ELSEVIER

Computer Networks 38 (2002) 61–74

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Effectiveness of optimal node assignments in wavelength division multiplexing networks with fixed regular virtual topologies

Fai Siu, Rocky K.C. Chang *

Department of Computing, The Hong Kong Polytechnic University, Room PQ 829, Hung Hom, Kowloon, Hong Kong

Received 19 June 2000; received in revised form 1 March 2001; accepted 24 May 2001

Responsible Editor: E. Modiano

Abstract

In this paper, we consider the optimal node assignment problem in wavelength division multiplexing lightwave networks, which is to optimally assign network nodes to the locations in a regular virtual topology through wavelength assignments. Unlike previous work, which concentrated on a single virtual topology, we consider this problem as a class of problems by formulating it as a quadratic assignment problem. As a result, our objective is of a wider scope: identify the factors responsible for effective (or ineffective) node assignments. Optimal node assignments are considered effective if they could significantly improve the performance given by a random node assignment. The performance metric considered here is the average weighted hop distance. Based on a set of carefully designed experiments and analyses, we have concluded that variability in virtual topologies' hop-distance distributions, variability in network traffic distributions, and pattern matching between distance and traffic matrices are major factors in determining the effectiveness of optimal node assignments. In particular, optimal node assignments are most effective for linear virtual topologies and clustered traffic patterns. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Optical networks; Wavelength division multiplexing; Optimal node assignment; Quadratic assignment problem; Combinatorial optimization; Simulated annealing

1. Introduction

In this paper, we consider wavelength division multiplexing (WDM) lightwave networks in which network nodes are equipped with tunable transmitters and receivers (transceivers). An attractive usage of the wavelength tuning capability is to overlay virtual topologies on multihop WDM networks [1]. The physical network at the optical layer for realizing multihop networks can be a WDM ring (or other physical topologies) based on all-optical add/drop multiplexers, or a broadcast-and-select WDM network based on a passive star coupler. In the case of WDM rings, an add/drop multiplexer S can establish a lightpath (or a logical link) to another add/drop multiplexer D by performing the following steps. First, S sets its transmitter to a certain wavelength λ. All the intermediate add/drop multiplexers between S and D

* Corresponding author. Tel.: +852-2766-7258; fax: +852-2774-0842.

E-mail address: csrchang@comp.polyu.edu.hk (R.K.C. Chang).

1389-1286/02/$ - see front matter
PII: S 1 3 8 9 - 1 2 8 6 (0 1) 0 0 2 4 0 - 7

on the physical network, if any, are set to let the λ signal pass through them. Finally, the receiver at D drops the λ signal to the attached network node (see Ref. [2] for more details). In the case of broadcast-and-select WDM networks, two nodes form a lightpath between them simply by tuning their transceivers to the same wavelength.

For both cases, virtual topologies can therefore be embedded on the physical WDM networks by properly assigning wavelengths to the network nodes' transceivers. Virtual topologies are also referred to as logical topologies in previous work. Each link (or hop) on the virtual topology represents a lightpath between a pair of nodes, and it is uni-directional. Virtual topologies could be irregular or regular. A virtual topology is regular if all the nodes in the topology have the same nodal degree. Notable examples of regular virtual topologies are linear bus [3], ring [4], Shufflenet [5,6], deBruijn [7], Manhattan street network (MSN) [8], and GEMNET [9]. In Fig. 1, we show (a) a wavelength assignment in a broadcast-and-select WDM network, and (b) the corresponding virtual topology.

Moreover, a virtual topology may be *reconfigured* to adapt to changes in the network environment, such as traffic distribution. The reconfiguration can be accomplished by retuning the transceivers at the network nodes. One class of problems concerned is to find an *optimal* virtual topology based on performance metrics, such as maximizing link flow and minimizing packet delay [1]. These problems do not require the post-reconfiguration virtual topology to be the same as the pre-configuration virtual topology. Therefore, these problems are referred to as *arbitrary regular topology optimization problems* [10]. Furthermore, the reconfigurations may be performed at regular intervals [11] or on demand basis [12]. The former usually makes small changes to the virtual topology at each step, whereas the latter usually makes significant changes to the virtual topology. These problems cannot be solved in polynomial time and heuristic algorithms were proposed.

In this paper, we consider another class of reconfiguration problems—*optimal node assignment problem* (ONAP), or fixed regular topology optimization problem. Unlike the other reconfiguration problem, ONAP requires a fixed virtual regular topology to be maintained after reconfigurations. In other words, the post-reconfiguration virtual topology must be the same as the pre-configuration virtual topology. Thus, for a given fixed virtual regular topology, ONAP is to seek the

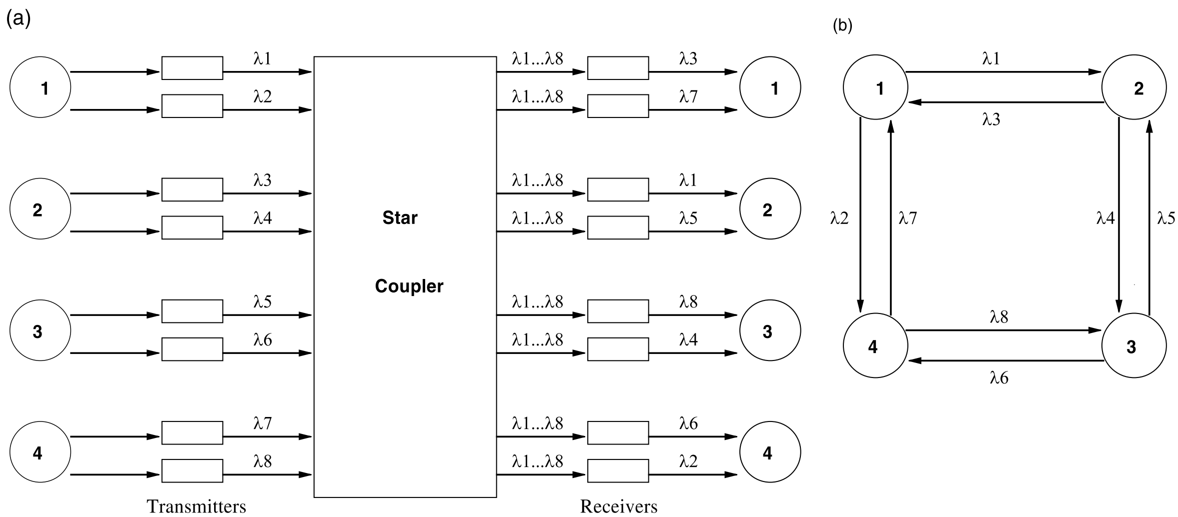


Fig. 1. Physical and virtual topologies of a broadcast-and-select WDM lightwave network.

best node assignment for the given topology based on performance metrics, such as maximum throughput or minimum delay.

Although the arbitrary regular topology optimization usually results in a higher performance improvement than that given by the optimal node assignment, an important advantage of the latter is the simplicity in packet routing. There are several packet routing problems associated with the arbitrary regular topologies. First, the routing algorithms are usually not scalable in terms of the number of messages required and the amount of routing information stored at the nodes. Second, the routing algorithms usually take certain time to converge, and this convergence time is incurred after every reconfiguration. Third, each node usually needs to perform table lookups to forward packets, thus incurring additional nodal processing time. The fixed regular topologies, on the other hand, is free of the aforementioned problems. In the LAN/MAN environment, the fixed regular topology is usually a well-understood structure, such as ring and torus. These topologies possess very simple routing algorithms. No routing information is stored at the nodes, and no message exchanges are necessary (except for periodic keepalive tests). Moreover, because the routing algorithm is associated with the topology itself, reassigning the nodes to other locations will not affect the routing algorithm in use (of course, the nodes' newly assigned locations need to be updated and made known to all other nodes).

2. Related work

The ONAP has been studied *separately* for a linear bus topology [3], a ring topology [4,11], and Shufflenet [5,6]. By reducing the problem to a minimum cut linear arrangement problem, it has been shown that the ONAP for the linear bus is an NP-complete problem [3]. As for the other two topologies, the authors only conjectured that these problems are also NP-complete but no formal proof was given [4–6]. Besides, it has been reported that the performance improvement obtained from reconfigurations decreases with the network size, but no explanations were given [3,6].

Unlike the previous studies of the ONAP for a particular virtual topology, we consider ONAP for any fixed regular virtual topologies as a class of problems. This is possible because we formulate the ONAP as a quadratic assignment problem (QAP) [13]. This formulation immediately shows that the ONAP is generally an NP-complete problem, because the QAP is known to be such, except for several special cases. As a result, our study is of a wider scope than the previous studies: Identify the factors responsible for the effectiveness (or ineffectiveness) of optimal node assignments for fixed regular virtual topologies. An optimal node assignment is considered effective if it can significantly improve the network performance when compared with a random node assignment, which is to assign nodes in a random fashion. This will help determine whether it is worthwhile to perform node reassignments at all. The results presented here are also expected to provide additional insights into previous results and for other fixed regular topologies that have not been studied before.

In particular, we will investigate the impacts of traffic patterns, virtual topologies, number of transceivers, and network sizes on the effectiveness of optimal node assignments. The performance metric selected here is the average of hop distance weighted by the proportion of traffic intensity. When the queuing delay is not significant, this metric also minimizes the packet delay, because propagation delay and node processing time are the dominating delay components [14]. Other possible objectives are to include the queuing delay component and maximizing the offered load, but we do not consider them in this paper [15,16]. We have chosen simulated annealing, which is known to produce better solutions than many other heuristics, to obtain suboptimal node assignments.

The rest of this paper is organized as follows: In Section 3, we first introduce the QAP formulation for the ONAP. In Section 4, we introduce a simulated annealing algorithm to solve the ONAP. In Section 5, we present numerical results and analyses from which we unearth the factors responsible for effective node assignments. Finally we conclude this paper in Section 6.

3. A quadratic assignment problem formulation

We consider a network of N nodes, each of which is equipped with P transceivers (P transmitters and P receivers). We configure these nodes into a fixed regular virtual topology by assigning proper wavelengths at the optical layer. The number of wavelengths required to configure a certain virtual topology depends on the virtual topology itself and the physical topology at the optical layer. For example, $N \times P$ wavelengths are needed for configuring any virtual topologies in a broadcast-and-select WDM network. However, this number could be much reduced in other physical topologies, because some wavelengths may be reused on different physical links. Impacts of wavelength restrictions on optimal virtual topology reconfigurations were discussed in Ref. [11]. In this paper, however, we assume that the number of wavelengths is plentiful ($N \times P$ is a sufficient number), so that we can configure any fixed virtual topology for the N nodes.

The objective of our ONAP is to configure the nodes into a fixed regular virtual topology, such that the average of hop distance weighted by the proportion of traffic intensity, or simply average weighted hop distance, is minimized. The hop distance refers to the number of links (wavelengths) required for a node to reach another node in the virtual topology. Clearly, the number of possible node assignments for a given virtual topology increases exponentially with N. In the following we present a mathematical formulation for the ONAP.

We first start with a *traffic matrix* to describe the traffic demands generated by the N network nodes. The traffic matrix is assumed to be known beforehand. We denote the traffic matrix by $\Gamma = [\gamma_{ij}]$, $i, j = 1, \ldots, N$, where γ_{ij} is the rate of traffic generated from node i and destined for node j, possibly measured in terms of bits/s. We further assume that $\gamma_{ii} = 0$, and γ is the sum of all γ_{ij}. In Section 5, we will use four different traffic patterns to study the effectiveness of optimal node assignments.

Since the objective function involves hop distance, we need to make assumptions about the routing algorithm employed by a virtual topology. The only assumption made for the routing algorithm is that it always gives the same hop distance from locations k to h in a virtual topology of N locations. The hop distance between the two nodes is denoted by l_{kh}, $k, h = 1, \ldots, N$. *A distance matrix* consists of the hop distances for all possible pairs of source and destination. We call this property *hop-distance invariant*. For example, we show in Fig. 2 an 2×4 MSN virtual topology, an (2,2) Shufflenet virtual topology, and their distance matrices. It is important to note that this assumption on the routing algorithm is realistic. Routing algorithms used for many regular topologies, including those considered in this paper, satisfy the hop-distance invariant property. Moreover, this assumption does not limit the routing algorithm to shortest-path ones. It also does not require the routing paths to be identical; that is, multiple equal-cost paths are allowed.

We also define binary-valued decision variables, x_{ik}, $i, k = 1, \ldots, N$. $x_{ik} = 1$ if node i is assigned to location k in the resulting virtual topology; otherwise, $x_{ik} = 0$. The final node assignment is therefore given by an *assignment matrix* $X = [x_{ik}]$. As a result, there is a lightpath between nodes i and j if $x_{ik} = 1$, $x_{jh} = 1$, and $l_{kh} = 1$. Otherwise, there will be multiple lightpaths between the two nodes. Finally, two constraints are clearly needed to ensure that each network node is assigned to only one location in the virtual topology, and each location in the virtual topology can be assigned with only one network node.

As a result, we present the mathematical formulation for the ONAP in Eqs. (1)–(4).

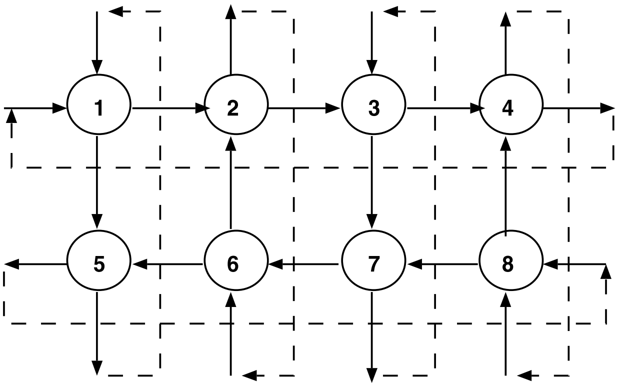
$$\min \frac{1}{\gamma} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{h=1}^{N} \sum_{k=1}^{N} \gamma_{ij} l_{kh} x_{ik} x_{jh}, \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{N} x_{ik} = 1 \quad \text{for } k = 1, 2, \ldots, N, \tag{2}$$

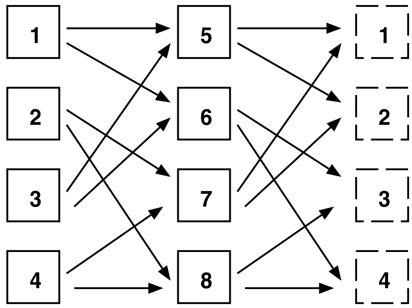
$$\sum_{k=1}^{N} x_{ik} = 1 \quad \text{for } i = 1, 2, \ldots, N, \tag{3}$$

$$x_{ik} \in \{0, 1\} \quad \text{for } i, k = 1, 2, \ldots, N, \tag{4}$$

where $El = (1/\gamma) \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{h=1}^{N} \sum_{k=1}^{N} \gamma_{ij} l_{kh} x_{ik} x_{jh}$ is the average weighted hop distance.



(a) An 2x4 MSN



(b) An (2,2) Shufflenet

To location

From location	1	2	3	4	5	6	7	8
1	0	1	2	3	1	2	3	2
2	3	0	1	2	2	1	2	3
3	2	3	0	1	3	2	1	2
4	1	2	3	0	2	3	2	1
5	1	2	3	2	0	3	2	1
6	2	1	2	3	1	0	3	2
7	3	2	1	2	2	1	0	3
8	2	3	2	1	3	2	1	0

unit : number of hops

(c) Distance Matrix of the 2x4 MSN

To location

From location	1	2	3	4	5	6	7	8
1	0	2	2	2	1	1	3	3
2	2	0	2	2	3	3	1	1
3	2	2	0	2	1	1	3	3
4	2	2	2	0	3	3	1	1
5	1	1	3	3	0	2	2	2
6	3	3	1	1	2	0	2	2
7	1	1	3	3	2	2	0	2
8	3	3	1	1	2	2	2	0

unit : number of hops

(d) Distance Matrix of the (2,2) Shufflenet

Fig. 2. An 2×4 MSN virtual topology, an (2,2) Shufflenet virtual topology, and their distance matrices.

The ONAP formulation is incidentally a QAP that seeks an assignment of N entities to N mutually exclusive locations in order to minimize a total quadratic interaction cost [13]. The QAP has found applications in varieties of problems, such as facility location problem, machine scheduling problem, keyboard design, and VLSI design problems [17]. As a result, many results and heuristic algorithms previously obtained for the QAP apply also to the ONAP, as we will see next.

4. A simulated annealing algorithm

The QAP is a combinatorial optimization problem, and the number of possible combinations grows exponentially with the problem size. It has been formally proved that the QAP is generally a strongly NP-hard problem [18]. However, polynomial solutions exist when the matrices have specific combinatorial properties, e.g., they belong to the Monge and anti-Monge matrices, Toeplitz and circulant matrices, sum and product matrices, and graded matrices [19]. Since the traffic matrix can take on any values, the ONAP is generally an NP-hard problem, regardless of the topologies. Thus the QAP formulation supports the conjectures for the ring and Shufflenet topologies given in Refs. [4,5].

Many heuristic algorithms have been proposed to solve QAP, and a comprehensive discussion of these methods is given in Ref. [19]. In particular, limited enumeration method, simulation method, and genetic algorithm may give better solutions, but their computational complexities are also much higher. Since our primary concern is to find a best node assignment, in this paper we employ a simulated annealing algorithm to generate (sub)optimal node assignments for fixed regular topologies. The algorithm is outlined in Fig. 3. We let $n(i)$ be node i's location in the virtual topology for $i = 1, 2, \ldots, N$, and Δ_{ij} be the reduction in El if the locations of two network nodes i and j are swapped ($n(i)$ and $n(j)$ are swapped). Therefore, $\Delta_{ij} > 0$ if there is a cost improvement after a node swapping, and $\Delta_{ij} \leqslant 0$, otherwise.

The simulated annealing algorithm overcomes the local optimality problem occurred in many heuristic algorithms. The main idea is to allow node swappings even when the cost of swapping is not improved. The probability of allowing node swappings under this situation is governed by a "temperature." Various simulated annealing algorithms reported in the literature differ in their choices of the initial temperatures, cooling rates, and stopping conditions. Here we adopt several guidelines established in the earlier studies to determine them. We set *MaxAttempt* to $10 \times N$, *MaxMove* to N [20], and the cooling rate α to 0.95 [21]. Our initial temperature T is computed based on the following formula suggested in Ref. [21]: $T = -\overline{\Delta_+}/\ln \chi$, where χ is the desired probability that a node swapping will be accepted for an initial solution, and $\overline{\Delta_+}$ is the average change of Δ_{ij} for those node swappings with $\Delta_{ij} > 0$ for the initial solution. We compute $\overline{\Delta_+}$ by randomly picking a number of neighbors of the initial solution, and computing the average cost change. Moreover, we set the value of χ to 0.6.

Complexity analysis of simulated annealing algorithms remains an open problem, and we are not aware of such studies, except for one that solved a maximum matching problem [22]. Moreover, previous studies showed that the simulated annealing procedure is sensitive to the annealing schedule and other control parameters [23].

5. Effectiveness of optimal node assignments

5.1. Numerical experiments

In this section we analyze how virtual topologies, traffic patterns, number of transceivers, and

1. Generate a random node assignment.
2. Initialize the temperature T.
3. While (*no_of_attempts* $<$ *MaxAttempt*) do
    While (*no_of_moves* $\leq$ *MaxMove*) do
        Randomly select two neighboring nodes i and j
        Compute Δ_{ij} if $n(i)$ and $n(j)$ were swapped.
        If $\Delta_{ij} > 0$, swap their locations; *no_of_attempts* $= 0$;
            increment *no_of_moves*
        If $\Delta_{ij} \leq 0$, swap their locations and increment *no_of_moves*
            with probability $e^{-\Delta_{ij}/T}$; increment *no_of_attempts*
    $T = T \times \alpha$

Fig. 3. A simulated annealing algorithm for the ONAP.

network size affect the performance gain obtained through assigning the nodes optimally. For the virtual topologies, we have selected MSN, Shufflenet, bi-directional ring, and bi-directional MSN (Torus) for the following reasons. First, all of them possess constant nodal degrees (constant number of transceivers), regardless of the network size, and the nodal degrees are either two (ring, MSN, and Shufflenet) or four (Torus). Second, all of them are basic topological structures upon which more complicated structures can be built. Third, they represent three common classes of topologies: linear (ring), mesh (MSN and Torus), and permutation (Shufflenet). Therefore, we believe that the conclusions derived from these four topologies will be useful for studying other topologies that are not considered in this paper.

We employ random, ring, clustered, and centralized traffic patterns for the investigation (see Fig. 4 for examples). Each γ_{ij}, $i \neq j$, for the random traffic pattern is independently generated according to a uniform distribution between 1 and 20. Each entry for the other three traffic patterns, on the other hand, takes on either a *high* or *low* traffic intensity. A high-intensity (low-intensity) entry is independently generated from a uniform distribution between 12 (1 for low-intensity) and 20 (7 for low-intensity). For the ring traffic patterns, $\gamma_{i(i+1)\mathrm{mod}\,N}$, $i = 1, \ldots, N$, is a high-intensity entry, and others, low-intensity entries. For the clustered traffic patterns, we generate two clusters of traffic: the *intra-cluster* and *inter-cluster*, and the former assumes a high intensity and the latter, a low intensity. For the centralized traffic patterns, the traffic to and from a *server node* is of high intensity, and the rest, low intensity. We choose node 5 to be the server node in Fig. 4.

We use $\mathrm{El}_{\mathrm{OA}}$ and $\mathrm{El}_{\mathrm{RA}}$ to denote the Els for node assignments obtained from the simulated annealing algorithm and for random node

From node \ To node	1	2	3	4	5	6	7	8
1	0	13	17	12	13	4	14	19
2	13	0	20	7	16	12	1	14
3	11	15	0	3	15	10	1	3
4	16	3	14	0	6	6	4	4
5	2	14	11	2	0	18	11	10
6	2	2	1	8	3	0	16	10
7	1	15	6	9	4	14	0	18
8	13	16	10	4	9	15	19	0

(a) Random Traffic Pattern

	1	2	3	4	5	6	7	8
1	0	14	7	1	3	1	3	3
2	6	0	14	2	2	2	1	3
3	1	6	0	14	4	4	4	5
4	2	2	4	0	12	8	4	7
5	1	5	2	1	0	16	2	4
6	1	5	4	7	1	0	12	7
7	4	2	5	3	3	6	0	14
8	15	4	5	7	5	5	1	0

(b) Ring Traffic Pattern

	1	2	3	4	5	6	7	8
1	0	13	17	12	4	7	1	3
2	15	0	14	15	6	6	4	4
3	14	15	0	15	1	6	4	4
4	15	19	18	0	2	4	5	5
5	5	4	7	1	0	12	20	13
6	2	1	6	2	14	0	20	16
7	5	4	4	1	12	12	0	13
8	5	7	7	6	14	15	15	0

(c) Clustered Traffic Pattern

	1	2	3	4	5	6	7	8
1	0	7	1	3	15	1	3	3
2	6	0	6	1	14	2	2	2
3	1	5	0	1	15	4	5	4
4	3	2	7	0	13	2	8	3
5	12	15	15	15	0	15	12	20
6	4	1	6	6	14	0	1	7
7	2	7	1	1	12	7	0	6
8	5	7	3	3	14	5	6	0

(d) Centralized Traffic Pattern

Fig. 4. Four types of traffic patterns (entries with values above 11 are highlighted).

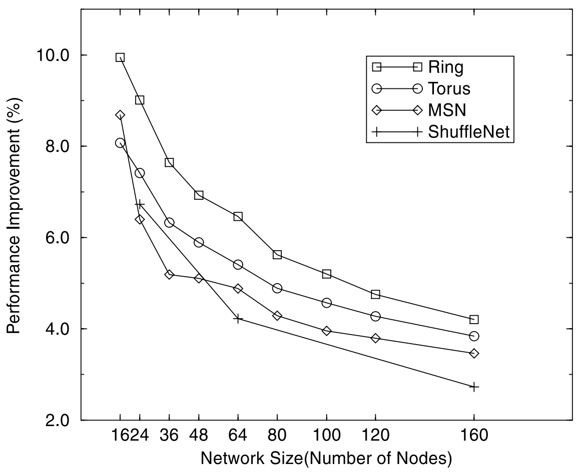


Fig. 5. PI for random traffic patterns.

assignments, respectively. The performance improvement (PI) obtained from the simulated annealing algorithm is computed according to

$$\mathrm{PI} = \frac{\mathrm{El}_{\mathrm{RA}} - \mathrm{El}_{\mathrm{OA}}}{\mathrm{El}_{\mathrm{RA}}} \times 100\%. \tag{5}$$

Figs. 5–8 show the numerical results for the four traffic patterns, and Figs. 9–12 for the four topologies. We have generated 50 samples for each type of traffic matrix, and used the same seed to generate the same set of random node assignments for each topology. The network size ranges from 16 to 160 nodes. The amount of data collected for

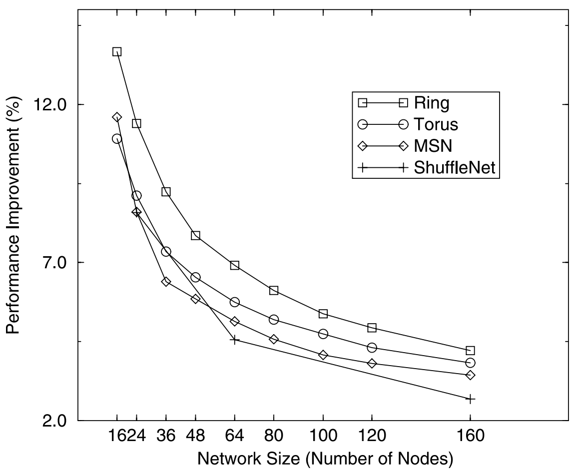


Fig. 6. PI for ring traffic patterns.

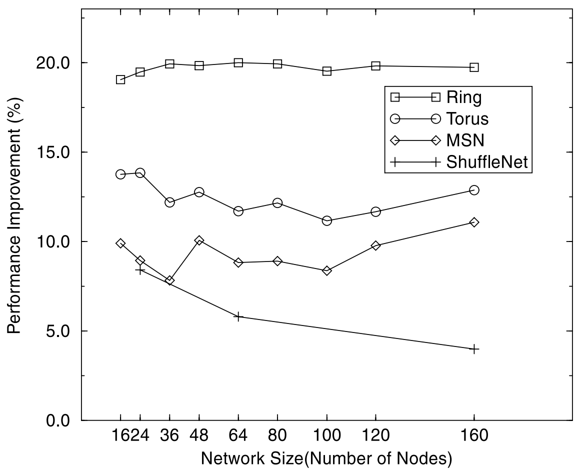


Fig. 7. PI for clustered traffic patterns.

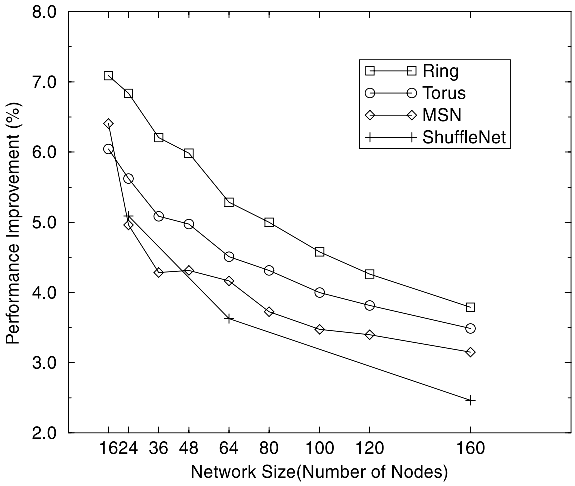


Fig. 8. PI for centralized traffic patterns.

an (p, k) Shufflenet is fewer than others due to the restriction on the number of nodes. In our simulation experiments we fix p to 2 and the network size grows rapidly from 24 ($k = 2$) to 160 ($k = 5$).

When comparing the PIs, it is important to point out that a higher PI does not imply that the corresponding case outperforms others in terms of the average weighted hop distance. It only says that optimal node assignments may significantly improve the performance of the case concerned. In other words, it is not worthwhile to perform optimal node assignments for the cases with low PIs.

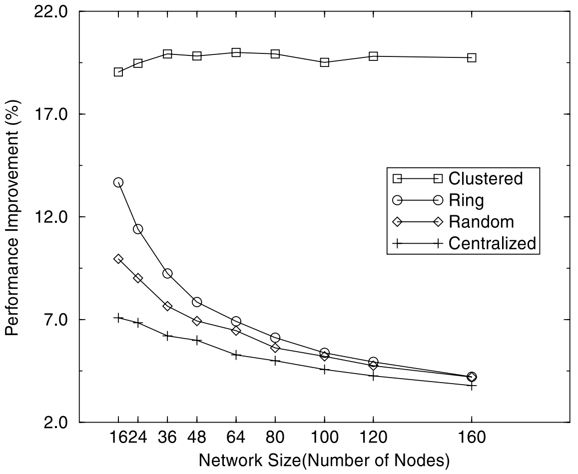


Fig. 9. PI for ring topology.

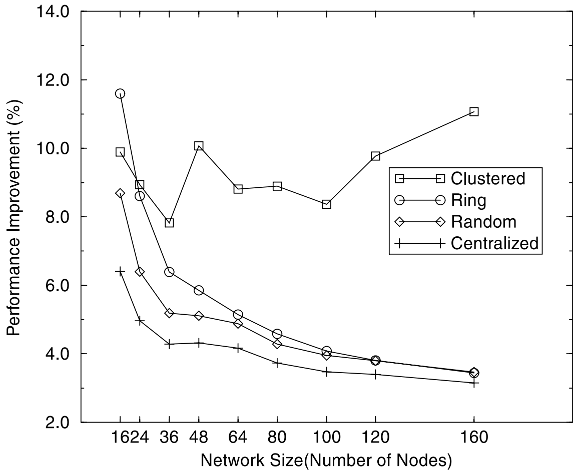


Fig. 11. PI for MSN topology.

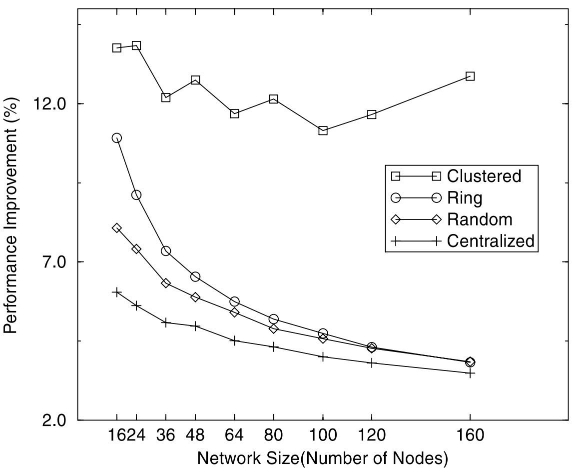


Fig. 10. PI for Torus topology.

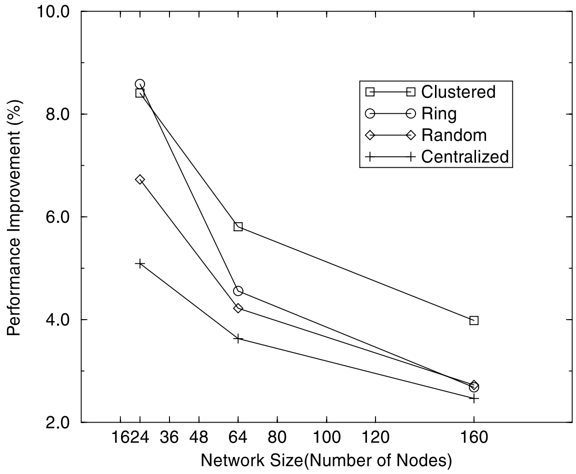


Fig. 12. PI for Shufflenet topology.

In the following we first list out several points observed from the numerical results, and then we interpret them in the next section. Points 1–3 can be observed from Figs. 5–8, and the rest from Figs. 9–12.

1. (Topology) Ring topology consistently gives the highest PI among the four topologies for each traffic pattern. In particular, Fig. 7 shows that as much as 20% PI can be obtained.
2. (Topology) When the network size is small (<64 nodes), Shufflenet and MSN yield similar PIs. But when the network size increases, MSN consistently gives a higher PI than Shufflenet.
3. (Number of transceivers) Torus generally performs better than MSN (except for $N = 16$), but the difference is only within a few percents.
4. (Traffic pattern) The PI attained for the clustered traffic pattern is generally higher than that for other three traffic patterns for all four topologies (except for $N = 16$ for MSN and Shufflenet topologies), and it is followed by ring, random, and centralized traffic patterns.

5. (Network size) The PIs for all traffic patterns, except for clustered traffic patterns, decrease with network size. They even converge to the same value of PI when N is sufficiently large. On the other hand, the PIs for the clustered traffic do not decline when the network grows in size, except when used with Shufflenet.

In relating to the previous studies on ONAP, the study on Shufflenet in Ref. [5] is closest to ours in terms of using the same objective function and similar traffic patterns. In particular, their traffic model was also based on a server node: The traffic from a nonserver node to any other node is a random number uniformly distributed between 0 and 1, but the traffic from a server node to other node is a random number uniformly distributed between 1 and a skew factor β. β was set to 10, 50, and 100 in their experiments. Their results for $\beta = 10$ and $N \geqslant 16$, which is closest to our experimental settings for the centralized traffic patterns, are comparable to ours: the PI is within 12% and it decreases with the network size. Furthermore, they showed that the PI increased with β, and the PI could reach 35% when $\beta = 100$.

In Ref. [4], the ONAP was studied for both unidirectional and bi-directional rings. The objective function considered there was throughput maximization. However, it is interesting to note that their findings agree with ours in that optimal node assignments were found to be most effective for clustered traffic patterns.

5.2. Discussion

In the following discussion, we are going to explain the aforementioned observations and give insight into the effectiveness of optimal node assignments under different scenarios. It is helpful to first understand how network performance can be improved through node reassignments. The general idea is to exploit the variance in the traffic intensity distribution and the variance in the hop-distance distribution. In general, El can be further reduced by assigning a relatively small hop distance (in the distance matrix) to a relatively high traffic intensity entry (in the traffic matrix). Since each location must be assigned to a node, the opposite is also true: Assigning a relatively large hop distance to a relatively low traffic intensity entry. This idea has been embedded into many heuristic algorithms, but the exact approach of implementing this idea varies. As a result, we expect that the explanation for the numerical results given in the following is independent of the choice of heuristic algorithm, provided that the algorithm gives a reasonably good solution to the ONAP.

5.2.1. Hop-distance distributions

For topologies with the same nodal degree, ring gives the best PI, then followed by MSN and Shufflenet. Given the same traffic matrix, one major factor in determining the PI is the variance of the topologies' hop-distance distributions—a high variance generally gives a high PI. In Table 1, we show Els for 24-node, 64-node, and 160-node topologies, assuming that γ_{ij} is the same $\forall i$, j and $i \neq j$. We also compute the standard deviation and a normalized (by the average hop distance) standard deviation (NSD) of the hop-distance distribution. For all cases, ring again gives the largest NSD and then followed by MSN and Shufflenet. This ordering is identical to the decreasing ordering for the PIs obtained in Figs. 5–8.

Table 1
Normalized standard deviations of hop-distance distributions

Topology	24 nodes			64 nodes			160 nodes		
	El	SD	NSD	El	SD	NSD	El	SD	NSD
Ring	6.00	3.49	**0.58**	16.00	9.25	**0.58**	40.00	23.10	**0.58**
Torus	2.50	1.19	**0.48**	4.00	1.73	**0.43**	6.50	2.78	**0.43**
MSN	3.17	1.34	**0.42**	4.94	1.97	**0.40**	7.38	2.83	**0.38**
Shufflenet	3.13	1.33	**0.42**	4.56	1.64	**0.36**	6.03	1.91	**0.32**

SD: standard deviation, NSD = SD/El.

As mentioned earlier, the general strategy is to exploit the variance in the hop-distance distribution. This exploitation becomes more effective when there is a high variance in the hop-distance distribution, because it is more likely to allocate smaller hop distances to the high-intensity entries by assigning the corresponding nodes closer to each other in terms of the hop distance. This explains the highest PI enjoyed by ring (observation 1). The observation 2 can be explained similarly based on the variance of hop-distance distribution. For 64 and 160 nodes, MSN gives a higher PI than Shufflenet because of a higher NSD for MSN. But for a smaller network size, e.g., 24 nodes, their NSDs are identical, therefore giving similar PIs.

5.2.2. Number of transceivers

From observation 3, Torus gives a better PI than MSN; however, the difference is not significant. The two share the same topological structure but with different nodal degrees. It turns out that the number of transceivers also affects the variances of hop-distance distributions of MSN and Torus. As shown in Table 1, Torus has a higher NSD than MSN although Torus has a lower average hop distance. But the difference is not as significant as that between Torus and ring. As a result, the impact of the number of transceivers on the effectiveness of optimal node assignment is not significant in this case.

5.2.3. Traffic intensity distributions

We have observed that the PI for the clustered traffic pattern is the highest when compared with the other three. For a given fixed regular topology, the PIs due to the traffic patterns are partially determined by the variance of the traffic intensity distribution—a high variance generally gives a high PI. In Tables 2 and 3, we compute the standard deviations for the four traffic patterns for a 80-node MSN and a 160-node MSN, respectively. We also tabulate the Els for a random node assignment and a (sub)optimal node assignment obtained from the simulated annealing algorithm, and the resulted PIs. Both tables show that the clustered traffic gives the highest variance and also the highest PI (observation 4).

Table 2
Performance improvement for a 80-node MSN

Traffic pattern	SD	El_{RA}	El_{OA}	PI (%)
Clustered	6.21	5.41	5.01	7.43
Ring	2.39	5.41	5.17	4.39
Random	5.77	5.43	5.20	4.38
Centralized	2.92	5.41	5.22	3.57

SD: standard deviation of a traffic distribution.

Table 3
Performance improvement for a 160-node MSN

Traffic pattern	SD	El_{RA}	El_{OA}	PI (%)
Clustered	23.33	5.41	4.89	9.53
Ring	8.57	5.43	5.14	5.43
Random	20.23	5.44	5.19	4.55
Centralized	10.15	5.41	5.21	3.87

SD: standard deviation of a traffic distribution.

The explanation for observation 4 is similar to that for the hop-distance distributions. As an extreme case, any node reassignment will not reduce the average weighted hop distance for uniform traffic distributions, where all the traffic matrix entries are identical. Having said that, we do not observe high PIs for the random traffic pattern in Tables 2 and 3 even though its standard deviation is comparable to that of clustered traffic. As we shall see in the next item, in addition to factors (A) and (C), we need to consider the combined effect of topologies and traffic patterns, to be explained next.

5.2.4. Pattern matching between traffic and distance matrices

Pattern matching between the distance and traffic matrices is another important factor in determining the effectiveness of optimal node assignments. In particular, we can base on this factor to further explain the ordering of the PIs for the four traffic patterns stated in observation 4.

We first consider the clustered traffic pattern. The PI for the clustered traffic pattern depends on how effective the topologies can be partitioned into two clusters, such that it only takes the nodes within a cluster a relatively short distance to reach each other. To quantify this factor for a fixed regular topology partitioned into two clusters, we have derived the intra-cluster average hop distance and inter-cluster average hop distance, labeled as

$\mathrm{El}_{\mathrm{intra}}$ and $\mathrm{El}_{\mathrm{inter}}$, respectively. Both the source and destination nodes are assumed to be in the same cluster when computing $\mathrm{El}_{\mathrm{intra}}$, and any node (except the source) in the cluster is equally likely to be the destination for the source. We also make the similar assumptions for computing $\mathrm{El}_{\mathrm{inter}}$. We use $\mathrm{El}_{\mathrm{inter}}/\mathrm{El}_{\mathrm{intra}}$ to measure how effective a topology is able to discriminate inter- and intra-cluster traffic. A high value would indicate that the topology is matched very well with a clustered traffic, and an optimal node assignment is expected to significantly improve the average weighted hop distance. We present the results for ring, Torus, and Shufflenet as follows:

- For a bi-directional ring of N nodes partitioned into two clusters of connected nodes, and the numbers of nodes in the clusters differ by at most one, $\mathrm{El}_{\mathrm{intra}}$ and $\mathrm{El}_{\mathrm{inter}}$ are given by

$$\mathrm{El}_{\mathrm{intra}} = \frac{\lfloor \frac{N}{2} \rfloor + 1}{3} \quad \text{or} \quad \frac{\lceil \frac{N}{2} \rceil + 1}{3}, \tag{6}$$

$$\mathrm{El}_{\mathrm{inter}} = \frac{1}{2}\left\lfloor \frac{N}{2} \right\rfloor + \frac{(\lfloor \frac{N}{4} \rfloor + 1)[2\lfloor \frac{N}{4} \rfloor + (\lceil \frac{N}{2} \rceil - 2\lfloor \frac{N}{4} \rfloor)(\lfloor \frac{N}{4} \rfloor + 2)]}{2\lceil \frac{N}{2} \rceil} \quad \text{for } N > 3, \tag{7}$$

  when N is a multiple of 4, $\lim_{N\to\infty} \mathrm{El}_{\mathrm{inter}}/\mathrm{El}_{\mathrm{intra}} = 9/4$.
- For a torus with N_x nodes on the horizontal dimension and N_y nodes on the vertical dimension partitioned into two clusters, the partition is performed vertically so that each cluster has $N_y \times N_x^c$ number of nodes, where $N_x^c = N_x/2$ when N is even, and $N_x^c = \lfloor N_x/2 \rfloor$ or $\lceil N_x/2 \rceil$ when N is odd. $\mathrm{El}_{\mathrm{intra}}$ and $\mathrm{El}_{\mathrm{inter}}$ are given by

$$\mathrm{El}_{\mathrm{intra}} = \frac{N_x^c}{N_x^c N_y - 1}\left\{ \left\lfloor \frac{N_y^2}{4} \right\rfloor + \frac{N_y(N_x^c - 1)}{2} \right\}, \tag{8}$$

$$\mathrm{El}_{\mathrm{inter}} = \frac{1}{N_y}\left\lfloor \frac{N_y^2}{4} \right\rfloor + \frac{1}{2}\left\lfloor \frac{N_x}{2} \right\rfloor + \frac{(\lfloor \frac{N_x}{4} \rfloor + 1)[2\lfloor \frac{N_x}{4} \rfloor + (\lceil \frac{N_x}{2} \rceil - 2\lfloor \frac{N_x}{4} \rfloor)(\lfloor \frac{N_x}{4} \rfloor + 2)]}{2\lceil \frac{N_x}{2} \rceil} \quad \text{for } N_x > 3, \tag{9}$$

  when $N_x = N_y$ and N_y is a multiple of 4, $\lim_{N\to\infty} \mathrm{El}_{\mathrm{inter}}/\mathrm{El}_{\mathrm{intra}} = 5/4$.
- For an (p, k) Shufflenet and p is even, the topology is partitioned into two equal halves horizontally. $\mathrm{El}_{\mathrm{intra}}$ and $\mathrm{El}_{\mathrm{inter}}$ are given by

$$\mathrm{El}_{\mathrm{intra}} = \frac{kp^k(p-1)(3k-1) - 2k(p^k-1) - 2k(p-1)}{2(p-1)(kp^k-2)}, \tag{10}$$

$$\mathrm{El}_{\mathrm{inter}} = \frac{kp^k(p-1)(3k-1) - 2k(p^k-1)}{2(p-1)kp^k}. \tag{11}$$

  Moreover,

$$\lim_{N\to\infty} \frac{\mathrm{El}_{\mathrm{inter}}}{\mathrm{El}_{\mathrm{intra}}} = 1.$$

The results above are very revealing and they explain the ordering of PIs in Fig. 7. The ring topology is outstanding in terms of discriminating the intra- and inter-cluster traffic and it is followed by Torus. Although we have not performed the analysis for MSN, which is much more complicated, it is expected to have a similar value as Torus. Shufflenet, on the other hand, does not discriminate the traffic at all in the way we partition the topology. There could be other less obvious ways of partitioning the topology in order to increase the value of $\mathrm{El}_{\mathrm{inter}}/\mathrm{El}_{\mathrm{intra}}$, but we do not expect that the value will be comparable to Torus.

On the other hand, topology partitioning does not seem to help for the other three traffic patterns. Nevertheless, the ring traffic pattern matches very well with all the four topologies in terms of their ability of embedding a uni-directional ring. The embeddings in the ring, MSN, and Torus are straightforward. Embedding of a uni-directional ring in Shufflenet is also possible. Thus, the internodal distance for each high-intensity entry is only one hop. However, these cases comprise only $1/N$ of all the traffic entries. As a result, this factor will benefit the ring traffic pattern, but the degree is expected to be lower than the case of the clustered traffic pattern.

The centralized traffic represents a greedy scenario in which every node attempts to make a shortest trip to a server node. Clearly, none of the fixed regular topologies can match well with this traffic pattern.

The fixed regular topologies are also unable to match with a random traffic pattern. This is reflected in the low PIs as shown in Tables 2 and 3 even though it exhibits the second highest standard deviation of the traffic intensity distribution.

5.2.5. Network size

It is well known that the difference between a worst QAP solution and an optimal QAP solution tends to zero as the problem size tends to infinity [24]. This asymptotic property was proved under the following conditions: (1) the entries in the distance matrix are mutually independent, (2) the entries in the traffic matrix are mutually independent, and (3) the entries in the distance matrix and that in the traffic matrix are mutually independent. It has been shown that for problem size of 50 nodes or more, the difference between an optimal solution and a worst solution is already very small [25].

However, the entries in the distance matrix for the ONAP are clearly not independent. Moreover, the entries in clustered, ring and centralized traffic matrices are also not independent. Nonetheless, we still observe a decline in PI as the network size increases (observation 5). We conjecture that this is also due to variances in the hop-distance and traffic distributions. We have observed from Table 1 that the NSDs of the hop-distance distributions for ring and Torus remain high as the network size increases, whereas Shufflenet and MSN give the largest decline in the NSD. Thus, it is expected that the PIs for Shufflenet and MSN decrease with the network size. On the other hand, the standard deviations for the ring and centralized traffic patterns decrease with the network size, thus causing a decline of PI when the network size grows. These two factors combined partially explain why the PIs for the clustered traffic pattern do not decline with the network size except when used with Shufflenet.

6. Conclusions

Formulating the ONAP as a QAP enables us to investigate the effectiveness of optimal node assignments for any fixed regular virtual topologies. We have identified three factors that have significant impacts on the effectiveness of optimal node assignments. They are the variances of virtual topologies' hop-distance distributions, the variances of traffic intensity distributions, and pattern matching between distance and traffic matrices. On the other hand, the impact from the number of transceivers is minimal for MSN and Torus.

Although we have not considered topologies and traffic patterns other than those considered in this paper, we believe that the findings obtained from this work have formed an adequate basis to predict results for other fixed regular topologies and traffic patterns. For example, we expect that optimal node assignments will not be effective for GEMNET, because this topology is essentially based on Shufflenet. On the other hand, it is important to perform optimal node assignments for linear bus topologies.

Acknowledgements

This work was partially supported by The Hong Kong Polytechnic University Central Research Grant 351/018. The authors would also like to thank Prof. Modiano and the three anonymous reviewers for their useful comments.

References

[1] B. Mukherjee, WDM-based local lightwave networks. Part II: multihop systems, IEEE Network Mag. 4 (1992) 20–32.
[2] A. Narula-Tam, E. Modiano, Load balancing algorithms for WDM-based IP networks, Proc. IEEE INFOCOM (2000) 1010–1019.
[3] S. Banerjee, B. Mukherjee, D. Sarkar, Heuristic algorithms for constructing optimized structures for linear multihop lightwave networks, IEEE Trans. Commun. 42 (1994) 1811–1826.
[4] S. Banerjee, B. Mukherjee, The photonic ring: algorithm for optimized node arrangements, Fiber Int. Opt. 12 (1993) 133–171.
[5] S. Banerjee, B. Mukherjee, Algorithms for optimized node arrangements in Shufflenet based multihop lightwave networks, J. High Speed Networks 4 (1995) 361–383.
[6] T.K. Chan, T.S. Yum, Active node placement in Shufflenets, Proc. IEEE INFOCOM (1994) 409–414.

[7] K.N. Sivarajan, R. Ramaswami, Multihop lightwave networks based on de Bruijn graphs, Proc. IEEE INFOCOM (1991) 1001–1011.
[8] N.F. Maxemchuk, Regular mesh topologies in local and metropolitan area networks, AT&T Tech. J. 64 (1985) 1659–1685.
[9] J. Iness, S. Banerjee, B. Mukherjee, GEMNET: a generalized, shuffle-exchange-based, regular, scalable, modular, multihop, WDM lightwave network, IEEE/ACM Trans. Networking 3 (1995) 470–476.
[10] J.-F.P. Labourdette, Traffic optimization and reconfiguration management of multiwavelength multihop broadcast lightwave networks, Computer Networks ISDN Sys. 30 (1998) 981–998.
[11] A. Narula-Tam, E. Modiano, Dynamic load balancing in WDM packet networks with and without wavelength constraints, IEEE J. Sel. Commun. (2000) 1972–1979.
[12] J.-F.P. Labourdette, A.S. Acampora, Logically rearrangeable multihop lightwave networks, IEEE Trans. Commun. 39(1991)1223–1230.
[13] G. Finke, R.E. Burkard, F. Rendl, Quadratic assignment problems, Ann. Discrete Math. 31 (1987) 61–82.
[14] J. Bannister, M. Gerla, Design of the wavelength-division optical network, Proc. IEEE ICC (1990) 962–967.
[15] B. Mukherjee, D. Banerjee, S. Ramamurthy, A. Mukherjee, Some principles for designing a wide-area WDM optical network, IEEE/ACM Trans. Networking 4 (1996) 684–696.
[16] J. Skorin-Kapov, J.-F.P. Labourdette, Rearrangeable multihop lightwave networks: congestion minimization on regular topologies, Telecommun. Sys. 9 (1998)113–132.
[17] R.E. Burkard, Some recent advances in quadratic assignment problems, Math. Programming (1984) 53–68.
[18] S. Sahni, T. Gonzalez, P-complete approximation problems, J. ACM 23 (1976) 555–565.
[19] E. Çela, The Quadratic Assignment Problem: Theory and Algorithms, Kluwer Academic Publishers, Dordrecht, 1998.
[20] D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon, Optimization by simulated annealing: an experimental approach. Part I: graph partitioning, Oper. Res. 37 (1989) 865–892.
[21] K. Nurmela, Constructing combinatorial design by local search, Research Report No. 27 in Series A, Department of Computer Science, Helsinki University of Technology, Finland, 1993.
[22] G.H. Sasaki, B. Hajek, The time complexity of maximum matching by simulated annealing, J. ACM 35 (1988) 387–403.
[23] M.R. Wilhelm, T.L. Ward, Solving quadratic assignment problems by simulated annealing, IIE Trans. 19 (1987) 107–119.
[24] W. Rhee, A note on asymptotic properties of the quadratic assignment problem, Oper. Res. Lett. 7 (1988) 197–200.
[25] J. Frenk, M.V. Houweninge, A.R. Kan, Asymptotic properties of the quadratic assignment problem, Math. Oper. Res. 10 (1985) 100–116.

Fai Siu received B.A. (Hons) and M.Phil. in Computing from The Hong Kong Polytechnic University in 1994 and 1997, respectively. From 1998 to 2000, she was an Assistant Computer Officer in the Computer Centre, The University of Hong Kong. She is currently with the Chinese Domain Name Corporation Ltd. Her technical interests are Internet and Java technology.

Rocky K.C. Chang received the B.Sc. degree in Electrical Engineering from Virginia Polytechnic Institute and State University in Blacksburg, Virginia, in 1983. He received the M. Engr. Degree in Electrical Engineering in 1985, the M.S. degree in Operations Research and Statistics in 1987, and the Ph.D. degree in Computer System Engineering in 1990, all from Rensselaer Polytechnic Institute, Troy, New York. From 1991 to 1993, he was with the Computer Science Department of IBM Thomas J. Watson Research Center, Yorktown Heights, New York. Since then he has been an Assistant Professor in the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (SAR). His research interests include performance evaluation of computer and communications systems, TCP/IP protocol design and analysis, and queueing theory. Dr. Chang is a member of IEEE and ACM.