Optimizing Processing of Query Subscription in An WDM Network Environment

Weifa Liang
Dept. of Computer Science
Australian National University
Canberra, ACT 0200, Australia

Maria E. Orlowska
Dept. of Computer Science & Electric Engineering
University of Queensland
St. Lucia, QLD 4072, Australia

Abstract

In this paper we examine query subscription processing in a Wavelength-Division Multiplexing (WDM) network environment. The users (clients) launch their queries through the Internet to an information center which consists of several servers. The information center then assigns proper servers to process the queries, and sends the query results back to the users. We first model the problem in terms of several optimization criteria such as query processing time required by the servers, the number of wavelengths used for the query result transmissions, and the other parameters contributing to the overall execution/transmission cost. we then present efficient heuristic algorithms to solve the problem.

1 Introduction

With the advent of telecommunications techniques, some leading experts predict that the Internet and Intranet in the next century will be dominated by the *Wavelength-Division Multiplexing* (WDM) optical network. This network has many incomparable advantages in contrast to the traditional electronic networks. The WDM network offers the possibility of interconnecting hundreds and thousands of users, covering local to wide areas, and providing capacities which substantially exceed those which a conventional network can provide. The network potentially provides data transmission rates several orders of magnitude higher than its electronic counterpart. The key to high speed in the network is to maintain the signal in optical form rather than electronic form. The high bandwidth of the fiber-optic links is utilized through Wavelength-Division Multiplexing technology which supports the propagation of multiple laser beams through a single fiber-optic link as long as each laser beam uses a distinct optical wavelength. Each laser beam, viewed as a carrier signal, provides transmission rates of 2.5–10 Gbps, thus the significance of WDM in very high speed networking is paramount. The major applications of the network are in video conferencing, scientific visualization and real-time medical imaging, supercomputing and distributed computing, etc [12, 13]. A comprehensive overview of the physical theory and applications of this technology can be found in the books by Green [4] and McAulay [9]. The further merger between telecommunications and information technology has fueled the development of a wide range of new information-centered applications [1]. Many such applications involve data dissemination, i.e., the delivery of data from a set of *producers* to a typically very large set of *consumers*. There are many such dissemination-based application examples including the stock and sports tickers of news wires, traffic information systems, electronic newsletters and multi-media entertainment delivery [2], etc.

1.1 Motivating example

To motivate this research, we use the following example to illustrate the main concepts. Imagine a cable-TV dissemination center of K servers which delivers movies to its consumers. There are a set of registered clients Q who use the service supplied by the center. Assume that there are 1000,000 clients who use this subscription service. We further assume that the center advertises its show in the morning (e.g., 8:00am) by giving a list of movies which are to be played at night (e.g., 9:00pm). Let the number of available movies be 100. As a registered client, you can choose one from the 100 movies to view. Now we can see, although it can be a 1000,000 queries, the number of distinct queries is at most 100 only, and most of the queries are identical. For this example, how to assign the 100 distinct queries to the K servers such that the query processing load among the K servers is as even as possible, and how to deliver the query results to the clients as economically as possible (reducing the number of wavelengths) are very important. If it is impossible to satisfy all clients' queries in one round, then, which policy would be the best to make the queries of the maximum number of clients be answered. Therefore, our objective is to process the query subscriptions efficiently by reducing the resource consumption in the system.

1.2 Our contributions

In this paper we first model the query subscription problem by giving conceptual and cost models in the WDM network environments. We then consider the server assignment problem by assigning the queries to the servers such that the load among the servers is as even as possible. We have shown that such an assignment is NP-hard [6]. We instead present a heuristic algorithm for the assignment. Subsequentially, we deal with the wavelength assignment problem which transmits the query results back to their clients. For this problem, we study two different versions of the problem. One is to find an assignment with the minimum number of rounds; and another is to find an assignment with the maximum throughput of the system in the case it is impossible to send all the query results back to the clients.

2 Problem Modeling

Now we introduce our abstract model formally. The model for query subscription services is shown in Fig. 1, where there are a large set of clients

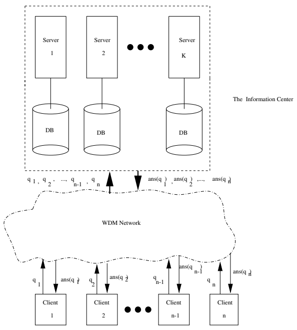

Figure 1: A subscription service conceptual model

$c_1, c_2, \ldots, c_n$ that request information from an information center which consists of K servers, where K is fixed. The clients and the center communicate with each other through an WDM optical network. Each subscription consists of a query and its timing requirement. For simplicity, we assume that all subscriptions have an identical timing requirement. Clients send their queries to the information center, and it periodically processes the queries against the databases. However, which server in the center can and will answer that query is *transparent* to the client. We also assume that a single query can not be further divided and assigned to different servers for processing.

For the network, let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ be the set of all different wavelengths, and let $\Lambda(c_i)$ be the set of wavelengths that client c_i can use, i.e., client c_i can receive the data if the data is transmitted using a wavelength in $\Lambda(c_i)$. Clearly, $\Lambda(c_i) \neq \emptyset$ and $\Lambda(c_i) \subseteq \Lambda$.

A lightpath is a path in the optical network from a source to a destination such that all links on the path are assigned with the same wavelength.

Following the definition, we now consider how to transmit the query results to clients after all queries have been processed. The center will assign the answer of each distinct query with a wavelength λ, establish the lightpaths for this answer, and inform all clients who have identical queries to use wavelength λ to receive their answers. When the client who issued q receives the wavelength message λ from the center, it switches to wavelength λ, and is ready for receiving the data from the network. Finally, the client can receive the answer which is transmitted with wavelength λ through the network correctly.

From the conceptual model, we can see that the query subscription processing consists of two phases. In the first phase, the clients submit their queries to the information center using the Web browser via the network. After the center receives the queries, it proceeds with the server scheduling for query processing, i.e., it will merge some identical queries into one, assign each distinct query to a server, and process a query by the server assigned to it. However, for each query it is the center to decide which server can answer the query. Such an assignment is not easy, since our objective is to minimize the absolute difference of loading between any two servers.

In order to compute the amount of time spent for processing a query, we need to estimate the size of the query result. Such estimation can be obtained using well known database system techniques [8]. Here we use $ans(q)$ to denote the estimated size of the answer of a query q. We further assume that the processing time of a query is linearly *proportional* to its size.

Let $q_{i_1}, q_{i_2}, \ldots, q_{i_l}$ be the queries processed by a server S. Then, the *load* $C(S)$ of server S is $C(S) = \sum_{j=1}^{l} ans(q_{i_j})$. Thus, in the first phase, our objective is to assign each distinct query to one of the K servers such that $T_{diff} = |C(S_a) - S(S_b)|$ is minimized for any two servers S_a and S_b in the center.

In the second phase, the query results are already constructed, and must be transmitted back to their clients via the network. Since the number of wavelengths in the system is the most critical resource, we need to assign the answer of each distinct query with a distinct wavelength at each round such that the number of rounds is as small as possible. If the number of rounds for transmission is fixed in advance, we need to identify which query answers should be sent back to their clients such that the maximum number of clients can receive their answers, i.e., maximize the throughput of the system. Also, if taking both the loading cost (at the server side) and the throughput of the system (at the network side) into consideration, we need finding a better wavelength assignment which trade-offs the two factors well.

3 Minimizing the Processing Time of Servers

Suppose that there are n clients in the network sending their queries to the center. Let $Q = \{q_1, q_2, \ldots, q_n\}$ be the set of queries, and Q' a subset of Q by deleting all duplicate queries except keeping a representative from Q. Clearly all the queries in Q' are distinct. Let $Q' = \{p_1, p_2, \ldots, p_d\}$. Assume that there are $n_i - 1$ other queries in Q which are identical to p_i, $1 \leq i \leq p$, then $\sum_{i=1}^{d} n_i = n$. It is easy to observe that, in order to answer all queries in Q, the servers only need to process all queries in Q' and broadcast the query results back to the network. Thus, we only need to assign the queries in Q' to the servers such that the absolute difference of loading among the servers is minimized. We refer to this load-balancing problem as *the server assignment problem.*

We have shown that the server assignment problem is NP-hard. In the following we present a heuristic algorithm for it. The algorithm is a greedy partition algorithm. Based on the framework of the greedy partition algorithm, we have also proposed two variants of the algorithm to improve the quality of the solution further [6]. The variants make use of simulated annealing and pair-exchange heuristic techniques respectively.

3.1 A greedy partition algorithm

The basic idea behind the proposed algorithm is as follows. Assume that there is a query partition $\mathcal{P} = \{Q_1, Q_2, \ldots, Q_K\}$, and every query $q_{i_j} \in Q_i$ can be answered by server S_i initially, $1 \leq i \leq K$. Each time we add a query to one of the subsets of $\mathcal{P}$ to form a new partition such that the cost of the new partition is the minimum among all possible valid partitions. This procedure is repeated until all queries are included in the partition.

Let $C(\mathcal{P})$ be the sum of the absolute difference of loading between any two servers. Then, $C(\mathcal{P}) = \frac{1}{2}\sum_{i=1}^{K}\sum_{i'=1}^{K}|S_i - S_{i'}|$, where $S_i = \sum_{q_{i_j} \in Q_i} ans(q_{i_j})$. Now consider add a new query q to $\mathcal{P}$. Assume that the answer of q can be derived by servers $S_{i_1}, S_{i_2}, \ldots, S_{i_p}$, $p \leq K$. Then, there are p possible valid K-partitions derived from $\mathcal{P}$ after adding q to it. Let $\mathcal{P}_{i_j}$ be such a valid K-partition by adding q to Q_{i_j} of $\mathcal{P}$, i.e., $\mathcal{P}_{i_j} = \{Q_1, Q_2, \ldots, Q_{i_j-1}, Q_{i_j} \cup \{q\}, Q_{i_j+1}, \ldots, Q_K\}$. Then, $C(\mathcal{P}_{i_j}) = C(\mathcal{P}) - \sum_{i'=1}^{K}|S_{i_j} - S_{i'}| + \sum_{i'=1}^{K}|(S_{i_j} + ans(q)) - S_{i'}|$. Our greedy algorithm is to choose the one from the p valid K-partitions which has the minimum cost. That is, we choose a $\mathcal{P}_{i_{j_0}}$ such that $\mathcal{P}_{i_{j_0}} = \min\{C(\mathcal{P}_{i_j}) \mid 1 \leq j \leq p\}$. Thus, q is added to the $Q_{i_{j_0}}$. The algorithm is presented as follows.

`Procedure Heurist_Algorithm`(Q', K, d);
/* Assume every server S_i can process at least a query. */
If there are K' servers which no queries can be processed by either one of them, then, set $K := K - K'$,
and look for a valid $(K - K')$-partition of d queries.
$UQ := Q'$;
/* the query set that has not been assigned to the servers */
$\mathcal{P} := \{Q_1, Q_2, \ldots, Q_K\}$;
/* where $Q_i = \{q \mid q$ can be answered by S_i, $q \in Q'\}$ */
$C(\mathcal{P}) := \frac{1}{2}\sum_{i=1}^{K}\sum_{i'=1}^{K}|\sum_{q' \in Q_i} ans(q') - \sum_{q'' \in Q_{i'}} ans(q'')|$;
while $UQ \neq \emptyset$ **do**
1. choose a query q from UQ randomly;
2. $UQ := UQ - \{q\}$;
/* assume that servers $S_{i_1}, S_{i_2}, \ldots, S_{i_p}$ can answer q */
3. **for** $j := 1$ **to** p **do**
compute the cost $C(\mathcal{P}_{i_j})$;
endfor;
4. let $\mathcal{P}_{i_{j_0}} = \min\{C(\mathcal{P}_{i_j}) \mid 1 \leq j \leq p\}$.
5. $\mathcal{P} := \{Q_1, Q_2, \ldots, Q_{i_{j_0}-1}$,
$Q_{i_{j_0}} \cup \{q\}, Q_{i_{j_0}+1}, \ldots, Q_K\}$;
6. $C(\mathcal{P}) := C(\mathcal{P}_{i_{j_0}})$;
endwhile
return ($\mathcal{P}$, $C(\mathcal{P})$)

Note that in Step 1 of `Heurist_Algorithm` we randomly choose a query from the query pool in order to avoid a local optimal solution in searching the solution space. It is easy to see that the server assignment problem is a constraint K-partition problem. Thus, some efficient, heuristic algorithms such as simulated annealing, and a pair exchange algorithms for solving a general K-partition problem can be adopted to solve this problem. Based on the framework of `Heurist_Algorithm`, we have presented two variants of the algorithm using simulated annealing and pair exchange techniques, respectively [6].

4 Wavelength Assignment

Since the WDM network has tremendous bandwidths and very fast transmission speed, we do not take into account the bandwidth consumption and the time delay for the message delivery through the network. However, the number of wavelengths in an optical network is a very critical resource. Therefore, we focus on assigning the wavelengths to the query results efficiently for transmission.

Recall that $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ is the wavelength set. Associated with each client (site) c_i, there is a subset $\Lambda(c_i) \subseteq \Lambda$, which means client c_i can send and/or receive data using any one wavelength in $\Lambda(c_i)$. Based on this assumption, we have the following questions. (1) If the answers of all queries are allowed to be sent back to their clients, then, how many rounds is needed. (2) If the number of rounds r is fixed in advance, then how to find a wavelength assignment plan such that the answers of the maximum number of queries (clients) are transmitted to their clients, i.e., find a wavelength assignment to maximize the throughput of the system. In the rest of this section we dedicate to solve the above problems.

We construct a bipartite graph $G_{XY} = (X, Y, E_{XY})$ as follows. Let $X = Q$ be the query set and Y be the wavelength set. There is an edge (x, y) in E_{XY} if the answer of query x can be sent using wavelength y. Having G_{XY}, for every vertex $x \in X$, *the neighboring vertex set* of x is defined as $N(x) = \{y \mid y \in Y \text{ and } (x, y) \in E_{XY}\}$. Given G_{XY}, *an equivalence class* of a vertex x in X is defined as follows: (1) for any $x \in X$, x is equivalent to itself; (2) for any two queries x_1 and x_2 in X, if x_1 and x_2 are identical and $N(x_1) = N(x_2)$, then they are in the same equivalence class. Denote by $[x] = \{x' \mid x' \text{ is equivalent to } x, \text{ and } x' \in X\}$ an equivalence class, where x is *the representative* of the equivalence class, and denote by $||[x]||$ the number of queries in $[x]$. Without loss of generality, we assume that X is partitioned into d equivalence classes $[x_i]$, $1 \leq i \leq d$. Then, $X = \cup_{i=1}^{d}[x_i]$. This equivalence class partition has the following nice property.

Theorem 1 *Let X be the query set of clients ($|X| = n$) and $Q' = \{x_1, x_2, \ldots, x_d\}$ be the set of the representatives of d equivalence classes of X and $X = \cup_{i=1}^{d}[x_i]$. Then, if there is a wavelength assignment scheduling which can transmit the answers of all queries in Q' back to their clients within r rounds, then there is a corresponding wavelength assignment scheduling which can transmit the answers of all queries in X back to their clients within r rounds.*

Theorem 1 implies that the wavelength assignment problem on X can be transformed to the wavelength assignment problem on a smaller set Q' without any additional wavelength (cost) consumption. In the rest of our discussion we assume that queries in different equivalence classes are distinct. In other words, all identical queries are in the same equivalence class. Obviously, if we assume $\Lambda = \Lambda(c_i)$ for all clients c_i, then $N(q_i) = N(q_j)$ for any two queries, and all identical queries are in the same equivalence class. It must be mentioned that the assumption is not true in the general case.

4.1 Algorithm for the optimal round assignment

Using G_{XY} and the equivalence class partition of X, we define a weighted bipartite graph $G_w = (X_w, Y_w, E_w, w)$ as follows. $X_w = \{x_1, x_2, \ldots, x_d\}$ is the set of representatives of all equivalence classes of X, and $Y_w = \Lambda$. There is an edge (x_i, y_j) in E_w with $x_i \in X_w$ and $y_j \in Y_w$ if $y_j \in \Lambda(c_i)$ for c_i. This edge indicates that client c_i and all other clients whose queries are identical to x_i can receive their answers, using wavelength y_j. The weight assigned to the edge $w(x_i, y_j)$ is $||[x_i]||$ — the number of queries in $[x_i]$.

Having G_w, we construct another weighted, bipartite graph $G^{(i)} = (X_w, Y^{(i)}, E^{(i)}, w^{(i)})$ as follows. $Y^{(i)} = \{y^{(1)}, y^{(2)}, \ldots, y^{(i)} \mid y \in Y_w\}$, $E^{(i)} = \{(x_a, y^{(j)}) \mid (x_a, y) \in E_w, \text{ and } 1 \leq j \leq i\}$. For each edge $(x_a, y^{(j)}) \in E^{(i)}$, the weight associated with it is $w^{(i)}(x_a, y^{(j)}) = w(x_a, y)$.

Since all the queries in Q must be answered, for a specific query, it is not important that which round it should be transmitted. The basic idea behind our algorithm is described as follows. In each round, a wavelength is only used to transmit the answer of a distinct query. Otherwise, if the answers of two distinct queries are transmitted by the same wavelength at the same round, then none of the clients of the two queries can receive their answers, by the definition of the WDM network. Thus, the wavelength assignment in each round is a matching of G_w, but this matching is not necessarily the maximum cardinality matching of G_w.

Let r be the minimum number of rounds needed to transmit all query answers to their clients. We have shown that there is a minimum i such that the endpoint set of a maximum cardinality matching in $G^{(i)}$ includes all vertices of X_w and $i = r$ [6]. The algorithm is presented as follows.

`Procedure Optimu_Rounds`(G_w)
$i := 0$; /* i is the minimum number of rounds */
repeat
    $i := i + 1$;
    construct a weighted bipartite graph $G^{(i)}$;
    /* when $i = 1$, $G^{(1)} = G_w$ */
    find a maximum matching $\mathcal{M}^{(i)}$ of $G^{(i)}$;
until all vertices in X_w are the endpoints of $\mathcal{M}^{(i)}$;
return $\mathcal{M}^{(i)}$.

Theorem 2 *Algorithm* `Optimu_Rounds` *runs in time $O(rm\sqrt{|X_w| + r \cdot |\Lambda|})$, and assigns the queries with the correct wavelengths. Also, the number of rounds needed for such an assignment is the minimum, where r is the minimum number of rounds needed, and m is the number of the edges in G_w and $|\Lambda| = |Y_w|$.*

4.2 Assignment with the maximum system throughput

In the previous subsection, although we assign a weight to each edge in G_w, we do not use the weight in our proposed algorithm. Now we consider the case where the number of rounds (e.g., r rounds) is fixed, then some of the query answers might not be sent back to their clients because of the limited numbers of wavelengths and rounds. Thus, this leaves us a problem, that is, which query answers should be sent to their clients such that the *throughput* of the system ρ is maximized, where ρ is defined as follows.

$$\rho \quad = \quad \frac{\textit{the number of queries answered}}{n} \qquad (1)$$

In the following we give a scheduling algorithm to maximize ρ. The algorithm basically is the same as the previous algorithm, the only difference is that here is to find a maximum weighted matching instead of a maximum cardinality matching in the previous algorithm. The algorithm is presented as follows.

`Procedure Maxim_Throughput`(G_w)
$i := 0$; /* i is the number of rounds */
repeat
    $i := i + 1$;
    construct a weighted bipartite graph $G^{(i)}$;
    /* when $i = 1$, $G^{(1)} = G_w$ */
    find a maximum matching $\mathcal{M}^{(i)}$ in $G^{(i)}$;
until (all vertices in X_w are the endpoints of $\mathcal{M}^{(i)}$)
        or $(i = r)$;
return $\mathcal{M}^{(i)}$.

As a result, the maximum weighted matching found is $\mathcal{M}^{(i)}$. The throughput of the system ρ' by this algorithm is the maximum one, where

$$\rho' \quad = \quad \frac{\sum_{(x,y)\in\mathcal{M}^{(i)}} w^{(i)}(x,y)}{n}. \qquad (2)$$

Theorem 3 *$\rho' = \rho$, where ρ is the best possible throughput of the system.*

So far, we process the queries of clients by first processing all queries at the servers, followed by transmitting the results back to the clients. However, as we know, for a given number of rounds r, although all queries are processed, some of the query answers never be transmitted back to their clients due to the limited number of wavelengths (bandwidth). Thus, it is not efficient in terms of the utilization of servers. A better way to deal with this case is presented as follows.

Step 1. Find a scheduling such that the maximum number of queries can be answered, i.e., find a scheduling such that the throughput of the system ρ is the maximum, using the above proposed algorithm.

Step 2. Let $\mathcal{M}^{(k)}$ be the result of the scheduling and Q' the vertex set in $\mathcal{M}^{(k)}$ which is also a subset of X_w, where $k = \min\{i, r\}$ and i is the returning result by our algorithm.

Step 3. Assign the queries in Q' to the servers by calling the load-balancing algorithm to distribute the queries in Q' to the servers and processing the queries. The answers are then returned to the clients whose queries are either in Q' or identical to the queries in Q'.

Therefore we have the following theorem.

Theorem 4 *Given a fixed number of rounds r, there is a wavelength assignment algorithm which delivers the maximum throughput of the system. The algorithm runs in time $O(rm \log(N_g \cdot C)\sqrt{N_g \alpha(rm, N_g) \log N_g})$, where $N_g = |X_w| + r \cdot |\Lambda|$, $C = \max\{w(x,y) \mid (x,y) \in E_w\}$, and m is the number of the edges of G_w and $|\Lambda| = |Y_w|$.*

4.3 Processing time vs throughput

In algorithm `Maxim_Throughput`, we treat the throughput of the system more importantly than the processing time of the servers. In some cases the servers may take a longer time to find the answers of the queries chosen by the wavelength assignment algorithm. As we know, both the number of wavelengths for data transmission and the processing time of the servers are important resources. In an ideal environment we wish to have such an assignment so that both resources are utilized efficiently. However, it seems that there is no trivial solution for this problem, though there is actually a trade-off between them.

Now, the problem is, given a fixed number of rounds r, how we can find a scheduling such that as many queries as possible will be answered and the total processing time for the chosen queries assigned to each server is minimized. Here we give a heuristic algorithm for this problem.

Using G_w, we first construct a vertex and edge weighted, bipartite graph $G_B = (X_w, Y_w, E_w, w_e, w_v)$ as follows. For every $(x,y) \in E_w$, $w_e(x,y) = w(x,y)$, which is the number of identical queries that use wavelength y, for each $x \in X_w$, $w_v(x) = ans(x)$ which is the cost for processing query x at a server, and for every $y \in Y_w$, $w_v(y) = 0$. Using graphs G_w and G_B, we then construct another auxiliary weighted, bipartite graph $G'_w = (X_w, Y_w, E_w, w')$ with $w'(x,y) = (1-\beta)w_e(x,y) + \beta \cdot w_v(x)$ for every $(x,y) \in E_w$, which trade-offs the number of queries to be answered and the processing time for each of the queries, where β is a fixed constant with $0 < \beta < 1$, which indicates which factor is more important than the other. Run algorithm `Maxim_Throughput`, using G'_w instead of G_w, a solution for the problem is then obtained. Thus,

Theorem 5 *Given a fixed number of rounds r, there is a heuristic assignment algorithm which delivers a solution for our problem by balancing the throughput of the system and the processing time in the servers. The algorithm runs in time $O(rm \log(N_g \cdot C)\sqrt{N_g \alpha(rm, N_g) \log N_g})$, where $N_g = |X_w| + r \cdot |\Lambda|$, $C = \max\{w(x,y) \mid (x,y) \in E_w\}$, and m is the number of the edges of G_w and $|\Lambda| = |Y_w|$.*

5 Conclusions

In this paper we have modeled the query subscription problem in WDM network environments by introducing a conceptual and cost models. The problem basically consists the server and wavelength assignment problems. For the server assignment problem, we have proposed a heuristic algorithm for it. For the wavelength assignment problem, we have studied its two versions. One is to find an assignment to minimize the number of rounds needed; and another is to find an assignment to maximize the system throughput if it is impossible to send the all query answers back to their clients due to the limited number of rounds. If all clients have the same wavelength set, the two versions of the problem are polynomially solvable. We have presented algorithms for them through a nontrivial reduction. Both algorithms are efficient.

References

[1] A. Crespo, O. Buyukkokten, and H. Garcia-Molina. Efficient query subscription processing in a multicast environment. *Technical Report*, DCS, Stanford Univ., 1998.

[2] M. Franklin and S. Zdonik. Dissemination-based information systems. *Bulletin of the Technical Committee on Data Engineering*, **19:**20–30,1996.

[3] H. N. Gobow and R. E. Tarjan. Faster scaling algorithms for general graph matching problems. *J. ACM.*, **38:**815–853, 1990.

[4] P. E. Green. *Fiber- Optic Communication Networks.* Prentice Hall, 1992.

[5] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, **2:**225–231, 1973.

[6] W. Liang and M. E. Orlowska. Optimizing processing of query subscription in an WDM network environment. *Technical Report*, DCS, Australian National Univ., 1999.

[7] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications.* Mathematics and its Applications. Reidel Publishing Company, 1987.

[8] M. V. Mannino, P. Chu, and T. Sager. Statistical profile estimation in database systems. *ACM Computing Surveys*, **20:**191–221, 1988.

[9] A. D. McAulay. *Optical Computer Architectures.* John Wiley, 1991.

[10] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimizations: Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[11] P. Raghavan and C. D. Thompson. Randomized rounding: provably good algorithms and algorithmic proofs. *Combinatorica* **7:**365–374, 1987.

[12] R. Ramaswami. Multi-wavelength lightwave networks for computer communication. *IEEE Communications Magazine*, **31:**78–88, 1993.

[13] R. J. Vitter and D. H. C. Du. Distributed computing with high-speed optical networks. *IEEE Computer* **26:**8–18, 1993.