heuristics are proposed to deal with the VN embedding problem, because they provide multiple solutions and allow us to choose the best from these solutions [8]. For example, Fajjari et al. proposed a novel ant colony optimization (ACO) based VN embedding algorithm [9] called VNE-AC. Cheng et al. and Zhang et al. employed particle swarm optimization (PSO)

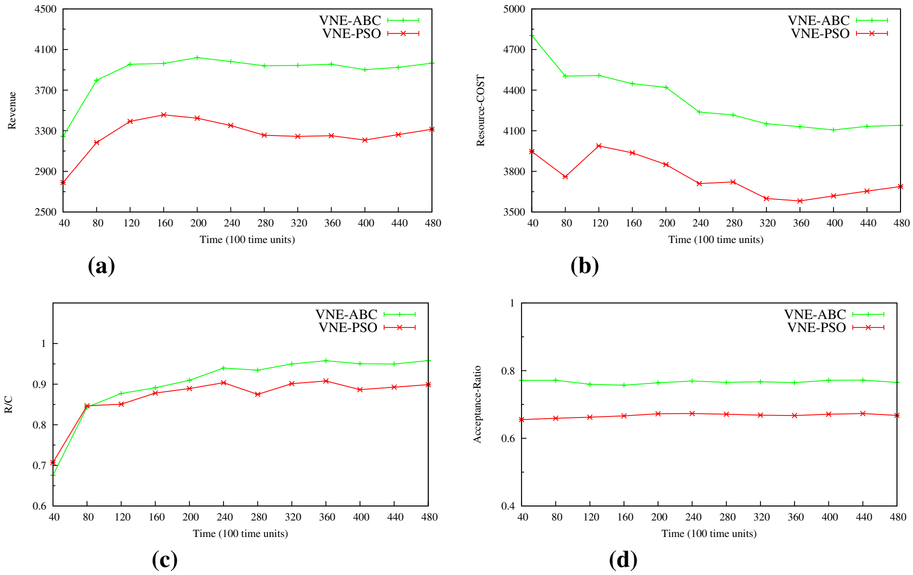

Fig. 4 Comparisons between our algorithm and the state-of-the-art algorithm (small-sized VNs). **a** The long-term average revenue. **b** The long-term average cost. **c** The RC-ratio. **d** Acceptance ratio

to maximize the revenues by trying to accommodate more VN requests [2, 30].

In this paper, to further improve the quality of the VN embedding solution, we devise an ABC based VN embedding algorithm to obtain near-optimal solutions in terms of the ratio of the long-term average revenue and the VN request acceptance.

VN embedding across multiple domains (i.e., InPs) has been studied in recent years. For example, Chowdhury et al. [4] present a policy-based inter-domain VN embedding framework that embeds end-to-end VNs in a decentralized manner. Houidi et al. [14] present an optimization problem and a heuristics algorithm for the problem.

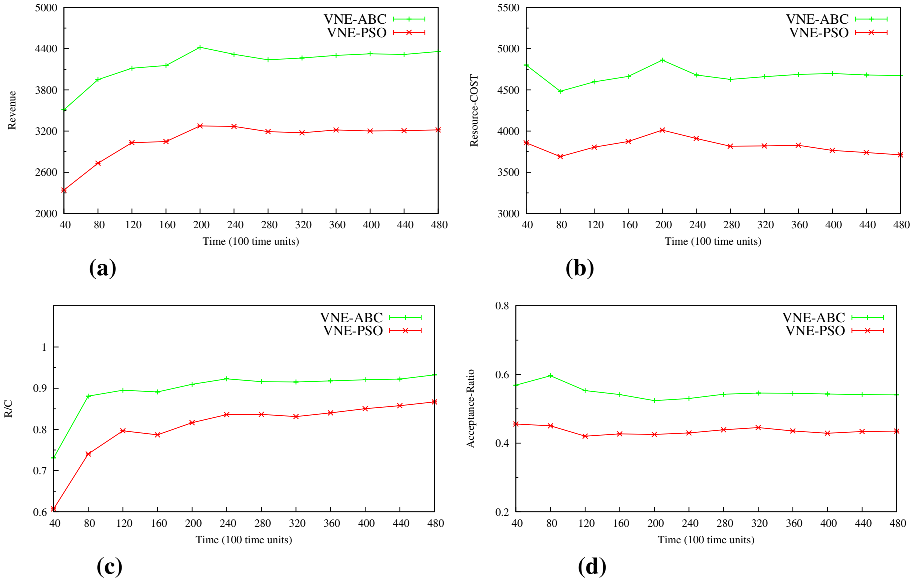

Fig. 5 Comparisons between our algorithm and the state-of-the-art algorithm (normal-sized VNs). **a** The long-term average revenue. **b** The long-term average cost. **c** The RC-ratio. **d** Acceptance ratio

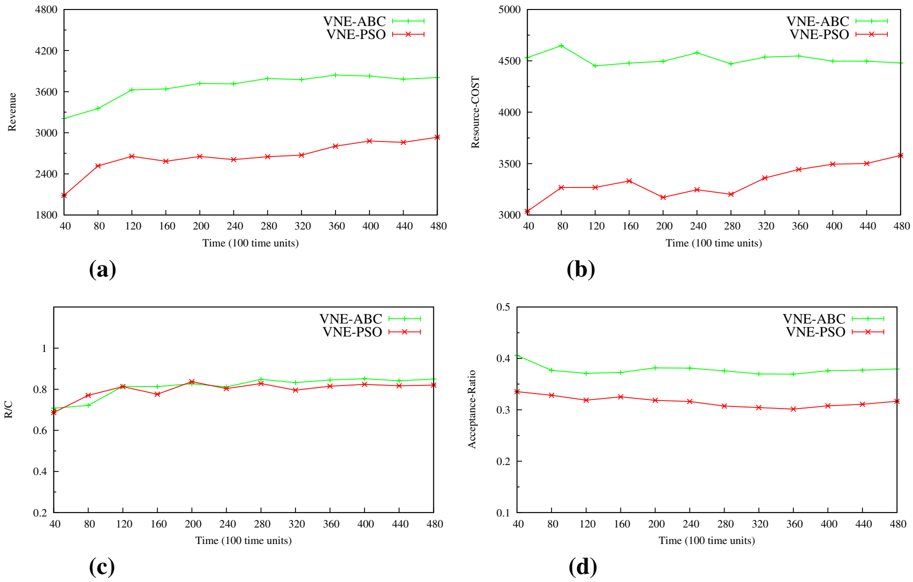

Fig. 6 Comparisons between our algorithm and the state-of-the-art algorithm (large-sized VNs). **a** The long-term average revenue. **b** The long-term average cost. **c** The RC-ratio. **d** Acceptance ratio

Recently, in our previous studies [19, 20, 29, 30], we considered another very important issue in VN embedding, i.e., energy conservation. How to leverage the ABC technique to further to optimize the energy consumption will be further considered in the future.

The authors can read the surveys [5, 11] of network virtualization and VN embedding for further reference. Besides the virtual network embedding allocation problem in cloud computing, there are also some hot topisc, e.g., security and privacy issues, which have attracted a lot of attention [12, 13, 18, 22–25, 27, 31]. Because of the page limit, we do not introduce them one by one.

6 Conclusions

In this paper, we propose another meta-heuristic (i.e., ABC) based algorithm to solve the VNE problem. Specifically, we redefine the parameters and operations of the parameters according to our problem. We then propose two strategies to further optimize the performance of the ABC based VN embedding algorithm. The first is preferential exchange to accelerate the speed of convergence. The second is the pheromone and sensitivity model strategy, which avoids local optimum solution to a large extent.

Through extensive simulations, we show that our algorithm can obtain nearly 35.4% more revenue than the state-of-the-art algorithm.

Acknowledgements
This work was supported in part by the following funding agencies of China: National Natural Science Foundation under Grant 61170274, 61602050 and U1534201.

Competing interests
The authors declare that they have no competing interests.

Received: 8 August 2016 Accepted: 1 November 2016
Published online: 29 November 2016

References

1. JF Botero, X Hesselbach, M Duelli, D Schlosser, A Fischer, H De Meer, Energy efficient virtual network embedding. Commun. Lett. IEEE. **16**(5), 756–759 (2012)
2. X Cheng, S Su, Z Zhang, K Shuang, F Yang, Y Luo, J Wang, Virtual network embedding through topology awareness and optimization. Comput. Netw. **56**(6), 1797–1813 (2012)
3. X Cheng, S Su, Z Zhang, H Wang, F Yang, Y Luo, J Wang, Virtual network embedding through topology-aware node ranking. ACM SIGCOMM Comput. Commun. Rev. **41**(2), 39–47 (2011)
4. M Chowdhury, F Samuel, R Boutaba, in *Proceedings of the second ACM SIGCOMM workshop on VISA*. Polyvine: policy-based virtual network embedding across multiple domains (ACM, New Delhi, 2010), pp. 49–56
5. NM Chowdhury, R Boutaba, A survey of network virtualization. Comput. Netw. **54**(5), 862–876 (2010)
6. MMK NChowdhury, MR Rahman, R Boutaba, in *INFOCOM*. Virtual network embedding with coordinated node and link mapping (IEEE, 2009), pp. 783–791
7. MMK NChowdhury, MR Rahman, R Boutaba, ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping. IEEE/ACM Trans. Netw. **20**(1), 206–219 (2012)
8. Oca Montes de MA, T Stutzle, Enden Van den K, M Dorigo, Incremental social learning in particle swarms. Syst. Man, Cybernet. Part B: Cybernet. IEEE Trans. **41**(2), 368–384 (2011)
9. I Fajjari, N Aitsaadi, G Pujolle, H Zimmermann, in *Communications IEEE International Conference on*. Vne-ac: Virtual network embedding algorithm based on ant colony metaheuristic (IEEE, Kyoto, 2011), pp. 1–6
10. J Fan, M Ammar, in *INFOCOM*. Dynamic topology configuration in service overlay networks: a study of reconfiguration policies (IEEE, Barcelona, 2006), pp. 1–12
11. A Fischer, J Botero, M Beck, H De Meer, X Hesselbach, Virtual network embedding: a survey. Commun. Surv. Tutor. **15**(4), 1888–1906 (2013)
12. Z Fu, K Ren, J Shu, X Sun, F Huang, Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Transactions on Parallel and Distributed Systems. **27**(9), 2546–2559 (2016)

13. Z Fu, X Wu, C Guan, X Sun, K Ren, Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans. Inform. Forens. Secur. **11**(12) (2016)
    pages=2706–2716,
14. I Houidi, W Louati, W Ameur, D Zeghlache, Virtual network provisioning across multiple substrate networks. Comput Netw. **54**(4), 1011–1023 (2011)
15. I Houidi, W Louati, D Zeghlache, in *Proceedings of IEEE ICC*. A distributed virtual network mapping algorithm, (Beijing, 2008), pp. 5634–5640
16. D Karaboga, An idea based on honey bee swarm for numerical optimization. Technical report (2005). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department
17. J Lischka, H Karl, in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*. A virtual network mapping algorithm based on subgraph isomorphism detection (ACM, Barcelona, 2009), pp. 81–88
18. Y Ren, J Shen, J Wang, J Han, S Lee, Mutual verifiable provable data auditing in public cloud storage. J. Internet Technol. **16**(2), 318 (2015)
19. S Su, Z Zhang, X Cheng, Y Wang, Y Luo, J Wang, in *IEEE INFOCOM WS-CCSES: Green Networking and Smart Grids*. Energy-aware virtual network embedding through consolidation, (Orlando, 2012), pp. 2708–2713
20. S Su, Z Zhang, AX Liu, X Cheng, Y Wang, X Zhao, Energy-aware virtual network embedding. IEEE Trans. Netw. **22**(5), 1607–1620 (2014). online published
21. G Sun, H Yu, L Li, V Anand, Y Cai, H Di, Exploring online virtual networks mapping with stochastic bandwidth demand in multi-datacenter. Photonic Network Commun. **23**(2), 109–122 (2012)
22. MA Tinghuai, Z Jinjuan, T Meili, T Yuan, A-D Abdullah, A-R Mznah, L Sungyoung, Social network and tag sources based augmenting collaborative recommender system. IEICE Trans. Inform. Syst. **98**(4), 902–910 (2015)
23. Z Xia, X Wang, X Sun, B Wang, Steganalysis of least significant bit matching using multi-order differences. Secur. Commun. Netw. **7**(8), 1283–1291 (2014)
24. Z Xia, X Wang, X Sun, Q Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans. Parallel Distributed Syst. **27**(2), 340–352 (2016)
25. Z Xia, X Wang, L Zhang, Z Qin, X Sun, K Ren, A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. IEEE Trans. Inform. Forensics Secur. **11**(11), 2594–2608 (2016)
26. M Yu, Y Yi, J Rexford, M Chiang. Rethinking virtual network embedding: substrate support for path splitting and migration, vol. 38, (2008), pp. 17–29
27. C Yuan, X Sun, R Lv, Fingerprint liveness detection based on multi-scale lpq and pca. Chin. Commun. **13**(7), 60–65 (2016)
28. S Zhang, Z Qian, J Wu, S Lu, L Epstein, Virtual network embedding with opportunistic resource sharing. Parallel Distrib. Syst. IEEE Trans. **25**(3), 816–827 (2014)
29. Z Zhang, S Su, X Niu, J Ma, X Cheng, K Shuang, in *IEEE GLOBECOM*. Minimizing electricity cost in geographical virtual network embedding, (2012), pp. 2609–2614
30. Z Zhang, X Cheng, S Su, Y Wang, K Shuang, Y Luo, A unified enhanced particle swarm optimization-based virtual network embedding algorithm. Int. J. Commun. Syst. **26**(8), 1054–1073 (2013)
31. F Zhangjie, S Xingming, L Qi, Z Lu, S Jiangang, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Trans. Commun. **98**(1), 190–200 (2015)
32. Y Zhu, M Ammar, in *INFOCOM*. Algorithms for assigning substrate network resources to virtual network components, (Barcelona, 2006)