Green Cloud Virtual Network Provisioning Based Ant Colony Optimization

Xiaolin CHANG, Bin WANG, Jiqiang LIU, Wenbo WANG
School of Computer and Information Technology
Beijing Jiaotong University, P.R. China
e-mail: { xlchang,08283021,jqliu, 12120458}@bjtu.edu.cn

Jogesh K. Muppala
Department of Computer Engineering
Hong Kong University of Science and Technology
e-mail: muppala@cse.ust.hk

ABSTRACT
Network virtualization is being regarded as a promising technology to create an ecosystem for cloud computing applications. One critical issue in network virtualization technology is power-efficient virtual network embedding (PE-VNE), which deals with the physical resource allocation to virtual nodes and links of a virtual network while minimizing the energy consumption in the cloud data center. When the node and link constraints (including CPU, memory, network bandwidth, and network delay) are both taken into account, the VN embedding problem is NP-hard, even in the offline case. This paper aims to investigate the ability of the Ant-Colony-Optimization (ACO) technique in handling PE-VNE problem. We propose an ACO-based heuristic PE-VNE algorithm, called E-ACO. E-ACO minimizes the energy consumption by considering the embedding power consumption in the node mapping phase and by making an implicit coordination between the node and link mapping phases. Extensive simulations are conducted to evaluate the performance of the proposed algorithm and investigate different energy-aware link embedding algorithms on the ability of E-ACO.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search–*Heuristic methods*; I.2.4 [**Computer Communication Networks**]: Distributed Systems–*Distributed applications*

General Terms
Algorithms, Experimentation, Performance, Design

Keywords
Cloud data center, Virtual network embedding, Ant colony optimization, Optimization, Energy consumption, Mixed integer programming.

1. INTRODUCTION
Network virtualization is being regarded as a promising technology to create an ecosystem for cloud computing applications[1][3][4]. Some network virtualization approaches such as in [2][3][4] have been proposed as a basis for enforcing QoS policies and performance guarantees in cloud data centers. One of the key issues in the network virtualization technology is the *virtual network embedding* (VNE), which deals with the mapping/embedding of VN requests onto specific physical nodes and paths/flows of the physical network. Significant studies have been carried out on a set of variants of cost-aware VNE problems (see [8]-[9] and references therein). These cost-aware VNE algorithms were designed for the scenarios of peak load instead of for the scenarios of current traffic load. They aimed to improve the Cloud Infrastructure Provider (InP) long-term revenue, expressed in form of VNs' resource demands including CPU and link bandwidth and so on.

The current link utilization in backbone networks of large Internet Service Providers (ISPs) is estimated at 30%-40% [6]. ISP's overprovisioning link bandwidth aims for peak traffic load or link failure. Such overprovision may lead to low energy efficiency. ISPs are now facing the challenge of minimizing the power/energy consumption [7]. In last years, efforts in the research community and the industries have been devoted to reduce the energy expenditure at both of the server and network levels. More recently, motivated by the fact that turning/powering/switching off unused devices can save energy in the context of the Internet [10], some efforts such as in [11]- [14] were devoted to deal with the power-efficient VNE (PE-VNE) problem. When the node and link constraints (including CPU, memory, network bandwidth, and network delay) are both taken into account, the VN embedding problem is NP-hard, even in the offline case [5]. Some authors such as in [11][13] formulated the VNE problem as mixed-integer programs and designed the CPLEX[25]/GLPK[24]-based exact PE-VNE algorithms. As the size of the network increases, the PE-VNE problem becomes more difficult due to the amount of nodes and links in both physical and virtual networks.

Meta-heuristic techniques have been applied successfully to design heuristic cost-aware VNE algorithms. The simulation results in both [15] and [16] demonstrated the better performance of metaheuristics-based VNE algorithms in terms of the InP's long-term revenue and embedding cost. This paper aims to investigate the ability of Ant-Colony-Optimization (ACO) technique in handling the PE-VNE problem and investigate the impacts of different energy-aware link embedding algorithms on the performance of the ACO-based PE-VNE algorithm in the following context (denoted as no-cycle undirected context in the rest of the paper):

- The physical network is modeled as a weighted undirected graph.
- There exists no cycle in the physical path/flow which hosts a virtual link. The cycle existence means the physical resource waste.

We summarize the main contributions in our work as follows:

- We propose a simple but effective ACO-based heuristic PE-VNE algorithm, called E-ACO. The key component in E-ACO is the adaptive node mapping algorithm. E-ACO can seamlessly apply any link mapping algorithm. It minimizes the VNE energy consumption without sacrificing InP's long-term revenue, compared to the existing heuristic PE-VNE algorithms. That is, E-ACO not only performs well in the peak-load situations in terms of InP's long-term revenue, but also performs well in the light-load situations in terms of InP's energy consumption. These features are achieved by considering power consumption in the node mapping phase and making an implicit coordination between the node and link mapping phases.
- We propose two CPLEX-based exact energy-aware link mapping (ELM) algorithms in the no-cycle undirected context. One is for the *path-unsplittable* situations and the other is for *path-splittable* situations. A virtual link can be either *path-unsplittable* or *path-splittable*. Path-splitting (allowing virtual link over multiple substrate paths) is beneficial to the bandwidth efficient utilization and also is beneficial to increase robustness to physical failures. However, it causes additional troubles, such as network management and dealing with out-of-order packets. For description convenience, we denote as *path-unsplittable* PE-VNE problem and *path-splittable* PE-VNE problem, respectively. We formulate the ELM problem in each kind of situations as a mixed integer problem in each kind of situations as a mixed integer program (MIP), denoted as US-ELM-MIP and S-ELM-MIP, respectively. To the best of our knowledge, we are the first to make MIP formulations in the no-cycle undirected context.
- We carry out extensive simulations to evaluate the performance of E-ACO. The simulation results in the static and dynamic networks demonstrate that the proposed algorithms perform better, compared to the existing heuristic PE-VNE algorithms.

The rest of the paper is organized as follows. Section 2 presents some related work. In Section 3 we first present the PE-VNE problem. Then we present the formulations of US-ELM-MIP and S-ELM-MIP. Section 4 describes the E-ACO algorithm and its performance is evaluated in Section 5. Section 6 presents the conclusions.

2. RELATED WORK

This section focuses on the existing work in relation to the PE-VNE problem and meta-heuristic technique based VNE algorithms.

PE-VNE algorithms. The authors in [13] and [14], respectively, proposed a generalized power consumption (GPC) model, which captured the fixed and variable power consumption of physical nodes and network equipments. The power consumption models considered in [11][12] were both a special case of the GPC model [14]. The authors in [11] formulated the *path-unsplittable* PE-VNE problem as a MIP, which was unsolvable by GPLK [24]. They then proposed a two-stage algorithm, denoted as EA-VNE in the rest of this paper. The authors in [12] proposed a CLPEX-based exact algorithm to handle the *path-splittable* PE-VNE problem. The authors in [11][12][13] all formulated the PE-VNE problem as a MIP in the situations where the physical network is modeled as a weighted directed graph. The authors in [14] also considered the *path-splittable* PE-VNE problem but formulated a MIP in the situation where the physical network is modeled as a weighted undirected graph. Through linear relaxation, the authors in [14] proposed a two-stage heuristic algorithm. Note that the MIP formulation proposed in [14] allows a cycle to exist in a physical flow hosting a virtual link.

Metaheuristics-based VNE algorithms. Metaheuristic techniques, see [16] and the references therein, have been explored to handle the cost-aware VNE problem. The authors in [18] proposed an ACO-based cost-aware VNE algorithm, which has a low performance [19]. E-ACO proposed in this paper applies a novel node mapping algorithm, which maps virtual nodes in an adaptive way in the dynamic network. E-ACO then can perform well in the scenarios of both on-peak and off-peak loads.

3. PE-VNE PROBLEM AND MIP FORMULATIONS OF ENERGY-AWARE LINK MAPPLING ALGORITHMS

This section first describes the network model and the PE-VNE problem. Then we describe the S-ELM-MIP and US-ELM-MIP formulations.

3.1 Network Model and Problem Description

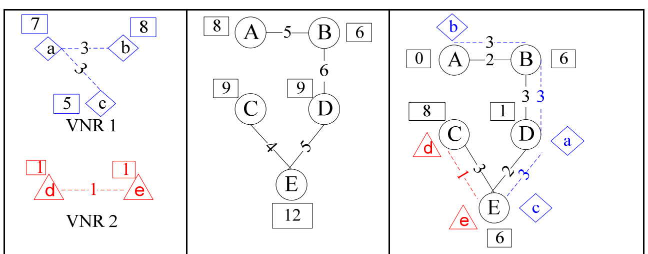
Both the physical network and the virtual network are modeled as a weighted undirected graph and are denoted by $G_S = (N^S, E^S)$ and $G_V = \left(N^V, E^V\right)$, respectively. Here N^S / N^V is the set of physical/virtual nodes and E^S / E^V is the set of physical/virtual links. The system resources of a physical node include memory, processing power, storage space and so on. Without loss of generality, this paper only considers the processing power. That is, each physical node $n^S \in N^S$ is associated with CPU resources $c(n^S)$ and geographical location $l(n^S)$. All the work presented in this paper can be applied directly to the situations where physical nodes have other resource demands besides CPU demand. When other physical resources are to be considered, we need to only add some constraints to the corresponding algorithms.

Each physical edge $e^S(v,w) \in E^S$ between physical nodes (v,w) is associated with bandwidth capacity. All the physical resources (i.e. bandwidth and CPU) in G_S are limited. Usually, a virtual node's (denoted as n^V) QoS (Quality of Service) requirements include the CPU demand $c(n^V)$ and a preferred value

d^{n^V} expressing how far this virtual node can be placed from the specified location $l(n^V)$. The QoS requirements of a virtual link $e^V(v,w)\in E^V$ between virtual nodes (v,w) include the bandwidth requirement $b(e^V)$. Embedding for a VN request is defined as a mapping from G_V to G_S with the following constraints:

- Each virtual node is mapped to a physical node in a one-to-one manner, and the virtual node QoS requirements are satisfied.
- Each virtual link $e^V(v,w)$ is mapped to a physical path (an unsplittable model) or a flow (a splittable model) in G_S between physical nodes which host v and w, respectively. At each physical edge $e^S(v,w)\in E^S$, the total physical bandwidth consumed by G_V must be less than the available bandwidth of this physical edge.

The available CPU capacity $A_N(n^S)$ of a physical node n^S is defined as $A_N(n^S) = c(n^S) - \sum_{\forall n^V\in\Theta(n^S)} c(n^V)$. Here $\Theta(n^S)$ denotes the set of the virtual nodes, which are hosted on the physical node n^S. The available bandwidth $A_E(e^S(v,w))$ of a physical edge $e^S(v,w)$ is defined as $b(e^S(v,w))$ minus the total bandwidth used by virtual links that pass through $e^S(v,w)$. Figure 1(c) illustrates the embedding of two VN requests (in Figure 1(a)) on the physical network (in Figure 1(b)).



(a)Two VN requests | (b)Physical network | (c) Physical network with two embedded VNs

Figure 1. An example of two VN requests' embedding

Table 1 gives the definition of the variables, used in the following. As in [14], we assume that:

① Whenever a physical edge is turned off, energy is saved in the pair of interfaces/ports on its endpoints (also turned off).

② A physical node is turned off only when all the network ports are turned off, namely all the physical edges connected to this node are turned off.

③ If a physical node is unused, then it is simply turned off and then has a zero power consumption.

Assume that the k^{th} VN (denoted as G_{Vk}) is served by InP from time $t_{G_{Vk}}^{start}$ and leaves at time $t_{G_{Vk}}^{start} + T_{G_{Vk}}^{d}$. Each virtual link $e_i^{Vk}(s_i^{Vk}, t_i^{Vk})$ is considered as a commodity from source physical node s_{ki} to destination physical node t_{ki}, $\left(\forall i\in\{1\ \ldots\ |E^{Vk}|\},\ s_{ki}, t_{ki}\in N^S,\ s_i^{Vk}, t_i^{Vk}\in N^{Vk}\right)$. Then, the energy consumption $E_u(G_{Vk})$ for serving the k^{th} G_{Vk} in unit time is defined as

$$\sum_{v\in N^S} P_1^v(y_v-\lambda_v) + \sum_{e^S(v,w)\in E^S} P_2^{e^S}(x_{vw}-\eta_{vw}) + \sum_{i\in\{1\ \ldots\ |E^{Vk}|\}} \left(P_3^{s_{ki}}\cdot c(s_i^{Vk}) + P_3^{t_{ki}}\cdot c(t_i^{Vk})\right) \quad (1)$$

Here, P_1^v, $P_2^{e^S}$ and P_3^u are constants. P_3^u represents the power consumption of unit CPU utilization of physical node u in unit time.

Table 1 Variable definition

Term	Definition
λ_v	A binary variable. $v\in N^S$. Denote whether a physical node is active or not before embedding G_{Vk}: 1 means active; otherwise 0.
η_{vw}	A binary variable. Denote whether a physical edge is active or not before embedding G_{Vk}: 1 represents active; otherwise 0. $e^s(v,w)\in E^s$
f_{vw}^{ki}	A flow variable denoting the total amount of loads on the physical edge $e^s(v,w)$ for the i^{th} virtual link of G_{Vk}. $i\in\{1\ \ldots\ \lvert E^{Vk}\rvert\}$.
g_{vw}^{ki}	A binary variable. Its value is 1 if $e^s(v,w)$ is on the path hosting the i^{th} virtual link of G_{Vk}. Otherwise, it is set to 0. $i\in\{1\ \ldots\ \lvert E^{Vk}\rvert\}$.
z_v	A binary variable. $v\in N^S$. Its value is 1 if physical node v is on the paths/flows hosting virtual links. Otherwise, it is set to 0.
x_{vw}	A binary variable. $v,w\in N^S$. After embedding G_{Vk}, its value is 1 if $e^S(v,w)$ is active; otherwise, it is set to 0.
y_v	A binary variable. $v\in N^S$. After embedding G_{Vk}, its value is 1 if the physical node v is active; otherwise it is set to 0.

Assume there are K active VNs in the interval T. For description convenience, we set $T=t_2-t_1$. The InP's average energy consumption over T is defined in Equation (2).

$$E_{avg}^{InP} = \frac{\sum_{k\in K}\sum_{t=0}^{T} E_u(G_{Vk}) L_{G_{Vk}}}{T} \quad (2)$$

where

$$L_{G_{Vk}} = \begin{cases} T, & t_{G_{Vk}}^{start} < t_1 \text{ and } T < T_{G_{Vk}}^{d} \\ T - (t_{G_{Vk}}^{start} - t_1), & t_{G_{Vk}}^{start} \ge t_1 \text{ and } T < T_{G_{Vk}}^{d} \\ T_{G_{Vk}}^{d} - (t_{G_{Vk}}^{start} - t_1), & t_{G_{Vk}}^{start} \ge t_1 \text{ and } T \ge T_{G_{Vk}}^{d} \\ T_{G_{Vk}}^{d} + (t_{G_{Vk}}^{start} - t_1), & t_{G_{Vk}}^{start} < t_1 \text{ and } T \ge T_{G_{Vk}}^{d} \end{cases} \quad (3)$$

Thus, for VNR1 in Figure 1(a), if {a→D, b→A, c→E, d→D, e→E }, energy consumption of embedding two VNs can be reduced compared to the embedding of Figure 1(c).

3.2 FORMULATIONS OF S-ELM-MIP and US-ELM-MIP

The S-ELM-MIP can be formulated as a linear integer programs with integer constraints in the following manner.

S-ELM-MIP

Objective:

minimize

$$\omega \cdot E_u(G_{Vk}) + \sum_{v\in N^S} z_v \quad (4)$$

Subject to:

Capacity constraints:

$$\sum_{\forall i\in\{1\ \dots\ |E^{Vk}|\}} \left(f_{vw}^{ki} + f_{wv}^{ki}\right) \le A_E(v,w)x_{vw}, \forall v,w\in N^S \quad (5)$$

Flow related constraints:

$$\sum_{w\in N^S} f_{vw}^{ki} - \sum_{w\in N^S} f_{wv}^{ki} = 0, \forall i\in\{1\ \dots\ |E^{Vk}|\}, \quad \forall v\in N^S \setminus \{s_{ki}, t_{ki}\} \quad (6)$$

$$\sum_{w\in N^S} f_{s_{ki}w}^{ki} - \sum_{w\in N^S} f_{ws_{ki}}^{ki} = b(e_i^{Vk}), \forall i\in\{1\ \dots\ |E^{Vk}|\} \quad (7)$$

$$\sum_{w\in N^S} f_{t_{ki}w}^{ki} - \sum_{w\in N^S} f_{wt_{ki}}^{ki} = -b(e_i^{Vk}), \forall i\in\{1\ \dots\ |E^{Vk}|\} \quad (8)$$

Binary constraints:

$$x_{vw} = x_{wv}, \forall v,w\in N^S \quad (9)$$

$$x_{vw} \le y_v, \forall v,w\in N^S \quad (10)$$

$$y_w \le \sum_{v\in N^S} x_{vw}, \forall v,w\in N^S \quad (11)$$

$$\lambda_v \le y_v, \forall v\in N^S \quad (12)$$

$$\eta_{vw} \le x_{vw}, \forall v,w\in N^S \quad (13)$$

$$\left(f_{vw}^{ki} == 0\right) + \left(f_{wv}^{ki} == 0\right) \ge 1, \forall i\in\{1\ \dots\ |E^{Vk}|\}, \quad \forall v,w\in N^S \quad (14)$$

$$z_v \ge \left(\left(\sum_{\forall i\in\{1\ \dots\ |E^{Vk}|\}} \sum_{w\in N^S} (f_{vw}^{ki} + f_{wv}^{ki})\right) \ge 0\right), \quad \forall v\in N^S \quad (15)$$

$$y_v = 1,\ \forall v\in\{s_{ki}, t_{ki}\}, \forall i\in\{1\ \dots\ |E^{Vk}|\}, \quad (16)$$

Domain constraints:

$$f_{vw}^{ki} \ge 0, \forall v,w\in N^S, \forall i\in\{1\ \dots\ |E^{Vk}|\} \quad (17)$$

$$x_{vw} \in \{0,1\}, \forall v,w\in N^S \quad (18)$$

$$y_v \in \{0,1\}, \forall v\in N^S \quad (19)$$

$$z_v \in \{0,1\}, \forall v\in N^S \quad (20)$$

Constraint set (4) enforce the capacity bounds of physical edges. Constraint sets (7-9) refer to the flow conservation conditions. Constraint sets (6,10) state the features of a undirected graph. Constraint sets (11-12) state that a physical node is active if and only if there exists at least a physical edge connecting this physical node is active. Constraint sets (13-14) state that if a physical node/edge is active before embedding G_{Vk}, it must be active after embedding G_{Vk}. Constraint sets (5,15,16) make sure that there is no circle in any path hosting a virtual link.

The mixed linear integer program of US-ELM-MIP is obtained by making some modifications to that of S-ELM-MIP: using Equations (22-25) to replace Equations (6-9) and using Equation (26) to replace Equation(18).

$$\sum_i \left(b(e_i^{Vk})\left(g_{vw}^{ki} + g_{wv}^{ki}\right)\right) \le A_E(v,w)x_{vw}, \forall v,w\in N^S \quad (21)$$

$$\sum_{w\in N^S} g_{vw}^{ki} - \sum_{w\in N^S} g_{wv}^{ki} = 0, \forall i\in\{1\ \dots\ |E^{Vk}|\}, \quad \forall v\in N^S \setminus \{s_{ki}, t_{ki}\} \quad (22)$$

$$\sum_{w\in N^S} g_{s_{ki}w}^{ki} - \sum_{w\in N^S} g_{ws_{ki}}^{ki} = 1, \forall i\in\{1\ \dots\ |E^{Vk}|\} \quad (23)$$

$$\sum_{w\in N^S} g_{t_{ki}w}^{ki} - \sum_{w\in N^S} g_{wt_{ki}}^{ki} = -1, \forall i\in\{1\ \dots\ |E^{Vk}|\} \quad (24)$$

$$g_{vw}^{ki} \ge 0, \forall v,w\in N^S, \forall i\in\{1\ \dots\ |E^{Vk}|\} \quad (25)$$

4. E-ACO ALGORITHM

In this section, we propose an ACO-based approach to this PE-VNE problem. The pseudo-code is shown in Algorithm1, which uses Algorithm2 to map virtual nodes. More details about E-ACO are given as follows.

Algorithm 1(E-ACO) : input=(G_V , G_S)

s1: **Generate the initial population of ants.** Initialize a certain number of ants by using Algorithm2 and check the feasibility of each ant's virtual node mapping.

s2: **Compute *pBest* and initialize *gBest***. Compute $f(x)$ of each ant. Set *pBest* and *gBest* according to of these ants' *f(x)*.

s3: **Update the pheromone trail**. Before reiterating the process from iteration t to $t + 1$, the pheromone trails of all physical nodes for each virtual node are first evaporated according to Equation (27).

$$\tau_{uv}(t+1) = \rho \cdot \tau_{uv}(t) \tag{27}$$

Here $\rho \in (0,1]$. Then, update the pheromone trail of each physical node (denoted as v) which hosts a virtual node (denoted as u) in *pBest* according to Equation (28).

$$\tau_{uv}(t+1) = \tau_{uv}(t+1) + \frac{\varphi_1}{pBw} \cdot \frac{\varphi_2}{pEn + \varphi_2} \tag{28}$$

Here *pBw* represents the total physical bandwidth used in *pBest*. *pEn* is defined in Equation (4), used in *pBest*. φ_1 and φ_2 are both constant parameters.

s4: **Evolution.** Update each ant solution by using Algorithm2 and check the solution's feasibility. We calculate the *fitness* function value of each ant and then update *pBest*. If $f(gBest) > f(pBest)$ then $gBest = pBest$.

s5 **Repeat s3-s4** until the maximum number of iterations is reached. If $f(gBest)$ is +∞, output there is no feasible solution. Otherwise, *gBest* is returned as the VNE solution.

Each ant i is associated with a solution, which is a $\left|N^V\right|$-length integer vector containing the virtual node mapping result, donated by $H_i = \left(h_i^1, h_i^2, \cdots, h_i^{\left|N^V\right|}\right)$. Here h_i^k denotes the physical node which hosts the k^{th} virtual node in the i^{th} ant. Whenever H_i is updated, its feasibility must be checked by using a link mapping algorithm to map all the virtual links in the VN. H_i is called as being feasible if all the virtual links finds their hosted physical paths/flows; otherwise H_i is infeasible. The *fitness* function $f(x)$ is defined Equation (4). If H_i is infeasible, the $f(x)$ value of this ant is set to +∞. Variable *pBest* is defined to denote the local best solution, namely the solution with the best *fitness* value in the current iteration. Variable *gBest* is defined to denote the best value obtained so far.

Each physical node j is associated with a pheromone trail for each virtual node i, denoted as τ_{ij} . τ_{ij} represents the desirability of assigning virtual node i to physical node j and is updated by Equation (27,28) in Algorithm1. The initial value of each pheromone trail is set to a large value (we set to 100 in our experiments) in order to increase the exploration space of solutions during the first iteration. At each iteration, every ant first updates the pheromone trail values in two steps (see Step 3 of Algorithm1): (i) Evaporate τ_{ij} of all substrate nodes. (ii) Reinforce τ_{ij} of the substrate nodes which contribute to the building of the local best solution (*pBest*).

Algorithm2 : input=(G_V , G_S)

s1: Set all physical and virtual nodes *untouched*. Calculate $NR(u)$, defined in Equation (29), for each virtual node *u*. *L*(*u*) denotes the virtual link set. One endpoint of each link in *L*(*u*) must be *u*.

$$NR(u) = \xi \frac{c(u)}{\sum_{w \in N^V} c(w)} + (1-\xi) \frac{\sum_{l \in L(u)} b(l)}{\sum_{w \in N^V} \sum_{l \in L(u)} b(l)} \tag{29}$$

s2: Enqueue all virtual nodes into a priority queue *PQ* in the descending order of $NR(u)$.

s3: Dequeue the virtual node (denoted as *u*) with the largest $NR(u)$. Construct a physical node list *CL*(*u*) for *u*. Each physical node in *CL*(*u*) must be *untouched* and satisfy the resource constraints (the rest CPU of physical node is larger than that of the virtual node, and the total outgoing available bandwidth resources of the physical node is larger than the total outgoing bandwidth demand of the virtual node *u*) of virtual node *u*. $NR(v)$ of each physical node in *CL*(*u*) is defined in Equation (30).

$$NR(v) = \xi \frac{A_N(v)}{\sum_{w \in CL(u)} A_N(w)} + (1-\xi) \frac{\sum_{l \in L(v)} A_E(l)}{\sum_{w \in CL(u)} \sum_{l \in L(w)} A_E(l)} \tag{30}$$

A physical node (denoted as *v*) from *CL*(*u*) is selected to host *u* with the probability p_{uv} ,defined in Equation (31)

$$p_{uv} = \frac{\left(\tau_{uv}\right)^{\alpha_1} \left(NR(v)\right)^{\alpha_2}}{\sum_{w \in CL(u)} \left(\left(\tau_{uw}\right)^{\alpha_2} \left(NR(w)\right)^{\alpha_2}\right)} \tag{31}$$

Here α_1 and α_2 are parameters that determine the relative importance of pheromone trail and $NR(u)$. When a physical node *v* is selected to host virtual node *u*, then both *u* and *v* are set to be *touched*.

s4: Repeat Step 3 until *PQ* is empty.

Every ant explores new solutions according to the new pheromone trails. The key idea behind E-ACO's power-aware node mapping is this pheromone tail updating method. In the peak-load situations, most of the physical nodes and edges are active and then *pEn* is zero or near zero. In the light-load situations, *pEn* affects more pheromone trail.

4.1 Time Complexity Analysis

Algorithm2 can be computed in polynomial time in terms of $|N^{Vk}|$, $|N^S|$, $|E^{Vk}|$ and $|E^S|$. Thus, if the link embedding algorithm is polynomial-time, E-ACO is a polynomial-time algorithm.

5. PERFORMANCE EVALUATION

In this section, we first describe the simulation environment and the algorithms to be compared. Then we present the simulation results.

5.1 Compared Algorithms

The performance of cost-aware VNE algorithms has been compared with energy-aware VNE algorithms in [11][12]. The CPLEX-based exact PE-VNE algorithms in both [12] and [13] were designed for the situations where the physical network is modeled as a directed graph. In addition, the acceptance ratio of the PE-VNE algorithm proposed in [14] is very low (in our unpublished work). Thus, this section only compares the performance of E-ACO and EA-VNE under the following three energy-aware link mapping algorithms:

- **SNP** algorithm. It is proposed in [14]. The key idea is getting *k-shortest* paths and on these paths searching for a path with the smallest number of active nodes which are inactive before embedding the VN.
- **S-CPLEX**. Use CPLEX solver to solve the S-ELM-MIP
- **US-CPLEX**. Use CPLEX solver to solve the US-ELM-MIP.

The parameters in E-ACO are set as follows. φ_1=5000. φ_2=500. α_1=α_2=1. ξ=0.5. The initial value of each τ_{ij} is set to 100. The population size is set to 5 and the number of iterations is set to 10 in all experiments. We set all P_1^u to 160W and set all $P_2^{e^S}$ to 1.1W, based on the experiment results in [20]-[22]. The parameters in EA-VNE are set as in [11]. ω in Equation (5) is set to $|N^S|+1$.

5.2 Simulation Environment

As in the existing PE-VNE literature, we use synthetic network topologies to evaluate the proposed algorithm. The substrate network and virtual networks are generated by using the GT-ITM tool [23]. The substrate network is configured to have 50 nodes in (25×25) grid, which are randomly connected with probability 0.5. Both physical node CPU and edge bandwidth capacities follow a uniform distribution from 50 to 100 units. We do two kinds of experiments, each having different VN configuration and performance metrics, described as follows.

Experiments of static VNRs (Experiment1). This kind of experiments investigates the impact of consolidation in the scenarios where 20 VNRs arrive and leave together. Each VN has 5 virtual nodes and the virtual node connectivity probability is 0.5. Both all virtual node' CPU and virtual link's bandwidth requirements are distributed uniformly in (0, β). We do simulations by varying β from 5 to 80, respectively. Under our simulation configurations, each algorithm can successfully embed these 20 VNs when β is less than 50. When β is not less than 50, both EA-VNE and E-ACO can embed successfully 20VNs only under S-CPLEX link mapping algorithm; the acceptance ration of E-ACO-US-CPLEX and E-ACO-SNP are similar, but less than that of EA-VNE-US-CPLEX and EA-VNE-SNP. The metrics considered include

(i) *Num of active nodes* and *Num of active edges*, which measure the number of active physical nodes and active physical edges, respectively, after these 20 VNRs are embedded.

(ii) *Power consumption*. Defined in Equation (4).

All simulation results are calculated at a confidence level of 95%. The confidence intervals are small and then are not displayed in the Figure 2-Figure 4, which depict the simulation results.

Experiments of on-line VNRs (Experiment2). Due to the large amount time in solving US-ELM-MIP and S-ELM-MIP, we only compare E-ACO-SPN and EA-VNE-SPN in dynamic scenarios, in which the VNs come and leave dynamically. The number of virtual nodes in a VNR is chosen uniformly between 2 to 5 and the virtual node connectivity probability is 0.5. CPU and bandwidth requirements are distributed uniformly in (0, 20) and (0,50) respectively. The VNR arrival rate is 4 VNRs per 100 time units. The lifetimes of the VNRs follow an exponential distribution. We do simulations by setting the VN lifetime to 500 (simulating a dynamic network with light load) and 1000 (simulating a dynamic network with high load) time units. Each simulation lasts 50000 time units and is repeated 10 times. The average value of these repetitions is presented as the simulation results in the corresponding figures. The metrics considered include

(i) *Average Num of active nodes* and *Average Num of active edges*, which measure the number of active physical nodes and the number of active physical edges of an interval *T* (set to 1000 time units), respectively.

(ii) *Acceptance ratio*, which measures the percentage of total VNRs accepted by an algorithm.

Figure 5-Figure 8 illustrate the results of *Average Num of active nodes* and *Average Num of active edges*. Table 2 depicts *Acceptance ratio*.

5.3 Evaluation Results

From the simulation results, we observe that:

- *Combining power consumption information in the node mapping and making coordination between the power-aware node and power-aware link mapping phases can effectively reduce the VN embedding power consumption and increase the VN acceptance ratio*. Figure2-Figure.8 show that EA-VNE-* performs worst in term of power consumption. Here, * denote S-CPLEX, US-CPLEX and SNP. The reason is that E-ACO-* considers power consumption in the node mapping phase. Then, when β is less than 50, E-ACO-* perform better in terms of *power consumption*. We also observe that E-ACO-US-CPLEX and E-ACO-U-CPLEX may not work better than E-ACO-SNP in the light-load situations (β is less than 50). This indicates the importance of combining power consumption information in the node mapping phase. Since the node mapping of EA-VNE-* ignores the power consumption information, the effect of the link mapping algorithm on the algorithm performance is very obvious, illustrated in Figure 4.

- *Num of active physical nodes/edges* or *Power consumption* can not completely be used to evaluate the performance of an algorithm. In ***Experiment1***, when β is 80, EA-VNE-SNP works better than E-ACO-SNP in the three metrics, namely *Num of active physical nodes/edges* or *Power consumption*. However, in the scenarios of β=80, EA-VNE-SNP only successfully embeds about 10 VNs. But E-ACO-SNP successfully embeds about 13 VNs.

  Note that the low power consumption of EA-VNE-US-CPLEX when β is more than 50 is due to its low acceptance ratio.

- *Running time.* We compute the time of each algorithm in embedding the first VN in ***Experiment1***. EA-VNE works about 8-9 times faster about than E-ACO under the same link mapping algorithm.

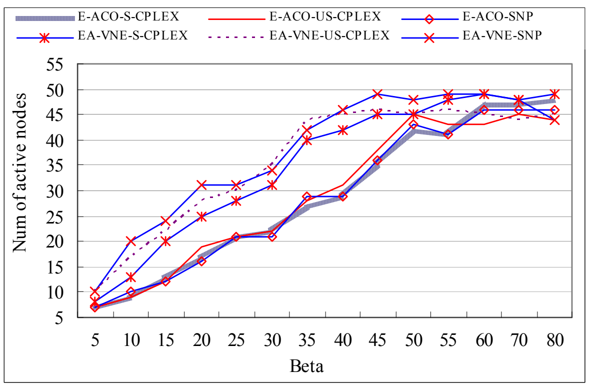


Figure 2 Effect of β on active physical node number

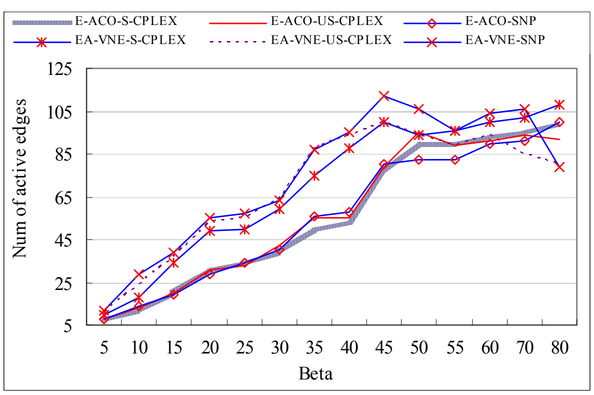


Figure 3 Effect of β on active physical edge number

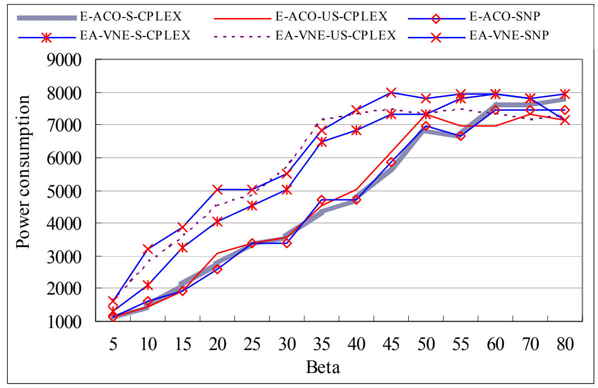


Figure 4 Effect of β on energy consumption

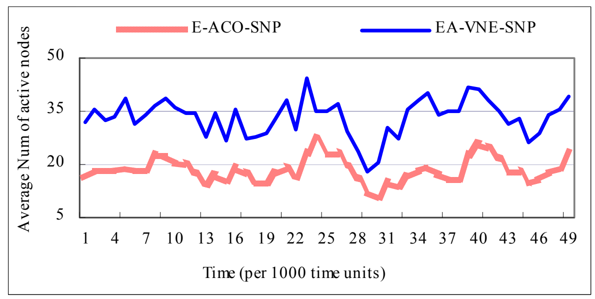


Figure 5 Average number of active nodes when VN average lifetime is 500 time units

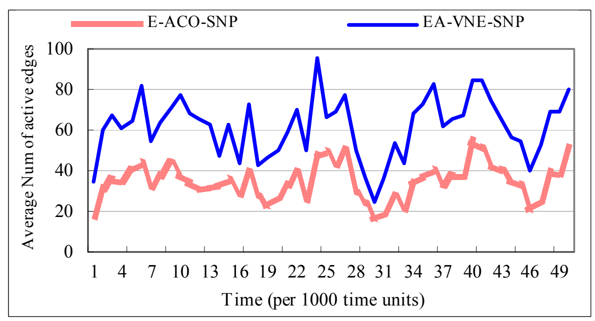


Figure 6 Average number of active edges when VN average lifetime is 500 time units

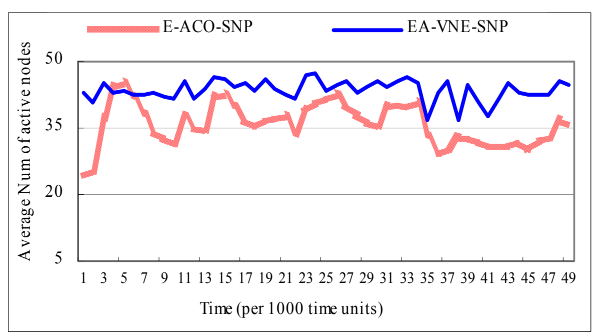


Figure 7 Average number of active nodes when VN average lifetime is 1000 time units

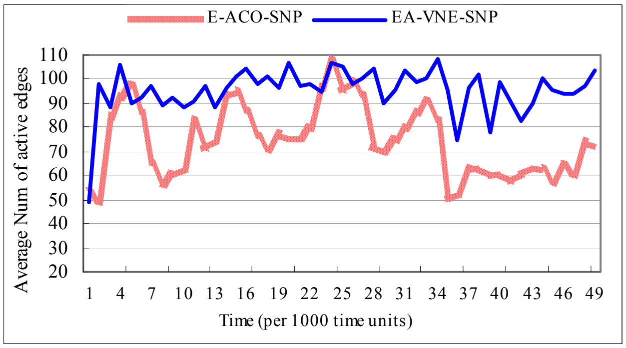


Figure8 Average number of active edges when VN average lifetime is 1000 time units

Table 2 Acceptance Ratio in *Experiment2*

Acceptance Ratio / Lifetime	E-ACO-SNP	EA-VNE-SNP
500 time units	100%	100%
1000 time units	99.5%	82.6%

6. CONCLUSIONS

This paper proposes a simple but effective ant-colony-optimization-based heuristic PE-VNE algorithm to trade off solution optimality with computation time. The proposed algorithm minimizes the energy consumption by considering the embedding power consumption in the node mapping phase and by making an implicit coordination between the node and link mapping phases. Extensive simulation results validate the E-ACO ability in minimizing the energy consumption without sacrificing the InP's long-term revenue, compared to the existing PE-VNE heuristic algorithms.

7. ACKNOWLEDGMENT

The work described in this paper has been supported in part by Beijing Municipal Natural Science Foundation (No. 4123103), Program for New Century Excellent Talents in University (NCET-11-0565), Program for Innovative Research Team in University of Ministry of Education of China (IRT201206).

8. REFERENCES

[1] http://nicira.com/

[2] T. Benson, A. Akella, A. Shaikh, and S. Sahu, "CloudNaaS: A Cloud Networking Platform for Enterprise Applications," In Proceedings of the ACM Symposium on Cloud Computing (SOCC), 2011.

[3] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards Predictable Data Center Networks," In Proc. of the Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM), 2011.

[4] T. Zou, R. Le Bras, M. Vaz Salles, A.lan Demers, and J. Gehrke, "ClouDiA: A Deployment Advisor for Public Clouds," In International Conference on Very Large Data Bases(VLDB) 2013.

[5] Y. Zhu, and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," In Proc. IEEE INFOCOM, 2006.

[6] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks:reducing energy consumption by shutting off cables in bundled links," In Proc. ACM SIGCOMM workshop on Green networking, 2010.

[7] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP Network Energy Cost: Formulation and Solutions," In IEEE/ACM Transactions on Networking, 2011.

[8] A. Belbekkouche, M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," In IEEE Communications Surveys Tutorials, (99):1–15, 2012.

[9] S. Zhang, Z. Qian, J. Wu, and S. Lu, "An opportunistic resource sharing and topology-aware mapping framework for virtual networks," In Proc. IEEE INFOCOM, 2012.

[10] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, "Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage," In Proc. NSDI, 2009.

[11] Z.B. Zhang, X. Cheng, Y.W. Wang, Y. Luo, and J. Wang, "Energy-aware virtual network embedding through consolidation, " In Proc. IEEE INFOCOM WS - CCSES, 2012.

[12] J.F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H.D. Meer, "Energy Efficient Virtual Network Embedding," In IEEE Communications Letters, 2012.

[13] E. Rodriguez, G. Alkmim, D.M. Batista, and N. L. Fonseca, "Green Virtualized Networks," In Proc. IEEE ICC 2012.

[14] B. Wang, X.L. Chang, J.Q. Liu, and Jogesh K. Muppala, "Reducing Power Consumption in Embedding Virtual Infrastructures, " In Proc. IEEE GlobeCom ManSEC, 2012.

[15] X. Cheng, S. Su, Z.B. Zhang, K. Shuang, F.C. Yang, Y. Luo, and J. Wang, "Virtual Network Embedding Through Topology Awareness and Optimization," In Elsevier Journal of Computer Newtorks, 2011.

[16] X.M Mi, X.L. Chang, and J.Q. Liu, "Embedding Virtual Infrastructure Based on Genetic Algorithm," In Proc. IEEE PDACT, 2012.

[17] Andreas Bley: Routing and Capacity Optimization for IP Networks. OR 2007: 9-16

[18] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "VNE-AC: Virtual Network Embedding Algorithm based on Ant Colony Metaheuristic," In Proc. IEEE ICC, 2011.

[19] N.Aitsaadi, G. Pujolle, and H. Zimmermann, "VNR Algorithm: A Greedy Approach for Virtual Networks Reconfigurations," In Proc. IEEE GLOBECOM, 2011.

[20] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," In Proc.ACM ISCA, 2007.

[21] R. Sohan, A. Rice, K.J. Mansley, and A.W. Moore, "Characterizing 10 Gbps Network Interface Energy Consumption," In The 35th Annual IEEE Conference on Local Computer Networks (LCN) Short Papers.

[22] V. Sivaraman, A. Vishwanath, Z. Zhao, and C. Russell, "Profiling per-packet and per-byte energy consumption in the netfpga gigabit router," In Proc. IEEE INFOCOM WKSHPS, 2011.

[23] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an Internetwork," In Proc. IEEE INFOCOM, 1996.

[24] "GNU Linear Programming Kit," http://www.gnu.org/software/glpk

[25] ILOG CPLEX. http://www.ilog.com/products/cplex/.