Implementation Issues and Experimental Study of a Wavelength Routing Algorithm for Irregular All–Optical Networks*

Athanasios Bouganis, Ioannis Caragiannis**, and Christos Kaklamanis***

Computer Technology Institute and
Dept. of Computer Engineering and Informatics
University of Patras, 26500 Rio, Greece

Abstract. We study the problem of allocating optical bandwidth to sets of communication requests in all–optical networks that utilize Wavelength Division Multiplexing (WDM). WDM technology establishes communication between pairs of network nodes by establishing transmitter–receiver paths and assigning wavelengths to each path so that no two paths going through the same fiber link use the same wavelength. Optical bandwidth is the number of distinct wavelengths. Since state–of–the–art technology allows for a limited number of wavelengths, the engineering problem to be solved is to establish communication between pairs of nodes so that the total number of wavelengths used is minimized.
In this paper we describe the implementation and study the performance of a wavelength routing algorithm for irregular networks. The algorithm proposed by Raghavan and Upfal [17] and is based on a random walk technique. We also describe a variation of this algorithm based on a Markov chain technique which is experimentally proved to have improved performance when applied to random networks generated according to the $G_{n,p}$ model.

1 Introduction

Optical fiber transmission is rapidly becoming the standard transmission medium for networks and can provide the required data rate, error rate, and delay performance for future networks. A single optical wavelength supports rates of gigabits-per-second (which in turn support multiple channels of voice, data, and video [18]). Multiple laser beams that are propagated over the same fiber on distinct optical wavelengths can increase this capacity much further; this is achieved through WDM (Wavelength Division Multiplexing). However, data rates are limited in opto–electronic networks by the need to convert the optical signals on the fiber to electronic signals in order to process them at the network nodes.

* This work has been partially supported by EU Esprit Project 20244 ALCOM–IT, the Greek General Secretariat of Research and Technology, and the Greek PTT.

** e-mail: `caragian@ceid.upatras.gr`, `http://www.ceid.upatras.gr/~caragian`

*** e-mail: `kakl@cti.gr`

J.S. Vitter and C.D. Zaroliagis (Eds.): WAE'99, LNCS 1668, pp. 258–270, 1999.

Networks using optical transmission and maintaining optical data paths through the nodes are called all–optical networks.

In such networks, we consider communication requests as ordered transmitter–receiver pairs of network nodes. WDM technology establishes connectivity by finding transmitter–receiver paths, and assigning a wavelength to each path, so that no two paths going through the same fiber link use the same wavelength. Optical bandwidth is the number of available wavelengths. As state–of–the–art optics technology allows for a limited number of wavelengths (even in the laboratory) the important engineering problem to be solved is to establish communication between pairs of nodes so that the total number of wavelengths used is minimized.

Theoretical work on optical networks mainly focuses on the performance of wavelength routing algorithms on regular networks or arbitrary networks using oblivious (predefined) routing schemes. We point out the pioneering work of Pankaj [15] who considered shuffle exchange, De Bruijn, and hypercubic networks. Aggarwal et al. [1] consider oblivious wavelength routing schemes for several networks. Raghavan and Upfal in [17] consider mesh–like networks. Aumann and Rabani [2] improve the bounded of Raghavan and Upfal for mesh networks and also give tight results for hypercubic networks. Rabani in [16] gives almost optimal results for the wavelength routing problem on meshes.

These topologies reflect architectures of optical computers rather than wide–area networks. For fundamental practical reasons, the telecommunication industry does not deploy massive regular architectures: backbone networks need to reflect irregularity of geography, non–uniform clustering of users and traffic, hierarchy of services, dynamic growth, etc. In this direction, Raghavan and Upfal [17], Aumann and Rabani [2], and Bermond et al. [4], among other results, focus on bounded–degree networks and give upper and lower bounds in terms of the network expansion. The wavelength routing problem in tree–shaped networks has also received much attention. Erlebach and Jansen [7,8] prove that several versions of the problem are NP–hard, while extensive study of a class of algorithms for such networks has been made in a series of papers [13,10,12,11].

In this paper we describe the implementation and experimentally study the performance of a wavelength routing algorithm for irregular (arbitrary) sparse networks. This algorithm proposed by Raghavan and Upfal in [17] and theoretical results were obtained for networks of bounded degree. The model of bounded degree networks reflects the irregularity property of real communication networks. We also describe a variation of this algorithm (algorithm MC) that is experimentally proved to have improved performance.

The paper is structured as follows. We give basic definitions on the optical model we follow in section 2. The description of algorithm RW and issues concerning its implementation are presented in section 3. Algorithm MC is described in section 4. We present experimental results in section 5. Some extensions of this work currently in progress are briefly discussed in section 6.

2 The Optical Model

We follow the notation proposed in [3]. A network is modeled as a graph $G = (V(G), E(G))$ where $V(G)$ is the set of nodes and $E(G)$ the set of fiber links. We denote by $P(x, y)$ a *path* in G that consists of consecutive edges beginning in node x and ending to node y. A *request* is an ordered pair of nodes (x, y) in G. An *instance* I is a collection of requests. Note that a given request may appear more than once in an instance.

A *routing* R for an instance I on network G is a set of paths $R = \{P(x, y)|(x, y) \in I\}$. The *conflict graph* associated to a routing R of instance I on network G is a graph $G_R = (V(G_R), E(G_R))$ such that each node in $V(G_R)$ corresponds to a path in R and two nodes x, y in $V(G_R)$ are adjacent (there exist an edge $(x, y) \in E(G_R)$) if and only if the corresponding paths in R share a fiber link of network G.

Let G be a network and I an instance of requests. The *wavelength routing problem* consists of finding a routing R for instance I and assigning each request $(x, y) \in I$ a wavelength, in such way that no two paths of R sharing a fiber link of G have the same wavelength. Intuitively, we can think the wavelengths as colors and the wavelength routing problem as a node–coloring problem of the corresponding conflict graph.

Early models of optical networks assumed that optical transmission through fibers is performed bidirectionally, so undirected graphs were used to model optical networks. It has since become apparent that directed graphs are essential to model state–of–the–art technology, so recent theoretical work on WDM optical networks focuses to the directed model. In this model, the network is a symmetric digraph (unless otherwise specified), i.e. between two adjacent nodes, there exist two opposite directed fiber links. Note that the definitions above apply to both models.

Given a routing R for an instance I on network G, we denote by $\boldsymbol{w}(G, I, R)$ the minimum number of wavelengths used for a valid wavelength assignment to paths in R. Obviously, $\boldsymbol{w}(G, I, R)$ is the chromatic number of the corresponding conflict graph G_R. We denote by $\boldsymbol{w}(G, I)$ the minimum $\boldsymbol{w}(G, I, R)$ over all routings R of instance I. The notation for the undirected model is $w(G, I, R)$ and $w(G, I)$, respectively.

Given a routing R for an instance I on network G, the *load* of a fiber link $\alpha \in E(G)$, denoted by $\boldsymbol{\pi}(G, I, R, \alpha)$, is the number of paths in R that contain α. We denote by $\boldsymbol{\pi}(G, I, R)$ the maximum load of any fiber link in $E(G)$, and by $\boldsymbol{\pi}(G, I)$ the minimum $\boldsymbol{\pi}(G, I, R)$ over all routings R of an instance I. The notation for the undirected model is $\pi(G, I, R, \alpha)$, $\pi(G, I, R)$ and $\pi(G, I)$, respectively. Obviously, $\boldsymbol{\pi}(G, I, R)$ is a lower bound for $\boldsymbol{w}(G, I, R)$, so $\boldsymbol{\pi}(G, I)$ is a lower bound for $\boldsymbol{w}(G, I)$.

Given a routing R for an instance I on network G, the *length* of a path $P(x, y) \in R$, denoted by $\boldsymbol{\delta}(G, I, R, P(x, y))$, is the number of consecutive fiber links of $E(G)$ contained in $P(x, y)$. The *dilation* of a routing R for instance I on network G, denoted by $\boldsymbol{\delta}(G, I, R)$ is the maximum length of any path in R. The average length of paths in R, denoted by $\tilde{\delta}(G, I, R)$ is defined in the obvious way. We can also define $\boldsymbol{\delta}(G, I)$ as the minimum $\boldsymbol{\delta}(G, I, R)$ over all routings R. In general computing $\boldsymbol{w}(G, I)$, $\boldsymbol{w}(G, I, R)$ and $\boldsymbol{\pi}(G, I)$ is NP–hard for most networks [9,7,8], while computing $\boldsymbol{\delta}(G, I)$ can be solved in polynomial time [6].

An important instance that has received much attention is the *k–relation*. A directed k–relation is defined as an instance I_k in which each node appears as a source or as a destination in no more than k requests. An undirected k–relation is defined as an instance I_k in which each node appears in no more than k requests (either as source or as destination). In the following sections, we consider k–relations that have the maximum number of requests, i.e. in a directed k–relation, each node appears as source or designation in exactly k requests, while in an undirected k–relation, each node appears (either as source or as destination) in exactly k requests. Note that under this definition of k–relations, given a network G, the number of requests in a directed k–relation is twice the number of requests in an undirected k–relation. A 1–relation I_1 is called permutation and has received much attention in the literature (not only within the context of wavelength routing).

3 The Algorithm of Raghavan and Upfal

In this section we describe the implementation of the algorithm proposed by Raghavan and Upfal [17] for approximating $\boldsymbol{w}(G, I_k)$ on bounded–degree all–optical networks (algorithm RW).

Given a k–relation I_k on a network G, the algorithm uses a random walk technique for finding a routing R for I on G and properly assigns colors to paths of R. The algorithm considers requests one by one, finds a path for each one of them and assigns a proper color to it. When a path has been established for a request of I and colored, it is never reestablished or recolored again.

We define a *random walk* on a graph $G = (V(G), E(G))$ (directed or not) as a Markov chain $\{X_t\} \subseteq V$ associated to a particle that moves from vertex to vertex according to the following rule: the probability of a transition from vertex v_i, of degree d_i, to vertex v_j is $1/d_i$ if $(i, j) \in E$, and 0 otherwise. The *transition matrix* of the random walk denoted by P is such that element P_{ij} (or P_{v_i,v_j}) is the probability that the particle moves from vertex v_i to vertex v_j. The *stationary distribution* of the random walk is a vector ρ such that $\rho = P\rho$. We define a *trajectory* W of length τ as a sequence of vertices $[w_0, w_1, \ldots, w_\tau]$ such that $(w_t, w_{t+1}) \in E(G)$ for $0 \leq t < \tau$.

Algorithm RW

1. Compute the transition matrix P and the stationary distribution ρ.
2. Compute a sufficient length L for trajectories.
3. For each request $(a_i, b_i) \in R$ do:
   (a) Choose a node $r_i \in V$ according to the stationary distribution.
   (b) Choose a trajectory W'_i (resp. W''_i) of length L from a_i to r_i (resp. from b_i to r_i) according to the distribution on trajectories, conditioned on the endpoints being a_i and r_i (resp. b_i and r_i).

(c) Connect a_i to b_i by the path $P(a_i, b_i)$, defined by W'_{a_i,r_i} followed by W''_{r_i,b_i}.
(d) Eliminate cycles of the path $P(a_i, b_i)$.
(e) Assign a proper color to $P(a_i, b_i)$.

Computing the length of trajectories. It is known [14] that since a random walk is an ergodic Markov chain, P^t converges to a unique equilibrium as $t \to \infty$. The length L of trajectories must be such that the power P^L is very close to $\lim_{t\to\infty} P^t$. Although this computation in [17] is $L = -3\frac{\log kn}{\log \lambda}$ where λ is the second largest eigenvalue of the transition matrix P in absolute value, we observed that P^t is very close to $\lim_{t\to\infty} P^t$ for smaller values of t. The computation $L = -\frac{1.5 \log n}{\log \lambda}$ from [14] suffices.

Random choice in step 3a. A node $r_i \in V$ is chosen according to the stationary distribution by simulating the casting an n–faced die with probabilities $\rho_i, i = 1, ..., n$ associated with the n faces.

Finding trajectories. The method proposed in [5] which is also implied in [17] for computing a random trajectory $W = [u = w_1, w_2, ..., w_t = v]$ of length t from node u to node v is to:

1. Choose a node w according to the following rule: let w be a neighbor of v. Then
$$Pr[w_{t-1} = w | w_t = v] = \frac{P^{(t-1)}_{u,w} P_{w,v}}{P^{(t)}_{u,v}}.$$
where $P^{(t)}$ denotes the t–th power of the transition matrix P.
2. Recursively, choose a random trajectory of length $t-1$ from u to w.

Eliminating cycles. This phase is not analyzed in [17]. We observed that the paths defined by two trajectories contain cycles, almost always. Cycles in a path $P(a_i, b_i)$ created by two trajectories W'_i and W''_i are emilinated by finding a shortest path between nodes a_i and b_i in the subgraph H_i defined by the two trajectories (i.e. $H_i = (V(H_i), E(H_i)$ s.t. $V(H_i) = \{v \in V(G) | v \in W'_i$ or $v \in W''_i\}$ and $E(H_i) = \{(v, u) \in E(G) | v, u \in W'_i$ or $v, u \in W''_i\})$.

Coloring paths. The algorithm we use for coloring paths is the obvious (greedy) one. We use a palette of colors $\chi_1, \chi_2, ..., \chi_{kn}$ and assign to a path $P(x, y)$ the color χ_c with minimum c s.t. χ_c has not been assigned to any established path that shares a fiber link with $P(x, y)$. The total number of colors (wavelengths) actually used is the maximum index of colors assigned to paths.

4 The Algorithm MC

The algorithm MC is a variation of the algorithm RW presented in the previous section. The main structure of algorithm MC remains the same with the one of algorithm RW. Also algorithm MC maintains the main characteristics of algorithm RW. Given a k–relation I_k on a network G, the algorithm uses a Markov chain technique for finding a routing R for I on G and properly assigns colors to paths of R. The algorithm considers requests one by one, finds a path for each one of them and assigns a proper color to it, so that once a path has been established for a request of I and colored, it is never reestablished or recolored again.

The main difference is that algorithm MC, while computing the path for a request of instance I, it considers a Markov chain $\{Z_t\} \subseteq V$ associated to a particle that moves from node to node. The definition of the Markov chain (i.e. the transition probabilities) is such that it takes into account the paths that have been already established in previous steps. This means that the transition matrix, the stationary distribution and the sufficient length of trajectories is recomputed for each request given as input to the algorithm. There are many different ways to define the Markov chain $\{Z_t\}$. In the following we present algorithm MC together with a simple strategy for computing the transition probabilities of the Markov chain $\{Z_t\}$. This strategy is the one used for performing the experiments presented in section 5.

Algorithm MC

1. For each request $(a_i, b_i) \in R$ do:
   (a) Recompute the transition matrix P' and the stationary distribution ρ'.
   (b) Recompute a sufficient length L for trajectories.
   (c) Choose a node $r_i \in V$ according to the stationary distribution.
   (d) Choose a trajectory W_i' (resp. W_i'') of length L from a_i to r_i (resp. from b_i to r_i) according to the distribution on trajectories, conditioned on the endpoints being a_i and r_i (resp. b_i and r_i).
   (e) Connect a_i to b_i by the path $P(a_i, b_i)$, defined by W_i' followed by W_i''.
   (f) Eliminate cycles of the path $P(a_i, b_i)$.
   (g) Assign a proper color to $P(a_i, b_i)$.

Recomputing the transition matrix. Consider a step of the algorithm. Assume that paths for $s < |I|$ requests have been already been established and colored. Let I_s denote the subset of instance I which contains the s already considered by the algorithm MC requests, and R_s the set of paths established for requests of I_s. We denote by $\boldsymbol{\pi}(G, I_s, R_s, \alpha)$ the load of a fiber link $\alpha \in E(G)$ at the current step.

Given a network $G = (V(G), E(G))$, we define a Markov chain $\{Y_t\} \subseteq V(G)$ associated to a particle that moves from node to node. The transition probability between non–adjacent nodes is $P''_{ij} = 0$. The definition of the transition probabilities from a node u_i to its neighbors, distinguishes between two cases:

1. Some of the fiber links adjacent to u_i are not used by any path of R_s. Let d be the degree of node u_i and m be the number of nodes adjacent to u_i, $u_1, ..., u_m$ such that the fiber links $(u_i, u_j), j = 1, ..., m$ have $\boldsymbol{\pi}(G, I_s, R_s, (u_i, u_j)) = 0$, for $1 \leq j \leq m$. It is $\boldsymbol{\pi}(G, I_s, R_s, (u_i, u_j)) > 0$, for $m + 1 \leq j \leq d$. Then, the

transition probability from node u_i to a neighbor node u_j is $P''_{ij} = 1/m$ if $j \leq m$, $P''_{ij} = 0$ otherwise.

2. All fiber links adjacent to u_i are used by paths in R_s. The transition probabilities P''_{ij} from node u_i to two neighbors u_j and u_l satisfy

$$\frac{P''_{ij}}{P''_{i,l}} = \frac{\boldsymbol{\pi}(G, I_s, R_s, (u_i, u_l))}{\boldsymbol{\pi}(G, I_s, R_s, (u_i, u_j))} \text{ s.t. } \sum_{(u_i,u_j) \in E(G)} P''_{ij} = 1.$$

This gives

$$P''^{-1}_{ij} = \boldsymbol{\pi}(G, I_s, R_s, (u_i, u_j)) \sum_{(u_i,u_k) \in E(G)} \frac{1}{\boldsymbol{\pi}(G, I_s, R_s, (u_i, u_j))}.$$

It is clear that, the Markov chain $\{Y_t\}$ constructed at a step of the algorithm, has the following main property: *for a transition from a node, it prefers the less loaded fiber links.* The algorithm MC uses a parameter $f \in (0, 1]$ and assigns to each transition of Markov chain $\{Z_t\}$ a probability

$$P'_{ij} = fP_{ij} + (1 - f)P''_{ij}.$$

Intuitively, $\{Z_t\}$ maintains the ergodicity feature of the random walk, while it encapsulates the main property of $\{Y_t\}$. Note that for $f = 1$, algorithm MC is identical to algorithm RW, since $\{Z_t\}$ is a random walk. Values close to $f = 0$ are not valid since the Markov chain $\{Y_t\}$ may not be ergodic.

Note that the transition probabilities of P' that must be recomputed at each step of the algorithm MC are only those associated to nodes of the path established in the previous step.

5 Experiments and Results

The algorithms have been tested using as input random networks and random k–relations. Random networks are constructed according to the $G_{n,p}$ model. For maintaining connectivity, a network given as input to the algorithms has an arbitrary Hamilton circuit. The parameters that effect the results of the wavelength routing algorithms are:

- Parameter n is the number of nodes of the random network. All the experiments presented in this paper have been made on networks with 200 nodes.
- Parameter c indicates that the probability that a fiber link (except those in the Hamilton path) exists in the network is $p = c/n$. This parameter gives a measurement for the density of the network.
- Parameter k indicates that the instance given to the algorithms as input is chosen randomly among all k–relations on a network of n nodes. For the experiments presented in this paper, we used small values for k (up to 4). Larger values increase the running time of the algorithms dramatically.

We first measure the number of wavelengths assigned to a k relation by the algorithm RW as a function of the density of the network. Note that, while the standard algorithm of Raghavan and Upfal as it was presented in [17] considers undirected paths only, the description given in section 3 applies to both the directed and the undirected model. We consider three types of k relations: (1) a random k–relation of undirected paths, (2) the corresponding symmetric k–relation of directed paths (i.e. for each request $(u, v) \in I_k$, $(v, u) \in I_k$ as well) and (3) a random k–relation of directed paths (non–symmetric). Note that the total number of requests in a undirected k–relation is half the number of requests of a directed k–relation. Moreover, the two versions of the wavelength routing problem in the directed and undirected model have important inherent differences. In particular, there exist a permutation I_1 on a n node graph which, although it can be routed with load $\pi(G, I_1) = 2$, it requires $w(G, I_1) = n$ wavelengths [1]. No such case has been observed in the directed model, and it has been conjectured that $\boldsymbol{w}(G, I) = O(\boldsymbol{\pi}(G, I))$ for every network G and instance I [3]. Surprisingly, algorithm RW has the same performance when applied to random undirected or directed requests on random (sparse) networks. The results for permutations, 2–relation and 4–relation are depicted in table 1.

Density	Undirected			Symmetric			Non–Symmetric		
3	19	29	55	19	31	54	18	30	55
5	14	23	36	13	22	36	13	23	36
10	9	14	21	9	14	22	8	14	21
15	5	10	14	6	10	15	6	10	15
20	5	8	11	5	7	12	6	7	11

Table 1. Routing undirected and directed requests on random networks of 200 nodes. The table contains the number of wavelengths used for random undirected, the corresponding symmetric directed, and random directed permutations, 2–relations, and 4–relations on 5 random networks of different density.

In the following we concentrate to directed instances. The results for the performance of algorithm RW on random permutations, 2–relations, and 4–relations according to the data contained in the three rightmost columns of table 1 are graphically represented in figure 1.

Furthermore, data of table 2 show that the number of wavelengths used by algorithm RW in each case is slightly greater than the load of the routings and is much smaller than the degree of the conflict graph. Also, for sparse networks (density 3 or 5), the dilation is smaller than the theoretical length of paths, meaning that all paths produced by connecting two trajectories contained cycles. This is not true for dense networks (density 10 or 20). For any case, most paths produced by connecting trajectories contained cycles, so the average length of paths is much smaller than the dilation. The difference between average length

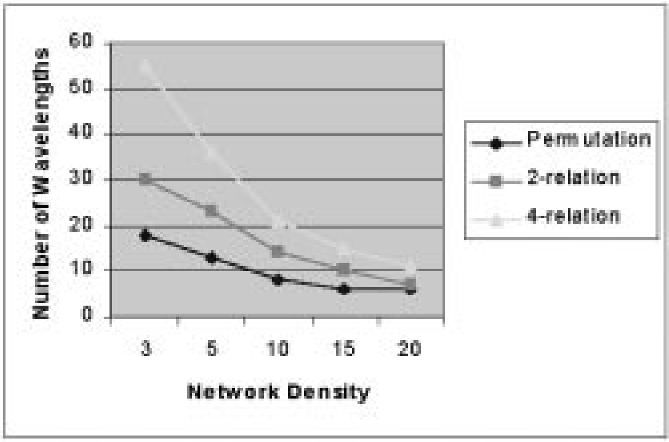

Fig. 1. Routing directed permutations, 2–relations, and 4–relations on random networks with 200 nodes using algorithm RW.

of paths and dilation (and the theoretical length of paths which is twice the length of trajectories) increases as the network density decreases.

Density	#Wav.	Load	Av. Length	Dilation	Traj. Length	CG Degree
3	20	11	17.73	35	22	76
5	15	9	17.04	27	15	54
10	8	6	12.53	18	9	24
20	7	5	10.60	12	6	13

Table 2. Routing a permutation on random networks with 200 nodes using algorithm RW. The columns of the tables contain results for the number of wavelengths used, the load, the average length of paths, the dilation, the length of trajectories, and the degree of the conflict graph of the produced routing for 4 random networks of different density.

We observed that algorithm MC improves algorithm RW concerning the number of wavelengths assigned to requests. This improvement can be about 20% of the performance of RW in some cases and is significant especially for instances with many requests (4–relations) on very sparse networks (density 3 or 5).

This is mainly due to the fact that the routings produced by algorithm MC are qualitatively better than those produced by algorithm RW. We performed experiments by running algorithm MC with parameter $f = 0.5$ for the same permutation on random networks of density 3, 5, 10, and 20. The conflict graph produced has smaller degree in any case; as a result, the number of wavelengths used is decreased related to the number of wavelengths used by algorithm RW.

A somewhat surprising result is that, for sparse networks, the average length of paths produced by algorithm MC is slightly smaller than the average length of paths produced by algorithm RW. Table 3 summarizes data produced by algorithm MC with $f = 0.5$.

Density	#Wav.	Load	Av. Length	Dilation	CG Degree
3	15	11	15.63	33	64
5	13	9	15.31	29	42
10	6	5	12.90	18	15
20	4	3	10.75	12	9

Table 3. Routing a permutation on random networks with 200 nodes using algorithm MC with parameter f set to 0.5. The columns of the tables contain results for the number of wavelengths used, the load, the average length of paths, the dilation, and the degree of the conflict graph of the produced routing for 4 random networks of different density. Corresponding rows of tables 2 and 3 represent results of experiments on the same network. Data in both tables were obtained for the same permutation.

Concerning the number of wavelengths used, which is the main performance metric we use for comparing our algorithms, the performance of algorithm MC is better in most cases as the parameter f decreases. Algorithm MC was executed with several values of parameter f (included $f = 1.0$, a value for which algorithm MC is identical to algorithm RW) on permutations and 4–relations on networks of different density. The results that correspond to these experiments are depicted in tables 4 and 5 and are graphically represented in figures 2 and 3, respectively.

Density	$f = 0.2$	$f = 0.4$	$f = 0.6$	$f = 0.8$	RW
3	14	16	16	17	20
5	13	12	13	12	14
10	5	6	7	7	8
20	4	4	4	5	5

Table 4. Routing permutations on networks with 200 nodes using algorithm MC. Data represent the number of wavelengths used, for several values of parameter f.

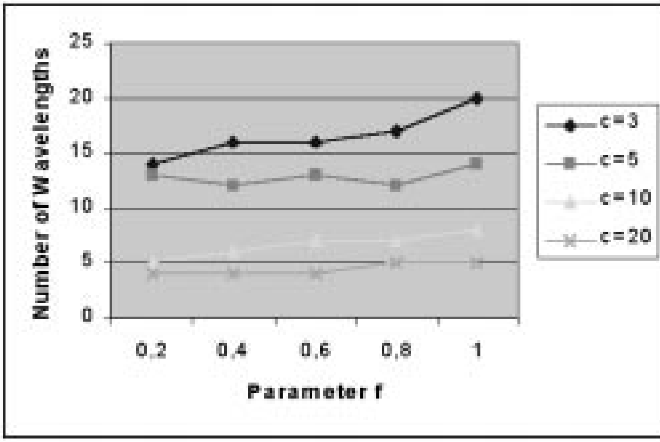

Fig. 2. Routing permutations on networks with 200 nodes using algorithm MC.

Density	$f = 0.2$	$f = 0.4$	$f = 0.6$	$f = 0.8$	RW
3	44	47	48	50	53
5	30	32	32	35	36
10	16	17	18	19	19
20	10	11	10	10	11

Table 5. Routing 4–relations on networks with 200 nodes using algorithm MC. Data represent the number of wavelengths used for several values of parameter f. Corresponding rows of tables 4 and 5 represent results of experiments on the same network.

6 Extensions

The algorithms RW and MC were implemented in the Matlab v5.0 environment and the experiments presented in this paper were conducted on a Pentium PC/200MHz. Recently, we completed the implementation and testing of the algorithms in C; we currently perform several experiments on a powerful Pentium III/500MHz running Solaris 7.

In experiments not included here, we experimentally studied the performance of modified MC algorithms that use more complex functions to define the transition probabilities of the Markov chain. In some cases, we observed that results can be further improved.

In addition to networks generated according to the $G_{n,p}$ model, we currently perform experiments on random regular networks with link faults; a model that also reflects irregularity of real networks and is closer to the theoretical model of bounded–degree graphs assumed in [17]. We plan to include all these results in the full version of the paper.

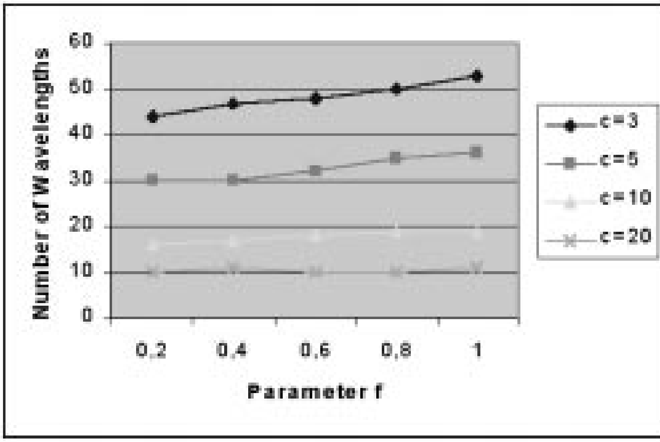

Fig. 3. Routing 4–relations on networks with 200 nodes using algorithm MC.

References

1. A. Aggarwal, A. Bar–Noy, D. Coppersmith, R. Ramaswami, B. Schieber, M. Sudan. Efficient Routing and Scheduling Algorithms for Optical Networks. In *Proc. of the 5th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA 94)*, pp. 412–423, 1994.
2. Y. Aumann, Y. Rabani. Improved Bounds for All Optical Routing. In *Proc. of the 6th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA 95)*, pp. 567–576, 1995.
3. B. Beauquier, J.–C. Bermond, L. Gargano, P. Hell, S. Perennes, U. Vaccaro. Graph Problems arising from Wavelength Routing in All–Optical Networks. *2nd Workshop on Optics and Computer Science (WOCS 97)*, 1997.
4. J.–C. Bermond, L. Gargano, S. Perennes, A. Rescigno, U. Vaccaro. Efficient Collective Communication in Optical Networks. In *Proc. of the 22nd International Colloquium on Automata, Languages, and Programming (ICALP 96)*, LNCS 1099, Springer–Verlag, pp. 574–585, 1996.
5. A.Z. Broder, A. Frieze, and E. Upfal. Existence and Construction of Edge Disjoint Paths on Expander Graphs. In *Proc. of the 24th Annual ACM Symposium on the Theory of Computing (STOC 92)*, pp. 140–149, 1992.
6. E. Dijkstra. A Note on Two Problems in Connection with Graphs. Num. Math., 1:269–271, 1959.
7. T. Erlebach, K. Jansen. Scheduling of Virtual Connections in Fast Networks. In *Proc. of the 4th Workshop on Parallel Systems and Algorithms (PASA 96)*, pp. 13–32, 1996.
8. T. Erlebach, K. Jansen. Call Scheduling in Trees, Rings, and Meshes. In *Proc. of the Hawaii Int. Conf. on Syst. Sciences (HICSS 97)*, 1997.

9. M.R. Garey, D.S. Johnson, G.L. Miller, C.H. Papadimitriou. The Complexity of Coloring Circular Arcs and Chords. *SIAM Journ. Alg. Disc. Meth.*, vol. 1, no. 2, (1980), pp. 216–227.
10. C. Kaklamanis, P. Persiano. Efficient Wavelength Routing on Directed Fiber Trees. In *Proc. of the 4th European Symposium on Algorithms (ESA 96)*, LNCS 1136, Springer Verlag, 1997, pp. 460–470.
11. C. Kaklamanis, P. Persiano, T. Erlebach, K. Jansen. Constrained Bipartite Edge Coloring with Applications to Wavelength Routing. In *Proc. of the 24th Internation Colloquium on Automata, Languages, and Programming (ICALP 97)*, LNCS 1256, Springer Verlag, 1997, pp. 493–504.
12. V. Kumar, E. Schwabe. Improved Access to Optical Bandwidth in Trees. In *Proc. of the 8th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA 97)*, 1997, pp. 437–444.
13. M. Mihail, C. Kaklamanis, S. Rao. Efficient Access to Optical Bandwidth. In *Proc. of the 36th Annual Symposium on Foundations of Computer Science (FOCS 95)*, 1995, pp. 548–557.
14. R. Motwani, P. Raghavan. Randomized Algorithms. *Cambridge University Press*, 1995.
15. R.K. Pankaj. Architectures for Linear Lightwave Networks. PhD. Thesis, Dept. of EECS, MIT, 1992.
16. Y. Rabani. Path Coloring on the Mesh. In *Proc. of the 37th Annual Symposium on Foundations of Computer Science (FOCS 96)*, 1996.
17. P. Raghavan, E. Upfal. Efficient Routing in All-Optical Networks. In *Proc. of the 26th Annual Symposium on Theory of Computing (STOC 94)*, 1994, pp. 133–143.
18. R. Ramaswami, K. Sivarajan. Optical Networks. *Morgan Kaufmann*, 1998.