ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Cost Effective Resource Mapping Based Requisition Partitioning Through Iterated Local Search

R. Hema, M. Siva Kumar, R. Devi

Department of CSE, Anna University, K.Ramakrishnan College of Technology, Trichy, Tamilnadu, India[1, 3]

Muthayammal College of Engineering, Rasipuram

Assistant Professor, Department of CSE, K.Ramakrishnan College of Technology, Trichy, Tamilnadu, India[2]

ABSTRACT--Network Virtualization Environment (NVE) affords the virtual resources across computers, server and datacenter that allow users to access resources. In Networked Cloud Environment such as the intra and inter cloud communication resources are allocated to users by resource mapping. The main objective of this paper is to overcome the Virtual Network Embedding problem (VNE) by using request partitioning and embedding approaches. In order to speed up the process and to reduce the embedding cost, these approaches are used. User request is partitioned and is given to the different Cloud Service Provider (CSP) for mapping the requested resources to users. The request partitioning is performed using Iterated Local Search. It facilitates the splitting of user request among eligible Cloud Service Provider. Resource mapping is performed in the embedded phase. This request partitioning and embedding approach allocates the cloud resources efficiently. Finally our proposed system confirm that the inter and intradomain resource provisioning and embedding cost effectively solve the VNE problem.

KEYWORDS—virtual network embedding, resource mapping, networked cloud mapping

I. INTRODUCTION

Network virtualization has gained considerable attention as it allows for deploying diverse network protocols and technologies customized for specific networked services and applications. In a network virtualization environment (NVE), the basic entity is a virtual network (VN), which is a logical topology, composed of virtual nodes and virtual links. The provisioning of a VN involves the mapping and embedding of virtual nodes onto physical ones and virtual links onto physical links or paths. Users of a cloud platform or a network infrastructure request their share of resources including CPU capacities, storage space, network bandwidth, etc., while the provider make their best effort to serve the requests, which are also known as virtual networks (VNs). The allocation of resources to VNs in such a virtualization environment is critical to both user's computation needs and the resource providers monetary gain. In brief, it introduces additional challenges related to 1) assigning the requested resources among the various physical resources belonging to different cloud service providers and 2) establishing the interconnection of these resources via appropriate inter cloud virtual network services. A major challenge in this respect is the VN embedding problem that deals with efficient mapping of virtual nodes and virtual links onto the substrate network resources. The VNE problem in various forms has been shown to be NP complete. A popular approach is to separate the node mapping problem from the link mapping problem. It is equivalent to the unsplittable multi-commodity flow problem when the virtual node mapping is given. To deal with the more general cases, several heuristics have been developed based on the reduction from various known NP-complete problems, such as subgraph isomorphism

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

detection and graph partitioning. In this paper, we explore inter domain resource mapping in a networked cloud environment via the introduction of a hierarchical framework. The key contributions of this work are as follows: Specifically, to deal with the inherent complexity and scalability of the problem under consideration, a request partitioning approach with the use of Iterated Local Search (ILS) meta heuristic is initially proposed and designed. In order to allow for a more realistic and comprehensive formulation of the corresponding resource mapping problem, we define the associated resource provisioning costs based on the availability of resources, namely the scarcity of the resource and the average utilization over a time window, in contrast to most of the existing works where either fixed or random costs are assumed. Detailed evaluations demonstrate that a great efficiency over a large number of VN requests and networked cloud sizes, with minimum computation time, can be obtained through the proposed framework. Then, the proposed algorithms for addressing the virtual resource mapping are described in detail. a thorough evaluation of the proposed overall solution on a simulated networked cloud environment is provided and critically compared against an exact request partitioning solution as well as a common intradomain VNE heuristic.

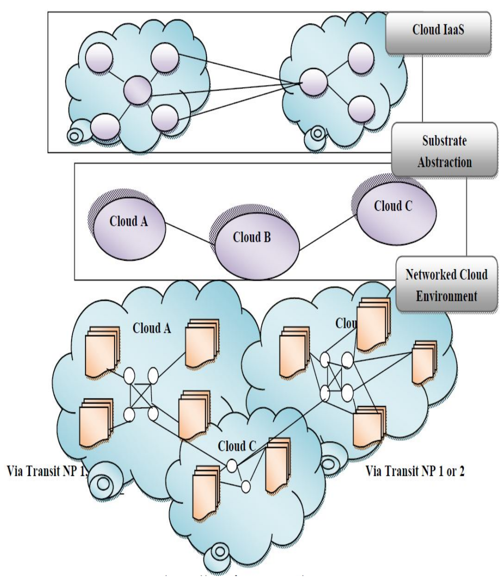

Fig 1.Networked Cloud Environment

II. RELATED WORK

A. VN provisioning model

Upon receiving VN requests from users, the VN providers are responsible for provisioning the VNs through matching, embedding and allocating available substrate resources to set up the required VNs. This section describes the general VN provisioning problem.

B. Substrate network model

A substrate network can be represented by a weighted undirected graph Gs = (Ns, Ls), where Ns is the set of substrate nodes and Ls is the set of substrate links. Substrate resources are described in terms of functional and non-functional attributes. Functional attributes define characteristics and properties of the substrate resources ns and ls including static parameters like node type (e.g. router, switch), node processor type and capacity (e.g. CPU, network processor), link/path type (e.g. VLAN, L3/L2 VPN), network interface type and number, geographic location, cost and so on. Non-functional attributes specify criteria and constraints related to the substrate resources including dynamic (real-time) parameters like available node capacity (CPU), available link capacity (bandwidth), actual QOS parameters, geographic coordinates, etc.

C. VN request model

This section models the VN requests expressed by VN users and sent to VN providers. The VN request is a set of virtual nodes interconnected via virtual links. The VN request topology is represented by a weighted undirected graph Gv = (Nv, Lv), where Nv is the set of required virtual nodes and Lv is the set of required virtual links. Each virtual node is associated with a minimum required capacity denoted by tnv as well as a set of node requirements (e.g. node type, location). Each virtual link lv 2 Lv is associated with a minimum required bandwidth capacity denoted by tlv as well as a set of link requirements (e.g. link type, QOS).

III. REQUSET PARTITIONING AND MAPPING

A. Intercloud request partitioning and intracloud resource mapping

Every cloud broker that receives an incoming request, groups the requested virtual resources according to their functional attributes, creating Virtual Resource Sets (VRS) sharing common characteristics. Functional attributes define characteristics and properties of the resources (e.g., resource type, operating system, virtualization environment, etc.), while non functional attributes specify criteria and constraints related to the resource (e.g., Virtual Machines: CPU, memory, storage, etc.). An one-to-many relation is established between the cloud broker and known CPs that are requested to state provisioning costs. CPs match physical resource candidates to the received VRSs and respond to the cloud broker with a resource provisioning cost per VRS. The cost is related to resource availability in the cloud. In order to perform partitioning in a realistic environment an appropriate heuristic is devised that can provide near optimal results in real time. The cloud broker concludes upon the most cost-effective request partitioning and sends the corresponding partial requests to the selected CPs for further processing.

B. Resource Mapping Framework

Within the cloud domain, a user usually interacts directly with a cloud service provider (CP) to request IaaS. In the context of networked clouds, users may interact indirectly with 1) CPs and 2) transit network providers that interconnect clouds with virtual links [3], since they will rely on a cloud Brokerage Service (denoted in the following as cloud broker) that will be responsible for providing cloud IaaS based on user's requirements. For the sake of simplicity in our work, without loss of validity, we use an abstracted view of the intercloud virtual links spanning multiple transit network providers as depicted in Fig. 1. In the proposed framework, a hierarchical approach is adopted, where virtual resource mapping on the networked cloud is a two-phase procedure: the Request Partitioning Phase where splitting of the request among eligible CPs takes place, and the Embedding Phase where the actual mapping takes place.

C. Embedding Phase

Upon receiving a partial request, the CP embeds it to its substrate resources using an appropriate intradomain VNE algorithm. In the proposed framework, we follow the basic principles of the methodology introduced in [7]. In this sense, virtual nodes are mapped to substrate nodes in a way that facilitates the mapping of virtual links to physical paths in the subsequent phase. However, the problem is extended and formulated for an arbitrary pool of heterogeneous shared

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

resources, while minimizing the overall number of hops for every virtual link mapped on the substrate [8]. In general, embedding is not an one size fits all process and must be tailored to the needs of the request, taking into consideration resource constraints and requested services (e.g., QoS aware embedding). Embedding costs are sent to the cloud broker that responds to the user with the overall mapping cost.

IV. APPROACHES

A. Iterated Local Search

The ILS metaheuristic iteratively builds a sequence of solutions generated by the repeated application of an embedded heuristic and perturbation of the local optimum found. The embedded heuristic is most commonly a problem-specific local search technique. The starting point for the search may be a randomly constructed candidate solution, or one returned by a greedy construction heuristic. Usually, employing a greedy solution results to better performance over less improvement steps. Local Search refers to the local improvements of the solution produced from the perturbation. Essentially, local search consists of moving from one solution to another according to some well-defined rules related to a neighborhood of solution, accepted moves within the neighborhood and termination criterion. Regarding acceptable movements in the search space, first improvement or best improvement can be used. Finally, common stopping criteria for local search algorithms are definition of a maximum number of iterations or terminating the search when no improvement has been observed for a number of iterations. Perturbation generates new starting points of the local search by modifying some local optimum solution. Perturbation guides the algorithm to escape from local optima hence to investigate other parts of the search space. The strength of the perturbation plays a significant role in the efficiency of the algorithm, where strength is usually defined as the number of solution components modified. For a very strong perturbation, ILS behaves like random restart while for a weak perturbation the algorithm's behavior is greedy. Finally, the solution from which the walk is continued is selected according to an appropriately defined Acceptance Criterion. The new local optimum is accepted usually with a fixed rule, e.g., accept only better solution. To enhance diversification, memory is incorporated on the decision process by keeping track of previous solutions and restarting the ILS when no improvement is noticed over a defined number of iterations. ILS metaheuristic is adopted for request partitioning in this paper, mainly due to its intrinsic simplicity and general applicability, overcoming at the same time performance scalability issues that arise when addressing the resource mapping problem in real time and for large incoming requests. The pseudocode of the ILS procedure with ILS is presented in Algorithm 1. To apply the general purpose ILS algorithm to the proposed problem, four procedures need to be fine tuned:

1. *Generate Initial Solution*: To select an appropriate initial solution for the particular ILS implementation, three cases where examined: (a) least cost node mapping, (b) single CP mapping, and (c) random initial solution. No significant improvement was noticed, both in terms of computing time and provisioning cost, therefore a random initial solution was selected.
2. *Perturbation*: In terms of perturbation, different strengths have been tested, where perturbation is accomplished by randomly remapping nodes within the set of interconnected clouds. The best performance was achieved when as many as 80 percent of the solution components were altered by the perturbation.
3. *Acceptance Criterion*: In order to select an appropriate memoryless acceptance criterion two different choices were examined: (a) only better solutions are accepted and (b) a simulated annealing type acceptance criterion is considered [24]. The first approach proved to be the most suitable. This choice of the acceptance criterion has the advantage that it implements a randomized descent in the space of locally optimal solutions [28].
4. *Local Search*: In the particular study an iterated descent algorithm for Local Search was applied employing a first improvement pivoting rule (see Section 2). A simple type of move is used to define a neighbourhood [10]; at each iteration a node is remapped to the various remaining CPs. A maximum number of iterations is used as a stopping criterion.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Algorithm 1. Iterated Local Search

s_i = *GenerateInitialSolution()*
s = *LocalSearch*(s_i)
while Stopping criterion is not met **do**
s' = *Perturbation(s, history)*
s_c= *LocalSearch(s')*
s = *AcceptanceCriterion*(s, s_c, *history*)
end while

B. VN Request Partitioning

The request partitioning problem is related to the graph partitioning problem. Graph partitioning or bisection is a common NP-complete problem with a large number of applications such as load balancing, mapping, etc. It refers to partitioning the vertices of a graph into a given number of disjoint subsets, so that the number of nodes in each subset is less than a given threshold and the number of cut edges is minimum. Kernighan and Lin [9] are among the most wellknown bisection heuristic. Metaheuristic approaches for graph partitioning include the use of simulated annealing or tabu search [10]. In the case of unbalanced partitions, the k cut problem and the multiway partitioning problem are fundamental partitioning problems. In the case of the k-cut problem the goal is to find a minimum weight set of edges whose removal separates the graph into k disconnected components [11]. When considering link weights the problem for an arbitrary k is NP-complete [11]. A number of solutions have been proposed on solving the k-cut problem [13], [14]. Minimum k-cut partition can be also solved via an appropriate max flow algorithm. Request partitioning within a networked cloud environment is presented in [3]. The proposed heuristic integrates a minimum k-cut algorithm followed by subgraph isomorphism. Intracloud resource provisioning costs are defined according to resource availability within the cloud. The cost of an intercloud virtual link is defined as a function of the transit network providers it traverses. In the context of network virtualization environment, Houidi et al. [12] split a VN request partitioning across multiple Infrastructure Providers (InPs) using both max-flow min-cut algorithms (Ford-Fulkerson algorithm) and linear programming techniques. Intraprovider provisioning costs are randomly set. The cost of peering links is also randomly set satisfying the constraint that it exceeds the cost of corresponding candidate intradomain links of the two endpoint peers. The approach described in [5] applies a request partitioning heuristic, that starts with an initial graph partitioning and reallocates the different parts of the VN via local improvements. Intradomain resource provisioning costs are randomly generated. The cost of interdomain links is assumed to be an order of magnitude higher than the average intradomain cost. It should be noted that in most of these approaches either fixed or randomly generated provisioning costs are taken into consideration. In contrast, in our proposed framework more realistic intra- and interdomain resource provisioning costs are included in the request partitioning problem definition. Such costs include resource availability, in terms of scarcity of the resource and the average utilization over a time window, as well as the load on the substrate network.

C. Virtual Network Embedding

Virtual network embedding within a single domain, with constraints on virtual nodes and virtual links, can be reduced to the NP-hard multiway separator problem [17]. As a result, a number of heuristic-based algorithms have been devised to solve the problem. Several of them decompose the problem into the node mapping phase and the link mapping phase, to reduce the overall complexity. Recent approaches tend to solve the two problems either simultaneously (e.g., [12]) or providing some type of coordination between the two phases (e.g., [7]). In most cases, VN requests are handled upon arrival (e.g., [7]). Dynamic (e.g., [18]) approaches support remapping of resources during the life-time of the request as opposed to static ones (e.g., [20], [7]). VNE algorithms suffer from scalability issues and hence request partitioning has been studied for mapping each part of a request to a different part of the substrate network. Distributed algorithms have

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) **Vol.2, Special Issue 1, March 2014**

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

been proven to enhance overall network resiliency as well as overcome scalability limitations and delays imposed by maintaining information centrally. Apart from heuristic algorithms, metaheuristics have been also applied for solving the VNE problem. It is noted that networked cloud environments supported in our proposed framework are more realistic in the sense that heterogeneous substrate resources (e.g., servers and routers) are taken into consideration, as opposed to existing approaches in the literature. Moreover, the proposed intradomain resource mapping formulation and approach takes into account quality of service (QoS) considerations as well, reflecting the need to establish low delay communication paths between noncollocated virtual resources.

D. Provisioning Cost Definition and Objective Function

The provisioning cost per CP for every requested VRS V^V_A is determined by matching set $V^S_{A,i}$. Specifically, it is based on the scarcity of nodes supporting the requested functional attributes in each CP's substrate, provided by

$$SC(V^S_{A,i}) = |V^S_{A,i}|.$$

In addition, every nonfunctional attribute of the matching substrate resources is taken into account, without however disclosing information about the CP's substrate. In that sense, the utilization of resource b denoted as $\bar{U}_b(V^S_{A,i})$ is estimated as the average of corresponding utilization values $U_b(CAP_b(n^S_{A,i}))$ of the entire set of candidate substrate nodes $n^s_{A,i} \in V^S_{A,i}$. Overall, the average(weighted) utilization for all functional attributes of matching substrate resources in $V^S_{A,i}$ is provided by

$$\bar{U}_B(V^S_{A,i}) = \bar{U}_b \sum_{\forall \in B} w_b \bar{U}_b \, (V^S_{A,i}).$$

Taking all the above into consideration the partitioning cost per CP for $V^S_{A,i}$ is given by

$$C(V^S_{A,i}) = \frac{\bar{U}_B(V^S_{A,i})}{SC(V^S_{A,i})}, \ \forall i \in I .$$

Similarly, provisioning cost per CP for every requested E^V_d is determined by matching set $E^S_{d,i}$. The particular cost is also associated with the remaining resources of link type d in each CP's substrate network

$$C(E^S_{d,i}) = \frac{\bar{U}(E^S_{d,i})}{SC(E^S_{d,i})} \bar{U}(E^S_{d,i}).$$

$\bar{U}(E^s_{d,i})$, $i \in I$ is the average link utilization of the entire set of candidate substrate links in set $E^S_{d,i}$. Moreover, $SC(E^S_{d,i}) = |E^S_{d,i}|$ determines the scarcity of the requested link type in the CP's substrate. The corresponding objective function with respect to provisioning cost is defined as follows:

$$\min \sum_{\forall i \in I} \sum_{\forall A} \sum_{\forall n^V{}_A \in V^V{}_A} C(V^S{}_{A,i}) z^i{}_n V_A$$
$$+ \min \sum_{\forall i \in I} \sum_{\forall A \in D} \sum_{\forall mn^V d \in E^V d} C(E^S{}_{d,i}) y^{ii}{}_{mn} V_d$$
$$+ \min \sum_{\forall i \in I} \sum_{\forall j \in I, j \neq i} \sum_{\forall d \in D} \sum_{\forall mn^V d \in E^V d} C^{ij}(E^V{}_d) y^{ii}{}_{mn} V_d$$

where the binary variable $Z^i{}_n{}^v{}_A$ = 1when the requested node $n^V{}_A$ is assigned to CP i. Similarly the binary variable $y^{ii}{}_{mn}V_d$ is set to 1. when the requested link $mn^V{}_d$ has been exclusively assigned to CP i. Finally, $y^{ii}{}_{mn}V_d$ = 1, $i \neq j$; when the requested link $mn^V{}_d$ spans multiple CPs, while i; j are the two endpoint CPs. It is noted that the first summation term in objective function (1) defines the node provisioning cost based on the resulting node assignment per CP. The second and third summation terms, in the same fashion, determine intra- and intercloud link provisioning cost, respectively.

V. CONCLUSIONS

Within the proposed framework, an ILS-based heuristic is employed to provide a cost-efficient resource allocation realizing the partitioning of the user request. In this work, the cloud resource mapping problem over a networked cloud computing environment is studied, by providing high-performance algorithms in terms of embedding effectiveness and runtime complexity. The proposed algorithm is compared against an exact method that provides minimal cost partitioning. Following VN partitioning, a sophisticated intra VNE algorithm based on linear programming is utilized so that each sub-VN is properly allocated in the selected cloud. The objective is to minimize the cost of embedding a request by performing load balancing and enhancing the cloud scalability. Finally, it should be noted that resource provisioning cost has been defined in this paper based on node scarcity and average utilization. Thus, utilization in this case depends linearly on the computing resources. An interesting approach would be to derive utilization based on more enhanced queuing models that could better reflect potential nonlinear behaviors. In future work will provide security and load balancing mechanisms.

ACKNOWLEDGMENTS

This work has been partially supported by the EC FP7 Programme under Grant No. 257867 - NOVI.

REFERENCES

[1] RAD Lab, "Above the Clouds: A Berkeley View of Cloud Computing," UC Berkeley Technical Report UCB/EECS 2009- 28, 2009.
[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," Recommendations of the National Institute of Standards and Technology, pp. 800-145, NIST Special Publication, 2011.
[3] Y. Xin, I. Baldine, A. Mandal, C. Heermann, J. Chase, and A. Yumerefendi, "Embedding Virtual Topologies in Networked Clouds," Proc. Sixth Int'l Conf. Future Internet Technologies (CFI' 11), June 2011, doi:10.1145/2002396.2002403.
[4] S. Paul, J. Pan, and R. Jain, "Architectures for the Future Networks and the Next Generation Internet: A Survey," Computer Comm., vol. 34, no. 1, pp. 2-42, Jan. 2011, doi:10.1016/j.comcom.2010. 08.001.
[5] F. Zaheer, J. Xiao, and R. Boutaba, "Multi-Provider Service Negotiation and Contracting in Network Virtualization," Proc. IEEE Network Operations and Management Symp. (NOMS), pp. 471-
478, June 2010, doi: 10.1109/NOMS.2010.5488487. [6] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual Network Embedding through Topology-Aware Node Ranking," SIGCOMM Computing Comm. Rev., vol. 41, no. 2,
pp. 38-47, Apr. 2011, doi:http://doi.acm.org/10.1145/1971162. 1971168.
[7] M. Chowdhury, M.R. Rahman, and R. Boutaba, "ViNEYard: Virtual Network Embedding Algorithms with Coordinated Node and Link Mapping," IEEE/ACM Trans. Networking, vol. 20, no. 1, pp. 206-219, Feb. 2012, doi: 10.1109/TNET.2011. 2159308.
[8] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervello-Pastor, and A. Monje, "On the Optimal Allocation of Virtual Resources in Cloud Computing Networks," Submitted 2011 (currently under revision), 2011.

[9] B. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," Bell Systems Technical J., vol. 49, no. 1, pp. 291-307, 1970.

[10] L. Tao, C. Zhao, K. Thulasiraman, and M.N.S. Swamy, "Simulated Annealing and Tabu Search Algorithms for Mulitway Graph Partitioning," J. Circuits Systems, and Computers, vol. 2, no. 2, pp. 159-185, 1992, doi:10.1142/S021812669200012X.

[11] O. Goldschmidt, "Polynomial Algorithm for the k-CUT Problem," Proc. 29th Ann. Symp. Foundations of Computer Science, pp. 444-451, Oct. 1988, doi: 10.1109/SFCS.1988.21960.

[12] I. Houidi, W. Louati, W.B. Ameur, and D. Zeghlache, "Virtual Network Provisioning Across Multiple Substrate Network," ELSEVIER Computer Networks, vol. 55, no. 2, pp. 1011-1023, 2011, doi.org/10.1016/j.comnet.2010.12.011.

[13] L.A. Sanchis, "Multiple-Way Network Partitioning," IEEE Trans. Computers, vol. 38, no. 1, pp. 62-81, Jan. 1989, doi: 10.1109/12.8730.

[14] S.J. Kang and B.R. Moon, "A Hybrid Genetic Algorithm for Multiway Graph Partitioning," Proc. Genetic and Evolutionary Computation Conf., 2000, doi: 10.1.1.152.8394.

[15] S. Kapoor, "On Minimum 3-Cuts and Approximating k-Cuts Using Cut Trees," Proc. Fifth Int'l IPCO Conf. Integer Programming and Combinatorial Optimization, vol. 1084, pp. 132-146, 1996, doi: 10.1007/3-540-61310-2_11.

[16] M. Stoer and F. Wagner, "A Simple Min-Cut Algorithm," J. ACM, vol. 44, no. 4, pp. 585-591, July 1997, doi:10.1145/263867.263872.

[17] D. Andersen, "Theoretical Approaches To Node Assignment," Unpublished Manuscript, http://www-2.cs.cmu.edu/dga/ papers/andersen-assign.ps, 2002.

[18] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration," ACM SIGCOMM Computer Comm. Rev., vol. 38, no. 2, pp. 17-29, Apr. 2008, doi:10.1145/1355734.1355737.

[19] J. Lischka and K. Holger, "A Virtual Mapping Algorithm Based on Subgraph Isomorphism Detection," Proc. ACM Workshop Virtualized Infrastructure Systems and Architectures (SIGCOMM '09), pp. 81-88, Aug. 2009, doi:10.1145/1592648.1592662.

[20] Y. Zhu and M.H. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," Proc. IEEE INFOCOM '06, pp. 1-12, Apr. 2006, doi:10.1109/INFOCOM. 2006.322.

[21] I. Houidi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, "Adaptive Virtual Network Provisioning," Proc. ACM SIGCOMM, pp. 41-48, Sept. 2010, doi:10.1145/1851399. 1851407.

[22] S. Zhang, Z. Qiant, S. Guo, and S. Lu, "FELL: A Flexible Virtual Network Embedding Algorithm with Guaranteed Load Balancing," Proc. IEEE Int'l Conf. Comm. (ICC '11), pp. 1-5, June 2011, doi: 10.1109/icc.2011.5962960.

[23] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "VNE-AC : Virtual Network Embedding Algorithm Based on Ant Colony Metaheuristic," Proc. IEEE Int'l Conf. Comm. (ICC), pp. 1-6, June 2011, doi: 10.1109/icc.2011.5963442.

[24] M. Gendreau and J. Potvin, Handbook of Metaheuristics, second ed. Springer, 2010.

[25] D.P. Bertsekas, Network Optimization: Continuous and Discrete Models. Athena Scientific, 1998.

[26] Local Search in Combinatorial Optimization. Princeton Univ. Press, 2003.

[27] R. Buyya, C.S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering It Services as Computing Utilities," Proc. IEEE Int'l Conf. High Performance Computing and Comm. (HPCC '08), pp. 5-13, Sept. 2008, doi: 10.1109/HPCC.2008.172.