An Artificial Bee Colony based Virtual Network Embedding Model

- A novel approach based on artificial bee colony (ABC) to address virtual network embedding in a multiple InP network virtualization environment is presented.
- The central feature of the proposed novel ABC approach based on SI is that it is intended to solve NP-hard problems.
- The proposed approach defines VNE among multiple InPs and provides an embedding algorithm based on ABC in a dynamic and unknown network virtualization environment, where VN requests arrive dynamically to be embedded.
- Compared with alternative approaches in the literature, the proposed approach has the benefits as following: (1) local search (2) improvisation of solution and (3) memory. To comprehend these advantages, ABC algorithmic approach is chosen to model the NP-hard VNE problem.

Formal description of VN embedding

- A set of VN requests at time ‘t’ is defined by a set $G_v(k, t)$where k = {1, 2, 3,…, n}
- The InPs manage various substrate networks which is defined by a set $G_s(j)$ where j = {1,2,3,..,m}
- The total unit of resources available with the substrate network at time ‘t’ is defined as in 23.

$$RG_s(j, t) = \{\alpha_1 Nr(j, t)+\alpha_2 Lr(j, t)\} \quad (23)$$

where, Nr(j, t) = total CPU resource available with substrate network j at time ‘t’ and

Lr(j, t) =total bandwidth resource available with substrate network j at time ‘t’

$\alpha_1$and α_2 weighted parameters

Formal description of VN embedding

- The total unit of resources requested by a virtual network request ‘k’ arriving at time ‘t’ is defined as in 24

$$RG_v(k, t) = \{\alpha_1 Vr(k, t) + \alpha_2 Er(k, t)\} \quad (24)$$

where, Vr(k, t)= total CPU resource requested by virtual network ‘k’ arriving at time ‘t’ and

Er(k, t) =total bandwidth resource requested by virtual network ‘k’ arriving at time ‘t’

- A decision variable X_{kj} is defined as

$$X_{kj} = \begin{cases} 1 & \textit{if virtual network k is mapped on physical network j} \\ 0 & \textit{otherwise} \end{cases} \quad \forall k\ 1 \le k \le n,\ \forall j\ 1 \le j \le m$$

- For embedding a VN request, the InP has to rent resources (CPU and bandwidth) from the substrate network. Thus the cost $C(G_v)$ of theVN request mapped on the substrate network is defined as the total cost of CPU and bandwidth resources allocated to those VN requests. The embedding cost $C(G_v)$ is calculated as follows:

$$C(G_v) = \left(\sum_{es \in L} \sum_{ev \in E} \beta * L_{r_{es}}^{ev} + \sum_{nv \in V} \gamma * V_r(nv) \right) \quad (25)$$

FORMAL DESCRIPTION OF VN EMBEDDING

Here, ${L_{r}}_{es}^{ev}$ denote the sum of substrate link bandwidth ${L_{r}}_{es}$ consumed by ev. βandγ represents the cost of unit bandwidth and CPU respectively.

- The objective of VNE is to minimize the cost of mapping VN requests on multiple substrate networks owned by multiple InPs. Thus the objective function is defined as,

$$\min \sum_{k=1}^{n} \sum_{j=1}^{m} \mathrm{C(Gv)}_{kj} X_{kj}$$

subject to $\sum_{k=1}^{n} \mathrm{RG_v(k,t)X_{kj}} \leq \mathrm{RG_s(j,\ t)}$ $\quad \forall \mathrm{j}\ 1 \leq \mathrm{j} \leq \mathrm{m}$

and$\sum_{j=1}^{m} X_{kj} = 1$ $\quad \forall \mathrm{k}\ 1 \leq \mathrm{k} \leq \mathrm{n}$

Artificial Bee Colony (ABC) Algorithm

- Artificial Bee Colony (ABC) is the most recent meta-heuristic algorithm which is defined by Dervis Karaboga. It is an iterative algorithm which is motivated by the intelligent behavior of honey bees. In the ABC model, the colony consists of three categories of bees: **employed bees, onlookers and scouts.** In the ABC algorithm, the first half of the swarm consists of employed bees, and the second half constitutes the onlooker bees. The proposed ABC approach for addressing the problem in this paper functions as follows.

1. Randomly generated food sources, which form the solutions, are associated with the employed bees initially.
2. After this in each iteration, the employed bee modifies the source position in her memory and determines new food source position in the neighborhood of currently associated food source.
3. The nectar value of the new food source is evaluated, which represents fitness. Provided that the nectar value of the new one is greater than that of the previous source, the employed bee moves to this new source position and forgets the old one.

Artificial Bee Colony (ABC) Algorithm

4. Otherwise, she keeps the position of the old food source in her memory and retains it. When all the employed bees have completed the search process, they share the nectar information of the sources with the onlooker bees through dancing.
5. Each onlooker bee at this stage, evaluates the nectar information and selects the food source depending on the probability proportional to the nectar amount of that food source.
6. The abandoned food sources are determined and replaced with new sources which are discovered by scouts.
7. In the ABC algorithm, it is assumed that for every food source only one employed bee exists. Therefore, the number of employed bees or the onlooker bees is equal to the number of solutions in the swarm.

Artificial bee colony initialization

- The initial bee colony is built using the initial virtual network embedding algorithm, where the solution is generated as follows.

(1)The next VN to be embedded on a substrate network managed and owned by InP is selected at each step.

(2) The substrate network on which the VN request will be embedded is determined.

(3) Repeat steps (1) and (2) until all the VN requests are embedded on the substrate networks.

Initial Virtual Network Embedding Algorithm

- **Step1: Initialization of VN requests and SNs**
- $G_v(k,t)$ = {VN requests at time 't'}, k = {1, 2, 3,…, n}
- $G_s(j)$ = {SNs}, j = {1, 2, 3,…, m}
- Let $P_j = \emptyset$, $\forall j\ 1 \le j \le m$ where P_j is the set of VNs arriving at time 't' which are assigned to SN j
- Create a list of SNs, L_k for each k^{th} VN having sufficient resources to embed the VN request.
- Initially $L_k = \{1,2,3,\ldots,m\} \forall_k$
- **Step2: Selecting an appropriate SN for each VN**
- While all the VN requests, $G_v(k, t)$ not mapped, repeat
- Randomly choose a substrate network j* from L_k
- Calculate the probability function depending on resource available with j^{th} substrate network and resource required by k^{th} VN :
- $\text{Probability}_{k,j} = \frac{RG_s(j,t)/RG_v(k,t)}{\sum_{l \epsilon L_k} RG_s(l,t)/RG_v(k,t)}$, $j \epsilon L_k$ The substrate network 'j' with the highest probability function $\text{Probability}_{k,j}$ is selected.
- Map the VN request 'k' on the substrate network j*: $P_{j*} = P_j \cup \{k\}$
- k=k+1
- if $\sum_{k \in P_{j_*}} RG_v(k,t) > RG_s(j^*, t)$, remove j* from any list. Repeat steps 5 to 11.
- **Step3: Register the assigned VN request 'k' at SN 'j'**
- $\sigma(k) = j$ if $k \epsilon P_j$

Proposed ABC based virtual network embedding methodology

- After building the initial bee colony through initial virtual network embedding algorithm, the solutions are generated.
- Each solution $S_h(h = 1,2,3,.....,SL)$ is a d-dimensional vector, where d is the number of optimization parameters and SL denotes the swarm size.
- The search processes of the employed, onlooker and scout bees undergo repeated cycles [C=1,2,…,MCN(maximum cycle number)] to generate improved position(solution) having a higher nectar value than of the previous source.
- The food sources (solutions) are evaluated on the basis of the fitness function after the potential solutions have been built.
- The probability of selecting the individual bees is determined by the fitness function (nectar value).
- The initial solutions that are constructed in the ABC based VNE algorithm, may produce infeasible solutions as well.
- Thus, a penalty term α_j ($\alpha_j > 0$)is introduced here and added to the fitness function to convert the constrained problem into an unconstrained one.

Proposed ABC based virtual network embedding methodology

- The fitness function is formulated as follows:

fitσ(b) =
$$\sum_{k=1}^{n}\sum_{j=1}^{m} \mathrm{C(Gv)}_{kj}\, X_{kj} + \sum_{j=1}^{m} \alpha_j \max\left(0, \sum_{k=1}^{n} \mathrm{RGv(k,t)}\, X_{kj} - \mathrm{RGs(j,t)}\right) \quad (26)$$

where, σ(b) = solution of employed bee **b**

fit σ(b) = fitness value of employed bee **b**

α_j= initial value of penalty parameter for j^{th} SN when it's capacity is overloaded

σ_{best} = best solution

- The solutions for each VN request embedding is probabilistically selected through roulette wheel selection mechanism using equation as follows. The probability p_h of selecting a solution S_h is then calculated.

$$p_h = \frac{\mathrm{fit}\,\sigma(h)}{\sum_{k=1}^{SL} \mathrm{fit}\,\sigma(k)} \quad (27)$$

- The onlookers determine the best food source among all the neighboring food sources which forms the new location of food source **k**.

Solution improvisation using local search

- As each virtual network embedding solution consists of **d** parameters, let $S_h = (S_{h1}, S_{h2,} I_, S_{hd})$ be a candidate solution.
- A new solution $NewS_h$ in the neighborhood of $S_{h,}$ is determined by randomly selecting a solution parameter j and another solution $S_i = (S_{i1}, S_{i2,} I_, S_{id})$.
- All the parameter values of $NewS_h$ are similar to S_h except for the randomly chosen parameter **j**, i.e. $NewS_h = (S_{h1}, S_{h2,} I_, S_{h,j-1}, W_{h,j}, S_{h,j+1}, L, S_{hd})$.
- The selected new solution is determined with the following equation 28.

$$NewS_{hj} = S_{hj} + q(S_{hj} - S_{ij}) \quad (28)$$

where **q** is a uniform variable within the range [-1,1]. The value for the parameter **j** is adjusted by setting it to a corresponding extreme value in that range, if it tends to fall outside the range.

Solution Updating

- The virtual network embedding process iteratively runs until the termination condition is reached.
- If the position of a solution is not improved in ABC algorithm till a predetermined number of cycles, it is assumed that a food source be abandoned.
- Let S_h be the abandoned food source, then scout bee searches for a new food source to replace S_h. This task can be defined as in equation 29.

$$S_h^j = S_{min}^j + rand[0,1](S_{max}^j - S_{min}^j) \quad j = 1,2,\ldots\ldots,d \quad (29)$$

DYNAMIC VIRTUAL NETWORK EMBEDDING ALGORITHM

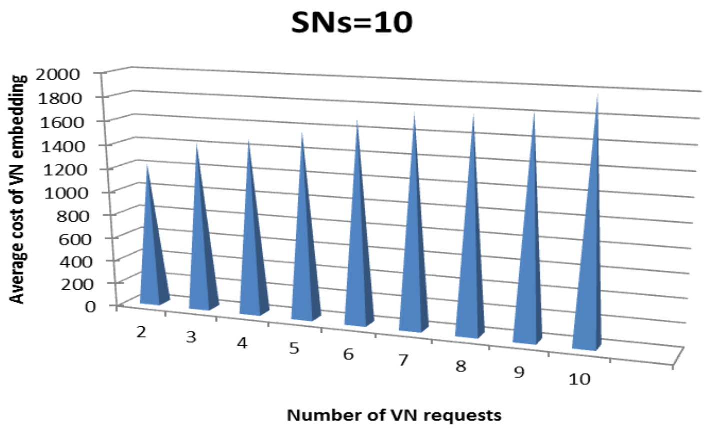
- **Step 1: Initialization**
- $G_s(j)$ = {substrate networks managed by InPs}, j = {1,2,3,..,m}
- $G_v(k, t)$= {VN requests arriving at time 't'}, k = {1, 2, 3,…, n}
- Construct the initial bee population S_h
- Initialize MCN, C=1
- **Step 2: VN Embedding**
- While C < MCN
- Generate the new population $NewS_h$ for the employed bees in the neighborhood of S_h using equation 28 and evaluate them.
- Apply greedy selection method between S_h and $NewS_h$
- Calculate the probability p_h to select the solutions S_h using equation 27. Generate new population $NewS_h$ for the onlooker bees from population S_h based on p_h for evaluation.
- Again apply greedy selection method between S_h and $NewS_h$
- Replace the abandoned solution if exists, with a new randomly produced solution S_h for the scout using equation 29.
- Best solution achieved so far is memorized by the bee
- C = C +1.
- **Step 3: Finish VN Embedding Process**
- Each VN request is embedded on the substrate networks owned and managed by the respective InPs.

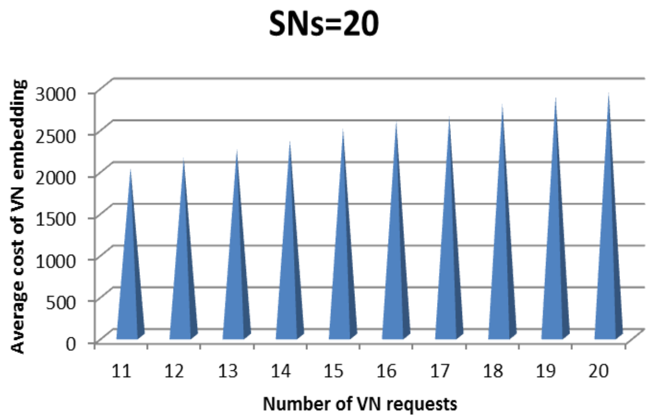
Experimental Results

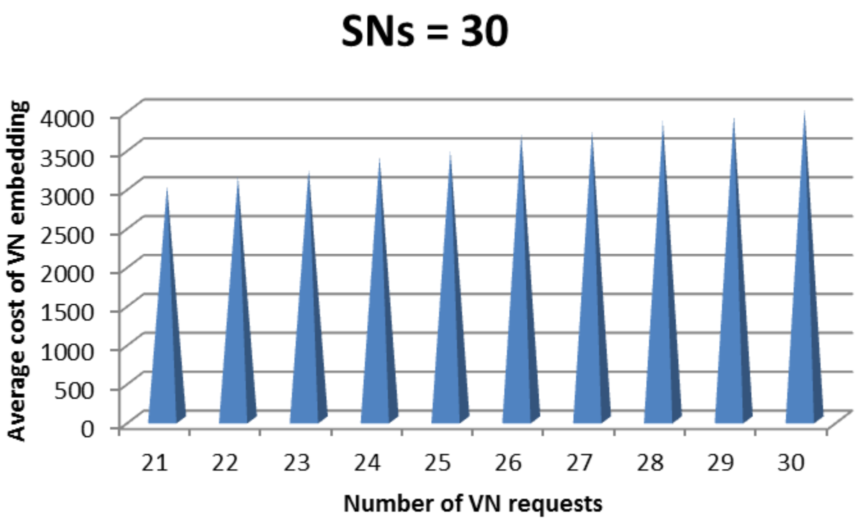
- The simulation study of the algorithm is done on three different metrics viz., cost of VN embedding, VN embedding time and VN acceptance ratio.
- The general input parameters used for the simulation are as follows:

1. The population size of artificial antis 50 and the MCNis taken as 100.
2. Size of substrate network varies uniformly in a range of 50 to 100.
3. Size of VN requests varies uniformly in a range of 2 to 20.
4. All pairs of substrate nodes and VN vertices are randomly connected with a probability of 0.5
5. Weights on the nodes and links of the SN are uniformly distributed between 50 and 100.
6. Weights on the vertices of the VN requests are uniformly distributed within a range of 0 to 20
7. Weights on the VN’s edges are uniformly distributed between 0 and 50.
8. $\alpha_j = 1$

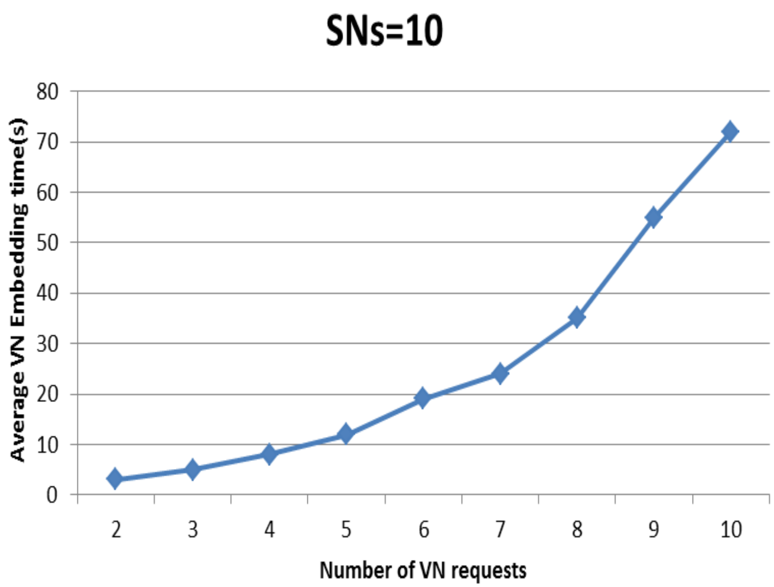
Observation on VN embedding cost

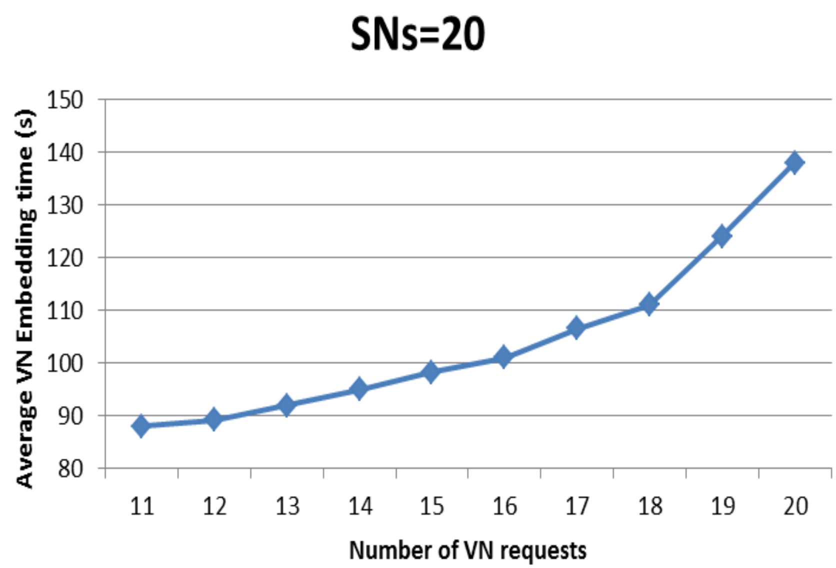


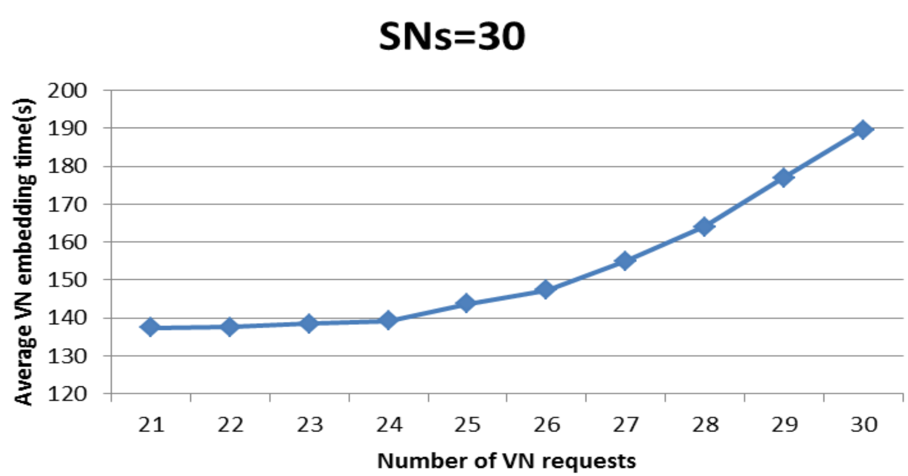




Observation on VN embedding time





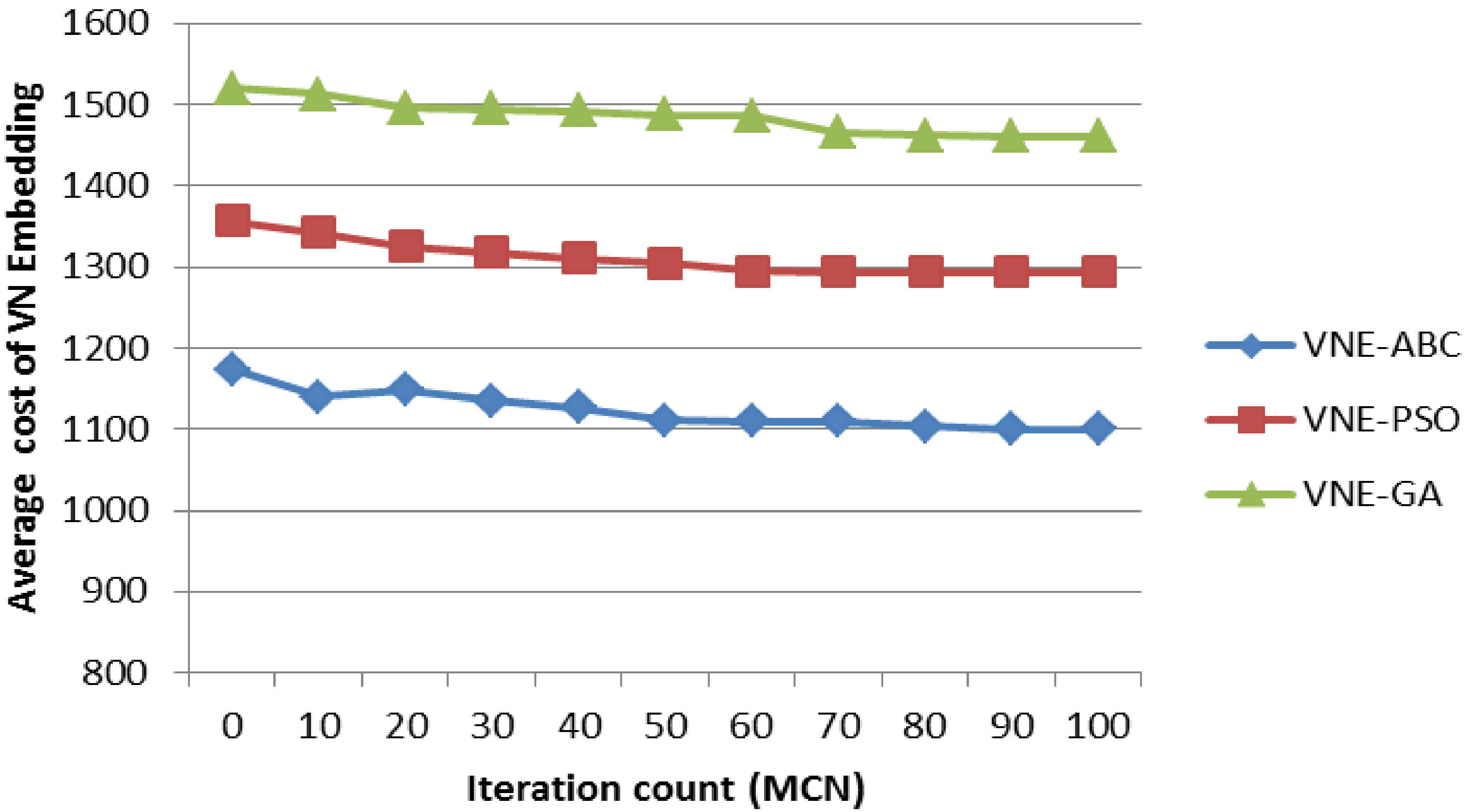


Observation for online VN requests

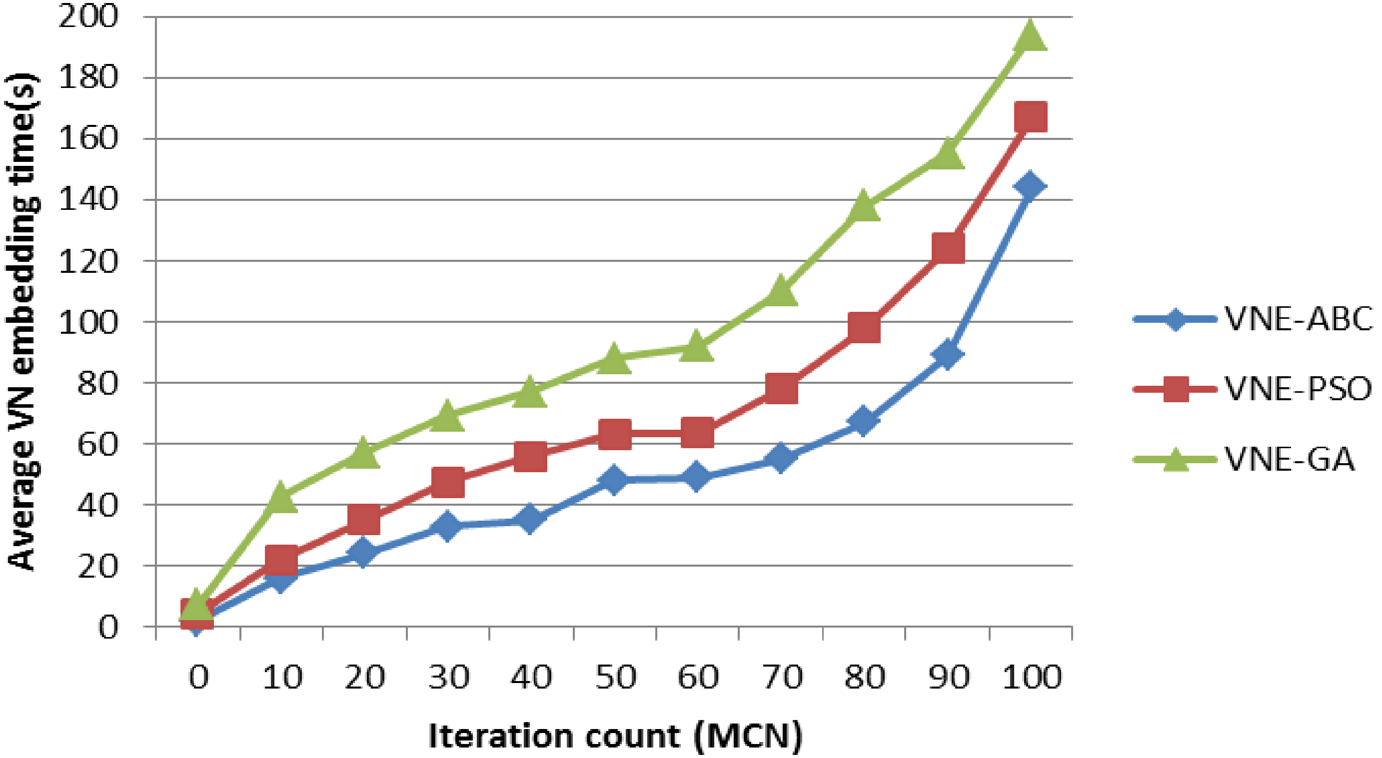
- Following are the assumptions made for online VN requests in a dynamic network virtualization environment.

1. The VN requests arrive in a Poisson process with an average rate of 4 VNs per 100 time units, and each one has an exponentially distributed lifetime with an average of 500 time units.
2. The simulation is run for about 50,000 time units.
3. For GA based VNE, the crossover probability is kept as 0.8 and the mutation probability is kept as 0.05. Roulette wheel selection is used.
4. For PSO based VNE, the inertia weight w is set to 0.8 and c1=c2= 1.0
5. The rest input parameters remain same as discussed before in the experimental evaluation.

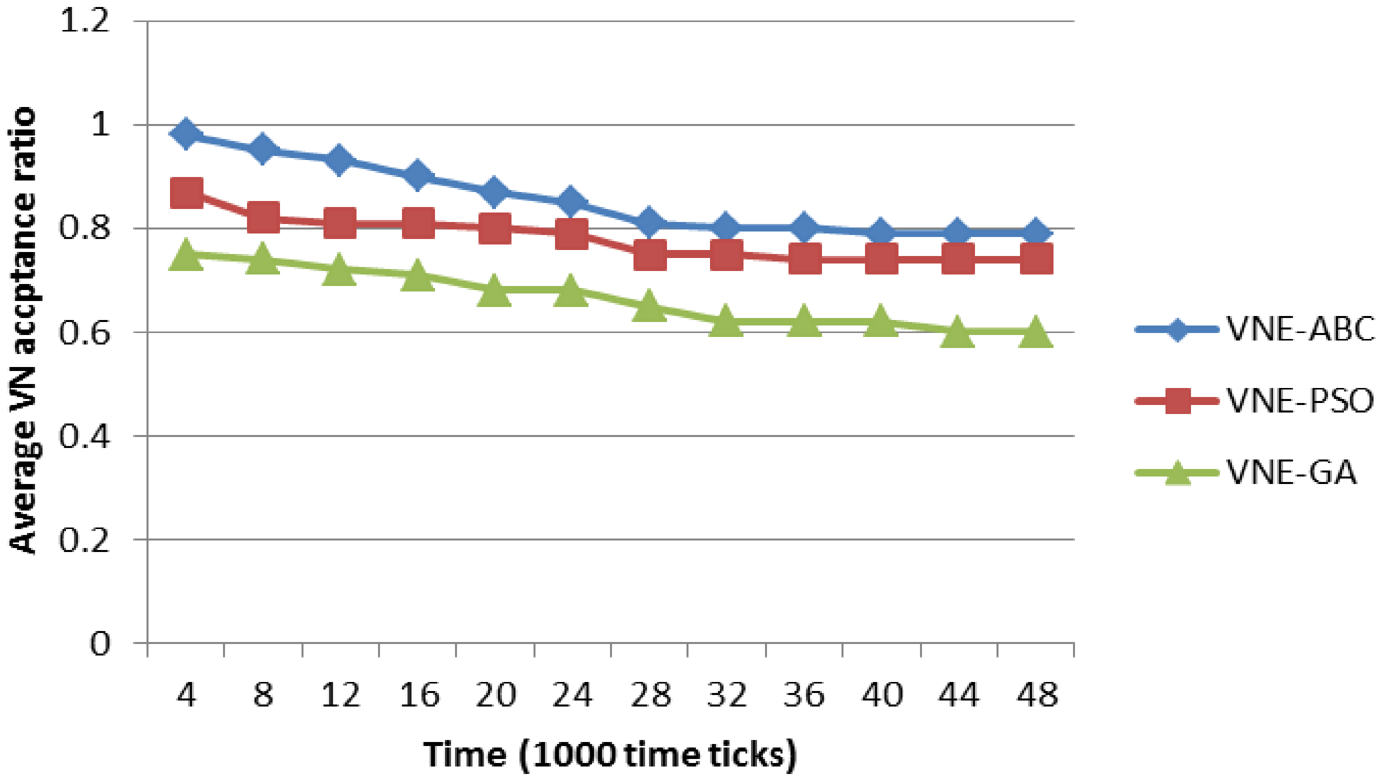
Average VN Embedding cost Comparison



AVERAGE VN EMBEDDING TIME COMPARISON



Average VN acceptance ratio comparison



Conclusion Of Model

- In this study, virtual network embedding problem in distributed multiple InP network virtualization environment is considered.
- Major nature inspired approaches to address the VN demands and complexity of VN embedding problem in multiple InP scenario were analyzed.
- A novel SI based approach which employs bee optimization algorithm was developed.
- The proposed approach uses ABC algorithm to solve virtual network embedding problem which effectively embed VN requests on multiple substrate networks.
- In the current work, the proposed ABC based virtual network embedding algorithm achieves global or near global solutions for embedding VN requests among multiple substrate networks managed by multiple InPs.


Thank You