PAPER

Optimal Wavelength Converter Placement in Optical Networks by Genetic Algorithm

Johannes Hamonangan SIREGAR[†], ***Nonmember***, **Hideaki TAKAGI**[††], ***Regular Member***, ***and*** **Yongbing ZHANG**[††], ***Nonmember***

SUMMARY In optical networks, wavelength converters are required to improve the efficiency of wavelength-division multiplexing. In this paper, we propose a genetic algorithm to determine the optimal locations of the nodes in the network where a given number of converters are placed. Optimality is achieved by the minimum wavelength blocking probability. Our algorithm is applied to two realistic networks constructed from the locations of major cities in Ibaraki Prefecture and from those in Kanto District in Japan and is shown to reach the nearly optimal solution in a limited number of generations. The accuracy is verified by simulation. The computational time is compared with that of an exhaustive search algorithm.

key words: *optical network, wavelength converter placement, genetic algorithm*

1. Introduction

In communication networks, the use of the optical technology has been popular for the need of wide bandwidth, high-speed transmission and a large number nodes. For optical transmission, wavelength division multiplexing (WDM) is proposed which has the ability to allocate many independent optical wavelengths on a single fiber link. One wavelength is dedicated to each channel between two adjacent nodes of the network. Interconnection between distant nodes is made possible by a set of available wavelengths over the path.

A lightpath is an optical communication path between a pair of source and destination nodes, and it may span multiple links. Because of the limitation on the number of wavelengths in each direct link, there may not be a through wavelength on all links of a path. With wavelength conversion capabilities inside the network, the nodes are capable of routing different wavelengths, which can be reused throughout the network to establish all the required connections [1]–[3]. A wavelength converter is a device that can be placed on the node where the wavelength conversion mechanism is assigned. It converts an incoming wavelength to a different outgoing wavelength. It can improve the utilization of wavelengths and lower the wavelength blocking probability compared with the case in which no wavelength conversion is allowed [3, Sect. 8.4.2].

We consider the converter placement problem as a kind of combinatorial optimization problem with constraints. Converters are placed in distinct nodes because the number of converters is limited. Each placement yields a blocking probability value of the network. The optimal solution will be found if a combined placement of converters has the minimum blocking probability value. The method of searching the optimal solution in a combinatorial problem is problematic due to the regional search of feasible solutions. Exploring the search space of feasible solutions to get an optimal solution has brought several algorithms to develop.

Approaches for the converter placement problem have been developed in exact as well as heuristic algorithms. Exact algorithms use exhaustive search that delivers an optimal solution, but they have to examine all combinations of converter placement [4], [5]. Heuristic algorithms are based on probabilistic performances with computational results such as random placement [1]. In [6], the authors use genetic algorithms (GAs) for placing full- and limited-range wavelength converters so as to minimize the blocking probability, which is evaluated by network simulator for each individual in the population. They conclude that "the approach of using GA . . . is extremely time consuming."

In this paper, we also propose a genetic algorithm to determine the set of nodes with full-range wavelength converters which gives the minimal blocking probability. Unlike [6], however, we employ the numerical evaluation of the approximate blocking probability values so that we can handle much larger networks than those in [6] in a relatively short time.

Genetic algorithms have been applied to various optimization problems, including those in the telecommunication field [6]–[8]. In general, the GA process uses a mechanism of natural selection from biology concept [9]–[11]. This is an evolution of individuals from one generation to the next, based on the elimination of weak individuals and the reproduction of strong individuals. The individuals are analogous to the possible solutions of the problem. The stronger individuals are related to the nearly optimal solutions that we search in optimization problems. These individuals will survive

Manuscript received May 11, 2001.

Manuscript revised January 16, 2002.

[†]The author is with the Doctoral Program in Policy and Planning Sciences, University of Tsukuba, Tsukuba-shi, 305-8573 Japan.

[††]The authors are with the Institute of Policy and Planning Sciences, University of Tsukuba, Tsukuba-shi, 305-8573 Japan.

over a number of generations until the strongest one remains, that is, an optimal solution of the problem is obtained. For reproducing individuals in the population, genetic operators such as selection, crossover, and mutation are used. These operators will explore more combinations of individuals which may lead to an optimal solution of the problem.

The rest of the paper is organized as follows. Section 2 contains formulation of the converter placement problem and evaluation of the blocking probability. The implementation of GA for the converter placement problem is given in Sect. 3. Results of experiments with GA and comparison of the GA solutions to those found by the exhaustive search algorithm and simulation are presented in Sect. 4. In Sect. 5 we make some concluding remarks.

2. Converter Placement Problem

In this section, we formulate the optimal converter placement problem for the WDM optical networks, show a procedure for evaluating the blocking probability, and discuss the exhaustive search algorithm.

2.1 Network Model

Following [5], we introduce a network model of the converter placement problem using graph theoretic terminology. We consider a directed graph $G = (V, L)$, where V is the set of vertices representing the network nodes and L is the set of directed edges representing the unidirectional fiber links in the network.

The traffic matrix of the network is given by $\{\lambda_{sd}\}$, where λ_{sd}, $s \neq d$, denotes the mean number of calls that arrive at source node s destined for destination node d per unit time. The call durations are assumed to be exponentially distributed with a mean taken as the unit of time. We assume that calls arrive at each node in a Poisson process. The path from node s to node d is predetermined for every pair of $s, d \in V$. Let F be the number of wavelengths on each direction of every link in the network. Every call requires a full wavelength on each link it traverses.

Let l_{ij} denote the directed link from node i to node $j \in V$. The link load ρ_{ij} per wavelength for link l_{ij} is defined as the probability that a given wavelength is occupied by a lightpath on link l_{ij}. Thus it is given by

$$\rho_{ij} = \frac{\sum_{s,d} \lambda_{sd}}{F}, \tag{1}$$

where the summation is taken over all (s, d) pairs such that the path from s to d traverses l_{ij}. It is assumed for the stability of the network that λ_{sd}'s are small enough so that $\rho_{ij} < 1$ for all $i, j \in V$.

2.2 Blocking Probability

Let N be the number of nodes in the network, out of which K nodes are equipped with converters. When a call arrives at a source node, a lightpath is established if at least one wavelength is available on every link it traverses on the path to the destination node, while the same wavelength must be used between the converter nodes. Otherwise it is blocked, for no alternative paths are allowed by assumption.

For the evaluation of blocking probabilities, we assume that the wavelength occupancy on each link is *statistically independent* of the occupancy of other wavelengths on the same link as well as that on other links. This "independence assumption" is commonly used in the analysis of blocking probability of optical networks [1]–[6], [12]. It yields the approximate values for the blocking probability very quickly (which is mandatory when used in each generation of GA). In Sect. 4 of this paper, we present the simulation result for a specific network example that validates the optimization by means of this assumption.

Let us first consider the path from node i to node j, which contains no converter nodes. Suppose that this path consists of successive links $l_{ii_1}, l_{i_1i_2}, \ldots, l_{i_nj}$, where $i_1, i_2, \ldots, i_n$ are the nodes between i and j along the path. A lightpath must use the same wavelength through these links. We define $\bar{\rho}_{xy} := 1 - \rho_{xy}$ for link l_{xy}, which is the probability that a given wavelength on l_{xy} is empty at an arbitrary time as well as at a time when a call arrives (due to a property of the Poisson process). Then, $\bar{\rho}_{ii_1}\bar{\rho}_{i_1i_2}\cdots\bar{\rho}_{i_nj}$ is the probability that a given wavelength is available on all links over the path from i to j. Hence the probability that a call successfully finds a lightpath on the path from i to j is given by

$$f(i, j) = 1 - (1 - \bar{\rho}_{ii_1}\bar{\rho}_{i_1i_2}\cdots\bar{\rho}_{i_nj})^F. \tag{2}$$

We next consider the path from source node s to destination node d that includes converters at nodes $c(1), c(2), \ldots, c(k)$. The set of these nodes is a subset of given converter placement C in the entire network. The probability of successfully establishing a lightpath on the path from s to d is given by

$$S_{sd}(C) = \prod_{i=0}^{k} f(c(i), c(i+1)), \tag{3}$$

where $c(0) := s$ and $c(k+1) := d$. Thus the blocking probability for the path from s to d is given by

$$P_{sd}(C) = 1 - S_{sd}(C). \tag{4}$$

Finally, the blocking probability over the entire network with converter placement C is given by

$$\Gamma(C) = \frac{\sum_{s,d \in V} \lambda_{sd} P_{sd}(C)}{\sum_{s,d \in V} \lambda_{sd}}. \tag{5}$$

This formula is used to select the optimal C that minimizes $\Gamma(C)$ in our genetic algorithm.

2.3 Exhaustive Search Algorithm

A straightforward way to solving the optimal converter placement problem is the exhaustive search of all combinations of converter locations. Given a network of N nodes with links and the number K of converters, it finds an optimal placement of converters that minimizes the blocking probability in the following steps.

Combinatorial enumeration: We produce the set of all $\binom{N}{K}$ converter placements.

Blocking probability calculation: For each combination, we calculate the blocking probability according to Eq. (5).

Optimal placement selection: We select a converter placement that yields the minimum blocking probability as the optimal placement.

This method surely leads to the globally optimal solution. Obviously, however, it is not an efficient way to solve the problem unless N and K are small.

3. Genetic Algorithm

In this section, we present a GA for the converter placement problem. Each individual in the population represents a possible converter placement as a string of bits. The objective is to find the placement that minimizes the blocking probability. The GA starts with creating the initial population. Better placements are selected over a number of generations. Operators such as crossover and mutation explore further possible placements. With proper constraint handling, the GA retains only feasible placements. When the stopping criterion is satisfied, an optimal placement is found.

3.1 Representation of Converter Placement

We use binary representation in encoding the possible solution. The converter placement is represented by an array of values 0 and 1. We can use this bit array for two alternative meanings.

In the first meaning of bit values, it represents the combination number in all converter placement combinations. For example, consider a network with 5 nodes and 2 converters to be placed. The number of combinations is $\binom{5}{2} = 10$, and $1 \le j \le 10$ corresponds to number j of the combinations. We construct the ordered set of 2-out-5 combinations as follows: number 1 is the converter placement at nodes 1 and 2, number 2 at nodes 1 and 3, and so on until number 10 at nodes 4 and 5. Thus we form the bit array of length 4. Since we have 10 combinations the size of search must not exceed 10. An infeasible solution occurs if the bit array has a value 0 or greater than 10.

The second meaning of bit values has a direct representation of the converter placement. For example, in a network with 5 nodes and 2 converters to be placed, vector solution $x = (x_j)$, $1 \le j \le 5$, means that if $x_j = 1$, node j is selected, and if $x_j = 0$ otherwise. The length of the array is 5, since there are 5 nodes in the network. The bit array 01100 means that it selects nodes 2 and 3 for converter placement. An infeasible solution occurs when the number of 1-bits exceeds the number of converters to be placed. The weakness of this representation can be solved by a constraint handling method. Another problem is that it produces sparse vectors when the number of converters is small in a network with a large number of nodes. For example, vector 0000011000 means that it selects nodes 6 and 7 out of 10 nodes in a network. Then this representation has high possibility to produce infeasible solutions.

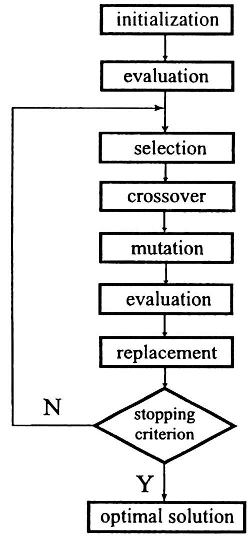


Fig. 1 Diagram of a GA process.

In our implementation of GA for the converter placement problem, we may employ either representation depending on the case. If the number of converters is small, it tends to produce many infeasible solutions; we then use the first representation. If the number of converters is large, we may not have too many infeasible solutions; we can then use the second representation.

3.2 GA Process

Let us elaborate each procedure of the GA process for searching the optimal combination of nodes with converters, as shown in Fig. 1.

Initialization: Each individual of the population is a possible solution to the problem. The GA starts with an initial population which is generated from the random seeds.

Evaluation of fitness value: The objective function to be minimized is the value of blocking probability given in Eq. (5). To maintain uniformity over the problem domain, we use a fitness value for the objective function normalized to a convenient range. The normalized objective function indicates the fitness of an individual that the selection uses to evaluate. Individuals with good fitness value will be selected for the next generation.

Selection: We use the tournament selection method for selecting two parents to produce new individuals for the next generation [9, p.121]. This selection leaves only those individuals with highest fitness values in the population.

Crossover: Crossover explores the diversity in the individual's bits. We use the uniform crossover operator by which a single child is created from two parents by copying each bit value from either parent [10, p.49]. The probability of crossover is given in the parameter setting. Using the random sequence of 0's and 1's generated with this probability, GA selects the parent from which each bit is copied.

Mutation: The mutation procedure is applied after the crossover on each child independently. Every child gets an opportunity for changing each bit value according to the probability of mutation. This probability is also given in the parameter setting. The mutation plays a role to restore lost genetic values when the population converges too fast [9, p.14].

Replacement: The new individual replaces the old individual or their parent in order to maintain a fixed population size if the fitness value of the new individual is higher than that of their parent. If the fitness value of the new individual is lower than that of their parent, the new individual is discarded and the next generation employs the old individual.

3.3 Constraints Handling

In our GA process, the initialization, crossover, and mutation procedures will breed new individuals with new fitness values. If they do not satisfy the constraint conditions, the GA produces infeasible solutions. We avoid producing infeasible solutions by changing the bits of binary representation in accordance to the constraint.

Let us refer back to the representation examples in Sect. 3.1. In the first meaning of representation, the feasible solution of bit array must be between 0001 and 1010 both inclusive. If the new individual does not fall in this region, the constraint is violated. To satisfy the constraint, some value must be added if the individual has a bit array 0000; some value must be subtracted from a bit array greater than 1010. In the second representation, the number of 1-bits denotes the number of converters. If it exceeds the given number of converters, the constraint is violated. To satisfy the constraint, some 1-bits must be changed to 0-bits. If the number of 1-bits is less than that of converters, some 0-bits must be changed to 1-bits. For changing the bit values, we manipulate each bit randomly until the constraint is satisfied.

The method of handling constraints modifies the GA process. It is incorporated in the initialization, crossover, and mutation procedures in order to produce only feasible solutions. Then the space of feasible solutions may include the optimal solution. The speed of convergence to the optimal solution depends on the probabilities of crossover and mutation. The crossover and mutation should maintain a certain degree of diversity in the population so as not to converge prematurely to some sub-optimal solution.

3.4 Stopping Criteria

A disadvantage in the optimization with GA is the difficulty of deciding when to stop [7]. Although statistical variables, such as average and best fitness values, are available in each generation, their values change almost unexpectedly as generations evolve. Stopping after a certain number of iterations with no improvement or when the change in average fitness is small may cause the algorithm to stop too early or too late.

Another stopping criterion may be that if the average fitness attains the value we expect then the iteration stops. In this case, the number of iterations may become large and the long computation may be needed.

In the present study, we have used a stopping criterion based on the number of generations. Our algorithm stops when the generation counter exceeds the preset maximum number of generations.

4. Experimental Results

In this section we show some numerical examples of using the GA for the converter placement problem. We have also conducted random-event simulation which simply counts the number of blocked calls out of all arriving calls over a long period. Recall that the evaluation of the blocking probability in each generation of our GA relies on the "independence assumption" mentioned in Sect. 2.2. Comparison with simulation shows that GA can attain nearly optimal solutions.

4.1 Construction of Network Examples

Before presenting the numerical results for specific network examples, some remarks may be in order for their construction and the preparation of the shortest paths, which are commonly used by GA and simulation.

Given an area in Japan, we have selected major cities in that area as the set of network nodes. The locations of those cities are extracted from NTT's public web site (http://www.ntt-east.co.jp/tariff/ryoukin/). Then, by the Delaunay triangulation [13, p.175] for the set of node locations, we get the set of links in the network. Thus we obtain a network topology. *We note that the resulting networks are by no means related to* *the real networks of NTT or any other companies.*

Given the network topology, we have applied Dijkstra algorithm [13, p.273] to determine the shortest path for every pair of the source and destination nodes. Because of high-speedness of optical transmission, we can assume that the actual length of each link between adjacent nodes does not affect the network performance. Therefore, we have simply used the number of links (hops) as the distance measure in the Dijkstra algorithm. A tie is broken in a specified manner.

4.2 Ibaraki Network

We first consider a network consisting of $N = 14$ nodes that represent major cities in Ibaraki Prefecture, which we call Ibaraki Network for the sake of convenience, as shown in Fig. 2.

In our experiment, we assume $K = 2$ and $F = 3$ for the number of converter nodes and the number of wavelengths on each link, respectively. We also assume that calls are generated at rate $\lambda_{sd} = 0.1$ for every pair of source and destination nodes in the network. The GA parameters are as follows:

- Population size = 20
- Probability of crossover = 0.6
- Probability of mutation = 0.00333
- Maximum number of generations = 20

We have conducted the experiment under Unix on a DEC Celebris GL workstation. Our GA starts with the initial generation which consists of 20 individuals according the parameter setting. Each individual represents a combination of converter nodes out of 14 nodes in the network. The fitness value is related to the blocking probability. An individual with the best value in each generation is the one with the smallest blocking probability among all individuals in the population. The average in each generation is the average value over the 20 individuals in the population.

The results are presented in Fig. 3, which shows the performance of the worst, best and average solutions. The x-axis represents the number of generations and the y-axis is the value of the objective function, i.e., the blocking probability. We see that the initial generation starts with the average value 0.0512, the worst 0.0545, and the best 0.0454. In generation 4, all individuals exhibit the same value, which is said that the GA has come to the *premature convergence* [9, p.74]. However, after generation 4, the GA improves the individuals' values by exploiting the search space with mutation. Then in generation 10 the optimal solution is reached with the value 0.0394, for which Daigo and Ishioka are the converter locations. We observe that the GA attains an optimal solution in a small number of generations. Selection of Daigo seems to be a peripheral effect due to the Delaunay triangulation.

In this case, there are only $\binom{14}{2} = 91$ combinations of converter placements. Therefore, we can calculate the blocking probabilities for all combinations by Eq. (5). We have also conducted simulation by generating 500,000 calls for each combination. The results are compared in Fig. 4. In both analysis and simulation, the converter placement in Daigo and Ishioka gives the minimum blocking probability among all combinations.

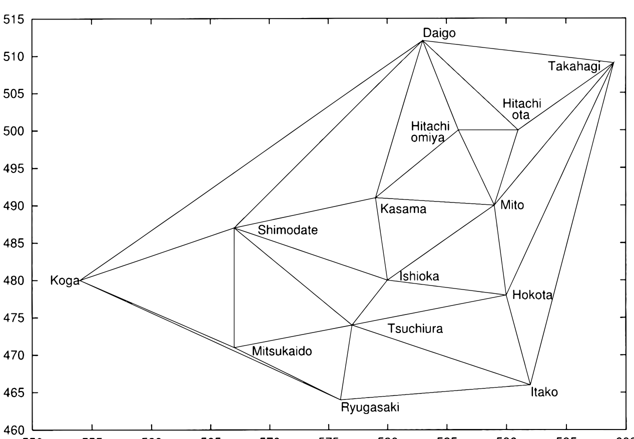


Fig. 2 Configuration of the Ibaraki Network.

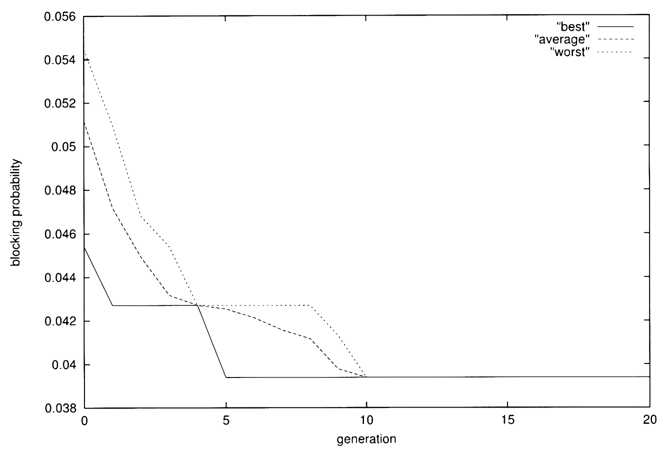


Fig. 3 GA performance for the Ibaraki Network.

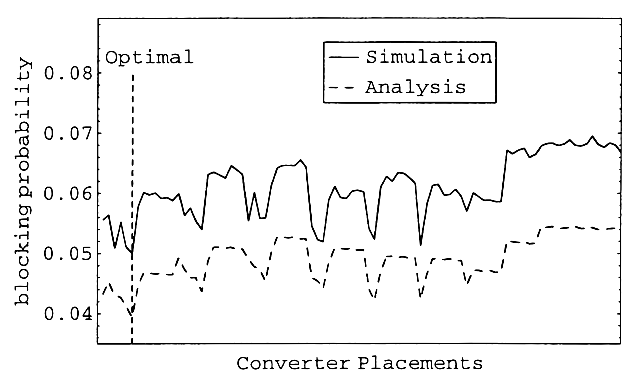


Fig. 4 Comparison of the blocking probability values by analysis and by simulation for the Ibaraki Network.

However, the values by simulation are generally larger than those by analysis. The reason is explained in the Appendix.

4.3 Kanto Network

Our second example is a network consisting of $N = 82$ nodes representing major cities in the Kanto District, which we call Kanto Network, as shown in Fig. 5.

We first consider the case with $K = 2$ and $F = 3$, and assume that calls are generated at rate $\lambda_{sd} = 0.005$ for every (s, d) pair. The GA parameters are as follows:

- Population size = 40
- Probability of crossover = 0.6
- Probability of mutation = 0.00333
- Maximum number of generations = 60

The performance of GA for this case is shown in Fig. 6, where the generation starts with the average value 0.1961, the worst value 0.1996, and the best value 0.1879. In generations 10 to 37, the GA comes to the premature convergence at 0.1764. However, after generation 38, the GA improves the individuals' values by exploitation. In generation 42, the GA finds the optimal solution at 0.1673 with converters at Chichibu and Kuroiso. While Chichibu (with 9 links) is inside, Kuroiso is on the peripheral of the network. This result agrees with the exhaustive search of $\binom{82}{2} = 3321$ combinations of converter placements.

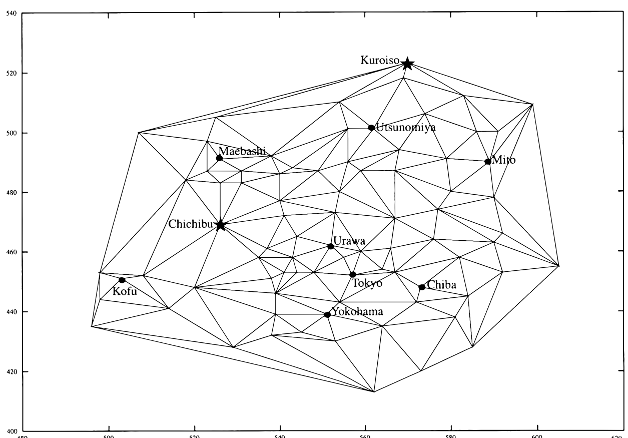


Fig. 5 Configuration of the Kanto Network.

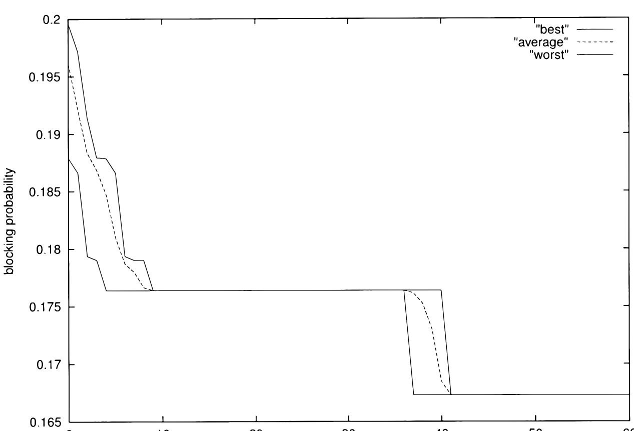


Fig. 6 GA performance for the Kanto Network with 2 converter nodes ($\lambda_{sd} = 0.005$).

Finally we study the dependence of the GA performance on the number K of the converters. In Fig. 7, we present the best value performance for $K = 0$, 2, 3, 5, 10, 20, 50, and 82 for the Kanto Network with $F = 3$ and $\lambda_{sd} = 0.01$. The GA parameters are the same as above except that the maximum number of generations has been extended to 200.

As expected, a larger number of converters can decrease the value of blocking probability. As the search space becomes larger, the convergence is attained more slowly. The value $K = 0$ corresponds to the case in which no converters are used in the network. It is an upper bound of the blocking probability. The value $K = 82$ corresponds to the case in which all nodes are equipped with converters. It provides a lower bound of the blocking probability.

Table 1 compares the computational time of our GA with the exhaustive search algorithm (ESA) for the Kanto Network with various numbers of converters. The computational time requirement of the ESA becomes large in proportion to the size of the search space

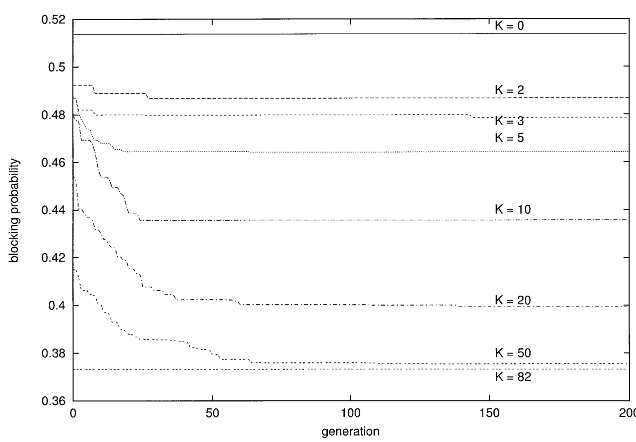


Fig. 7 GA best value performance for the Kanto Network with K converter nodes ($\lambda_{sd} = 0.01$).

Table 1 Run time of exhaustive search algorithm (ESA) and GA for the Kanto Network with K converter nodes.

K	Search Space	ESA (*min.*)	GA (*min.*)
2	3321	$1'\ 6''$	$5'\ 2''$
3	88560	$28'45''$	$5'32''$
5	2.72853×10^7	7.45 days	$6'\ 8''$
10	2.13928×10^{12}	-	$7'20''$
20	6.20877×10^{18}	-	$9'\ 2''$
50	5.93991×10^{22}	-	$12'30''$

which grows exponentially with the number of converters. In contrast, the computational time requirement of GA barely grows with the number of converters. This experiment shows that the computational time of GA achieves considerable improvement over ESA.

5. Conclusion

The purpose of our GA method is to produce an optimal solution for a certain population of individuals in a limited number of generations. This method has been tested to obtain the optimal converter placement for the network examples consisting of major cities in Ibaraki and Kanto areas. We have compared the GA performance (accuracy and computational time) with the ESA and the simulation.

A major difference between GA and ESA is in the size of the search space for the optimal solution. The ESA compares the blocking probability values for all combinations of converter placement in a network. For large networks where the number of combinations grows, it needs explosive computational time for searching the optimal solution. On the other hand, since GA restricts trials to feasible solutions it does not need to compare all the blocking probability values.

The GA tries to reach an optimal solution for a certain population in a number of generations. Using the probability model in crossover and mutation to produce more diversity in the population, there is a higher chance of producing the solution with the optimal value. It has been shown by our experiment that GA can produce a nearly optimal solution in a limited number of generations by spending less time than the ESA.

Our GA evaluates the blocking probability by a procedure that relies on the independence assumption about the wavelength occupancy on each link. This leads only to approximate values. However, our goal is to identify the optimal converter placement quickly, not the exact evaluation of the blocking probability values by precise computation. By comparison with simulation, we have shown that the optimal placement selected by GA is nearly correct.

Acknowledgment

The authors wish to thank Dr. Konosuke Kawashima of NTT Advanced Technology Corporation for providing them with the web site URL of the NTT office locations. They are also grateful to Dr. Sashisekaran Thiagarajan of the Iowa State University, who kindly answered some questions by e-mail. The simulation was conducted by Mr. Masaki Murata. This work has been carried out by the support of the Grant-in-Aid for Scientific Research (C), Project No. 11650407, from the Japan Society for the Promotion of Science.

References

[1] B. Ramamurthy and B. Mukherjee, "Wavelength conversion in WDM networking," IEEE J. Sel. Areas Commun., vol.16, no.7, pp.1061–1073, Sept. 1998.
[2] K.-C. Lee and V.O.K. Li, "A wavelength-convertible optical network," J. Lightwave Technology, vol.11, nos.5/6, pp.962–970, May/June 1993.
[3] R. Ramaswami and K.N. Sivarajan, Optical Networks: A Practical Perspective, Second Edition, Morgan Kaufmann, San Francisco, 2002.
[4] S. Subramaniam, M. Azizoglu, and A.K. Somani, "On the optimal placement of wavelength converters in wavelength-routed networks," Proc. INFOCOM 1998, pp.902–909, 1998.
[5] S. Thiagarajan and A.K. Somani, "An efficient algorithm for optimal wavelength converter placement on wavelength-routed networks with arbitrary topologies," Proc. INFOCOM 1999, pp.916–923, 1999.
[6] C. Vijayanand, M.S. Kumar, K.R. Venugopal, and P.S. Kumar, "Converter placement in all-optical networks using genetic algorithms," Computer Communications, vol.23, pp.1223–1234, 2000.
[7] C. Gazen and C. Ersoy, "Genetic algorithms for designing multihop lightwave network topologies," Artificial Intelligence in Engineering, vol.13, pp.211–221, 1999.
[8] I.E. Kassotakis, M.E. Markaki, and A.V. Vasilakos, "A hybrid genetic approach for channel reuse in multiple access telecomunications networks," IEEE J. Sel. Areas Commun., vol.18, no.2, pp.234–242, Feb. 2000.
[9] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[10] L. Davis, ed., Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.
[11] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.
[12] R.A. Barry and P.A. Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers," IEEE J. Sel. Areas Commun., vol.14, no.5, pp.858–867, June 1996.
[13] J. O'Rourke, Computational Geometry in C, Cambridge University Press, New York, 1994.

Appendix: Approximation by Independence Assumption

As a simple example, suppose that we have load λ on a unidirectional directed link with F wavelengths. The mean call duration is unity. Then the blocking probability is given by the well-known *Erlang's B formula*:

$$P_{B,\text{exact}} = \frac{\lambda^F}{F!} \Big/ \sum_{j=1}^{F} \frac{\lambda^j}{j!}.$$

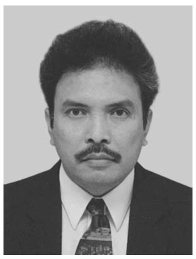
On the other hand, the approximate analysis based on the independence assumption of Sect. 2.2 yields

$$P_{B,\text{approx}} = \left(\frac{\lambda}{F}\right)^F.$$

Thus, when λ is small, we have

$$P_{B,\mathrm{exact}} \approx \frac{\lambda^F}{F!} > P_{B,\mathrm{approx}}.$$

This indicates that the blocking probability values by simulation ($\approx$ exact values) are larger than those by the approximate analysis, as in Fig. 4.



Johannes Hamonangan Siregar is graduate student at the Doctoral Program in Policy and Planning Sciences of the University of Tsukuba. He received B.S. in Mathematical Science from the University of Indonesia, Jakarta, and M.Ed. in Applied Mathematics from Yokohama National University. He is interested in combinatorial optimization problems, genetic algorithms, and computer communication networks.

Hideaki Takagi is Professor at the Institute of Policy and Planning Sciences of the University of Tsukuba, and currently Chair of the Institute. He received B.S. and M.S. in Physics from the University of Tokyo, and Ph.D. in Computer Science from the University of California, Los Angeles. Prior to the current position, he was Consultant Researcher at IBM Research, Tokyo Research Laboratory. He is interested in enumerative combinatorics, probability models, and performance evaluation of computer and communication systems. He is IEEE Fellow and IFIP Silver Core holder.

Yongbing Zhang is Associate Professor at the Institute of Policy and Planning Sciences of the University of Tsukuba. He received B.S. in 1984 in Electrical Engineering from the Polytechnic University, Kanagawa, Japan, and M.S. and Dr.Eng. in Computer Science in 1989 and 1992, respectively, both from the University of Electro-Communications, Tokyo. From 1992 to 1996, he was a Research Associate with the Department of Computer Science at the University of Electro-Communications. In 1996, he joined the Institute of Policy and Planning Sciences, University of Tsukuba. His current research interests include distributed/parallel computer systems, communication networks, and performance evaluation.