Computer Communications xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Generating desirable network topologies using multiagent system

Q1 Noriaki Kamiyama *

NTT Service Integration Laboratories, Tokyo 180-8585, Japan

ARTICLE INFO

Article history:
Received 19 March 2011
Received in revised form 30 July 2012
Accepted 31 July 2012
Available online xxxx

Keywords:
Network topology
AHP
Multiagent system

ABSTRACT

Designing network topologies requires simultaneous consideration of multiple criteria, such as network cost and reliability. So, the author applied the analytic hierarchy process, a way to make a rational decision considering multiple criteria, to network topology evaluation. However, the time required to construct the candidate topology set greatly increases as the network scale grows. Therefore, the author proposed to generate candidate topologies within a practical time frame for large-scale networks by limiting the positions for putting links to a small set of candidates. However, the diversity of the obtained candidate set is limited because the links are always put at certain link positions and are never put at a majority of the other link positions in all the candidate topologies generated. Therefore, this paper proposes to use of a multiagent system, in which each agent autonomously behaves to maximize each criterion, for generating a candidate topology set with high diversity within a practical time frame for large-scale networks.

© 2012 Published by Elsevier B.V.

1. Introduction

For network carriers and ISPs operating and managing physical network resources, one important problem is how to design a network topology. Recently, network virtualization technique in which network resources can be flexibly reserved for each network service has been widely investigated [27]. Using this technique, ISPs can flexibly design their network infrastructure for each service, so developing optimal design method of network topologies becomes more important for ISPs. For a backbone network topology, we should carefully consider both the connectivity between any pair of edge nodes and the redundancy for maintaining the connectivity in case of node or link failure. To improve the redundancy, increasing the routes between each edge node pair by providing more intermediate nodes and links is desirable. However, the increase in nodes and links will also increase equipment and operating costs. For users, avoiding congestion at intermediate nodes and having a shorter path length to reduce the packet network delay is desirable. If we decrease the number of nodes and links to reduce the network cost, the flexibility of path design is degraded, so suppressing the path length becomes difficult. Therefore, when designing a network topology, we need to consider multiple incompatible criteria with different units, such as cost, reliability, and path length.

There are many works designing network topologies. One proposed a physical topology design minimizing the total physical link count under the condition that connectivity between all pairs of nodes is maintained in the case of a single physical link failure [24]. Ramaswami and Sivarajan [19] and Krishnaswamy [17] proposed a logical topology design minimizing the maximum link load in a wavelength-routed optical network. A design method minimizing the average hop count of wavelength paths was proposed in [1], and another method maximizing overall throughput in a wavelength-routed optical network was proposed in [30]. Chattopadhyay et al. [6] and Gersht and Weihmayer [7] presented heuristic approaches using a branch-and-bound method or a greedy method to solve the cost minimization problem with a constraint on the delay between nodes. Steiglitz et al. [23] presented a heuristic method using a local search that solves the cost minimization problem with the constraint that all node pairs have more than a specified number of disjoint routes. Wille et al. [29] depicted heuristic approaches using a tabu search and generic algorithm for solving the same problem with the constraint that the connectivity between any pair of nodes is maintained for any single-node failure. However, all these works consider only a single criterion as the optimization target.

As an approach that considers multiple criteria, the concept of the Pareto frontier is well known [26], and one study applied this concept to logical topology design [10]. Assume that there are M criteria, $V_1, \ldots, V_M$, and let $V_{m,x}$ denote the mth criterion of candidate x. We can say that candidate x is better than candidate y in the Pareto sense only if $V_{m,x} \leqslant V_{m,y}$ for any m and there exists criterion m that satisfies $V_{m,x} < V_{m,y}$. (Assume that smaller values are desirable in all criteria.) All candidates that are surpassed by no other candidates are the optimum solution set, i.e., the Pareto frontier. However, a large number of candidates are regarded as the Pareto frontier, so it is difficult to effectively limit the optimum candidates and select one network topology to use.

* Tel.: +81 422 59 4763; fax: +81 422 59 6364.
E-mail address: kamiyama.noriaki@lab.ntt.co.jp

0140-3664/$ - see front matter © 2012 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.comcom.2012.07.019

The analytic hierarchy process (AHP) is a way to make a rational decision considering multiple criteria [9,20]. Using AHP, we can reflect the relative importance of each criterion in the evaluation result. AHP considers all the related factors in a hierarchical structure and quantifies qualitative factors, such as the importance of each criterion, using paired comparison. Therefore, we have applied AHP to network topology evaluation to consider multiple criteria simultaneously [11]. When evaluating network topologies using AHP, we need to construct a set of topology candidates prior to evaluation. However, the time required to construct a candidate set increases in the order of $2^{N \times N}$ as the number of nodes N increases; therefore, it is difficult to construct a set of topology candidates within a practical time frame for large-scale networks.

In general, enumerating all candidates satisfying certain conditions without replications is known as an enumeration problem [8]. In such a problem, it is important to reduce the required calculation time while satisfying both completeness, i.e., enumerating all candidates satisfying the condition without any omissions, and uniqueness, i.e., enumerating candidates without duplications. There are mainly two approaches for enumeration algorithms: a binary partition and a reverse search [25]. We applied the binary partition method to the construction of candidate topologies [12]. However, it is difficult to construct candidate topologies within a practical time frame for large-scale networks with about 10 or more nodes when using the binary partition method [13].

To generate candidate topologies within a practical time frame for large-scale networks, we should take another approach, i.e., generating only some candidate topologies instead of generating all the candidate topologies satisfying the conditions. In this approach, generating desirable and diverse candidate topologies is important to suppress the influence on the AHP result. Based on this approach, we proposed generating candidate topologies by limiting the candidate positions for locating links in a small set [13].

To satisfy the connectivity requirement between nodes, this method first constructs a topology in which some links are added to the minimum spanning tree. Next, this method selects candidate positions where we can put links. Although we can dramatically reduce the time required to construct the candidate topology set by using this method, the diversity of the generated topologies is low. This is because that links are always put at the positions constructing the initial topology, whereas links are never put at a large part of positions in all the generated candidates. Therefore, the results of applying AHP to the generated candidate set are expected to be largely different from those obtained by applying AHP to all the candidate topologies that can be constructed.

A multiagent system (MAS) is used for investigating the environment in which multiple agents behave autonomously, such as the ecosystems of animals and social systems [22,28]. MAS is mainly used to analyze the environment resulting from the autonomous behavior of multiple agents or to investigate the control method for generating a desirable environment for the whole system. Systems such as ecosystems and social systems that can be investigated by MAS often show high robustness against changes of environment or failures, and this robustness seems to be derived from the diversity of the systems as a result of dynamic interaction among the agents [28]. Therefore, if we regard the evaluation criteria of AHP as agents and simulate MAS in which each agent autonomously adds or removes links at any candidate position to optimize its evaluation criterion, we can expect to construct candidate topologies with high diversity, which are evaluated highly by AHP.

This paper proposes to construct a candidate topology set by using MAS and investigates its effectiveness by numerical evaluation.[1] In Section 2, we summarize the evaluation method using AHP for network topologies. In Section 3, we briefly describe the construction method for candidate topologies that limits the candidate positions for putting links, proposed in [13]. We describe the proposed construction method for candidate topologies using MAS in Section 4 and show the numerical results in Section 5. Finally, we conclude the paper in Section 6.

2. Topology evaluation using AHP

2.1. Overview of AHP

In a decision-making problem, there are normally three kinds of elements, i.e., *problem P*, *evaluation criteria V*, and *alternative plan G*. As shown in Fig. 1, AHP considers the relationship among these elements as a hierarchical structure and link-related elements. Evaluation criteria V can take multiple layers, $V^1, V^2, \ldots$ By calculating the relative strength (weight) for each pair of related elements, AHP derives the score S_i of each alternative plan G_i.

We need to quantify the relative importance of each criterion V against a problem P. This is achieved by comparing the elements on each level in pairs using AHP. For the two elements X_i and X_j in layer c, the numerical value listed in Table 1 selected by the decision maker is set to a_{ij}, the relative importance of X_i against X_j. By defining w_i as the true weight of X_i, we ideally have $a_{ij} = w_i / w_j$. Let $\boldsymbol{A}$ and $\boldsymbol{w}$ denote a matrix of pairwise comparisons a_{ij} and a vector of w_i, respectively. By multiplying $\boldsymbol{A}$ by $\boldsymbol{w}$, we obtain $\boldsymbol{Aw} = n\boldsymbol{w}$, where n is the number of elements in the layer. Therefore, $\boldsymbol{w}$ is the principal eigenvector and n is the maximum eigenvalue.

In practice, consistently setting a_{ij} for all pairs of elements is difficult, so we need to judge the degree of inconsistency. Letting λ_{max} denote the maximum eigenvalue of $\boldsymbol{A}$, we have $\lambda_{max} \geqslant n$ [9,20]. We can then judge the degree of inconsistency using the *consistency index* (C.I.) defined by

$$\text{C.I.} = \frac{\lambda_{max} - n}{n - 1}.$$

λ_{max} decreases as the degree of consistency increases, and $\lambda_{max} = n$ when $\boldsymbol{A}$ is a consistent matrix. Hence, the degree of consistency increases as C.I. decreases. For each size of matrix n, random matrices are generated and their mean C.I. value, called the *random index* (R.I.), is computed. The *consistency ratio* (C.R.) is defined as the ratio of C.I. to R.I., i.e., C.R. = C.I./R.I., and C.R. is a measure of how a given matrix compares to a purely random matrix in terms of their C.I. A C.R. less than or equal to 0.1 is typically considered acceptable [9].

Let w_{ij}^c denote the weight of the ith element in layer c against the jth element in layer $c-1$. We also define $\boldsymbol{\Phi}_i^c$ as the element set in layer $c-1$ related to X_i^c, the ith element in layer c. S_i^c, the score of X_i^c against problem P, is then derived as

$$S_i^c = \sum_{j: X_j^{c-1} \in \boldsymbol{\Phi}_i^c} w_{ij}^c S_j^{c-1}.$$

In layer 1, S_i^1 is equal to the weight of each element against problem P. We can recursively obtain S_i^c in the order of $c = 2, 3, \ldots$ and finally derive S_i, the score of alternative plan G_i. Plans with large S_i are desirable.

2.2. Applying AHP to network topology evaluation

When we apply AHP to network topology evaluation, the target problem P, which is *choosing optimum network topologies* in this case, is located in the top layer (layer 0), the evaluation criteria V_i are located in the middle layer (layer 1), and the candidate topologies G_i are located in the bottom layer (layer 2) as shown in Fig. 2. Let N_1 and N_2 denote the numbers of evaluation criteria and candidate topologies, respectively.

[1] A shorter version of this manuscript was presented in [14].

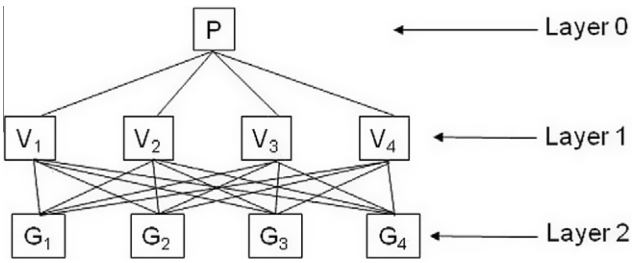


Fig. 1. Layered structure in AHP. AHP considers the relationship among these elements as a hierarchical structure and link-related elements.

Table 1
Scale of measurement for AHP.

Numerical values	Definition
1	Equally important or preferred
3	Slightly more important or preferred
5	Rather more important or preferred
7	Much more important or preferred
9	Extremely more important or preferred
2, 4, 6, 8	Intermediate values to reflect compromise
Reciprocals	Used to reflect dominance of second alternative over first

The pairwise comparisons described in Section 2.1 enable us to derive the scores (weights) of the elements in layer 1, i.e., the evaluation criteria, for the problem P. If all the criteria have numerical values, pairwise comparisons are not necessary to obtain the weights of the elements in layer 2, i.e., the topology candidates, for each element in layer 1. Let V_{ij} denote the jth criterion of candidate i. Because AHP evaluates elements with higher weights more highly, it derives weights based on X_{ij}, which is the reciprocal of V_{ij}, i.e., $X_{ij} = 1/V_{ij}$. The weights of the elements in layer 2, w^2_{ij}, need to satisfy the normalized condition, so we have $w^2_{ij} = X_{ij}/\sum_{k=1}^{N_2} X_{kj}$. Because the number of decision candidates in AHP has been normally up to around seven [21], this weight setting has been reasonable. However, the number of topology candidates is huge even in a moderately sized network, so the denominator of this equation becomes huge, and the difference in the weights w^2_{ij} among the candidates becomes far smaller compared with the difference in the weights w^1_{ij} among the evaluation criteria. As a result, AHP tends to simply choose the candidates with desirable values of the weighted criteria.

To solve this problem, we have proposed using the normalized value of Y_{ij}, i.e., a linear-transformed value of X_{ij}, rather than X_{ij} itself, for the weights [11]. In other words, we define Y_{ij} as $Y_{ij} = a(X_{ij} + b)$, where a and b are arbitrary real numbers. The weights w^2_{ij} are derived as

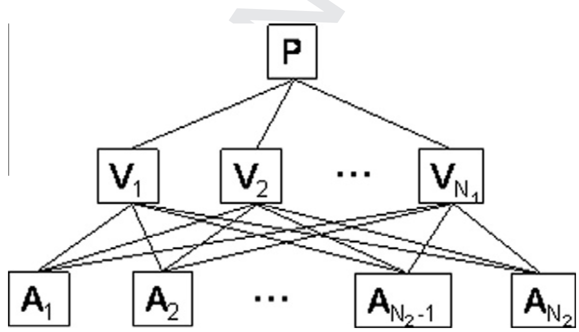


Fig. 2. AHP structure in network topology evaluation.

$$w^2_{ij} = \frac{a(X_{ij} + b)}{\sum_{k=1}^{N_2} a(X_{kj} + b)} = \frac{X_{ij} + b}{\sum_{k=1}^{N_2} X_{kj} + bN_2}. \quad (3)$$

Because of the normalization, w^2_{ij} is independent of a, so we set $a = 1$ hereafter. Moreover, to make all the weights take a positive value or zero, we set b to the minimum value of X_{ij} among all the candidates multiplied by -1, i.e., $b = -min_i\{X_{ij}\}$. The difference in the weights is increased by the linear-transformation. In particular, we can dramatically decrease the weights for candidates with a large criterion value, i.e., a small value of X_{ij}, and thus we can avoid choosing topologies having terrible values for some criteria as optimum topologies [11].

2.3. Constructing topology candidates

We need to construct the candidate topologies before evaluating them using AHP. We consider nodes and links as the generalized elements of networks. In addition, we assume that the node location and the traffic demand matrix among nodes are given and that we can set links to any position between nodes. In other words, L, the number of candidate positions where we can set links, is given by $L = N(N-1)/2$, where N is the number of nodes. We also assume that all links are bidirectional and that packets are transmitted in both directions on each link. By selecting locations where we set links from the L candidate positions, we can construct a network topology; therefore, the total number of topology candidates we can construct is 2^L. However, connectivity between all node pairs is the minimum requirement for network topologies, so we consider constructing network topologies that satisfy the connectivity between all pairs of nodes in the normal operation, i.e., the state without failures of any nodes or links.

In the Internet, moreover, we normally see failures of various links [18], and network topologies are required to maintain connectivity at these failures. About 70% of unintentional failures, excluding maintenance ones, originate from a single link failure (SLF) [18]. We define ξ, the average ratio of traffic whose connectivity is lost at SLFs, as

$$\xi = \frac{1}{M}\sum_{e \in \boldsymbol{E}} \sum_{i,j \in \boldsymbol{P}_e} r_{ij},$$

where M and $\boldsymbol{E}$ are the number and set of links constructing the topology, $\boldsymbol{P}_e$ is the set of node pairs whose connectivity is lost at the SLF of link e, and r_{ij} is the ratio of the traffic amount from node i to node j against the total amount of traffic within the network. We consider generating only topologies satisfying $\xi \leqslant \alpha_\xi$, where α_ξ is an arbitrarily given upper limit of ξ, as well as the connectivity constraint in the normal operation. We assume that r_{ij} is proportional to the product of the populations of nodes i and j. Let U_k and U denote the population of node k and the total population of all nodes. We set $r_{ij} = r_i r_j$, where r_k is the population ratio of node k, i.e., $r_k = U_k/U$. Let C denote the two constraints that candidate topologies need to satisfy: the connectivity between all pairs of nodes in the normal operation and $\xi \leqslant \alpha_\xi$.

2.4. Evaluation criteria

When evaluating network topologies using AHP, we can consider any evaluation criteria simultaneously. However, to obtain the candidate topology set within a practical time frame for large-scale networks, we need to develop a desirable construction method for candidate topologies according to the evaluation criteria used in the AHP. In this paper, as the candidates of evaluation criteria, we consider four criteria: (i) ζ, the total link length, (ii) ϵ,

UNCORRECTED PROOF

the average hop distance between nodes weighted by the traffic ratio, (iii) v, the average end-to-end packet delay weighted by the traffic ratio, and (iv) ξ, the average ratio of traffic whose connectivity is lost at the SLFs.

ζ is related to the cost and is defined as $\zeta = \sum_{e \in \boldsymbol{E}} d_e$, where d_e is the length of link e. ϵ is related to the user quality and is defined as $\epsilon = \sum_{i,j \in \boldsymbol{V}} r_{ij} h_{ij}$, where $\boldsymbol{V}$ is the node set and h_{ij} is the shortest hop distance between nodes i and j.

Let t_{sd} denote the end-to-end delay from source node s to destination node d, and only consider the queuing delay at links on the path in t_{sd}. Assuming the M/M/1 queuing model, the queuing delay at link e, t_e is given by $t_e = \tau\rho_e/(1-\rho_e)$, where τ is the packet transmission delay and ρ_e is the utilization of link e [16]. We assume that the transmission capacity of all links is B and the total traffic demand T is $T = \kappa B$ where κ is a given parameter. Let Q_e denote the set of source and destination node pairs whose path takes link e, and we define X_e as $X_e = \sum_{s,d \in Q_e} r_{sd}$. Using X_e, ρ_e is given by $\rho_e = \kappa B X_e / B = \kappa X_e$. Therefore, we obtain t_{sd} as $t_{sd} = \sum_{e \in P_{sd}} \tau\kappa X_e/(1-\kappa X_e)$, where P_{sd} is the set of links on which the path from node s to node d takes. Because τ is identical in all the source and destination node pairs and all the topology candidates, we set $\tau = 1$. We define v as

$$v = \sum_{s,d \in \boldsymbol{V}} r_{sd} \sum_{e \in P_{sd}} \frac{\kappa X_e}{1 - \kappa X_e}.$$

$\xi$is defined by (4).

We assume that packets are transferred on the shortest hop routes using OSPF (open shortest path first). Smaller values are more desirable for all the four criteria. However, as the number of links decreases, ζ decreases, whereas ϵ, v, and ξ increase in general because the diversity of route settings between nodes is degraded. Therefore, there is a negative correlation among these criteria, and it is difficult to construct ideal network topologies in which both ζ and ϵ are close to the minimum values. So, the topology evaluation using AHP is effective. In the following sections, we assume that the two criteria, ζ and ϵ, are used to evaluate candidate topologies, and we set $V_1 = \zeta$ and $V_2 = \epsilon$. In Section 5.5, we investigate the case when using v instead of ϵ, and three criteria, ζ, ϵ, and ξ.

3. Constructing topology set by limiting candidate positions to set links

The required calculation time grows proportionally to 2^x as the number of candidate positions for setting links x increases, so one possible approach to construct the candidate topologies within a practical time frame is to bound x to a smaller value than the total possible count L. In Ref. [13], we proposed to always set links at the minimum number of locations to satisfy the connectivity constraint and to limit the candidate positions where we can set links in addition to these mandatory link positions. In this section, we briefly summarize this method.

3.1. Selecting locations to always set links

A minimum spanning tree (MST) is a topology minimizing the total link cost and satisfying the connectivity between all nodes when the cost of each link is given. We generate an MST by using the Prim algorithm [4] using d_{ij}/r_{ij} as the link cost between nodes i and j, where d_{ij} is the distance between nodes i and j.

Although the obtained topology T_a satisfies the connectivity between all nodes in normal operation, the connectivity at the SLFs is not considered in T_a. Therefore, we add the least number of links to T_a for ξ to be bounded below an arbitrarily given design parameter α_ξ. Let x_a denote the number of links on T_a, i.e., $x_a = N - 1$. We derive ξ in the topology obtained by adding one link to each of the $L - x_a$ candidate positions to which links can be added, and we add one link to the candidate position with the minimum ξ. This process is repeated until $\xi \leqslant \alpha_\xi$ is satisfied; let T_b denote the obtained topology.

3.2. Selecting candidate locations for adding links

When constructing T_b, we set links to the least number of positions to satisfy the constraint for connectivity and ξ. This results in generating topologies with a desirable (i.e., small) V_1, the total link length. However, the other evaluation criterion, V_2, is not considered when making T_b. As the number of links increases, the diversity of routes between a node pair increases. As a result, V_2, the weighted average hop distance between nodes, tends to decrease. Therefore, by adding candidate locations for locating links in T_b, we can construct a candidate set including diverse topologies in V_2.

Let x denote the number of candidate positions to which we add links, and let $\boldsymbol{E}_c$ denote the set of these x candidate positions. We construct candidate topologies by considering all the combinations of setting a link to any location included in $\boldsymbol{E}_c$, so the constructed candidate set $\boldsymbol{T}_d$ consists of 2^x topologies in which links definitely exist at x_b link positions in T_b. Initially, we set $\boldsymbol{E}_c$ to an empty set. We calculate ϵ for each of the obtained topologies by adding one link to any $L - x_b$ candidate location for setting links between any node pair, excluding x_b links in T_b, and add the location e_1 to $\boldsymbol{E}_c$, where ϵ is minimized when adding one link to e_1. This process is repeated x times.

4. Constructing candidate topologies using multiagent system

When using the method proposed in [13] and described in Section 3, the generated topologies are limited to ones in which links are always set at positions constructing T_b, and no links exist at a large part of the locations, excluding those constructing T_b or included in $\boldsymbol{E}_c$. Therefore, only topologies with similar shapes are generated. This results from limiting the candidate locations for setting links from L to a much smaller number x. In this section, we propose a construction method for candidate topologies with high diversity within a practical time frame, without limiting the candidate positions for link settings.

4.1. Multiagent system (MAS)

A multiagent system (MAS) is used for analyzing the phenomena caused in large-scale and robust systems such as the ecosystems of animals and social systems of humans [22,28]. MAS consists of an environment and multiple agents, which are entities behaving autonomously and selfishly. Agents influence the environment as a result of behaving autonomously according to the sensed result of the environment state. By describing the interaction among agents, we can analyze the behavior of the entire system using computer simulation.

MAS is mainly used to analyze the states of the entire system achieved as a result of the autonomous behaviors of agents. Moreover, it is also used to appropriately design the systems to have high robustness and flexibility. Systems such as ecosystems and social systems that MAS mainly targets show high robustness against changes in the environment or failures, and this robustness seems to be derived from the diversity of the systems resulting from dynamic interaction among the agents [28]. Therefore, we can expect that the environment states generated by MAS at each time position are diverse ones in which the requirements of all the agents are reflected. Thus, if we regard the evaluation criteria of AHP as agents and simulate the MAS in which each agent autonomously adds or

UNCORRECTED PROOF

removes links at any candidate position to optimize its evaluation criterion, we can expect to construct candidate topologies with high diversity, which are evaluated highly by AHP.

4.2. Environment and agents

To generate candidate topologies using MAS, we define the environment of MAS as a topology that can be constructed for a given node set. The set of environment states, $\mathscr{S}$, consists of $Q = 2^L$ topologies, i.e., $\mathscr{S} = \{s_1, s_2, \ldots, s_Q\}$. We set the number of agents to the number of evaluation criteria in AHP and relate the ith agent A_i to the ith criterion V_i. MAS can be constructed by defining the behavior of each agent against the current state of environment, and we define the behavior of agent A_i as optimizing the related criterion V_i. For the optimization behavior of each agent, we consider removing any one link among ones constructing the current environment, i.e., topology, or adding one link at any candidate position without a link in the current environment. As a result of behavior of each agent, i.e., decreasing or increasing a link, the environment is influenced and changes from one state (topology) to another.

As mentioned in Section 2.4, we assume that candidate topologies are evaluated by AHP on the basis of two evaluation criteria: ζ, the total link length, and ϵ, the weighted average hop distance between nodes. Therefore, we provide two agents A_1 and A_2, and let A_1 and A_2 optimize ζ and ϵ, respectively. ζ never decreases for the addition of any link, whereas it never increases for the deletion of any link. In contrast, ϵ never increases for the addition of any link, whereas it never decreases for the deletion of any link. Therefore, agent A_1 always removes one link, whereas agent A_2 always adds one link. A_1 removes a link with the maximum value of link length d_e from the links included in $\boldsymbol{E}_n$, where $\boldsymbol{E}_n$ is the link set constructing the current state of environment, s_n. Let $\overline{\boldsymbol{E}}_n$ denote the set of candidate link positions where links are not set in s_n. A_2 calculates ϵ for the topology in which a link is added at each of the candidate positions included in $\overline{\boldsymbol{E}}_n$, and it adds one link at the candidate position giving the minimum value of ϵ.

We consider outputting the time series of the environment state as the candidate topology set, so all the environment states need to satisfy the constraint C. Topologies obtained by adding a link at any candidate position in a topology satisfying the constraint C obviously also satisfy that constraint, so agent A_2 does not need to consider the constraint C when selecting a position to which to add a link. However, topologies obtained by removing a link from a topology satisfying the constraint C do not always satisfy that constraint, so agent A_1 removes one link with the maximum length with the constraint that C is still satisfied after the link is removed. If there are multiple candidate links with the maximum length, one link is randomly selected from them. In the same way, if there are multiple candidate positions with the minimum value of ϵ when A_2 selects a position to which to add a link, one position is randomly selected from them.

4.3. Operation of proposed MAS method

As the initial state of environment, s_0, the proposed MAS method constructs the topology T_b satisfying the constraint C by the method described in Section 3.1. We assume that agents behave, and the state of environment changes at discrete time instances. Therefore, the MAS randomly and independently selects agent A_1 or A_2 that takes action at each turn. Let θ denote the probability that A_2 is selected at each turn and θ be a design parameter that determines the behavior of the entire system of the proposed MAS method. Because there are two agents, the probability that A_1 is selected at each turn is $1 - \theta$. If there is no environment state (topology) that satisfies the constraint C and has not appeared as a result of the action of the selected agent at each turn, the other agent takes action. If all the agents cannot take any action, one state of environment that has already appeared is randomly selected and the simulation process is re-started from the selected state.

On the proposed MAS, we repeat the selection and action of agents $K - 1$ times, and we can obtain K candidate topologies satisfying the constraint C by outputting the states of environment for $K - 1$ turns as well as the initial state, i.e., $s_0, s_1 \ldots, s_{K-1}$. In the candidate topology set generated by the proposed MAS method, links could be set at any candidate positions, and there are no positions where links are always set. Therefore, the proposed MAS method is expected to construct candidate topologies with higher diversity than those obtained by the method proposed in Ref. [13].

When generating candidate topologies, we need to output them without duplication. Therefore, we need to check whether the new state of environment has already appeared when determining the action of each agent at each turn. The simplest way to check the duplication of environment state (topology) is to store states appearing and outputted at all turns in a table and to check all entries stored in the table at each turn. However, the required memory size and the calculation time linearly increase as the number of topologies generated increases, so this naive approach is difficult to apply when generating a large number of candidate topologies.

A Bloom filter (BF) is used to judge whether a key is a member of a set using a limited amount of memory and a limited number of memory accesses [2]. BF consists of k hash functions h_i with b bit hash space and a bitmap of 2^b bits that is reset to zero at the initial state. We assign an integer 1 to L for each of the L candidate positions for setting links and make a bitmap with L bit length in which each bit takes unity or zero when a link is set or is not set at the corresponding candidate position in the current state of environment. Using the BF, the proposed MAS method can judge that the target state of environment is a new topology that has not yet appeared, if there are one or more bits being set to unity among the k positions on the bitmap corresponding to k hash values $h_i(s)$ obtained from the current state of environment s. After k bits of the bitmap are checked, all of these k bits are set to unity.

By using the BF to check the uniqueness of the generated topologies, the MAS can perfectly avoid outputting duplicate topologies. However, there is a possibility that the MAS falsely judges that a new topology has already appeared because k bits of the bitmap corresponding to k hash values are possibly set as a result of updates of other topologies. Here, we design the BF parameters k and b to make η, the average loss probability of new topologies, less than or equal to δ, which is an arbitrary given parameter [15].

Let η_n denote the loss probability of a new topology after n updates of the BF bitmap. We then have $\eta = \sum_{n=0}^{K-2} \eta_n / (K-1)$ because we update the BF bitmap $K - 1$ times. From Ref. [5], the optimum value of k minimizing η_n is given by $k = 2^b \ln 2 / n$. Therefore, the optimum value of k changes as BF updates proceed. However, for simplicity, fixing k and setting k to k^* minimizing η are desirable. The optimum value of k decreases as n increases, and k_{K-2}, the minimum value of optimum k, almost agrees with k^* [15], so we simply set k as

$$k = \frac{2^b}{K-2} \ln 2.$$

k needs to take an integer, so we round k to the closest one.

After the 2^b-bit bitmap for n different topologies is updated, the probability of an arbitrary bit in the bitmap being set is $1 - (1 - k/2^b)^n$. Therefore, we have $\eta_n = \{1 - (1 - k/2^b)^n\}^k$, and from (6), we obtain η as

UNCORRECTED PROOF

Q1 Please cite this article in press as: N. Kamiyama, Generating desirable network topologies using multiagent system, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.07.019

$$\eta = \frac{1}{K-1}\sum_{d=0}^{K-2}\left\{1-\left(1-\frac{\ln 2}{K-2}\right)^d\right\}^{\frac{2^b \ln 2}{K-2}}.$$

From (7), we can obtain the minimum value of b satisfying $\eta \leqslant \delta$ for the given K. Moreover, from (6), we can set k.

4.4. Reducing calculation time

Now, we discuss the required calculation time to generate K candidate topologies using the proposed MAS method. We need $O(N^2)$ time to check the connectivity between all pairs of nodes in the normal operation, and we need $O(yN^2)$ time to derive ξ, where y is the number of links constructing the topology. Because the number of candidate positions for setting links is given by $L = N(N-1)/2, O(N^2)$ time is necessary to judge the uniqueness of the target topology by using BF. Moreover, $O(N^3)$ time is required to derive ϵ because we need to obtain the shortest-hop routes between all pairs of nodes. Therefore, at each turn, we need $O(y(N^2 + N^2 + yN^2)) = O(y^2N^2)$ and $O(N^2N^3) = O(N^5)$ time to decide the actions of A_1 and A_2, respectively.

To select the action of agent A_2, the MAS needs $O(N^5)$ time, so the required calculation time rapidly increases as the network scale N grows. Thus, we investigate a method to reduce the calculation time required to determine the action of A_2. We can expect a large reduction of ϵ by setting a link between nodes i and j with large traffic ratio r_{ij}. To confirm this, Fig. 3 plots the correlation coefficient between ϵ on the topologies in which one link is added between each pair of nodes without a link and the product of the relative population of two nodes connected with the added link, against the node count N. We use 35 networks whose topologies are publicly available at the CAIDA web site, excluding one full-mesh network [3], and the figure shows the results for each of the 35 networks.

We observe a negative correlation between the two properties in all 35 networks. Although the correlation coefficient tends to be smaller in smaller-scale networks, the correlation coefficient is widely different among networks with similar node counts. We show the topologies of networks 2 and 5 in Fig. 4 as examples; we see that no hub node with high degree exists in network 2, whereas network 5 has some hub nodes. The relative population of hub nodes tends to be large in networks with hub nodes, e.g., network 5, so we can largely reduce ϵ by setting links at the positions connected with hub nodes. We can expect large reduction of ϵ in many topologies by setting a link between nodes with large populations, although the reduction effect depends on the shapes of topologies that appeared in the process of the proposed MAS method.

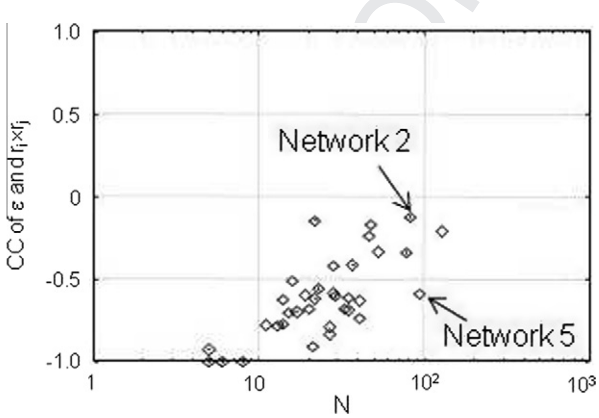


Fig. 3. Correlation coefficient between ϵ on the topologies in which one link is added between each pair of nodes without a link in original topologies and the product of the relative population of two nodes connected with the added link. By adding link between nodes with larger population, we can expect larger reduction of ϵ.

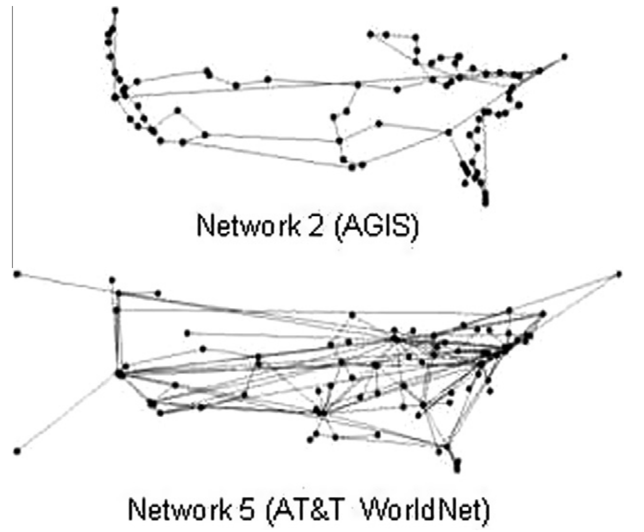


Fig. 4. Example of network topologies. No hub node with high degree exists in network 2, whereas network 5 has some hub nodes.

Hence, to reduce the calculation time when determining the action of A_2, we repeat the addition of one link at a candidate position in descending order of r_{ij} and judge by using BF whether the obtained topology has not appeared before, until we obtain a new topology. By using this method, we can reduce the calculation time required to select the action of A_2 to $O(N^3)$ time, and the total calculation time of the proposed MAS method is $O(Ky^2N^2 + KN^3) = O(KN^4)$, assuming that y, the link count of the generated topology, is similar to N.

4.5. Possibility of generating all candidates

In the proposed MAS method, candidate topologies are generated by adding or removing one link from the initial topology s_0. Therefore, some topologies satisfying the constraint C might never be generated using the proposed MAS method. However, the following theorem is formed.

Theorem. *Possibility of generating all candidates satisfying constraint C.*

From s_0, the initial topology satisfying the constraint C, any topologies satisfying the constraint C can be generated within a finite number of turns of the proposed MAS method.

Proof. Starting from topology i satisfying the constraint C, we assume that topology j satisfying the constraint C is generated by repeating the action of adding or removing one link. We define the topology set T_{ij} as that of topologies generated between topologies i and j. We can consider multiple combinations of topologies as T_{ij}. If all the possible T_{ij} include one or more topologies that do not satisfy the constraint C, topology j will never be generated after topology i is generated with the proposed MAS method because the environment of the proposed MAS method must satisfy the constraint C. On the other hand, if all the topologies in one or more T_{ij} satisfy the constraint C, topology j can be generated within a finite number of turns from topology i with the proposed MAS method.

Now, we consider generating topology m satisfying the constraint C within a finite number of turns after the initial topology s_0 with the proposed MAS method. We define $\boldsymbol{G}_1$ as the set of links that exist in s_0 but not in m. On the contrary, we also define $\boldsymbol{G}_2$ as the set of links that exist in m but not in s_0. First, let us consider the case when $\boldsymbol{G}_1 = \phi$. We can find T_{s_0m} consisting of topologies generated by adding links of $\boldsymbol{G}_2$ individually to s_0. Topologies generated by adding links to s_0 satisfying the constraint C also satisfy the constraint C, so all the topologies in this T_{s_0m} also satisfy the constraint C.

Next, let us consider the case when $\boldsymbol{G}_1 \neq \phi$. Let m' denote the topology that is obtained by adding links of $\boldsymbol{G}_2$ to s_0 and m' satisfy the constraint C. We can also obtain m' by adding links of $\boldsymbol{G}_1$ individually to topology m, and there exists $T_{m,m'}$ consisting of only topologies satisfying the constraint C because topology m satisfies the constraint C. Therefore, we can find $T_{s_0m'}$, which contains topology m' and consists of only topologies satisfying the constraint C.

Therefore, we can generate any topology satisfying the constraint C within a finite number of turns starting from the initial topology s_0 satisfying the constraint C with the proposed MAS method. □

5. Numerical evaluation

In this section, we show the results of numerical evaluation when setting $\alpha_\xi = 0.02$ and $\delta = 0.01$.

5.1. Influence of agent selection probability

Using the node location and the node population of Nap.-Net.LLC, whose topology is publicly available at the CAIDA web site [3], we analyze the influence of θ, the selection probability of agent A_2 at each turn, on the topology set generated. There are $L = 15$ candidate positions for setting links because the node count of Nap.Net.LLC is $N = 6$. Fig. 5 shows the cumulative distribution (CD) of the two evaluation criteria V_1 and V_2 of the topologies generated by the proposed MAS method for three values of θ when generating $K = 100$ candidate topologies. The figure also depicts the CD of two criteria when all the topologies satisfying the constraint C were constructed by checking all 2^L possible patterns of link settings. Although we can construct $2^{15} = 32{,}768$ topologies, only 14,718 of them satisfied the constraint C.

Although only a limited number of topologies, i.e., less than 1% of 14,718 possible candidates, were constructed when setting $K = 100$, we confirm that the proposed MAS method constructed many desirable topologies with small values of V_1 and V_2. As θ

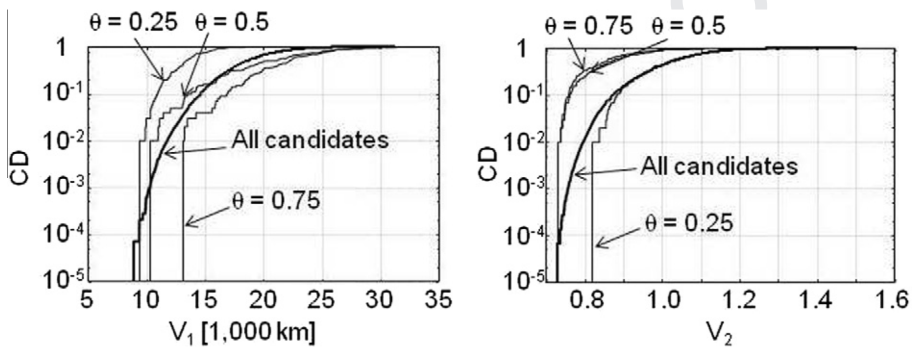


Fig. 5. Cumulative distribution of V_i in candidate set obtained by proposed MAS method when generating $K = 100$ topologies. Although only a limited number of topologies, i.e., less than 1% of 14,718 possible candidates, were constructed, the proposed MAS method constructed many desirable topologies with small values of V_1 and V_2.

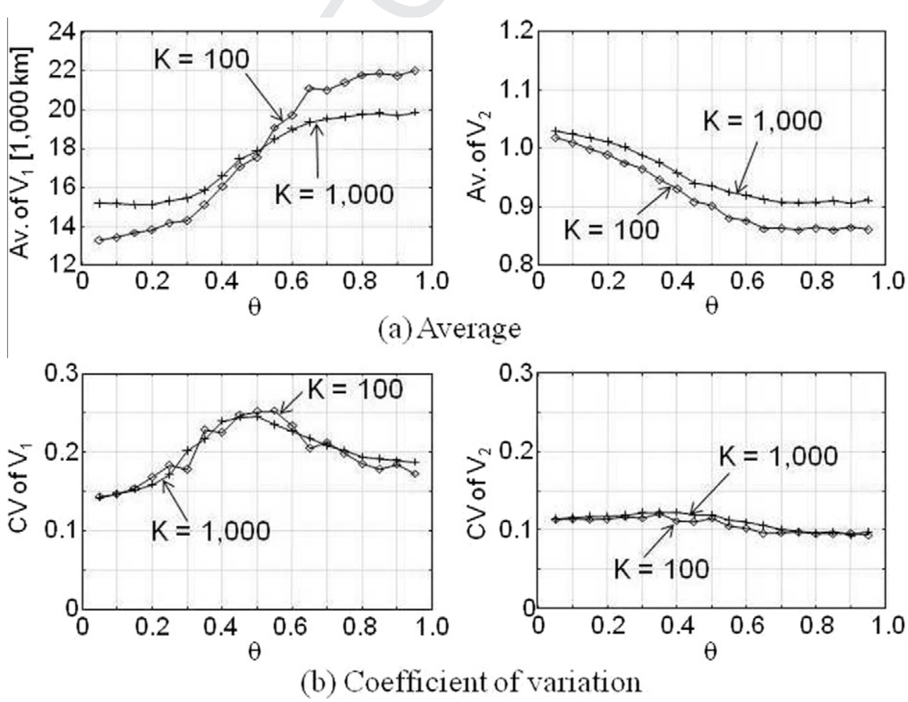


Fig. 6. (a) Average of V_i in obtained candidate set. The average of V_1 increased, whereas the average of V_2 decreased as θ increased. (b) CV of V_i in obtained candidate set. The CV of V_2 in the generated topology set was almost constant. On the other hand, the CV of V_1 in the generated topology set took the maximum value at around $\theta = 0.5$ and decreased as θ approached zero or unity.

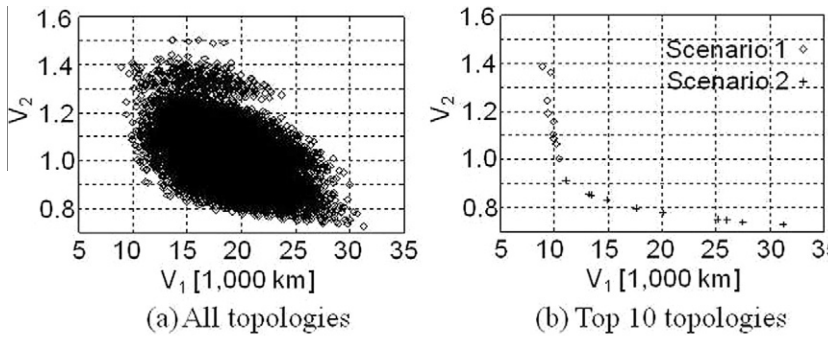


Fig. 7. (a) Scattergram of V_1 and V_2 of all 14,718 candidates satisfying constraint C. There are topologies with diverse values of V_1 and V_2. (b) Scattergram of V_1 and V_2 of top 10 candidates when AHP applied to all 14,718 candidates. It is effective to construct candidate topologies with a desirable value for the weighted evaluation criterion and various values for the other evaluation criterion to suppress the influence on the AHP result when generating a limited number of candidate topologies.

decreased, topologies with smaller V_1 and larger V_2 tended to be generated because agent A_1, which tried to optimize V_1, had more chance to take action. On the other hand, as θ increased, agent A_2 had more chance to take action, so topologies with smaller V_2 and larger V_1 tended to be generated.

Fig. 6(a) plots the average of V_1 and V_2 of the obtained candidate topologies against θ when setting $K = 100$ or 1000. We also show the coefficient of variation (CV) of each evaluation criterion of the obtained topology set in Fig. 6(b). We constructed 10 topology sets using the proposed MAS method for each value of θ, and we show the average results of these 10 trials. We also confirm that the average of V_1 increased, whereas the average of V_2 decreased as θ increased. This tendency strengthened as the generated topology count K decreased, so the influence of θ on the topology set generated was stronger as we set K to a smaller value to reduce the calculation time. The CV of V_2 in the generated topology set was almost constant. On the other hand, the CV of V_1 in the generated topology set took the maximum value at around $\theta = 0.5$ and decreased as θ approached zero or unity. As a result of the addition and removal of links being balanced, the diversity of V_1 in the generated candidate topology set was maximized when $\theta = 0.5$.

5.2. Optimality of constructed topology set

By appropriately setting K, the number of candidate topologies generated, in the proposed MAS method, we can obtain the candidate topology set within a practical time frame even for large-scale networks. However, the evaluation result of AHP depends on the candidate topology set, and it is ideal to apply AHP to the candidate set including all topologies satisfying the constraint C. Because only a part, not all, of the topologies satisfying the constraint C are constructed as a candidate set when setting K to a small value, the topologies evaluated highly by AHP will deviate from the ideal ones when constructing all the candidates. Therefore, in this section, we compare the AHP result when constructing the candidate set using the proposed MAS method with that when generating all the candidates satisfying the constraint C. It is difficult to construct all the candidates within a practical time frame for large-scale networks, so we also use the node location and node population of Nap.Net.LLC consisting of only six nodes. We consider two AHP scenarios. Scenario 1 is the case where cost is more important than quality, whereas Scenario 2 is the case where quality is more important than cost. We set the weights of evaluation criteria as

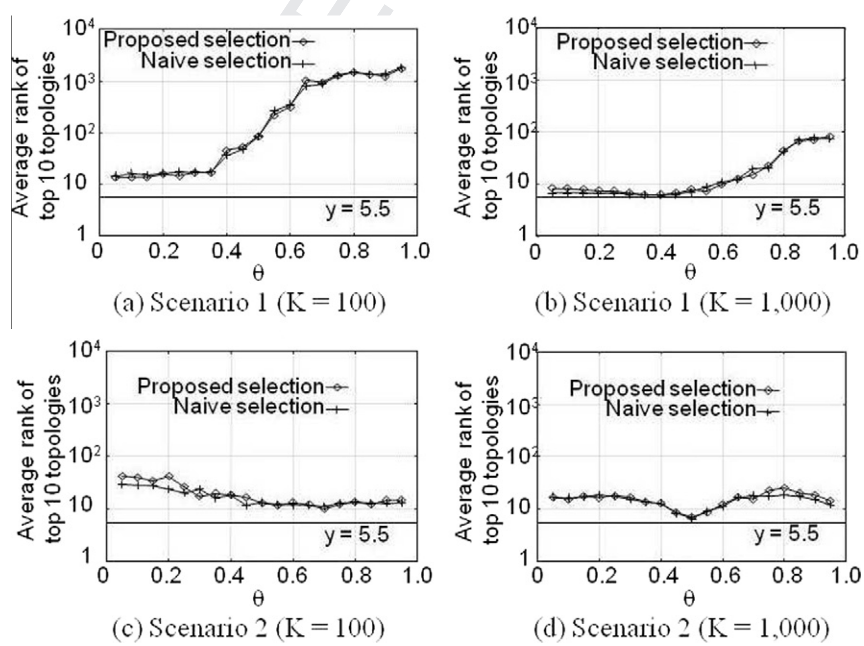


Fig. 8. Average rank of top 10 topologies when AHP was applied to candidate set generated by proposed MAS method. The influence of trying to set a link between nodes with a large population ratio when determining the A_2 action on the AHP result was negligible. In Scenario 1, the influence of limiting the number of candidate topologies generated in the proposed MAS method on the AHP result was negligible when θ was smaller than about 0.4. In Scenario 2, the average rank of the top 10 topologies was close to the ideal value 5.5 in the wide range of θ, and it took the minimum value at around $\theta = 0.5$.

Q1 Please cite this article in press as: N. Kamiyama, Generating desirable network topologies using multiagent system, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.07.019

$S_1^1 = 0.75$ and $S_2^1 = 0.25$ in Scenario 1 and as $S_1^1 = 0.25$ and $S_2^1 = 0.75$ in Scenario 2.

Fig. 7(a) shows a scattergram of V_1 and V_2 for all the 14,718 topologies satisfying the constraint C. Although we see a weak negative correlation between the two evaluation criteria, there are topologies with diverse values of V_1 and V_2. Fig. 7(b) depicts a similar scattergram for the top 10 topologies when the AHP was applied to all 14,718 candidates. In Scenario 1 weighting V_1, topologies with small V_1 and various V_2 were evaluated highly. In contrast, in Scenario 2 weighting V_2, topologies with small V_2 and various V_1 were evaluated highly. Therefore, we confirm that it is effective to construct candidate topologies with a desirable value for the weighted evaluation criterion and various values for the other evaluation criterion to suppress the influence on the AHP result when generating a limited number of candidate topologies.

Next, we plot the average rank of the top 10 topologies when AHP was applied to the candidate set generated by the proposed MAS method against θ in Fig. 8. In this figure, the y-axis is the average rank of these 10 topologies when all 14,718 candidates were evaluated by AHP. For each value of θ, we repeated 10 trials of the proposed MAS method and show the average results of these 10 candidate sets. When the evaluation ranks of the top 10 topologies when AHP was applied to the candidate set constructed by the proposed MAS method completely agreed with those when the AHP was applied to all the candidates, the average rank was $\sum_{i=1}^{10} i/10 = 5.5$, so we also show the ideal Result 5.5 in the figure. As mentioned in Section 4.4, the proposed MAS method tries to add a link between nodes with a large relative population when determining the action of agent A_2 to reduce the required calculation time. To see the influence of this modification on the AHP result, we also show the average rank of the top 10 candidates (denoted as *Naive selection*) when AHP was applied to the candidate set generated by deriving ϵ for all the possible candidate positions for setting links in the MAS. We confirm that the average rank of the top 10 candidate topologies when AHP was applied to the candidate set generated by the proposed MAS method was close to that of Naive selection, and the influence of trying to set a link between nodes with a large population ratio when determining the A_2 action on the AHP result was negligible.

As seen in Figs. 8(a) and (b), in Scenario 1, the average rank of the top 10 topologies was close to the ideal value 5.5, and the influence of limiting the number of candidate topologies generated in the proposed MAS method on the AHP result was negligible when θ was smaller than about 0.4. However, when θ was greater than about 0.4, the average rank of the top 10 topologies rapidly increased as θ increased and the AHP result was largely influenced by limiting the generated candidate topologies. This is because, as θ increased, the average of V_1 in the constructed candidate set increased, whereas the CV of V_2 in the constructed candidate set was almost constant, as seen in Fig. 6. In Scenario 1, it is effective to generate topologies with small V_1 over a wide range of V_2 as the candidate set, so the AHP result is desirable when setting θ to a small value. Therefore, we set $\theta = 0.25$ when generating the candidate topology set in Scenario 1 hereafter because we set the weight of V_2 to 0.25 in Scenario 1.

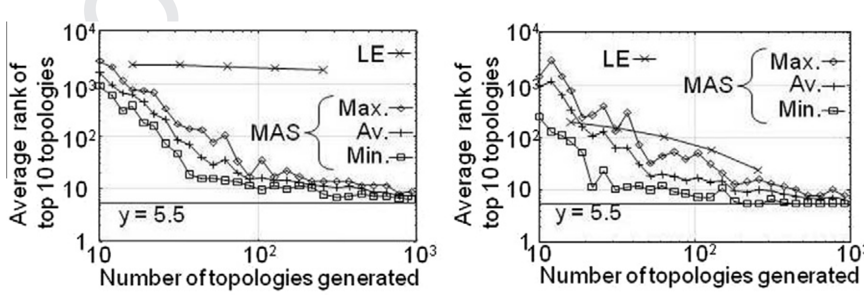
In Scenario 2, in contrast, as seen in Fig. 8(c) and (d), the average rank of the top 10 topologies was close to the ideal value 5.5 in the wide range of θ, and it took the minimum value at around $\theta = 0.5$. This is because the average of V_2 in the constructed candidate set gradually decreased as θ increased when θ was small, whereas the CV of V_1 in the constructed candidate set was maximized at around $\theta = 0.5$, as seen in Fig. 6. In Scenario 2, it is effective to generate topologies with small V_2 over a wide range of V_1 as the candidate set, so the AHP result was desirable when setting θ to around 0.5. Therefore, we set $\theta = 0.5$ when generating the candidate set in Scenario 2 hereafter.

5.3. Comparison with other candidate generation method

To confirm the superiority of the proposed MAS method for constructing the candidate topology set using MAS, we compared the MAS method with a previously proposed method [13] briefly summarized in Section 3 (denoted as *LE (limit edge) method*). Fig. 9 plots the average rank of the top 10 topologies against the number of candidates when AHP was applied to the candidate sets constructed using the MAS and LE methods, respectively. The y-axis is also the rank of these 10 topologies when all 14,718 candidates were evaluated using AHP. In the MAS method, we set θ to 0.25 or 0.5. Because the constructed candidate set is different for each trial in the MAS method, we show the minimum, maximum, and average values of the average rank in 10 trials of the MAS method. Although we can set the candidate count directly by parameter K in the MAS method, we can only set the candidate count to the power of 2 in the LE method because 2^x candidates are generated for a given x, the number of candidate positions for link setting.

When cost is more important than quality, i.e., Scenario 1, in the wide range of the candidate count, the MAS method can generate desirable candidate topologies with a smaller average rank. The superiority of the MAS method to the LE method was high, and the MAS method generated a candidate set superior to that of the LE method, even when the candidate count in the LE method was set as greater than that in the MAS method. When quality is important, although the superiority of the MAS method to the LE method decreases, the MAS method is still superior to the LE method when both can generate similar numbers of candidate topologies. The LE

UNCORRECTED PROOF



(a) Scenario 1 (b) Scenario 2

Fig. 9. Average rank of the top 10 topologies against the number of candidates when AHP was applied to the candidate sets constructed by the MAS and LE methods, respectively. It is desirable to use the MAS method when cost is important, even when the LE method can generate more candidates than the MAS method. When quality is important, although the superiority of the MAS method to the LE method decreases, the MAS method is still superior to the LE method when both methods can generate similar numbers of candidate topologies.

method is more effective than the MAS method only when the LE method can generate many more candidates than the MAS method and quality is more important than cost.

5.4. Comparison with other topology design method

We can design network topology by selecting a desirable topology evaluated highly by using AHP from the candidate set constructed using the proposed MAS method. To see the effectiveness of this topology design framework, we compared the results of this framework with those obtained by the topology design method using the generic algorithm (GA) proposed in [29]. The GA is a stochastic optimization heuristic in which explorations in solution space are carried out by limiting the population genetics stated in Darwin theory of evolution. We need to represent a solution to the problem as a genome consisting of binary strings. At each generation, the population comprises a group of N_p individuals (chromosomes) generated by the parent selection, genetic operations (crossover and mutation), and replacement.

We used the node location and population of Nap.Net.LLC in which there are $L = 15$ candidate positions for locating links. Each chromosome consisting of $L = 15$ bit binary strings was each candidate topology. We set the unique number from 1 to 15 at each of the $L = 15$ candidate positions for setting links, and there was a link at candidate position x if the binary string at x was unity; otherwise, there was no link at position x. Each chromosome was evaluated based on the objective function f, and we used V_1 or V_2 as f. Initially, N_p candidates satisfying constraint C were randomly selected. To produce N_p chromosomes at generation t, the following procedure was repeated [29]. First, two chromosomes were randomly selected from the population of generation $t-1$, and the chromosome with smaller f was selected. This was repeated twice, and we obtained two parent chromosomes. Next, with probability p_c, the selected parent chromosomes were recombined. This crossover operation was performed by randomly selecting the crossover point between unity and L and exchanging the portions of the two parent chromosomes beyond this point. The generated chromosome was inserted to the new generation chromosome set if it satisfied constraint C. Mutation was used to change the value of a gene to prevent the convergence of the solutions to bad local optima. With probability p_m, one bit of the gene at the randomly selected position was inversed. If the obtained chromosome satisfied constraint C, it was inserted to the new generation set. Finally, the N_p chromosomes with the smallest f were selected from N_p chromosomes of generation $t-1$ and N_p chromosomes of the new generation set as the population of generation t. The procedure of making each generation was repeated T times, and some candidates with the smallest f were outputted as the best candidate set.

Fig. 10(a) shows the average rank of V_1 of the 10 candidates with the smallest V_1 in the population at each generation for three values of N_p when the objective function f was V_1. Fig. 10(b) also depicts the average rank of V_2 of the 10 candidates with the smallest V_2 in the population at each generation when f was V_2. Because it is desirable to set p_c closet to unity and p_m to a small value [29], we set $p_c = 0.95$ and $p_m = 0.01$. We also set T to 100. When $N_p = 10$, the average rank of the top 10 candidates at each generation did not monotonically decrease, and the obtained candidates at the Tth generation were not desirable. However, even when we set N_p to 100, which was still much smaller than 14,718, the total candidate count satisfied the constraint C, we can obtain the desirable candidates close to the ideal ones with the average rank of 5.5.

Fig. 11 shows the topologies of the top six candidates with the smallest V_1 at the Tth generation when using the GA with $f = V_1$ and $N_p = 100$. Fig. 12 also shows the topologies of the top six candidates when AHP was applied to the candidate set generated using the proposed MAS method in Scenario 1. Although the obtained topologies from the GA were similar to those obtained using AHP and the proposed MAS method, we can also obtain more diverse candidates, such as ranks 5 and 6, when using AHP and the proposed MAS method. This is because we can consider the quality criterion V_2 even when V_1 was weighted in evaluation when using AHP. On the other hand, just a single criterion V_1 was considered when using the GA. This tendency was more noticeable when V_2 was optimized. Fig. 13 shows the topologies of the top six candidates with the smallest V_2 at the Tth generation when using the GA with $f = V_2$ and $N_p = 100$, and Fig. 14 shows the topologies of the top six candidates when AHP was applied to the candidate set generated using the proposed MAS method in Scenario 2. When using the GA, all the obtained candidates were close to the full-mesh topology in which links were provided at all the L candidate positions. This clarified that if we design the network topology considering just a single criterion, the other criteria seriously degrade in the obtained topologies in many cases. Although AHP also tends to emphasize candidates with excellent values in limited criteria, we can consider all the criteria simultaneously and obtain more moderate results.

5.5. Results using other evaluation criteria

As mentioned in Section 2.4, we assume two evaluation criteria: $V_1 = \zeta$, the total link length, and $V_2 = \epsilon$, the average hop distance between nodes weighted by the traffic ratio. To further investigate the effectiveness of the proposed method, we show the results when using another quality criterion, ν, the average end-to-end packet delay weighted by the traffic ratio, as V_2 instead of ϵ. Moreover, we also show the results when considering the three evaluation criteria, ζ, ϵ, and ξ.

First, we show the results when using the two criteria, $V_1 = \zeta$ and $V_2 = \nu$. The maximum value of X_e in the initial topology s_0 was 0.344, and we set κ to $\kappa = 0.95/0.344 = 2.76$. In other words,

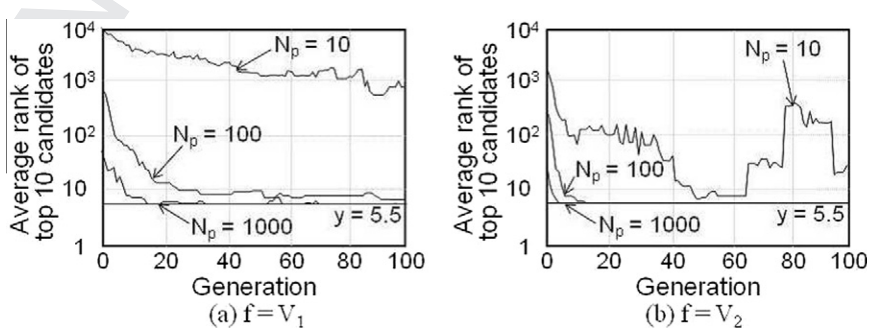


Fig. 10. Average rank of the top 10 topologies at each generation of GA. Even when N_p was just 100, we can obtain the desirable candidates close to the ideal set.

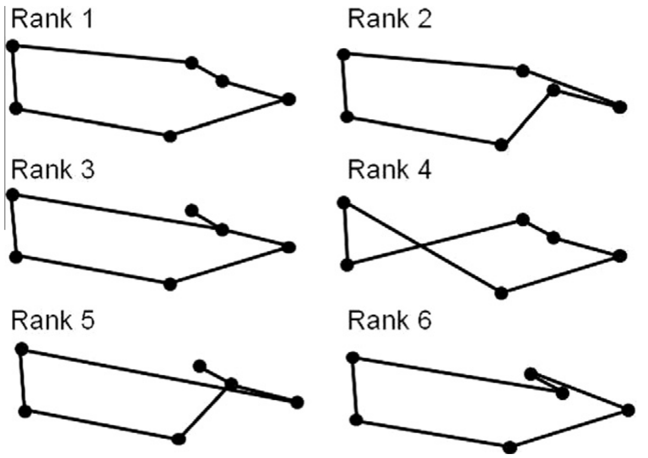


Fig. 11. Topologies of the top six candidates with the smallest V_1 at the Tth generation when using GA with $f = V_1$ and $N_p = 100$. V_2 were seriously degraded in the obtained topologies.

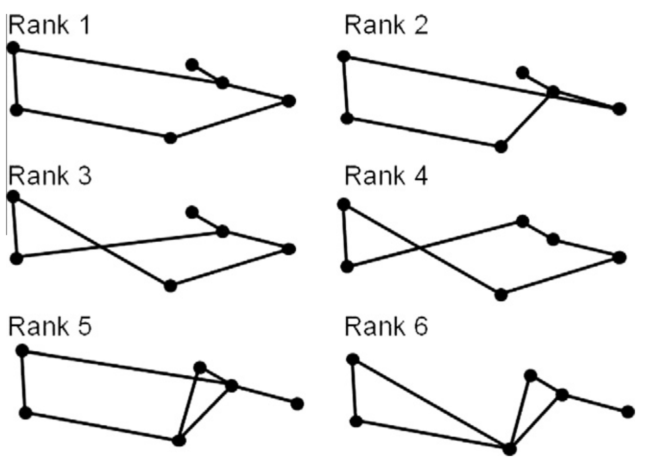


Fig. 12. Topologies of the top 6sixcandidates when AHP was applied to the candidate set generated by the proposed MAS method in Scenario 1. We can consider all the criteria simultaneously and obtain more moderate results.

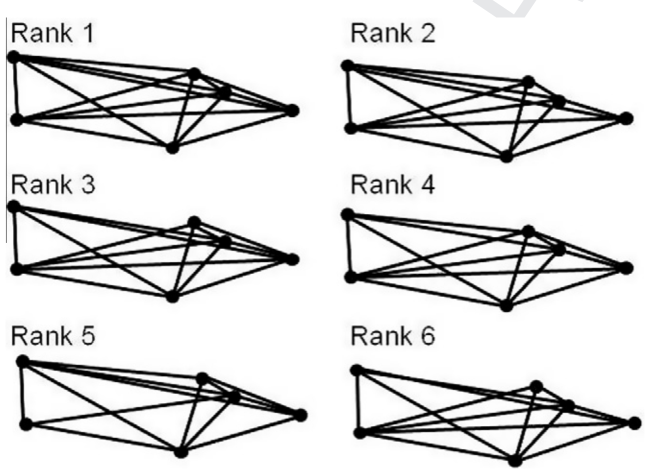


Fig. 13. Topologies of the top six candidates with the smallest V_2 at the Tth generation when using GA with $f = V_2$ and $N_p = 100$. V_1 were seriously degraded in the obtained topologies.

we set the maximum link utilization in s_0 to 0.95. We removed the topologies with links of ρ_e greater than or equal to unity from the

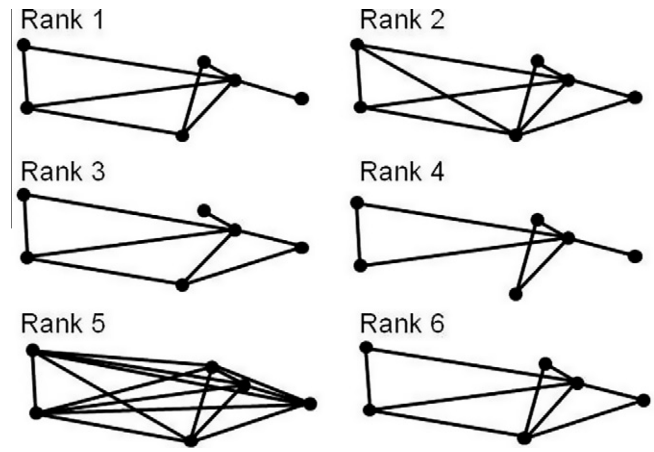


Fig. 14. Topologies of the top six candidates when AHP was applied to the candidate set generated by the proposed MAS method in Scenario 2. We can consider all the criteria simultaneously and obtain more moderate results.

candidate set. In the node locations and populations of Nap.-Net.LLC, 13,109 topologies satisfied constraint C and the constraint in which the utilization of all the links is less than unity. We also set the weights of evaluation criteria as $S_1^1 = 0.75$ and $S_2^1 = 0.25$ in Scenario 1 and as $S_1^1 = 0.25$ and $S_2^1 = 0.75$ in Scenario 2.

Figs. 15(a) and (b) plot the average rank of the top 10 topologies against θ when AHP was applied to the candidate set generated using the proposed MAS method. In Scenario 1, the influence of limiting the number of candidate topologies generated in the proposed MAS method on the AHP result was negligible when θ was smaller than about 0.5. In Scenario 2, the average rank of the top 10 topologies was close to the ideal value 5.5 when θ was larger than about 0.5 and K was 1000. Fig. 15(c) also plots the average rank of the top 10 topologies against K when setting $\theta = 0.3$ in Scenario 1 and $\theta = 0.7$ in Scenario 2. The average rank of these candidates was less than 10 when K was larger than about 200 in Scenario 1 and K was larger than about 700 in Scenario 2. We confirm that the influence of limiting the candidate count K by using the proposed MAS method on the AHP result was also negligible when the end-to-end delay was evaluated as one of the criteria. However, we needed to generate more candidates to suppress the influence on the AHP result in Scenario 2.

Next, we show the results when considering the three evaluation criteria, $V_1 = \zeta, V_2 = \epsilon$, and $V_3 = \xi$. For the constraint that candidate topologies must satisfy, we considered only the connectivity between all node pairs in normal operation, and 26,704 topologies satisfied this constraint using the node location and population of Nap.Net.LLC. ξ never increases with the addition of any link, whereas it never decreases with the deletion of any link. Therefore, agent A_3 calculates ξ for the topology in which a link is added at each of the candidate positions included in $\overline{\boldsymbol{E}}_n$, and it adds one link at the candidate position giving the minimum value of ξ. For the given parameter θ, we set the probabilities of selecting agents A_1, A_2, and A_3 to $1 - \theta, \theta/2$, and $\theta/2$, respectively.

We assume three AHP scenarios. Scenario 1 is where V_1 is more important than the other criteria, and we set the weights of evaluation criteria as $S_1^1 = 0.6, S_2^1 = 0.2$, and $S_3^1 = 0.2$. Scenario 2 is where V_2 is more important than the other criteria, and we set $S_1^1 = 0.2, S_2^1 = 0.6$, and $S_3^1 = 0.2$. Scenario 3 is where V_3 is more important than the other criteria, and we set $S_1^1 = 0.2, S_2^1 = 0.2$, and $S_3^1 = 0.6$.

Fig. 16(a)–(c) shows the average rank of the top 10 topologies against θ when AHP was applied to the candidate set generated using the proposed MAS method in each of the AHP scenarios. In Scenario 1, the average rank of the top 10 topologies was close to the ideal value of 5.5 when θ was less than about 0.6. In Scenarios

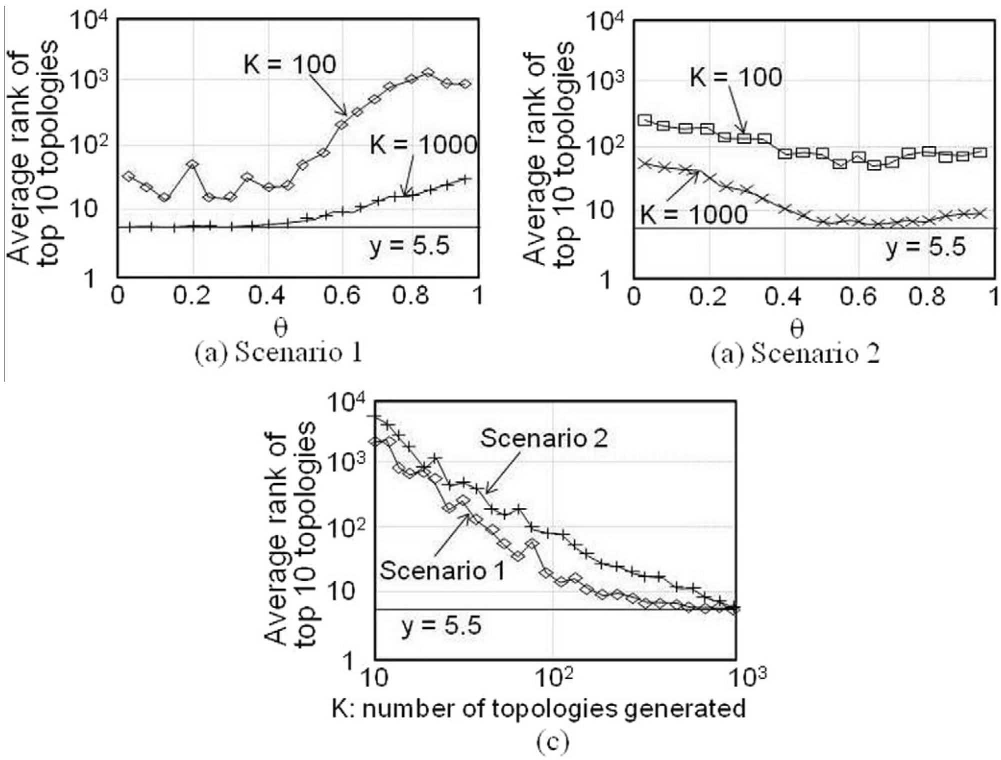


Fig. 15. Average rank of the top 10 topologies when AHP was applied to candidate set generated by the proposed MAS method using the weighted average end-to-end delay as V_2. In Scenario 1, the influence of limiting the number of candidate topologies generated in the proposed MAS method on the AHP result was negligible when θ was smaller than about 0.5. In Scenario 2, the average rank of the top 10 topologies was close to the ideal value 5.5 when θ was larger than about 0.5 and K was 1000. We needed to generate more candidates to suppress the influence on the AHP result in Scenario 2.

2 and 3, the influence of limiting the number of generated candidate topologies on the AHP result was also negligible when θ was around 0.6 and $K = 1000$. Fig. 16(d) also plots the average rank of the top 10 topologies against K, the number candidates generated. We set $\theta = 0.4$ in Scenario 1 and $\theta = 0.6$ in Scenarios 2 and 3. In all three AHP scenarios, the average rank of the top 10 topologies decreased as K increased, and it approached unity. We confirmed that the proposed MAS method can effectively generate a limited number of candidate topologies while suppressing the influence on the AHP result even when considering the three evaluation criteria.

5.6. Evaluation on various networks

In this section, we show the result of applying the MAS and LE methods to the node locations and node populations of 36 networks of commercial ISPs whose topologies are publicly available at the CAIDA web page [3]. Table 2 summarizes the names, node counts N, and link counts M of these networks. Although these 36 networks consist of various-scale networks in which the node count N is from 5 to 126, we cannot construct all the candidate topologies within a practical time frame for networks with N exceeding 6, and we cannot investigate the average rank of the top 10 candidates when applied to all the candidate topologies, as in Sections 5.2 and 5.3. However, as described in Section 5.3, the MAS method is always superior to the LE method when cost is more important than quality, whereas the superiority of each method depends on the number of candidate topologies generated when quality is more important than cost. Therefore, we compare K_{max}, the maximum number of candidate topologies that the MAS or LE method can generate, with the constraint that the upper allowable limit of the calculation time is 600 s.

Fig. 17 plots K_{max} of each method against N for each network when setting $\theta = 0.5$ in the MAS method. We constructed the candidate set on a PC with a 2.6 GHz Pentium 4 CPU and 1 GB memory. For UUNET with $N = 126$, it took more than 600 s to construct the initial topology T_b by both the MAS and LE methods, so we show the results for the other 35 networks. For the two networks with $N = 5$, GetNet International and ipf.net, and one network with $N = 6$, Nap.Net.LLC, both methods can generate all the topologies satisfying the constraint C within 600 s. For the other 32 networks, only a part of the topologies can be generated within 600 s, and K_{max} decreased as N increased. The LE method can generate more candidate topologies when N is small, whereas the MAS method can generate more candidate topologies when N is large.

The total amount of calculation required in the MAS method is $O(KN^4)$, as described in Section 4.4. On the other hand, the total amount of calculation time in the LE method is $O(xN^5 + 2^xN^3)$ for a given x, the number of candidate positions for setting links [13]. The number of topologies generated in the LE method is 2^x, so the total amount of calculation time of the LE method is $O(\ln KN^5 + KN^3)$ by setting $K = 2^x$. For networks with small N, a large value is allowed for K, so the calculation time in the MAS method tends to be larger than that of the LE method because $O(\ln KN^5 + KN^3) \simeq O(KN^3)$. In contrast, for networks with large N, only a small value is allowed for K, so the calculation time of the LE method tends to be larger than that of the MAS method because $O(\ln KN^5 + KN^3) \simeq O(\ln KN^5)$. The number of candidate topologies that can be generated is especially small in large-scale networks, so it is important to increase the candidate count for large-scale networks. Therefore, we can conclude that the MAS method that

Q1 Please cite this article in press as: N. Kamiyama, Generating desirable network topologies using multiagent system, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.07.019

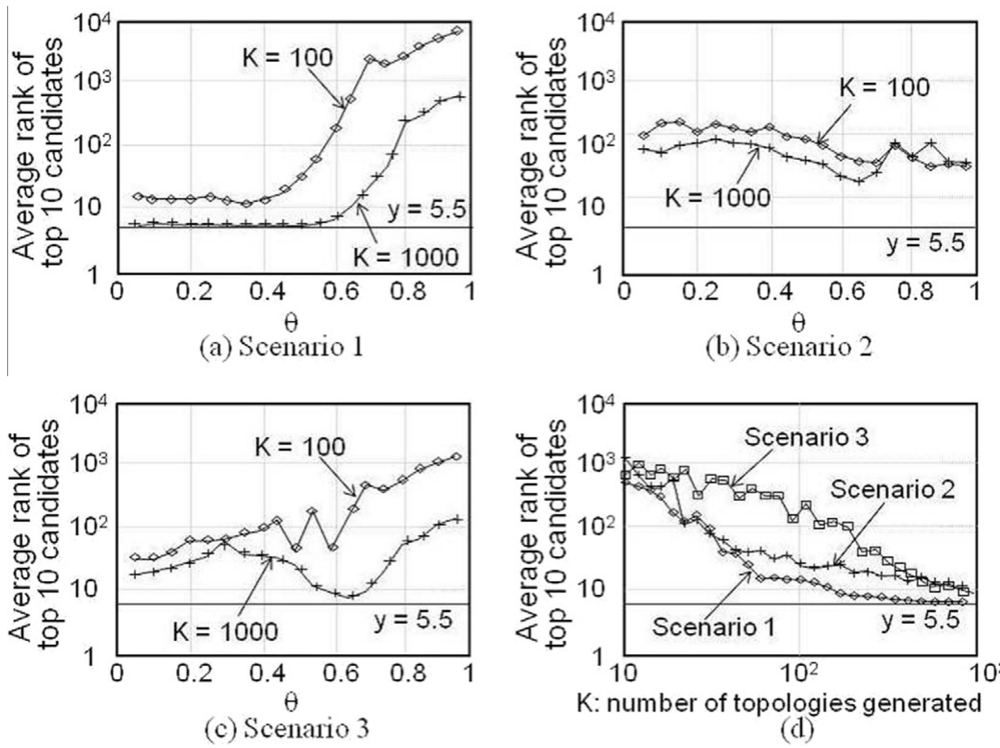


Fig. 16. Average rank of the top 10 topologies when AHP was applied to candidate set generated by the proposed MAS method using the three evaluation criteria. The proposed MAS method can effectively generate a limited number of candidate topologies while suppressing the influence on the AHP result even when considering the three evaluation criteria.

Table 2
36 ISP backbone networks.

	Network name	Node count N	Link count M
1	above.net	22	25
2	AGIS	82	92
3	Allegiance Telecom	53	88
4	At Home Network	46	55
5	AT&T WorldNet	93	154
6	BBN Planet	41	49
7	Cable & Wireless	19	33
8	CAIS Internet	37	44
9	CompuServe Network Services	16	23
10	CRL Network Services	35	50
11	DataXchange Network Inc.	8	24
12	EPOCH Networks Inc.	29	30
13	EUnet	28	30
14	Exodus	14	19
15	Genuity	48	53
16	GeoNet Communications Inc.	13	15
17	GetNet International	5	6
18	GlobalCenter	9	36
19	GoodNet	27	58
20	IDT Corp	15	18
21	ipf.net	5	5
22	iSTAR Internet Inc.	20	22
23	MindSpring	41	45
24	Nap.Net.LLC	6	7
25	Netrail Incorporated	17	21
26	PSINet	78	110
27	Qwest	14	26
28	RNP	27	35
29	Savvis Communications	28	56
30	ServInt Internet Services	23	34
31	Sprint	22	39
32	Telstra Internet	21	24
33	UUNET	128	321
34	Verio	35	72
35	VisiNet	11	13
36	XO Communications	33	38

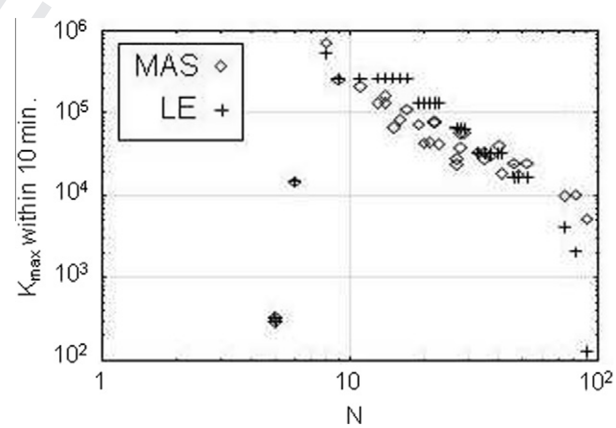


Fig. 17. Maximum number of candidate topologies that the MAS or LE method can generate, with the constraint that the upper allowable limit of the calculation time is 600 s. The LE method can generate more candidate topologies when N is small, whereas the MAS method can generate more candidate topologies when N is large.

can generate more candidate topologies for large-scale networks is superior to the LE method.

6. Conclusion

When evaluating network topologies by using AHP, we need to construct the candidate topology set prior to the evaluation. We proposed a method generating diverse candidate topologies using the multiagent system (MAS) within a practical time frame. Although MAS is a method analyzing the achieved environment as a result of interoperation among multiple agents acting autonomously, we can generate many topologies that are evaluated highly by AHP within a limited time length by correlating topologies with the environment states and evaluation criteria with the agents. Through a numerical evaluation, we confirmed that the proposed construction method for candidate topologies using MAS can generate more diverse topologies and suppress the influence of limiting the candidate count on the AHP result compared with the method previously proposed by the authors that limited the candidate position for setting links to reduce the calculation time.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.comcom.2012.07.019.

References

[1] D. Banerjee, B. Mukherjee, Wavelength-routed optical networks: linear formulation resource budgeting tradeoffs and a reconfiguration study, IEEE/ACM Trans. Netw. 8 (5) (2000) 598–607.
[2] B.H. Bloom, Space/time trade-offs in hash coding with allowable errors, Commun. ACM 13 (7) (1970).
[3] CAIDA web page, Available from: <http://www.caida.org/>.
[4] W.J. Cook et al., Combinatorial Optimization, John Wiley & Sons Inc., 1998.
[5] L. Fan, P. Cao, J. Almeida, A.Z. Broder, Summary cache: a scalable wide-area web cache sharing protocol, IEEE/ACM Trans. Network. 8 (3) (2000).
[6] N.G. Chattopadhyay, T.W. Morgan, A. Raghuram, An innovative technique for backbone network design, IEEE Trans. Syst. Man Cybernet. 19 (5) (1989).
[7] A. Gersht, R. Weihmayer, Joint optimization of data network design and facility selection, IEEE J. Selected. Areas Commun. 8 (9) (1990) 1667–1681.
[8] L.A. Goldberg, Efficient Algorithms for Listing Combinatorial Structures, Cambridge University Press, New York, 1993.
[9] B.L. Golden et al., The Analytic Hierarchy Process, Springer-Verlag, 1989.
[10] G. Huiban, G.R. Mateus, A multiobjective approach of the virtual topology design and routing problem in WDM networks, in: ICT International Conference on Telecommunications, 2005.
[11] N. Kamiyama, D. Satoh, Network topology design using analytic hierarchy process, in: IEEE ICC, 2008.
[12] N. Kamiyama, Efficiently constructing candidate set for network topology design, in: IEEE ICC, 2009.
[13] N. Kamiyama, Construction of candidate topologies for large-scale networks, in: IEEE GLOBECOM, 2009.
[14] N. Kamiyama, Constructing candidate network topologies using multiagent system, in: IEEE GLOBECOM, 2010.
[15] N. Kamiyama, T. Mori, R. Kawahara, Simple and adaptive identification of superspreaders by flow sampling, in: IEEE INFOCOM, Mini-Symposium, 2007.
[16] L. Kleinrock, Queueing Systems Volume I: Theory, Wiley-Interscience Publication, 1975.
[17] R.M. Krishnaswamy, K.N. Sivarajan, Design of logical topologies: a linear formulation for wavelength-routed optical networks with no wavelength changers, IEEE/ACM Trans. Netw. 9 (2) (2001).
[18] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, C. Diot, Characterization of failures in an IP backbone, in: IEEE INFOCOM, 2004.
[19] R. Ramaswami, K.N. Sivarajan, Design of logical topologies for wavelength-routed optical networks, IEEE J. Selected Areas Commun. 14 (5) (1996).
[20] T.L. Saaty, Fundamentals of Decision Making and Priority Theory, RWS Publications, 1994.
[21] T.L. Saaty, M.S. Ozdemir, Why the magic number seven plus or minus two, Math. Comput. Modell. 38 (2003) 233–244.
[22] T. Schelling, Micromotives and Macrobehavior, Norton & Company, 1978.
[23] K. Steiglitz, P. Weiner, D.J. Kleitman, The design of minimum-cost survivable networks, IEEE Trans. Circ. Theory CT-16 (4) (1969) 455–460.
[24] A.N. Tam, E. Modiano, A. Brzezinski, Physical topology design for survivable routing of logical rings in WDM-based networks, IEEE J. Selected Areas Commun. 22 (8) (2004).
[25] T. Uno, Enumeration algorithms and speeding up, in: 11th RAMP Symposium, 1999.
[26] H.R. Varian, Microeconomic Analysis, W.W. Norton & Company, 1992.
[27] Y. Wang, E. Keller, B. Biskeborn, J. Merwe, J. Rexford, Virtual routers on the move: live router migration as a network-management primitive, in: ACM SIGCOMM, 2008.
[28] G. Weiss, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, The MIT Press, 1999.
[29] E.C.G. Wille, M. Mellia, E. Leonardi, M.A. Marsan, Topological design of survivable IP networks using metaheuristic approaches, Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 191–206.
[30] K. Zhu, B. Mukherjee, Traffic grooming in an optical WDM mesh network, in: IEEE ICC, 2001.

UNCORRECTED PROOF