A Genetic Algorithm-Based Methodology for Optimizing Multiservice Convergence in a Metro WDM Network

Hyo-Sik Yang, Martin Maier, Martin Reisslein, *Member, IEEE*, and W. Matthew Carlyle

***Abstract*—We consider the multi-objective optimization of a multi-service arrayed-waveguide grating-based single-hop metro WDM network with the two conflicting objectives of maximizing throughput while minimizing delay. We develop and evaluate a genetic algorithm based methodology for finding the optimal throughput-delay tradeoff curve, the so-called Pareto-optimal frontier. Our methodology provides the network architecture (hardware) and the Medium Access Control (MAC) protocol parameters that achieve the Pareto-optima in a computationally efficient manner. The numerical results obtained with our methodology provide the Pareto-optimal network planning and operation solutions for a wide range of traffic scenarios. The presented methodology is applicable to other networks with a similar throughput-delay tradeoff.**

***Index Terms*—Arrayed-waveguide grating, genetic algorithm, medium access control protocol, metropolitan area network, multi-objective optimization, Pareto-optimal, wavelength-division multiplexing (WDM).**

I. Introduction

OPTICAL single-hop wavelength division multiplexing (WDM) networks have the potential to provide high throughput and low delay connectivity in metropolitan and local area settings, as demonstrated by recent studies [1]–[6]. The throughput-delay performance of these single-hop WDM networks is typically very sensitive to the setting of the *architecture parameters* (e.g., degree of underlying arrayed-waveguide grating (AWG), degree of employed combiners and splitters) and the medium access control (MAC) *protocol parameters* (e.g., length of frames in timing structure, number of control slots, node back-off probability). For good network performance, these parameters must be set properly, which is a challenge due to the large search space of possible parameter combinations and the typically computationally demanding evaluation of a particular parameter combination. Importantly, in single-hop WDM networks, the objectives to maximize the throughput while minimizing the delay are typically conflicting. With certain combinations of parameter settings, the networks achieve a small delay and moderate throughput, which is perfectly suited for *delay-sensitive* traffic with moderate throughput requirements, such as voice traffic. On the other hand, certain combinations of parameter settings achieve a large throughput but introduce some moderate delays, which is perfectly suited for *throughput-sensitive* traffic that can tolerate some delays, such as Internet (FTP, HTTP, e-mail) and Frame Relay traffic. Typically, these different types of traffic dominate during different times of the day, as illustrated in Fig. 1(a)–(c) [7]. During office hours, voice traffic dominates the network load. Whereas Internet and Frame Relay traffic play a major role in the evening and at night, respectively. By carrying these heterogeneous traffic types in a single converged network the utilization of the network resources can be significantly increased, as illustrated in Fig. 1(d). The resulting multi-service network enables revenue-generating services in an efficient and cost-effective way [8], [9]. This is very important especially in cost-sensitive metropolitan and local area networks.

The challenge of multi-service convergence lies in i) providing the different types of small delay—moderate throughput and large throughput—moderate delay service at different times of the day in a given fixed installed network, and ii) providing these different service types efficiently, e.g., achieving the largest possible throughput in the small delay—moderate throughput regime. Optimizing the parameter setting in single-hop WDM networks for multi-service convergence thus gives rise to a so-called multi-objective optimization problem. This multi-objective optimization problem does not have a single solution; instead, the solution is a Pareto-optimal tradeoff curve between throughput and delay. Roughly speaking, this tradeoff curve gives the smallest achievable delay as a function of the desired throughput, or conversely, the largest achievable throughput as a function of the tolerable delay. Finding the optimal tradeoff curve as well as the combinations of parameter settings that attain this optimal tradeoff curve is a challenging problem. This is due to the large search space of parameter combinations and the typically demanding evaluation of an individual parameter combination. The optimal tradeoff curve, however, is crucial for 1) the planning and provisioning of new networks, i.e., to determine the best architecture (hardware) parameters, and 2) the efficient operation of installed network hardware. The Pareto-optimal throughput-delay tradeoff curve

Manuscript received April 23, 2002; revised February 5, 2003. This work was supported in part by the National Science Foundation under Grant No. Career ANI-013322 and the German Federal Ministry of Education and Research under the TransiNet Project.

H.-S. Yang is with the Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287–7206 USA (e-mail: yangkoon@asu.edu).

M. Maier is with the Telecommunication Networks Group, Technical University Berlin, 10587 Berlin, Germany (e-mail: maier@ee.tu-berlin.de).

M. Reisslein is with the Telecommunication Research Center, Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-7206 USA (e-mail: reisslein@asu.edu).

W. M. Carlyle was with the Department of Industrial Engineering, Arizona State University, Tempe, AZ 85287-7206 USA. He is now with the Operations Research Department, Naval Postgraduate School, Monterey, CA 93943 USA (e-mail: mcarlyle@nps.navy.mil).

Digital Object Identifier 10.1109/JLT.2003.811564

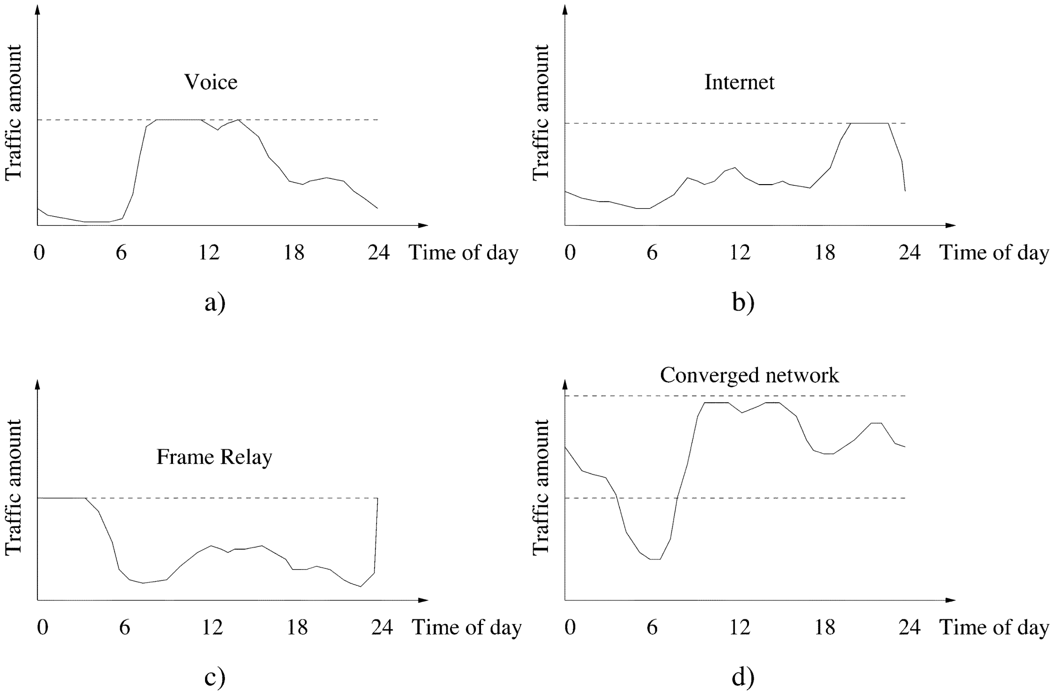


Fig. 1. Different types of traffic dominate during different times of the day.

can thus be used in a two-step optimization process as follows. First, we optimize a *new* network by finding the optimal architecture (hardware) parameter values. Second, after fixing the architecture, we optimize the protocol (software) parameters for an *existing* architecture. Specifically, we operate the network at different points of its Pareto-optimal throughput-delay tradeoff curve according to the traffic type that dominates at a given time of the day. The network protocol parameters are tuned to provide varying degrees of i) small delay (and moderate throughput) service, or ii) large throughput (and moderate delay) service as the traffic changes with the time of the day. This tuning requires detailed knowledge of the optimal tradeoff curve, which can be precomputed with our methodology and stored in tables for fast look-up.

In this paper, we develop a genetic algorithm based methodology for solving the multi-objective optimization problem of maximizing throughput and minimizing delay in single-hop WDM networks. We consider the arrayed-waveguide grating (AWG)-based network [2] as an example throughout this paper. Our methodology finds the optimal tradeoff curve and the parameter combinations attaining the curve in a computationally efficient manner. Our work enables network planners to select the (hardware) network architecture parameters that give the best performance. In addition, our methodology enables the operators of (fixed) installed network hardware to optimally tune the throughput-delay performance along the optimal tradeoff curve by changing the (software) network MAC protocol parameters.

While we focus on the AWG-based network [2] in this work, our methodology applies analogously to networks with a similar throughput-delay tradeoff. Our genetic algorithm based approach takes an analytic characterization of the mean throughput and the mean delay of the network as input. This analytic characterization may involve highly nonlinear equations (or possibly systems of equations); we only require that the equations can be solved numerically. Our methodology may also be applied to networks that are analytically intractable and require simulations to obtain the (mean) throughput and the (mean) delay. The computational effort required to obtain the optimal throughput-delay tradeoff curve for a given traffic load with our approach depends on the effort required to evaluate the throughput and the delay for a particular combination of network parameters and the size of the exhaustive search space. The number of parameter combinations that our approach needs to evaluate to obtain the optimal tradeoff curve is usually on the order of thousand times smaller than the exhaustive search space. In typical scenarios, our approach requires less than one day of CPU time on a 933 MHz PC to find the optimal tradeoff curve, whereas the exhaustive search would require several years of CPU time.

This paper is organized as follows. In the following section we review the related work on optimizing optical WDM networks, including works that employ genetic algorithm based approaches. In Section II, we formulate the multi-objective optimization problem of maximizing throughput while minimizing delay. We briefly review the AWG-based single-hop WDM network [2], which is used as an example throughout the paper. We give the two objective functions (throughput and delay), we identify the decision variables in the optimization and discuss the constraints on the decision variables. In Section III, we develop our genetic algorithm based methodology for finding the Pareto-optimal throughput-delay tradeoff curve. First, we briefly review the notion of multi-objective optimization and explain why we base our solution methodology on genetic algorithms. We then discuss and evaluate in detail the individual components of our methodology. In Section IV we apply our methodology to the AWG-based single-hop WDM network and study its optimal throughput-delay tradeoffs in detail. We summarize our conclusions in Section V.

A. Related Work

We now give a brief overview of the literature on optimization in optical WDM networks, which may be broadly categorized into studies addressing i) wide-area wavelength-routed mesh WDM networks (typically envisioned as Internet backbone networks), ii) WDM ring networks, and iii) WDM networks with a physical star topology (typically employed in the metro/local area with a central passive star coupler (PSC) or AWG). The design and operation of wavelength-routed mesh (wide area) WDM networks have been optimized extensively, including aspects such as the routing and wavelength assignment, as well as the design of optimal logical topologies, see for instance [10]–[13], and references therein. Also, optimality issues in planning and operation of survivable wavelength-routed WDM networks have been thoroughly investigated, see for instance [14], [15] and references therein. The optimal placement of wavelength converters in WDM mesh networks is studied in [16], while [17] studies the optimal amplifier placement. The optimal setting of physical parameters in optical networks, such as the power budget and detection thresholds, have also been investigated, e.g., [18], [19]. General strategies for the optimal planning of optical networks are explored in [20].

WDM ring networks (including SONET/SDH rings) have received a great deal of attention and a wide range of aspects of ring networks, including the placement of add–drop multiplexers, traffic grooming strategies, the provisioning of wavelengths and hardware components to ensure network survivability, as well as MAC protocols and wavelength assignment have been optimized, see for instance, [21]–[24].

WDM networks with a physical star topology are typically studied in the context of single-hop networks [25] or multi-hop networks [26]. For multi-hop networks, much research has gone into the design of optimal virtual topologies (see for instance the survey [26]). For single-hop networks most optimization efforts have focused on the optimal scheduling, see for instance [27] and [28]. Our optimization methodology is orthogonal to these studies in that our methodology optimizes the architecture and MAC protocol parameters of the network without assuming any particular scheduling mechanism. (To fix ideas a simple FCFS scheduling policy is used in [29], where the mean throughput and the mean delay of the network considered in this paper are derived.) A unique aspect of our work is that we jointly optimize the network *architecture* (hardware) and the MAC *protocol* parameters (software). Generally, the existing works, in isolation optimize either hardware or software parameters. We also note that most of the existing literature on single-hop WDM networks considers networks based on a central PSC, which is a broadcast device and hence does not allow for spatial wavelength reuse. In contrast, we consider a network based on an AWG, which provides wavelength-sensitive routing and thus allows for spatial wavelength reuse. This allows for increased concurrency and as we demonstrate in this paper, makes the AWG based network a promising candidate for efficiently achieving multi-service convergence in metro area networks. (The wavelength routing property of the AWG has recently also been exploited in other networking contexts, e.g., in optical packet switches [30].)

Another distinguishing feature of our work is that we explicitly consider a multi-objective optimization problem, whereas most of the existing literature focuses on optimizing a single objective function. Optical network optimization with multiple conflicting objectives is considered only by a few studies. In [31] reconfiguration policies to accommodate changing traffic (routing) patterns or the failure of network components in a PSC-based single-hop WDM network are studied. It is found that maximizing the degree of load balancing and minimizing the number of transceiver retunings are conflicting objectives. The problem is formulated in a Markov decision process framework, which is used to evaluate reconfiguration policies. The reconfiguration policy that achieves the desired balance between the two conflicting objectives is determined by selecting proper cost functions and weights for the objectives. In [24] it is noted that minimizing the number of nodes (optical add–drop multiplexers) and minimizing the number of rings in a stack of WDM rings are conflicting objectives; the tradeoff is quantified and a heuristic for finding a spectrum of designs is developed. Similarly, in [22], [23] it is observed that the objectives to minimize the number of optical add–drop multiplexers and to minimize the number of wavelengths in a WDM ring network are conflicting and a number of designs that strike different balances between the objectives are proposed. In [32] a multi-objective optimization problem to find the wavelength assignment in a mesh WDM network that minimizes the path lengths while maximizing the fiber utilizations is formulated and solved using genetic algorithms.

A wide range of optimization methods are employed in the reviewed optical network optimization studies. Some use traditional optimization methods that are guaranteed to find the global optimum, such as integer linear programming, employed for instance in [10], [15]. However, due to the complexity of the problems and the prohibitive computational effort required for solving them with traditional methods, novel algorithms and heuristics are developed (e.g., [13]) and heuristic algorithms, such as Tabu-search (e.g., in [11]), simulated annealing (e.g., in [12]), and genetic algorithms (in [17], [32]–[36]) are applied. We note that the use of evolutionary (genetic) algorithms in the design of general wide area mesh network topologies that minimize the network cost is studied in [37]. Genetic algorithms are compared with simulated annealing for optimizing the topological design of a network in [38] and it is found that genetic algorithms give better performance than simulated annealing. The existing studies employing genetic algorithms for optical network optimization typically optimize a *single* objective, e.g., minimize the number of amplifiers [17], minimize the network cost [35], [36], or maximize the number of connections while satisfying power constraints [33]. In contrast, in this paper we

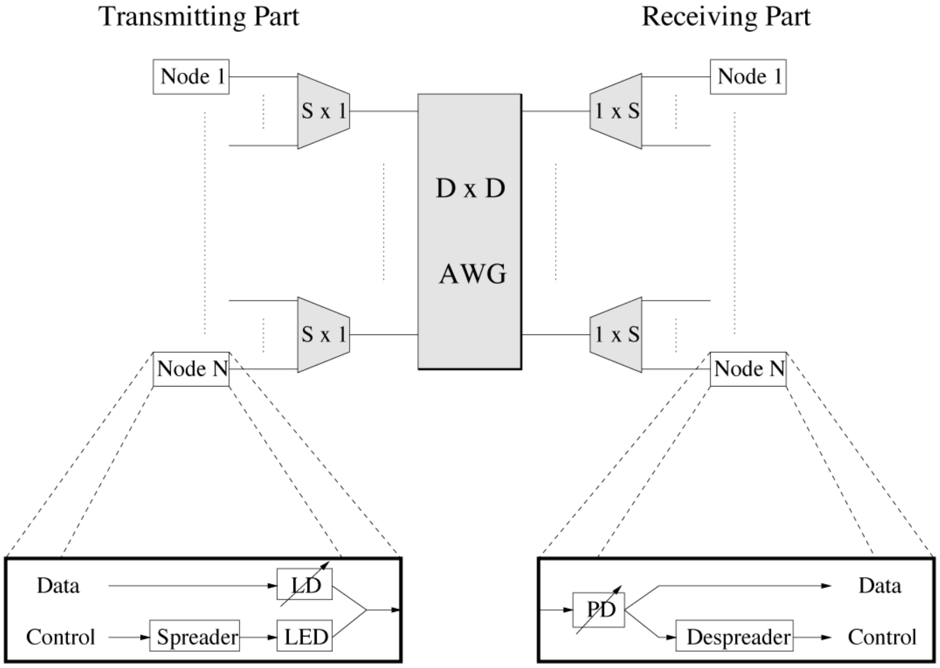


Fig. 2. Architecture of AWG based WDM network.

consider a *multi-objective* optimization problem—minimize delay while maximizing throughput.

II. Formulating the Multi-Objective Optimization Problem

In this section we formulate the multi-objective optimization problem of maximizing throughput while minimizing delay in single-hop WDM networks. We first review the AWG-based single-hop WDM network [2], which we use as an example network throughout this paper.

A. Overview of AWG-Based Single-Hop WDM Network

The basic architecture of the single-hop WDM network [2] is based on a $D \times D$ AWG, as shown in Fig. 2. At each AWG input port, a wavelength-insensitive $S \times 1$ combiner collects data from S attached nodes. Similarly, at each AWG output port, signals are distributed to S nodes by a wavelength-insensitive $1 \times S$ splitter. (An Erbium Doped Fiber Amplifier (EDFA) is placed at the output of each combiner and the input of each splitter to compensate for the splitting/combining and fiber losses.) Each node is composed of a transmitting part and a receiving part. The transmitting part of a node is attached to one of the combiner ports. The receiving part of the same node is located at the opposite splitter port. The network connects $N = D \cdot S$ nodes. At each AWG input port we exploit R adjacent Free Spectral Ranges (FSR's) of the AWG, each FSR consists of D contiguous wavelengths. The total number of wavelengths at each AWG input port is $\Lambda = D \cdot R$. The network runs an attempt-and-defer type of MAC protocol, i.e., a data packet is only transmitted after the corresponding control packet has been successfully transmitted. In the MAC protocol, time is divided into cycles. Each cycle consists of D frames. Each frame contains F slots. The slot length is equal to the transmission time of a control packet. Each frame is partitioned into the first M, $1 \leq M < F$, slots and the remaining $(F - M)$ slots. In the first M slots, control signals are transmitted based on a modified slotted ALOHA protocol and all nodes must be tuned (locked) to one of the Light Emitting Diode (LED) slices carrying the control information. (This LED slice broadcast mechanism can also be used to quickly update the protocol parameters in all network nodes. By looking up the appropriate parameter settings in a table precomputed with our methodology and broadcasting them to the nodes with the LED slices in one single hop, the network is able to adapt almost instantly to changing traffic conditions and throughput-delay requirements.) In every frame within the cycle, the nodes attached to a different AWG input port send their control packets. Specifically, all nodes attached to AWG input port o, $1 \leq o \leq D$, (via a common combiner) send their control packets in frame o of the cycle. During the first M slots of frame o, control and data packets can be transmitted simultaneously by the nodes attached to AWG input port o. Transmissions from the other AWG input port cannot be received during this time interval. In the last $(F - M)$ slots of each frame, no control packets are sent. The receivers are unlocked, allowing transmission between any pair of nodes. This allows for spatial wavelength reuse. In the considered traffic scenario, a node that is not backlogged generates a new packet with probability σ at the beginning of its transmission cycle. The generated packet is long (has size F slots) with probability q, and is short (has size $K = F - M$ slots) with probability $1 - q$. The parameters of the considered network architecture and MAC protocol, as well as the traffic parameters are summarized in Table I.

B. Objective Functions: Throughput and Delay

The two key performance metrics of single-hop WDM networks, such as the AWG-based network reviewed in the preceding section, are the mean throughput and the mean delay. The

TABLE I
PARAMETERS OF NETWORK ARCHITECTURE AND MAC PROTOCOL

	Network Architecture (Hardware) Parameters
N	Number of nodes in the network
Λ	Number of usable wavelengths at each AWG port (Tuning range of transceivers)
D	Degree of AWG
R	Number of FSRs ($R = \Lambda/D$)
S	Degree of combiner and splitter ($S = N/D$)
	Protocol (Software) Parameters
F	Number of slots in a frame
M	Number of reservation slots in a frame
K	Length of short packets in slots ($K = F - M$)
p	Re–transmission probability of control packet in ALOHA contention
	Traffic Parameters
σ	Packet generation probability (for idle node at beginning of cycle)
q	Probability that a given data packet is long (i.e., occupies F slots)
	Performance Metrics (Objective Functions)
TH	Average network throughput in transmitting nodes per slot (or equivalently in packets/frame)
$Delay$	Average packet delay in slots

typical goal of the optimization of single-hop WDM networks is to maximize the throughput while minimizing the delay. For the reviewed AWG-based network, the mean throughput and the mean delay have been derived in [29] as functions of the parameters summarized in Table I. (The derivation in [29] considered the case $M < F$, i.e., $K > 0$. In our optimization, we allow for $M \leq F$, i.e., $K \geq 0$; the objective functions for the special case $M = F$ are derived in [39].) We briefly review here these two objective functions of our optimization.

The average throughput of the network is defined as the average number of transmitting nodes in a slot and is given by

$$TH = D^2 \cdot \frac{F \cdot E[\mathcal{L}] + K \cdot E[\mathcal{S}]}{F \cdot D} \tag{1}$$

where $E[\mathcal{L}]$ is the expected number of successfully scheduled long packets (of size F slots) from a given (fixed) AWG input port to a given (fixed) AWG output port per cycle (of length $F \cdot D$ slots), and $E[\mathcal{S}]$ is the expected number of successfully scheduled short packets (of length $K = F - M$ slots) from a given (fixed) AWG input port to a given (fixed) AWG output port per cycle. (We note that the throughput given by (1) may also be interpreted as the average number of transmitted data packets per frame; for convenience we will use this packets/frame interpretation in our numerical work in Sections III and IV) $E[\mathcal{L}]$ and $E[\mathcal{S}]$ are evaluated by modeling the control packet contention and the data packet scheduling, and then establishing a set of equilibrium equations for the network. In brief, the arrival rate of control packets to a given control slot is expressed as

$$\beta = \frac{S}{M}[\sigma v + p(1 - v)] \tag{2}$$

where v is the fraction of idle (i.e., not backlogged) nodes in steady state. The number of successful (i.e., not collided) control packets destined to a given AWG output port in a given frame is expressed as

$$\begin{aligned} P(Z = k) = &\binom{M}{k} \left(\frac{\beta e^{-\beta}}{D}\right)^k \left(1 - \frac{\beta e^{-\beta}}{D}\right)^{M-k}, \\ k =&0, 1, \ldots, M. \end{aligned} \tag{3}$$

The probability that a given control packet corresponds to a long data packet (either newly generated by an idle node, or retransmitted by a backlogged node) is denoted by $\tilde{q}$; note that typically $\tilde{q} > q$ since long data packets are more difficult to schedule and thus typically require more retransmissions than short packets. The analysis of the data packet scheduling results in

$$\begin{aligned} E[\mathcal{L}] =&\tilde{q} \left\{ R - \sum_{k=0}^{\min(R,M)} P(Z = k)(R - k) \right\} : \\ =&\tilde{q} \cdot \varphi(\beta) \end{aligned} \tag{4}$$

and

$$\begin{aligned} E[\mathcal{S}] =&(1 - \tilde{q}) \left[R - \sum_{k=0}^{R} (R - k) \cdot P(Z = k) \right] \\ &+ \sum_{j=1}^{M-R} \gamma_j \sum_{m=j}^{M-R} \sum_{k=m+R}^{M} \binom{k - R}{m} \\ &\cdot (1 - \tilde{q})^m \tilde{q}^{k-R-m} \cdot P(Z = k) := h(\tilde{q}, \beta) \end{aligned} \tag{5}$$

where γ_j accounts for the "packing" of the short packets into the schedule and is given by a nonlinear function of the network and traffic parameters and $\tilde{q}$. Finally, in equilibrium, the numbers of serviced long and short packets are equal to the numbers of newly generated long and short packets, which, after some algebraic manipulations, results in the equations

$$\tilde{q} = q \cdot \frac{S\sigma v}{D \cdot \varphi(\beta)} \qquad (6)$$

and

$$(1-q) \cdot \frac{S\sigma}{D} \cdot v = h\left(\tilde{q}, \beta\right). \qquad (7)$$

(7) is solved numerically and the obtained v is inserted in (2) to obtain β, which in turn is used in (4) to obtain $\varphi(\beta)$. These quantities are in turn used to obtain $\tilde{q}$ from (6), and finally $E[\mathcal{L}]$ from (4) and $E[\mathcal{S}]$ from (5).

The mean packet delay is defined as the average time period in slots from the generation of the control packet corresponding to a data packet until the transmission of the data packet. The average delay in the network in slots is

$$\text{Delay} = \left\{ \frac{S}{D \cdot (E[\mathcal{L}] + E[\mathcal{S}])} - \frac{1-\sigma}{\sigma} \right\} \cdot D \cdot F. \qquad (8)$$

C. Decision Variables and Constraints

We now identify the decision variables in our optimization problem and identify the constraints on the decision variables. We select the AWG degree D as the (independent) decision variable for the network (hardware) architecture; we determine the other architecture parameters R and S (see Table I) as functions of D (and the given N and Λ), as discussed shortly. Generally, the decision variable D can take any integer satisfying

$$D \geq 2 \quad \text{and} \quad D \leq \Lambda \qquad (9)$$

where Λ is the maximum number of wavelength channels accommodated by the fast tunable transceivers employed in the considered network. In other words, Λ is the maximum tuning range of the employed transceivers divided by the channel spacing and is thus very technology dependent. [To use transceivers with a negligible tuning time (and a small tuning range) we set $\Lambda = 8$ in our numerical investigations in Sections III and IV.] We also note that the number of ports of commercially available photonic devices is typically a power of two. We can easily incorporate this constraint by restricting D to the set $\{2, 4, 8, \ldots\}$.

The number of used FSRs R depends on the (independent) decision variable D and the given tuning range Λ of the transceivers. Generally, R must be an integer satisfying $R \cdot D \leq \Lambda$, i.e., $R \leq \Lambda/D$. The larger R, the more parallel channels are available between each input-output port pair of the AWG, and hence the larger the throughput. Therefore, we set R to the largest integer less than or equal to Λ/D, i.e., $R = \lfloor \Lambda/D \rfloor$. We note that the tuning range Λ and degree D are typically powers of two for commercial components. Hence, Λ/D is a power of two for practical networks, and we may write $R = \Lambda/D$. The combiner/splitter degree S depends on the decision variable D and the given number of nodes in the network N. In determining the combiner/splitter degree S, it is natural to assume that the nodes are equally distributed among the D AWG input/output ports; i.e., each input/output port serves at least $\lfloor N/D \rfloor$ nodes. This arrangement minimizes the required combiner/splitter degree S, which in turn minimizes the splitting loss in the combiners/splitters. Hence, we set $S = \lceil N/D \rceil$.

We now turn to the protocol (software) parameters; see Table I. We identify three decision variables; these are F, M and p. Generally, the number of slots per frame F can take any positive integer, i.e., $F \geq 1$, while the number of control slots per frame can take any positive integer less than or equal to F, i.e., $1 \leq M \leq F$. (Note that in case $M = F$, the length of the short packets degenerates to zero. In this case only large packets contribute to the throughput; the objective functions for this case are given in [39].) We note that the size of the packets to be transported may impose additional constraints on F and M. With a given maximum packet size, F must be large enough to accommodate the maximum size packet in a frame. If short packets have a specific size requirement, $F - M$ should be large enough to accommodate that packet size. For our numerical work in Sections III and IV, we do not impose packet size requirements. Instead, we let the genetic algorithm determine the F and M values that give the optimal throughput-delay performance, subject only to $F \geq 1$ and $1 \leq M \leq F$. The packet re-transmission probability p may take any real number in the interval $[0, 1]$. To reasonably limit the search space we restrict p to [0, 0.05, 0.10, 0.15,…, 1.0] in our numerical work.

D. Network Cost Considerations

Minimizing the total network cost could be a third objective, in addition to the maximize throughput and minimize delay objectives introduced in Section II-B. We note that the genetic algorithm methodology could accommodate the third objective in a straightforward fashion, it would make the solution space three dimensional. Specifically, we would obtain an optimal throughput-delay tradeoff plane for a given (acceptable) cost level. We did not include network cost minimization in our optimization model because we are primarily interested in uncovering the fundamental performance limitations and tradeoffs in the metro WDM network. Network cost—while an important consideration—is typically not considered a fundamental performance metric for a network. In addition, network costs tend to be highly variable. The costs of the hardware components in the considered network are expected to drop significantly once they are extensively mass produced.

Even though we did not include cost minimization in our optimization model, we now briefly discuss the impact that the cost minimization objective would have on the problem and its solution. Generally, the total network cost is the sum of capital expenditures (cost of network hardware and installation) and operational expenditures (cost of network management). With the current component pricing structure, the hardware cost of the network increases linearly with the AWG degree D. This is because i) there is typically a per-port charge for an AWG, and ii) the number of required EDFAs increases linearly with D. (The cost of the splitters/combiners is typically insignificant. Also, the number of transceivers depends only on the number of network nodes.) The cost of installation is roughly fixed (and independent of the decision variables), as is the network management cost. Thus the total network cost is approximately a linear

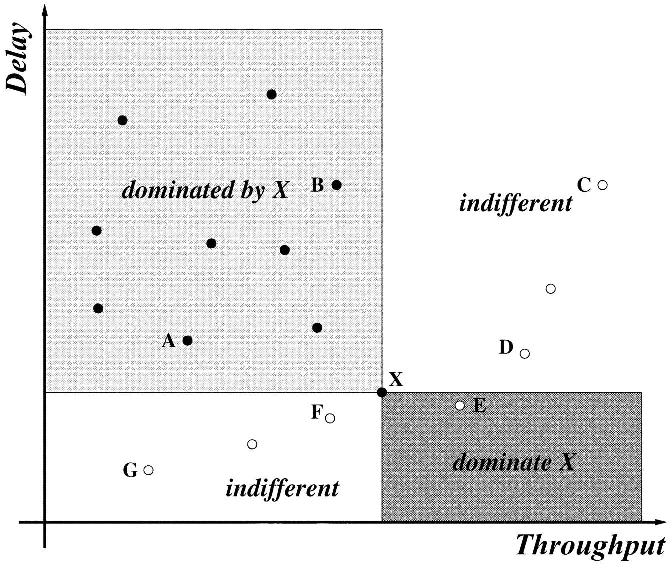


Fig. 3. Illustration of Pareto-optimal solutions for maximize throughput-minimize delay problem.

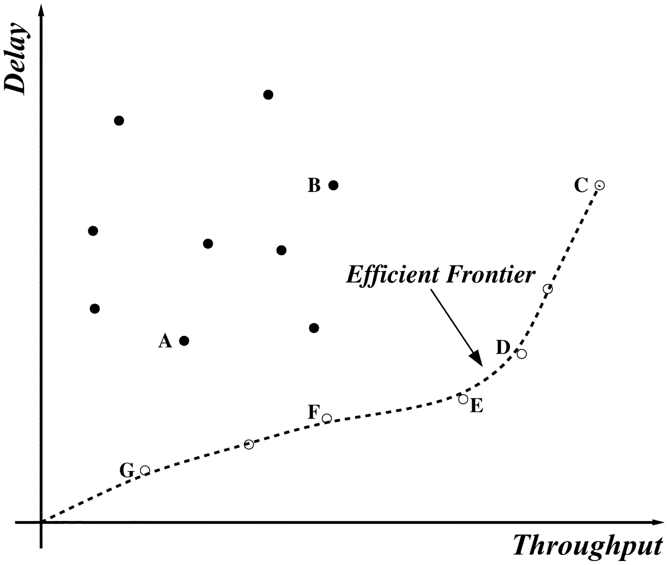


Fig. 4. Illustration of efficient frontier for maximize throughput-minimize delay problem.

function of the AWG degree D. Since D is typically a power of two, the genetic algorithm methodology would give optimal throughput-delay planes for each $D = 2, 4, \ldots$. This three dimensional solution gives the best throughput-delay tradeoff for a given acceptable cost level.

III. Genetic Algorithm Based Methodology

In this section we discuss the difficulties in optimizing the multiple objectives of maximizing throughput while minimizing delay. We point out why we base our solution methodology on genetic algorithms. We describe our genetic algorithm solution approach to the multi-objective optimization problem formulated in the previous section and evaluate the performance of our approach.

A. Why Evolutionary Algorithm (Genetic Algorithm)?

The familiar notion of an optimal solution becomes somewhat vague when a problem has more than one objective function, as is the case in our metro WDM network optimization. A solution (i.e., set of decision variables D, F, M, and p) that gives very large throughput may also give large delay and thus rate poorly on the minimize delay objective. The best we can do is to find a set of optimal tradeoff solutions, i.e., solutions that give the largest achievable throughput for a given tolerable delay, or equivalently the smallest achievable delay for a required throughput level. After a set of such optimal tradeoff solutions is found, a user can then use higher-level considerations, such as the traffic patterns illustrated in Fig. 1, to make a choice. A feasible solution to a multi-objective optimization problem is referred to as *efficient point* or *Pareto-optimal* solution [40]. As illustrated in Figs. 3 and 4, we have two objectives—maximizing throughput, and minimizing delay. The region which is shaded in light gray is said to be *dominated* by the point X. All points in the region, e.g., A and B have larger delay and smaller throughput than the point X. Clearly, the point X is superior to the points A and B. Thus all points in the light gray rectangle are dominated by point X. All points in the dark gray rectangle, e.g., the point E, are said to *dominate* the point X. Since all points in the dark gray rectangle have larger throughput and smaller delay than X. The point E is superior to the point X. Based on the concept of Pareto dominance, the optimality criterion for multi-objective problems can be introduced. Consider the points C, D, E, F and G. These points are unique among all the points in the plot in that each of them is not dominated by any other point. The set of these solutions is termed as *Pareto-Optimal* solution set or *Efficient Frontier*. The efficient frontier corresponding to Fig. 3 is shown in Fig. 4.

The goal of multi-objective optimization is to find such a feasible efficient frontier. Classical methods for generating the Pareto-optimal solution set aggregate the objectives into a single, parameterized objective function. The parameters of this function are not set by the decision maker, but systematically varied by the optimizer [41]. In contrast to classical search and optimization algorithms, evolutionary algorithms use a *population* of solutions in each iteration, instead of a single solution. Since a population of solutions is processed in each iteration, the outcome of an evolutionary algorithm is also a population of solutions for the conflicting objective functions. These multiple optimal solutions can be used to capture multiple efficient points of the problem [40].

We now proceed to develop a methodology for efficiently finding the Pareto-optimal solutions (optimal tradeoff curve) of the multi-objective problem of maximizing throughput while minimizing delay in single-hop WDM networks. Our solution methodology is based on genetic algorithms, which are members of the family of evolutionary algorithms.

B. Basic Operation of Genetic Algorithm

The basic structure of a genetic algorithm is illustrated in Fig. 5. In the genetic algorithm, we consider a population of individuals. Each individual is represented by a string of the decision variables, i.e., D, F, M, and p (as well as the corresponding objective function values TH and *Delay*). In the terminology of

```
Genetic Algorithm()
{
    $t = 0$;                               //start with an initial generation
    init_population $\mathcal{P}(t)$;
    //initialize a usually random population of individuals
    evaluate $\mathcal{P}(t)$;
    //evaluate fitness of all individuals of initial population
    while not terminated do {             //evolution cycle;
        $t \leftarrow t + 1$;                      //increase the generation counter
        $\mathcal{P}'(t) =$ select_parents $\mathcal{P}(t)$; //select a mating pool for offspring production
        recombine $\mathcal{P}'(t)$;               //recombine the 'chromosome' of selected parents
        mutate $\mathcal{P}'(t)$;                  //perturb the mated population stochastically
        evaluate $\mathcal{P}'(t)$;                //evaluate fitness of new generation
        $\mathcal{P}(t) \leftarrow \mathcal{P}'(t)$;
    }
}
```

Fig. 5. Basic structure of a genetic algorithm.

genetic algorithms the string of decision variables is referred to as *chromosome*, while each individual decision variable is referred to as *gene*. The quality of an individual in the population with respect to the two objective functions is represented by a scalar value, called *fitness*. After generating the initial population (by randomly drawing the decision variables for each individual from uniform distributions over the respective ranges of the decision variables), each individual is assigned a fitness value. The population is evolved repeatedly, generation by generation, using the crossover operation and the mutation operation. The crossover and mutation operations produce offspring by manipulating the individuals in the current population that have good fitness values. The crossover operation swaps portions of the chromosomes. The mutation operation changes the value of a gene. Individuals with a better fitness value are more likely to survive and to participate in the crossover (mating) operation. After a number of generations, the population contains members with better fitness values. The Pareto-optimal individuals in the final population are the outcome of the genetic algorithm. Each operation is discussed in detail in the following subsections.

C. Fitness Function

The fitness function is typically a combination of objective functions. We evaluate three commonly used types of fitness function. We generate $G = 20$ generations, each with a population size of $P = 200$ to compare the quality of the fitness functions. We set the probability of crossover to 0.9 and the probability of mutation to 0.05, which are typical values. We compare the genetic algorithm outputs with the true Pareto-optimal solutions which were found by conducting an exhaustive search over all possible combinations of the decision variables. We fix $\sigma = 0.6$ and $q = 0.1$ for this evaluation. All results presented in this paper assume a channel spacing of 200 GHz, i.e., 1.6 nm at 1.55 μm. Thus, we can use 7–10 wavelengths at each AWG input port with fast tunable transceivers with a tuning range of 10–15 nm [29]. For all subsequent results, the number of wavelengths is fixed at eight, i.e., $\Lambda = 8$. D can take the values 2, 4, and 8. Thus, the corresponding R values are 4, 2, and 1. We fix the number of nodes in the network at $N = 200$. To reasonably limit the search space of the genetic algorithm, we restrict F to be smaller than 400 slots in this paper. We note that with a large F, the considered network generally achieves larger throughput values (at large delays), however, the computational effort for evaluating a given parameter combination increases as F increases. For the exhaustive search, we therefore limit F to values less than or equal to 200 slots.

First, we evaluate the Vector Evaluated Genetic Algorithm (VEGA), which is easy to implement. The VEGA algorithm divides the population into two subpopulations according to our two objective functions. The individuals in each subpopulation are assigned a fitness value based on the corresponding objective function. When using only one objective function to determine the fitness values of the individuals in a subpopulation, it is likely that solutions near the optimum of an individual objective function are preferred by the selection operator. Such preferences take place in parallel with other objective functions in different subpopulations. The main disadvantage of VEGA is that typically after several generations, the algorithm fails to sustain diversity among the Pareto-optimal solutions and converges near one of the individual solutions. Indeed, as reported in Table II, the VEGA finds only 15 Pareto-optimal solutions; the efficient frontier spanned by these solutions is plotted in Fig. 6. We observe, however, that the VEGA efficient frontier is overall quite close to the true efficient frontier (found by exhaustive search).

Next, we evaluate the Weight Based Genetic Algorithm (WBGA) which uses the weighted sum of the objective functions as fitness function. The main difficulty in WBGA is that it is hard to choose the weight factors. We use the same weight factor of 1/2 for each objective function. Since the mean delay

TABLE II
NUMBER OF PARETO-OPTIMAL SOLUTIONS IN FINAL POPULATION FOR GENETIC ALGORITHM BASED SEARCH WITH $F \leq 400$; EXHAUSTIVE SEARCH FOR $F \leq 200$ GIVES 580 PARETO-OPTIMAL SOLUTIONS

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.6$			$\sigma = 0.8$		
q	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
$D = 2$	148	132	133	108	84	158	31	102	121	23	105	135
$D = 4$	0	1	8	2	65	4	86	46	5	102	46	3
$D = 8$	0	0	0	0	2	2	1	4	1	0	4	1
Total	148	133	141	110	151	164	118	152	127	125	155	139

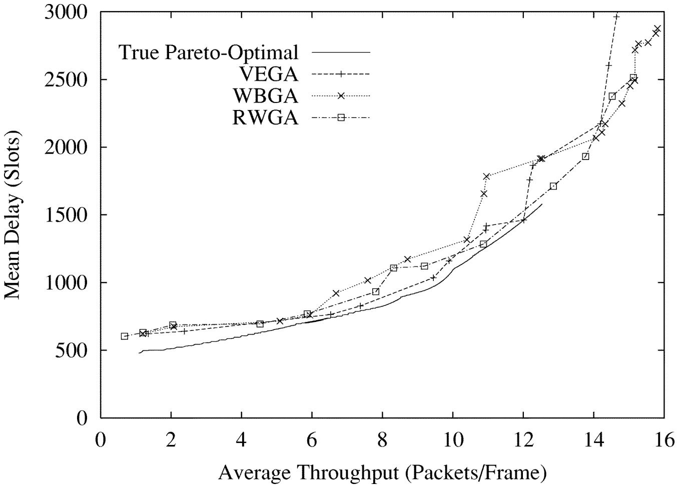


Fig. 6. Efficient frontiers obtained with different fitness functions without elitism for $F \leq 400$ and with exhaustive search for $F \leq 200$.

should be minimized in our problem, we use the negative delay as the second objective function. The fitness function used is

$$Fitness = \frac{1}{2} \cdot TH - \frac{1}{2} \cdot Delay. \tag{10}$$

Our goal is to maximize the average throughput while minimizing the mean delay. Thus, with the WBGA approach, the larger the fitness value, the better. We observe from the results given in Fig. 6 and Table II that the WBGA finds more Pareto-optimal solutions than VEGA. However, the WBGA efficient frontier has parts (particularly in the throughput range from 7–13 packets/frame) that are distant from the true efficient frontier. We note that the average network delay given in (8) in units of slots is on the order of thousands of slots in typical scenarios, whereas the average throughput is typically on the order of one to 16 packets per frame. To achieve a fair weighing of both throughput and delay in the fitness function, we use the delay in unit of cycles (where one cycle corresponds to $D \cdot F$ slots) in the evaluation of the fitness in (10) (and the following fitness definition in (11)); with this scaling, the delay is on the order of 1 to 20 cycles in typical scenarios.

Finally, we evaluate the Random Weight Genetic Algorithm (RWGA) which weighs the objective functions randomly. A new independent random set of weights is drawn each time an individual's fitness is calculated. We use the fitness function

$$Fitness = \varepsilon \cdot TH - (1 - \varepsilon) \cdot Delay \tag{11}$$

where ε is uniformly distributed in the interval (0, 1). We observe from Fig. 6 that the RWGA efficient frontier is relatively far from the true efficient frontier in the throughput range from 8–10 packets/frame. Also, the RWGA finds only a relatively small number of Pareto-optimal solutions.

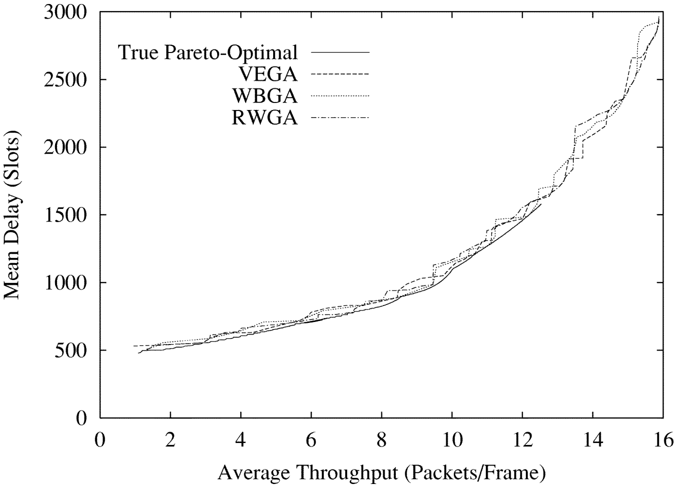


Fig. 7. Efficient frontiers obtained with different fitness functions without elitism for $F \leq 400$ and with exhaustive search for $F \leq 200$.

We now study the concept of *elitism*. Elitism is one of the schemes used to improve the search; with elitism the good solutions in a given generation are kept for the next generation. This prevents losing the already found good solutions in the subsequent crossover operation(s), which may turn good solutions into bad solutions. For each generation we determine the Pareto-optimal solutions by comparing the throughput and delay achieved by the individuals in that generation. (Note that the thus determined Pareto-optimal solutions are not necessarily the true Pareto-optimal solutions to the optimization problem, rather they are Pareto-optimal with respect to the other individuals in the considered generation.) The determined Pareto-optimal solutions are kept for the next generation; they are not subjected to the crossover operation, they are, however, subjected to the mutation operation (as explained in Sections III-E and III-F). If we find that a Pareto-optimal solution from a previous generation is no longer Pareto-optimal solution in a new generation, i.e., it is dominated by some other individual in the new generation, then this old Pareto-optimal solution is discarded.

The results obtained with elitism are given in Fig. 7 and Table II. We observe that the number of Pareto-optimal solutions in the final population is dramatically larger and the efficient frontiers are closer to the true efficient frontier of the problem. From Fig. 7, it appears that all schemes with elitism perform quite well, with RWGA hugging the true efficient frontier most closely. This observation is corroborated by comparing the number of Pareto-optimal solutions in the final population in Table II, which indicates that RWGA gives the best performance. According to the observations made in this section, we use RWGA with elitism throughout the remainder of this paper.

D. Population Size and Number of Generations

The population size trades off the time complexity (computational effort) and the number of optimal solutions. In order to accommodate all Pareto-optimal solutions, the population should be large enough. However, as the population size grows, the time complexity for processing a generation increases (whereby the most computational effort is typically expended on evaluating the throughput and delay achieved by an individual to determine its fitness value). On the other hand, for a smaller population, the time complexity for the population decreases while the population may lose some Pareto-optimal solutions. As a result, the smallest population size which can accommodate all Pareto-optimal solutions is preferable.

For schemes that employ elitism, we categorize the population in generation t into three groups: i) The *elite group* of size $P_e(t)$ which contains the Pareto-optimal solutions from the preceding generation $t-1$, ii) the *reproduction group* of size $P_p(t)$ which is reproduced from the individuals with good fitness values in the preceding generation $t-1$ through crossover (see Section III-E), and iii) the *random group* of size $P_r(t)$ which is generated randomly (by drawing the decision variables from uniform distributions over their respective ranges). The random group is required to prevent the algorithm from getting stuck in local optima. The population size should accommodate these three groups appropriately. Furthermore, the size of the reproduction group and the random group need to be carefully considered. If the reproduction group is too large, the solution may get stuck in a local optimum. If the size of the random group is too large, we may spend most of the time calculating the fitness values of solutions that are very distant from the efficient frontier. However, the population size should at least be larger than the elite group. To find the proper population size, we evaluate the adopted RWGA with elitism for the population sizes $P = 150$, 200, and 300. We initially set the size of the reproduction group to one half of the population size, i.e., $P_p^{\text{init}} = P/2$. Once the number of Pareto-optimal solutions in a generation $t-1$ exceeds P_p^{init}, i.e., $P_e(t) > P_p^{\text{init}}$, we set the size of the reproduction group to $P_p(t) = P - P_e(t)$ in the next generation. Thus $P_p(t) = \min(P_p^{\text{init}}, P - P_e(t))$. If the number of Pareto-optimal solutions in a generation $t-1$ is less than $P - P_p^{\text{init}}$, we set the size of the random group to $P_r(t) = P - P_p^{\text{init}} - P_e(t)$ in the next generation, otherwise we set $P_r(t) = 0$; i.e., $P_r(t) = \max(0,\ P - P_p^{\text{init}} - P_e(t))$. Thus, the more Pareto-optimal solutions there are in the preceding generation, the fewer randomly generated individuals are in the next generation. (If the number of Pareto-optimal solutions in a generation exceeds P_p^{init}, the succeeding generation does not contain randomly generated individuals.) For the following evaluation, the parameters Λ, σ, q, and the ranges of D, F, M, and p are set as given in Section II-C. For comparison, we set the number of generations to $G = 20$, 15, and 10, respectively. Thus, the total number of considered individuals is $P \cdot G = 3000$ in all cases. The results are shown in Fig. 8. We observe from Fig. 8 that all three efficient frontiers hug the true Pareto-optimal frontier quite closely, with all three curves having "humps" around a throughput of 14 packets/frame. The number of Pareto-optimal solutions obtained for the population sizes $P = 150$, 200, and 300 are 87, 104 and 70, respectively. The population size of $P = 150$ does not perform very well in our network optimization because it typically cannot accommodate all the Pareto-optimal solutions. This is because the elite group takes up almost two thirds of the population. With a population size of $P = 300$ (and only $G = 10$ generations to ensure a fair comparison) the evolution of the generations does not settle down as much as for 20 and 15 generations and therefore gives only 70 Pareto-optimal solutions (although the efficient frontier has a relatively small "hump"). Overall, we conclude that all three considered population sizes give fairly good results. We choose $P = 200$ for the following experiments in this paper as it appears to accommodate all three population groups in a proper fashion. In Fig. 9 we plot the efficient frontiers obtained with different initial sizes $P_p^{\text{init}} = 50$ and 100 of the reproduction group (with $P = 200$, fixed). The number of Pareto-optimal solutions for $P_p^{\text{init}} = 50$ and 100, are 85 and 115, respectively. We observe from Fig. 9 that both efficient frontiers are quite close to the true Pareto-optimal frontier. We set $P_p^{\text{init}} = 100$ for all the following experiments in this paper.

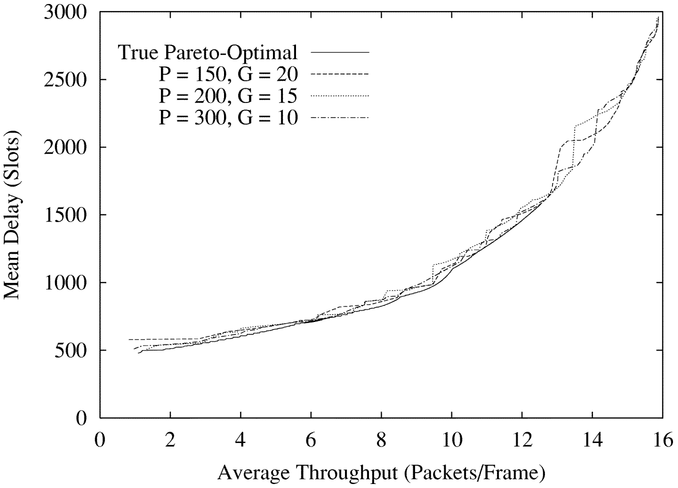


Fig. 8. Efficient frontiers for different population sizes P with $P \cdot G = 3000$, fixed.

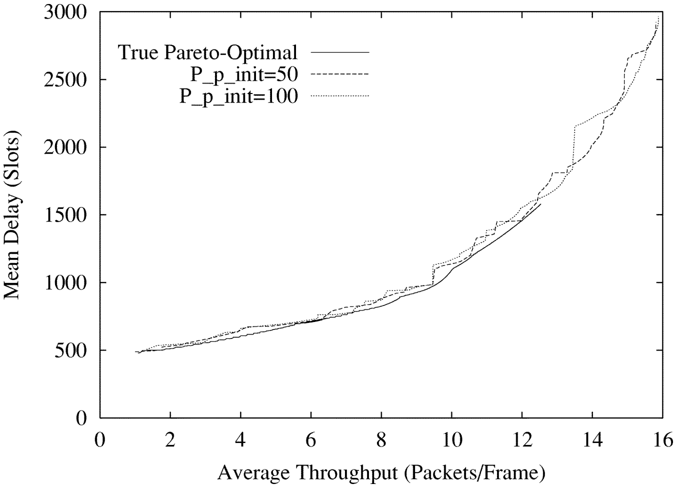


Fig. 9. Efficient frontiers for different initial sizes P_p^{init} of the reproduction group (Population size $P = 200$, fixed).

We now investigate the impact of the number of generations G. In Fig. 10, we plot the size of the elite group $P_e(t)$ as a

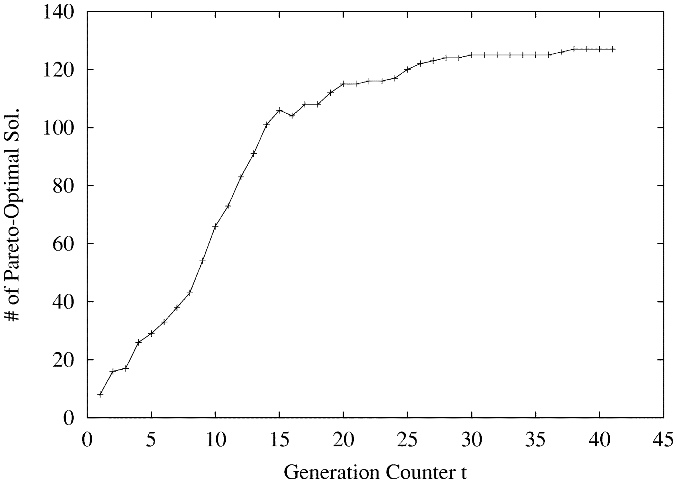


Fig. 10. Size of elite group $P_e(t)$ as a function of generation counter t.

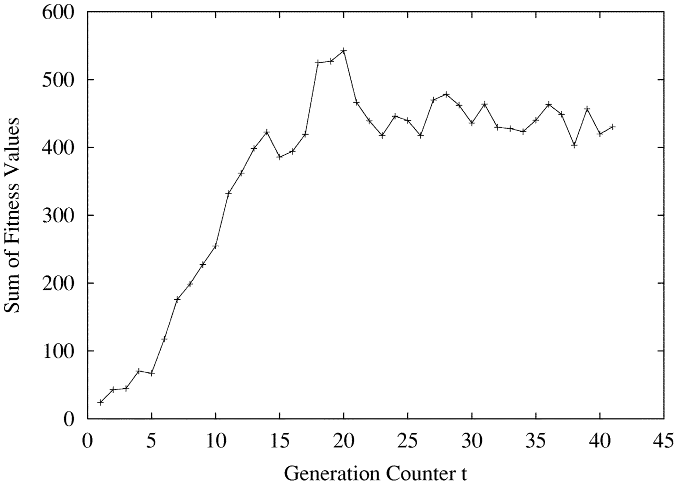


Fig. 11. Sum of fitness values of individuals in elite group as a function of the generation counter t.

function of the generation counter t. Recall that $P_e(t)$ is defined as the number of Pareto-optimal solutions in generation $t-1$; thus $P_e(1)$ is the number of Pareto-optimal solutions in the initial generation $t=0$. In Fig. 11, we plot the sum of the fitness values of the individuals in the elite group $P_e(t)$ as a function of the generation counter. We observe from Fig. 10 that the number of Pareto-optimal solutions in a generation first steadily increases and then settles on a fixed value as the generations evolve. (The slight drop around the fifteenth generation is because we found a Pareto-optimal solution which dominates several earlier Pareto-optimal solutions.) We observe from Fig. 11 that the sum of the fitness values of the Pareto-optimal solutions in a generation first increases quickly, then fluctuates, and finally settles down as the generations evolve. This behavior is typical for genetic algorithm based optimization and is due to the random nature of the evolution of the population. To allow for the evolution to settle down sufficiently, we set the total number of generations to $G = 40$. According to the decisions made in this section, we set the population size to $P = 200$, the number of generations to $G = 40$, and the initial size of the reproduction group to $P_p^{\text{init}} = 100$.

E. Crossover Operation

The crossover operation swaps parts of the chromosomes of the fittest individuals in the current generation to produce offspring with large fitness values for the reproduction group in the next generation. In our crossover operation the individuals in the generation $t-1$ are sorted in decreasing order of their fitness values (whereby the individuals from all three groups, i.e., elite group, reproduction group, and random group, are considered). A mating pool is formed from the first $P_p(t)$ individuals in the ordering. Parts of the chromosomes of the individuals in the mating pool are then exchanged (swapped) with a fixed crossover probability. We chose to swap their M values because we have observed that M (with D, F, and p fixed) tends to explore potential solutions in the vicinity of the parents (as is also evidenced by the tables in the Appendix, which are discussed in detail in Section IV). More specifically, the first $P_p(t)$ individuals in the ordering, i.e., the mating pool, are processed as follows. We take the first two individuals in the ordering. With the crossover probability (which we fix at the typical value 0.9), we swap their M values, i.e., we put the M value of the first individual (in the ordering) in place of the M value of the second individual, and vice versa. The other three decision values, D, F, and p, in the individuals' chromosomes remain unchanged. (Note that in our problem the swapping of M while keeping D, F, and p in place may result in a chromosome that violates the constraint $M \leq F$. If this situation arises, we discard the violating M value and randomly draw a new M from a uniform distribution over [1, F].) With the complementary crossover probability (0.1), the chromosomes of the two individuals remain unchanged. The two individuals (irrespective of whether their chromosomes were swapped or not) then become members of the reproduction group in the next generation. We then move on to the third and fourth individuals in the ordering, and swap their M values with probability 0.9, move them to the reproduction group in the next generation, and so on. We note that the elite group of the next generation is formed from the Pareto-optimal individuals in the current generation, irrespective of whether these individuals are in the mating pool of the current generation. (An individual may appear twice in the next generation if it is Pareto-optimal in the current generation and participates in the crossover operation without having the M value changed. Only one copy of such a "duplicate" individual is processed in the next generation, the other copy is discarded.)

F. Mutation Operation

The mutation operation keeps diversity in the population by changing small parts in the individuals' chromosomes with a given (typically small) mutation probability. We mutate each individual in the elite group, the reproduction group and the random group with a mutation probability of 0.05 (a typical value). The mutation is typically performed by flipping a bit in the binary representation of the individual's chromosome. The location of the bit is typically drawn from a uniform distribution over the length of the chromosome. We chose not to use bitwise mutation because bitwise mutation would frequently produce offspring that are distant from the parents. Instead, we implement the mutation operation by randomly drawing an M value

TABLE III
NUMBER OF PARETO-OPTIMAL SOLUTIONS WITH $D = 2, 4$, AND 8

VEGA	WBGA	RWGA	VEGA with Elitism	WBGA with Elitism	RWGA with Elitism
15	23	13	55	82	115

from a uniform distribution over $[1, F]$. This operation does not result in constraint violations, yet tends to keep the population sufficiently diverse.

After the mutation operation, we evaluate the average throughput and mean delay achieved by the individuals (in all three groups, i.e., elite group, reproduction group, and random group) in the new generation and start the next evolution cycle; as illustrated in Fig. 5. In this new evolution cycle, we select again the individuals with the largest fitness values for the crossover operation, which gives the reproduction group of the next generation. We also determine again the Pareto-optimal individuals to form the elite group in the next generation.

IV. NUMERICAL RESULTS

In this section, we employ the genetic algorithm based methodology developed in the preceding section to optimize the AWG-based single-hop WDM network. We determine the settings of the network architecture parameter D and the protocol parameters F, M, and p that give Pareto-optimal throughput-delay performance. We use the random weight genetic algorithm (RWGA) with elitism with the parameter settings found in the preceding Section, ie., a population size of $P = 200$, $G = 40$ generations, crossover probability 0.9, and mutation probability 0.05. Data packets can have one of two lengths. A data packet is F slots long with probability q, and $K = F - M$ slots long with probability $(1 - q)$. To reasonably limit the search space we restrict F to be no larger than 400 slots. The number of nodes in the network is set to $N = 200$ and the transceiver tuning range is fixed at $\Lambda = 8$ wavelengths.

In the first set of optimizations, we determine the Pareto-optimal performance for different (but fixed) combinations of traffic load σ and fraction of long packet traffic q. Specifically, we optimize the network for a light traffic scenario with $\sigma = 0.1$, a medium traffic scenario with $\sigma = 0.3$, and heavy load scenarios with $\sigma = 0.6$ and $\sigma = 0.8$. For each traffic load level, we consider the fractions $q = 0.1$, 0.5, and 0.9 of long packet traffic. In these optimizations we determine the free decision variables D, F, M, and p that give the Pareto-optimal solutions.

To put the optimizations for fixed σ and q in perspective, we also conduct an optimization where the traffic load σ and the fraction q of long packet traffic are free decision variables (in addition to D, F, M, and p). This optimization gives the best achievable network performance, which we refer to as *network frontier*. Loosely speaking, the network frontier gives the Pareto-optimal performance when the network is "fed optimally" with traffic. (To find the network frontier, we exchange (swap) σ as well as M in the crossover operation and use a population size of $P = 400$ rather than $P = 200$ to accommodate the larger chromosome.) Some detailed solutions for the network frontier are given in Table IV in the Appendix.

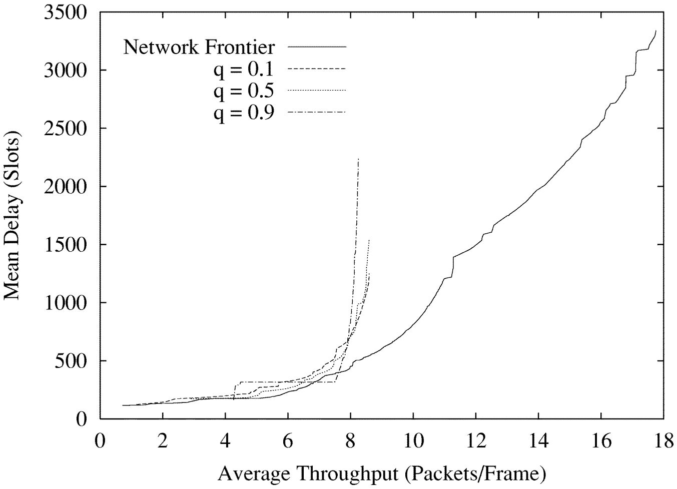


Fig. 12. Efficient frontiers for light traffic load $\sigma = 0.1$ for different fractions q of long packet traffic and network frontier (with σ and q as free decision variables).

Due to space constraints, we present throughout only a few representative individual solution in this paper. We refer the interested reader to [39] for the full table listings which have as many rows as there are Pareto-optimal solutions.

A. Pareto-Optimal Performance for Light Traffic Load

Fig. 12 shows the Pareto-optimal throughput-delay frontier for a light traffic load of $\sigma = 0.1$ for $q = 0.1$, 0.5, and 0.9 (along with the network frontier). Table V and Table VI in the Appendix give some representative individual Pareto-optimal solutions. The numbers of Pareto-optimal solutions with each $D = 2, 4$, and 8 are shown in Table III. We observe from Fig. 12 that for a small fraction q of long packets the network is able to achieve relatively small delays (of less than 1500 slots) even for large throughputs (of 8 packets/frame and more). When the fraction q of long packet traffic is large, however, the smallest achievable delays become very large (up to 2250 slots) for large throughputs. This is because the considered network allows for the scheduling of at most $R (= \Lambda/D)$ long packets in a cycle (consisting of D frames) at each of the D AWG input ports. (There are also $(D - 1) \cdot R$ transmission slots exclusively for short packets in a cycle at each AWG input port; in addition short packets can fill up the R long packet transmission slots.) With a larger fraction of long packets, the probability increases that a data packet fails in the scheduling and requires re-transmission of the corresponding control packet, resulting in larger delays.

We also observe that the light traffic scenario is able to achieve the small delay (and small throughput) part of the network frontier. This is because a small number M of control slots is sufficient to ensure reasonably large success probabilities in the control packets contention when the probability σ of an idle node generating a new packet at the beginning of a cycle is small. The small M in turn allows for small frame length F, and thus short cycle length $D \cdot F$, which results in small delays.

We observe that there are some instances where the Pareto-optimal frontier for $q = 0.9$ dominates the network frontier, e.g., around a throughput of 7.7 packets/frame. This is due to the stochastic nature of the genetic algorithm, which finds a very close approximation of the true optimal frontier in a computationally efficient manner. By definition, the true network frontier cannot be dominated by the true frontier for a fixed σ or q; finding these true frontiers, however, is computationally prohibitive.

We observe from Tables III, V, and VI that for the considered light traffic load $\sigma = 0.1$, most of the Pareto-optimal solutions have $D = 2$. However, for a larger fraction q of long packet traffic the number of Pareto-optimal solutions with $D = 4$ increases. We observe from the Table VI that $D = 4$ is the best choice to achieve low delay service. This is because the long packets are more difficult to schedule and therefore tend to require more re-transmissions of control packets, resulting in increased mean delay. Recall that a control packet is discarded if the corresponding data packet cannot be scheduled. This makes the control packet contention a bottleneck when the packet scheduling becomes difficult. With larger D, fewer nodes $S = N/D$ contend for the M control slots available to them every Dth frame. This increases the probability of successful control slot contention, thus relieving the control packet contention bottleneck. Note that the control packet contention bottleneck could also be relieved by reducing the re-transmission probability p. However, we see from the results in Table VI that this strategy is not selected (except in the 9th row of Table VI when the transition from $D = 4$ to $D = 2$ occurs). The reason for this is that the smaller p would result in a relative large increase in the mean delay, making it preferable to increase D and keep p large (the first eight rows of Table VI).

Generally, we observe from Table V and Table VI that the Pareto-optimal solutions with larger throughput are achieved for larger F. The Pareto-optimal M values, on the other hand, remain in the range 30–60 for $q = 0.1$ and $q = 0.5$ and are typically 30–80 for $q = 0.9$, even for very large F. Upon close inspection we discover an interesting underlying trend in the F and M solutions as we move along the efficient frontier from small to large throughput values. The frame length F typically makes a jump to a new value (e.g., from $F = 44$ to 59 in the fourth row of Table V) and stays around the new value for a few solutions. For F (almost) fixed, several distinct Pareto-optimal solutions are obtained for decreasing M values (from $M = 49$ to 30 for F around 59 in Table V). Once F makes a jump (to values around 100 in line 20), M is reset to a larger value (of 50 in line 20). The explanation of this behavior is as follows. For large M, the probability of successful control packet contention is large, and the probability of control packet re-transmission is small, giving small delays. However, for large M, the length $K = F - M$ of a short packet is small, resulting in a small contribution of a short packet to the throughput (1). Now as M decreases (for F fixed), control packet re-transmission becomes more likely, increasing the mean delay, while the contribution of a short packet to the throughput increases. We also observe from the tables that for optimal network operation the re-transmission probability p should be in the range from 0.75 to 1.0.

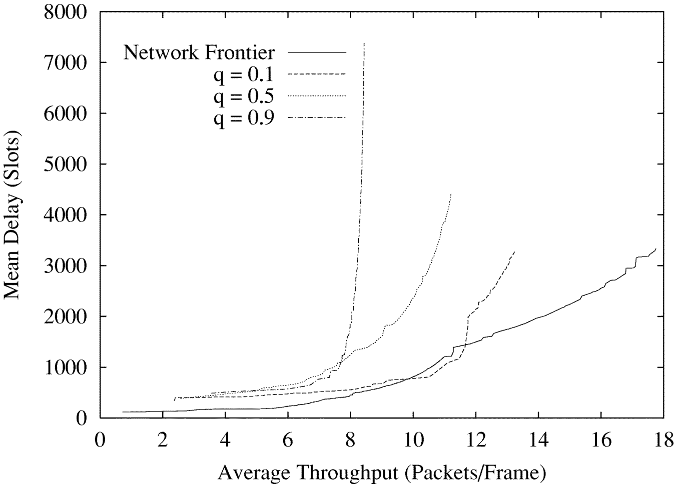


Fig. 13. Efficient frontiers for medium traffic load $\sigma = 0.3$ for different fractions q of long packet traffic and network frontier (with σ and q as free decision variables).

B. Pareto-Optimal Performance for Medium Traffic Load

Fig. 13 shows the Pareto-optimal solutions for a medium traffic load of $\sigma = 0.3$. The numbers of Pareto-optimal solutions with $D = 2$, 4, and 8 are shown in Table III and samples of the individual Pareto-optimal solutions are given in Tables VII–IX in the Appendix. We observe from Fig. 13 that the differences in performance for the different fractions q of long packet traffic are more pronounced for the larger traffic load $\sigma = 0.3$, compared to the light traffic load $\sigma = 0.1$ shown in Fig. 12. For $\sigma = 0.1$, the efficient frontiers for $q = 0.1$ and $q = 0.5$ roughly overlap and give both a smallest achievable delay of roughly 715 slots for a throughput of 8 packets/frame. For $\sigma = 0.3$, on the other hand, the efficient frontier for $q = 0.1$ clearly dominates, giving a smallest achievable delay of roughly 555 slots for a throughput of 8 packets/frame, whereas the corresponding smallest achievable delay for $q = 0.5$ is more than twice as large. This increasing gap in performance is again due to the fact that long packets are more difficult to schedule and thus tend to cause larger delays. The smaller delay of 555 slots for $\sigma = 0.3$, compared to 715 slots for $\sigma = 0.1$ is achievable because with the larger σ, the throughput level of eight transmitting nodes per slot is reached with smaller sized packets (i.e., smaller F and smaller $K = F - M$), thus reducing the cycle length and in turn the delay. We observe from Tables VII–IX that small delays are again achieved for large D values. For $q = 0.5$ and $q = 0.9$, the first few Pareto-optimal solutions at the top of the tables have $D = 8$, then $D = 4$ is optimal as we go down the tables to larger delays. As in the case of $\sigma = 0.1$, this behavior is due to the control packet contention and data packet scheduling bottlenecks. From Table III we observe that there is no clear trend in the number of solutions with $D = 2$ and $D = 4$. This appears to be due to the stochastic nature of the genetic algorithm approach, which finds a large total number of solutions for $q = 0.5$, with many solutions being tightly spaced in the region where $D = 4$ is optimal. As before, larger throughput is optimally achieved for large F. The optimal settings of M

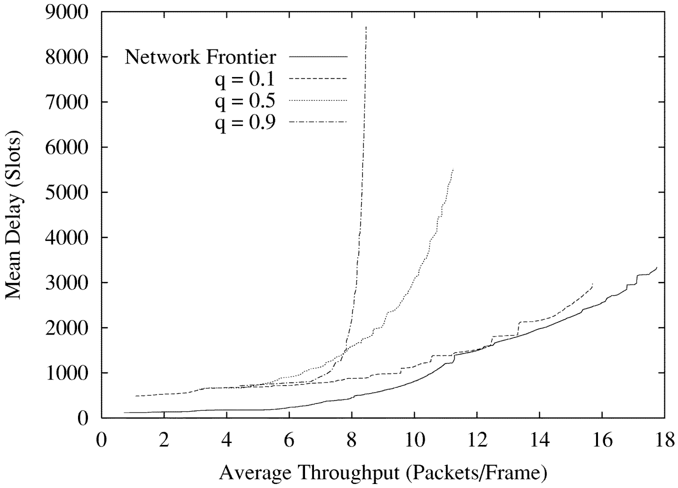


Fig. 14. Efficient frontiers for heavy traffic load $\sigma = 0.6$ for different fractions q of long packet traffic and network frontier (with σ and q as free decision variables).

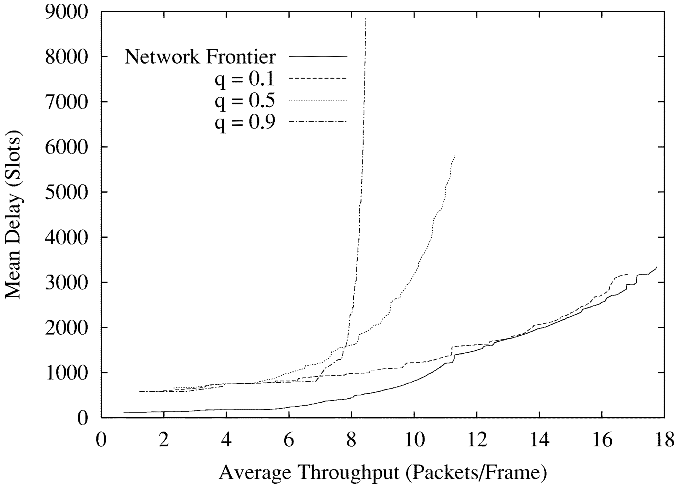


Fig. 15. Efficient frontiers for heavy traffic load $\sigma = 0.8$ for different fractions q of long packet traffic and network frontier (with σ and q as free decision variables).

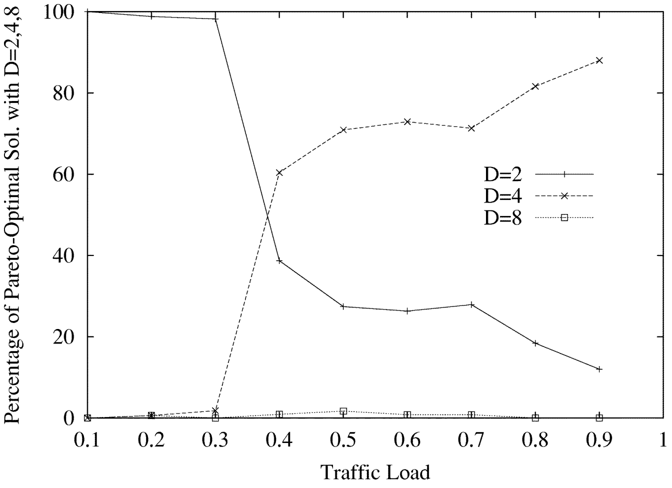


Fig. 16. Percentage of Pareto-optimal solutions with $D = 2$, 4, and 8 as a function of the traffic load σ (fraction of long packet traffic $q = 0.1$).

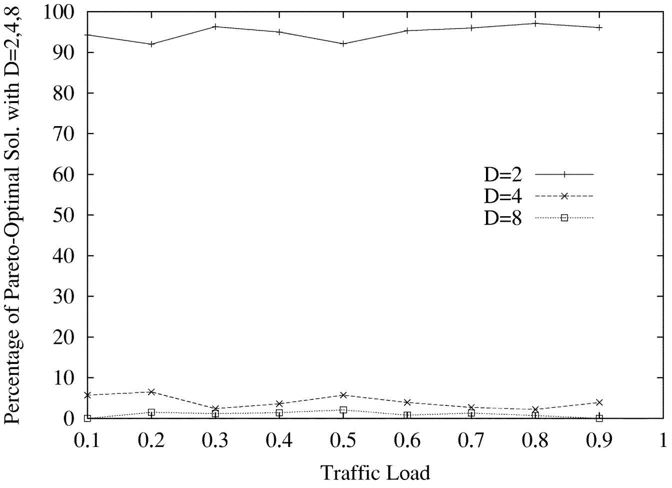


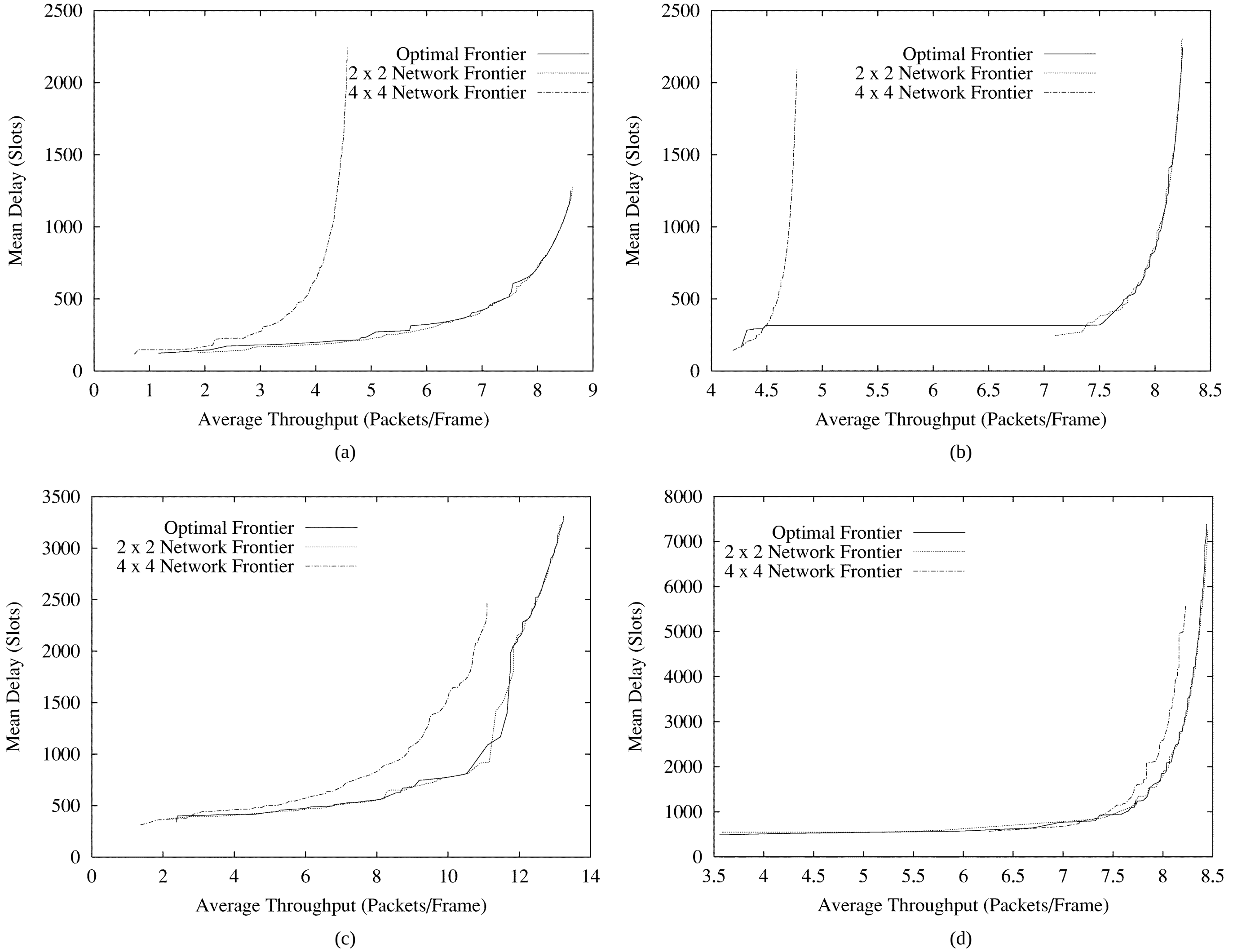
Fig. 17. Percentage of Pareto-optimal solutions with $D = 2$, 4, and 8 as a function of the traffic load σ (fraction of long packet traffic $q = 0.9$).

are typically in the range from 60–80. The optimal settings of p are mostly 0.95 for $q = 0.1$ and $q = 0.5$. For $q = 0.9$, the optimal p settings are typically 0.7. This smaller p setting for a medium load of predominantly long packet traffic is better as it somewhat abates the control packet contention bottleneck at the expenses of slightly larger delays, as discussed above.

C. Pareto-Optimal Performance for Heavy Traffic Load

Figs. 14 and 15 show the Pareto-optimal solutions for a heavy traffic load of $\sigma = 0.6$ and $\sigma = 0.8$, respectively. The number of Pareto-optimal solutions with $D = 2,\ 4$, and 8 are given in Table III. The complete parameter vectors corresponding to the Pareto-optimal solutions are given in Tables X–XII. We observe from the figures and the tables that both considered heavy load scenarios give similar results with the $\sigma = 0.8$, $q = 0.1$ scenario attaining the larger throughput region of the network frontier. We notice that with an increasing fraction q of long packet traffic, the number of Pareto-optimal solutions with $D = 2$ increases, while the number of solutions with $D = 4$ decreases. There are two primary effects at work here. On the one hand, a larger D allows for a larger throughput. To see this, note that the considered network allows for the scheduling of at most $R\ (= \Lambda/D)$ long packets at each of the D AWG input ports within one cycle (consisting of D frames); for a total of at most $D \cdot R = \Lambda$ scheduled long packets per cycle in the entire network. The network also allows for the scheduling of at most $(D-1) \cdot R$ short packets at each of the D AWG input ports within one cycle; for a total of at most $D \cdot (D-1) \cdot R = \Lambda \cdot (D-1)$ scheduled short packets per cycle in the network (in addition short packets may take up long packet transmission slots). Thus, for a larger D the network allows for the scheduling of more short packets and thus for an overall larger throughput; this is a result of the spatial reuse of all Λ wavelengths at all D AWG ports.

On the other hand, a larger D increases the delay in the network (provided the frame length F is constant). This is because a larger cycle length $D \cdot F$ increases the delay incurred by the



(a)

(b)

(c)

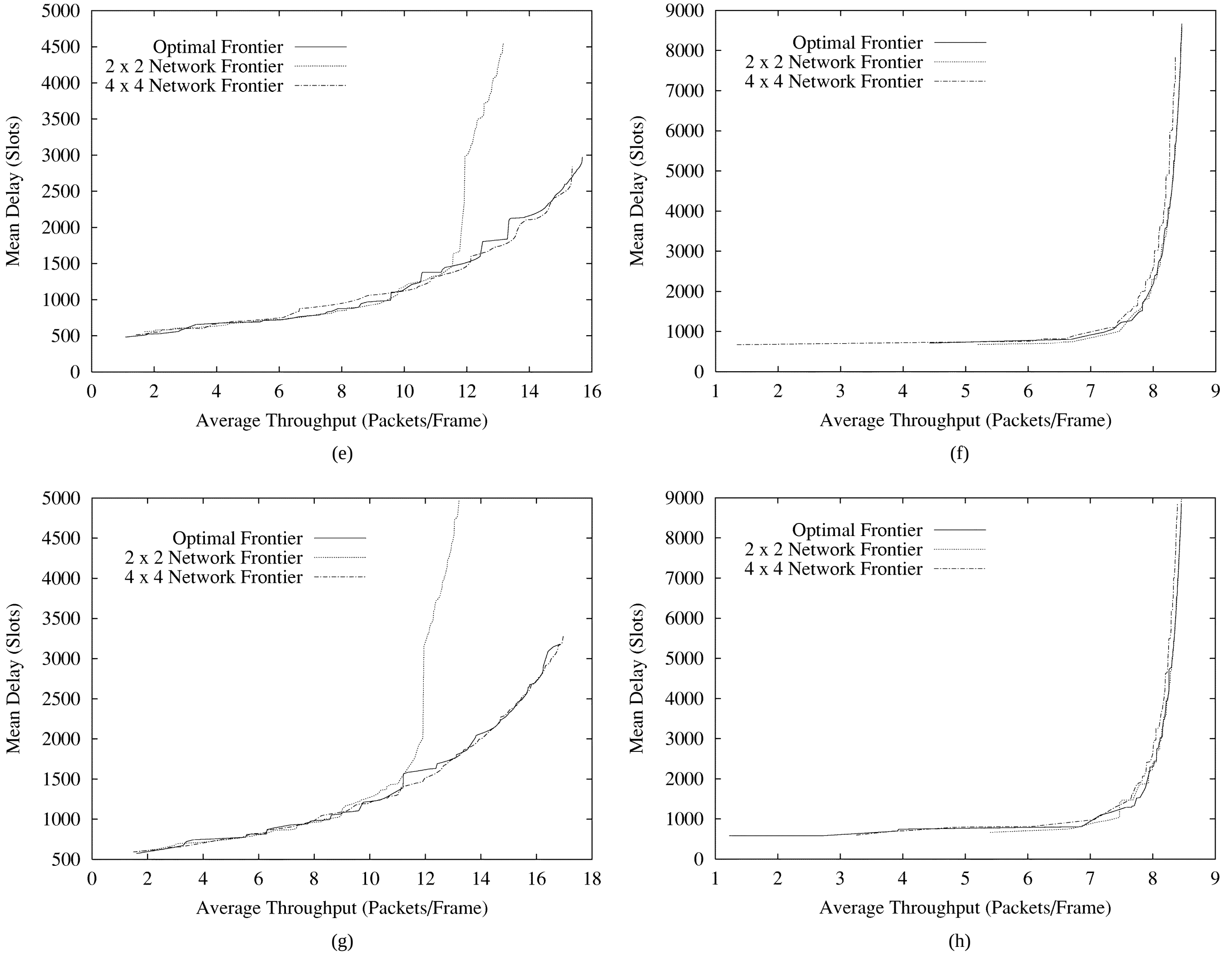
(d)

Fig. 18. Optimal frontier (with D a free decision variable), 2 × 2 network frontier (with $D = 2$, fixed), and 4 × 4 network frontier (with $D = 4$, fixed) for different (fixed) traffic loads σ and fractions q of long packet traffic. (a) $\sigma = 0.1, q = 0.1$. (b) $\sigma = 0.1, q = 0.9$. (c) $\sigma = 0.3, q = 0.1$. (d) $\sigma = 0.3, q = 0.9$.

control packet pre-transmission coordination and re-transmissions, which operate on a cycle basis. These throughput and delay effects combine to make $D = 2$ the better choice when long packets dominate (i.e., when q is large), since short packets make only a small contribution to the throughput. We also observe from Table X and Table XI that even when q is small, $D = 2$ is a good choice for delay sensitive traffic. Although we see that some Pareto-optimal solutions in the small delay range have $D = 4$. This indicates that both a 2×2 AWG and a 4×4 AWG based network can achieve small delays for traffic consisting mostly of short packets, provided the protocol parameters F, M, and p are set properly. On the other hand, only a 4×4 AWG based network achieves the large throughputs on the efficient frontier for small q (i.e., predominantly short packet traffic). As before, we observe that the Pareto-optimal solutions with larger throughput values have larger frame lengths F. Also, as before, the Pareto-optimal solutions have typically between $M = 60$ and 110 control slots per frame. We note, however, some differences in the optimal setting of the re-transmission probability p in this heavy traffic load scenario compared to the light/medium load scenario. As before for $q = 0.1$ the optimal p setting is typically in the range of 0.9–1.0. For $q = 0.5$ and $q = 0.9$, however, the optimal p is now typically in the range from 0.6 to 0.95.

D. Pareto-Optimal Planning of the Network Architecture

We now study the proper setting of the AWG degree D in detail. The setting of this network architecture (hardware) parameter has a profound impact on the network performance, as the results discussed so far illustrate. Importantly, once the network hardware for a particular D value has been installed, it is very difficult and costly to change D; whereas the protocol parameters F, M, and p can easily be changed by modifying the network protocol (software). For this reason, the proper setting of D warrants special attention. We have observed so far that for predominantly long packet traffic (i.e., large q), $D = 2$ is the best choice for all levels of traffic load σ. For predominantly short packet traffic (i.e., small q), on the other hand, the choice is not so clear. For light traffic loads, $D = 2$ is the best choice,



(e)

(f)

(g)

(h)

Fig. 18. (*Continued.*) Optimal frontier (with D a free decision variable), 2 × 2 network frontier (with $D = 2$, fixed), and 4 × 4 network frontier (with $D = 4$, fixed) for different (fixed) traffic loads σ and fractions q of long packet traffic. (e) $\sigma = 0.6$, $q = 0.1$. (f) $\sigma = 0.6$, $q = 0.9$. (g) $\sigma = 0.8$, $q = 0.1$. (h) $\sigma = 0.8$, $q = 0.9$.

whereas for heavy traffic loads, $D = 4$ turns out to be the best choice.

To explore the optimal setting of D as a function of the traffic load σ, we plot in Figs. 16 and 17 the percentage of Pareto-optimal solutions with $D = 2$, 4, and 8 for $q = 0.1$ and $q = 0.9$, respectively. We observe from Fig. 16 that for σ less than 0.4, most Pareto-optimal solutions have $D = 2$, whereas for σ larger than 0.4, most Pareto-optimal solutions have $D = 4$. The explanation of this behavior is as follows. For light traffic loads, $D = 2$ is preferred as it achieves smaller delays while at the same time providing sufficient resources for control packet contention and data packet scheduling. (Recall that $S = N/D$ nodes at an AWG input port content for the M control slots available to them in one frame (out of the D frames in a cycle), and that spatial wavelength reuse provides for $\Lambda \cdot (D - 1)$ transmission slots for short packets.) As the traffic load increases, however, the control packets contention and data packet scheduling become increasingly bottlenecks which are relieved for larger D.

E. Pareto-Optimal MAC Protocol Tuning (Network Operation) for Fixed Network Architecture

Next, we fix the AWG degree D at $D = 2$ and $D = 4$, and allow only the protocol parameters F, M, and p to vary (i.e., only F, M, and p are decision variables, D is fixed). We employ our genetic algorithm based methodology to obtain the Pareto-optimal throughput-delay frontiers in these settings; we refer to these efficient frontiers as the2 × 2 *network frontier* and the 4 × 4 *network frontier*, respectively. We compare the thus obtained efficient frontiers with the efficient frontier obtained when both the hardware parameter D and the software parameters F, M, and p are decision variables, which we refer to as *optimal frontier*. We compare the 2 × 2 frontier and the 4 × 4 frontier with the optimal frontier in Fig. 18(a)–(h). Samples of the corresponding Pareto-optimal solutions for the combination $\sigma = 0.6$, $q = 0.1$ are tabulated in Table XIII and Table XIV. We refer the interested reader to [39] for the other tables, which we cannot include here because of space constraints. A number of observations are in order. First, as expected the 2×2 frontier approximately coincides with the optimal frontier for light to medium loads of predominantly short packet traffic, and all load levels of predominantly long packet traffic. For heavy loads of predominantly short packet traffic, on the other hand, the 4×4 network frontier achieves the optimal frontier, as we expect from our earlier results. We also observe that there are some instances where the optimal frontier is dominated by the 2×2 network frontier or the 4×4 network frontier, e.g., in Fig. 18(c) around a throughput of 11.5 packets/frame. These instances are again due to the stochastic nature of the employed genetic algorithms. By definition, the 2×2 network frontier and the 4×4 network frontier cannot dominate the true optimal frontier, which however could only be found by a computationally prohibitive exhaustive search. The genetic algorithm methodology finds a very close approximation of the true optimal frontier in a computationally efficient manner.

Fig. 18(a)–(h) give also a number of surprising results, which we would not expect, based on our earlier observations. First, the 4×4 network is able to come close to the optimal frontier for medium and heavy loads of predominantly long packet traffic, which is a surprise given the results in Table III and Fig. 17. The 4×4 network achieves this by properly tuning its three protocol parameters, F, M, and p, as detailed in the corresponding tables in [39]. Overall, the 4×4 network shows some flexibility in achieving good performance close to the optimal frontier for medium to heavy loads of both short and long packet traffic by properly tuning the protocol parameters (in software). For light traffic loads, however, the 4×4 network is not able to come close to the optimal frontier. The 2×2 network, on the other hand, appears to be more flexible than the 4×4 network. By properly tuning its protocol parameters, the 2×2 network is able to come fairly close to the optimal frontier even for heavy loads of short packet traffic [see Fig. 18(e) and (g)]. Overall, our results indicate that the 2×2 network is the best choice for achieving efficient multi-service convergence in a metro WDM network. The 2×2 network frontier approximately coincides with the optimal frontier for all load levels of long packet traffic and for light to medium loads of short packet traffic. For heavy loads of short packet traffic, the 4×4 network attains the optimal frontier. But the 2×2 network is able to come fairly close to the optimal frontier, simply by adjusting its protocol parameters in software.

V. Conclusion

We have developed a genetic algorithm based methodology for the multi-objective optimization problem of maximizing throughput while minimizing delay in an AWG-based metro WDM network. Our methodology finds the Pareto-optimal throughput-delay tradeoff curve in a computationally efficient manner. The optimal tradeoff curve can be used to optimally provide varying degrees of small delay (and moderate throughput) or large throughput (and moderate delay) packet transport services. Our methodology thus facilitates efficient multi-service convergence for increased cost-effectiveness in metropolitan and local area networks.

Specifically, for the AWG based network considered as an example throughout this paper, we find that a network based on a 2×2 AWG is most flexible in efficiently providing different transport services under a wide range of traffic loads and packet size distributions. In addition, using an AWG with the minimum degree of $D = 2$ minimizes the network cost (see Section II-D) which is an important consideration in cost-sensitive metro WDM networks.

For a fixed network hardware the different transport services are achieved by optimally tuning the MAC protocol parameters (software) according to the found Pareto-optimal solutions. In particular, small frame lengths in the timing structure of the AWG network's MAC protocol give Pareto-optimal performance with small delay (and moderate throughput), while large frame lengths achieve optimal performance with large throughput (and moderate delays). The optimal number of control packet contention slots per frame is typically in the range from 30 to 80, the specific optimal values for a given traffic load and required throughput-delay performance are available in tables in [39]. The optimal control packet re-transmission probabilities are close to one for light traffic loads and in the range from 0.6–0.75 for heavy loads.

The developed genetic algorithm methodology can be applied in analogous fashion to networks with a similar throughput-delay tradeoff. The methodology is especially useful for the multi-objective optimization of networks with complex, highly nonlinear characterizations of the network throughput and delay.

Appendix
Tables for Pareto-Optimal Solutions

TABLE IV
Network Frontier: Pareto-Optimal Solutions With σ and q as Free Decision Variables

D	F	M	p	σ	q	TH	$Delay$
4	17	16	0.90	0.10	0.10	0.71	115.8
2	32	30	0.90	0.10	0.10	1.44	120.1
2	38	37	0.80	0.10	0.15	1.61	130.0
2	41	36	0.90	0.10	0.10	1.98	132.8
2	37	31	0.90	0.10	0.10	2.27	134.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	171	54	1.00	0.15	0.20	9.63	716.7
2	173	55	1.00	0.15	0.20	9.63	717.9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
4	400	106	0.70	0.95	0.05	17.72	3294.3
4	400	105	0.70	0.95	0.05	17.73	3303.3
4	400	104	0.70	0.95	0.05	17.74	3312.5
4	400	102	0.70	0.95	0.05	17.75	3331.5
4	400	101	0.70	0.95	0.05	17.75	3341.4

TABLE V
Pareto-Optimal Solutions for $\sigma = 0.1$ and $q = 0.1$

D	F	M	p	TH	Del
2	40	39	0.90	1.16	123.5
2	37	34	0.75	1.60	133.1
2	44	38	0.75	2.09	146.7
2	59	49	0.75	2.41	173.2
2	59	48	0.75	2.56	174.7
2	59	47	0.75	2.70	176.2
2	60	46	0.75	2.95	181.0
2	59	44	0.75	3.12	181.6
2	59	42	0.75	3.40	185.9
2	59	39	0.75	3.81	193.7
2	59	37	0.75	4.07	200.1
2	59	34	0.75	4.46	212.4
2	69	39	0.90	4.66	213.2
2	69	38	0.90	4.77	216.3
2	59	31	0.75	4.81	229.8
2	65	34	0.75	4.90	234.0
2	59	30	0.75	4.93	237.4
2	99	52	0.90	5.09	270.8
2	68	28	0.90	5.69	280.1
2	111	50	0.90	5.71	313.3
2	111	49	0.90	5.78	315.5
2	111	48	0.90	5.86	317.9
2	111	47	0.90	5.93	320.4
⋮	⋮	⋮	⋮	⋮	⋮
2	218	40	0.90	7.91	676.0
2	220	40	0.90	7.93	682.2
⋮	⋮	⋮	⋮	⋮	⋮
2	380	39	0.90	8.59	1194.2
2	379	38	0.90	8.59	1208.2
2	380	38	0.90	8.598	1211.4
2	380	36	0.90	8.59	1251.4

TABLE VI
Pareto-Optimal Solutions for $\sigma = 0.1$ and $q = 0.9$

D	F	M	p	TH	Del
4	24	21	1.00	4.27	162.3
4	49	48	0.80	4.32	282.3
4	49	46	0.80	4.34	284.1
4	49	44	0.80	4.35	286.4
4	40	22	0.80	4.38	286.8
4	49	40	0.80	4.38	291.5
4	51	35	1.00	4.46	292.5
4	55	35	1.00	4.49	315.4
2	46	40	0.75	7.35	315.5
2	50	43	1.00	7.50	318.8
⋮	⋮	⋮	⋮	⋮	⋮
2	171	64	1.00	8.06	1006.0
2	171	58	1.00	8.06	1020.1
⋮	⋮	⋮	⋮	⋮	⋮
2	361	82	1.00	8.23	2070.3
2	390	78	1.00	8.25	2246.3

TABLE VII
Pareto-Optimal Solutions for $\sigma = 0.3$ and $q = 0.1$

D	F	M	p	TH	Del
4	45	40	1.00	2.37	338.8
2	79	78	1.00	2.38	369.4
4	49	43	0.65	2.40	399.2
2	92	87	0.95	3.28	405.2
2	64	57	0.90	3.57	412.0
⋮	⋮	⋮	⋮	⋮	⋮
2	261	63	1.00	12.06	2168.7
⋮	⋮	⋮	⋮	⋮	⋮
2	400	64	1.00	13.24	3307.2

TABLE VIII
Pareto-Optimal Solutions for $\sigma = 0.3$ and $q = 0.5$

D	F	M	p	TH	Del
8	17	16	0.95	2.57	382.7
8	23	21	0.95	2.98	409.9
4	36	29	0.95	4.96	531.2
4	49	43	0.65	5.24	592.4
4	59	55	0.95	5.49	596.1
⋮	⋮	⋮	⋮	⋮	⋮
2	123	70	0.95	8.70	1449.4
⋮	⋮	⋮	⋮	⋮	⋮
2	397	89	0.95	11.21	4433.6

TABLE IX
Pareto-Optimal Solutions for $\sigma = 0.3$ and $q = 0.9$

D	F	M	p	TH	Del
8	15	14	0.65	3.55	486.2
8	21	20	0.70	4.18	517.2
4	27	25	0.70	5.96	569.5
4	36	34	0.70	6.69	638.5
4	43	40	0.65	6.90	729.3
4	45	39	0.70	6.96	761.5
2	40	39	0.65	7.32	799.2
2	46	40	0.75	7.37	925.1
⋮	⋮	⋮	⋮	⋮	⋮
2	176	63	0.65	8.25	3293.9
⋮	⋮	⋮	⋮	⋮	⋮
2	397	75	0.65	8.44	7381.4

TABLE X
PARETO-OPTIMAL SOLUTIONS FOR $\sigma = 0.6$ AND $q = 0.1$

D	F	M	p	TH	Del
8	24	23	0.95	1.08	482.2
4	45	44	0.80	1.71	510.5
4	50	49	0.80	1.80	522.3
4	44	41	0.80	2.19	530.2
4	51	46	0.80	2.76	558.5
4	49	43	0.65	2.89	582.7
4	84	77	0.95	3.34	655.7
2	106	99	0.80	4.14	676.1
4	84	70	0.95	4.62	685.2
2	94	80	0.80	5.39	691.4
2	82	66	0.55	5.50	712.5
2	94	76	0.80	6.05	720.7
2	106	86	0.80	6.45	745.4
⋮	⋮	⋮	⋮	⋮	⋮
4	234	78	0.95	13.30	1839.6
⋮	⋮	⋮	⋮	⋮	⋮
4	394	87	0.95	15.69	2972.7

TABLE XI
PARETO-OPTIMAL SOLUTIONS FOR $\sigma = 0.8$ AND $q = 0.1$

D	F	M	p	TH	Del
4	40	39	0.90	1.60	572.4
4	49	43	0.65	2.95	648.8
4	66	61	0.95	3.07	656.9
4	67	61	0.90	3.27	672.0
2	75	69	0.35	3.40	721.8
2	70	62	0.95	3.65	743.4
2	75	64	0.35	4.39	754.4
2	67	53	0.35	4.85	763.0
2	82	66	0.55	5.53	775.8
2	75	57	0.35	5.57	812.8
2	111	93	0.95	6.27	815.3
⋮	⋮	⋮	⋮	⋮	⋮
4	259	88	0.95	14.65	2195.9
⋮	⋮	⋮	⋮	⋮	⋮
4	390	99	0.95	16.85	3180.8

TABLE XII
PARETO-OPTIMAL SOLUTIONS FOR $\sigma = 0.8$ AND $q = 0.9$

D	F	M	p	TH	Del
8	4	4	0.10	1.22	580.7
4	9	9	0.10	2.74	582.7
2	15	12	0.10	3.91	697.8
2	16	12	0.10	3.93	744.3
2	31	28	0.25	6.86	806.6
4	45	40	0.90	7.17	1099.1
4	49	48	0.65	7.32	1159.4
2	55	53	0.90	7.56	1286.7
2	56	54	0.65	7.66	1292.9
⋮	⋮	⋮	⋮	⋮	⋮
2	147	79	0.90	8.16	3336.1
2	153	81	0.90	8.17	3469.3
2	153	78	0.90	8.18	3473.8
⋮	⋮	⋮	⋮	⋮	⋮
2	392	94	0.90	8.46	8851.2

TABLE XIII
PARETO-OPTIMAL SOLUTIONS WITH $D = 2$ FOR $\sigma = 0.6$ AND $q = 0.1$

F	M	p	TH	Del
43	42	0.30	1.69	556.4
44	42	0.40	1.98	566.4
46	44	0.30	2.02	573.2
48	46	0.30	2.05	577.7
45	42	0.30	2.24	582.3
54	51	0.55	2.47	584.4
68	66	0.90	2.51	594.2
44	39	0.30	2.68	606.6
71	66	0.80	3.28	612.8
85	78	0.90	3.96	633.6
96	88	0.95	4.31	649.8
⋮	⋮	⋮	⋮	⋮
267	61	0.70	12.14	3137.5
⋮	⋮	⋮	⋮	⋮
392	65	0.55	13.17	4542.7

TABLE XIV
PARETO-OPTIMAL SOLUTIONS WITH $D = 4$ FOR $\sigma = 0.6$ AND $q = 0.1$

F	M	p	TH	Del
34	33	0.85	1.42	514.3
36	34	0.60	1.74	526.5
46	44	0.60	1.92	545.3
48	46	0.60	1.94	551.6
45	42	0.60	2.14	551.8
37	33	0.60	2.23	556.5
54	51	0.60	2.23	580.5
38	32	0.60	2.67	588.9
54	49	0.60	2.68	595.2
48	41	0.60	3.05	599.4
49	40	0.90	3.56	600.2
⋮	⋮	⋮	⋮	⋮
179	91	1.00	10.94	1298.3
⋮	⋮	⋮	⋮	⋮
378	91	0.90	15.37	2844.0

ACKNOWLEDGMENT

The authors are grateful to B. Kim of Arizona State University for sharing his insights on genetic algorithms and assisting us in the implementation of our methodology in a C program.

REFERENCES

[1] B. Mukherjee, "WDM optical communication networks: Progress and challenges," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1810–1824, Oct. 2000.

[2] M. Maier, M. Reisslein, and A. Wolisz, "High performance switchless WDM network using multiple free spectral ranges of an arrayed-waveguide grating," in *Proc. SPIE Terabit Optical Networking: Architecture, Control, and Management Issues*, Boston, MA, Nov. 2000, pp. 101–112.

[3] F. Jia and B. Mukherjee, "MultiS-net: A high-capacity, packet-switched, multi-channel, single-hop architecture and protocol for a local lightwave network," *J. High Speed Networks*, vol. 5, pp. 221–241, 1999.

[4] K. Bengi, "Performance of single-hop WDM LAN's supporting real-time services," *Photonic Network Communications*, vol. 1, no. 4, pp. 287–301, Dec. 1999.

[5] A. Bianco, E. Leonardi, M. Mellia, and F. Neri, "Network controller design for sonata—A large-scale all-optical passive network," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 2017–2028, Oct. 2000.

[6] A. Okada *et al.*, "All-optical packet routing by an out-of-band optical label and wavelength conversion in a full-mesh network based on a cyclic-frequency AWG," in *OFC 2001 Tech. Dig.*, Anaheim, CA, Mar. 2001, Paper ThG5.

[7] J. Spaeth, "Dynamic wavelengths under decentralized management," in TransiNet Workshop, Berlin, Germany, Oct. 2001.

[8] K. Kumaran, M. Mandjes, D. Mitra, and I. Saniee, "Resource usage and charging in a multi-service multi-QoS packet network," in MIT Workshop on Internet Service Quality Economics, Dec. 1999.

[9] P. Streilein and J. John, "Enabling revenue-generating services—The evolution of next-generation networks," *Bell Labs Tech. J.*, vol. 6, no. 1, pp. 3–12, Jan. 2001.

[10] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *IEEE/ACM Trans. Networking*, vol. 8, pp. 684–696, Oct. 2000.

[11] A. Grosso, E. Leonardi, M. Mellia, and A. Nucci, "Logical topologies design over WDM wavelength routed networks robust to traffic uncertainties," *IEEE Commun. Lett.*, vol. 5, pp. 172–174, Apr. 2001.

[12] B. Mukherjee, D. Banerjee, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Trans. Networking*, vol. 5, pp. 684–696, Oct. 1996.

[13] Z. Zhang and A. S. Acampora, "A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use," *IEEE/ACM Trans. Networking*, vol. 3, pp. 281–288, June 1995.

[14] A. Fumagalli and L. Valcarenghi, "IP restoration vs. WDM protection: Is there an optimal choice?," *IEEE Network*, vol. 14, pp. 34–41, Nov./Dec. 2000.

[15] M. Sridharan, M. V. Salapaka, and A. K. Somani, "A practical approach to operating survivable WDM networks," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 34–46, Jan. 2002.

[16] G. Xiao and Y.-W. Leung, "Algorithms for allocating wavelength converters in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 7, pp. 545–557, Aug. 1999.

[17] L. Zhong and B. Ramamurthy, "Optimization of amplifier placements in switch-based optical network," in *Proc. IEEE ICC '01*, vol. 1, Helsinki, Finland, June 2001, pp. 224–228.

[18] C. L. Lu, R. T. Hofmeister, P. Poggiolini, and L. G. Kazovsky, "Power budget optimization of a WDM network with MSCM control using semiconductor optical amplifier," *IEEE Photon. Technol. Lett.*, vol. 9, pp. 115–117, Jan. 1997.

[19] D. Remondo, R. Srinivasan, V. F. Nicola, W. Van Etten, and H. E. P. Tattje, "Adaptive importance sampling for performance evaluation and parameter optimization of communication systems," *IEEE Trans. Commun.*, vol. 48, pp. 557–565, Apr. 2000.

[20] N. Geary, A. Antonopoulos, E. Drakopoulos, and J. O'Reilly, "Analysis of optimization issues in multi-period DWDM network planning," in *Proc. IEEE INFOCOM '01*, vol. 1, Anchorage, AK, Apr. 2001, pp. 152–158.

[21] A. L. Chiu and E. H. Modiano, "Traffic grooming algorithm for reducing electronic multiplexing costs in WDM ring networks," *J. Lightwave Technol.*, vol. 18, pp. 2–12, Jan. 2000.

[22] O. Gerstel, R. Ramaswami, and G. H. Sasaki, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Trans. Networking*, vol. 8, pp. 618–630, Oct. 2000.

[23] H. Ghafouri-Shiraz, G. Zhu, and Y. Fei, "Effective wavelength assignment algorithms for optimizing design costs in SONET/WDM rings," *J. Lightwave Technol.*, vol. 19, pp. 1427–1439, Oct.r 2001.

[24] A. S. T. Lee, D. K. Hunter, D. G. Smith, and D. Marcenac, "Heuristic for setting up a stack of WDM rings with wavlength reuse," *J. Lightwave Technol.*, vol. 18, pp. 521–529, Apr. 2000.

[25] B. Mukherjee, "WDM-based local lightwave networks part I: Single-hop systems," *IEEE Network Magazine*, vol. 6, no. 3, pp. 12–27, May 1992.

[26] ——, "WDM-based local lightwave networks part II: Multihop systems," *IEEE Network*, vol. 6, pp. 20–32, July 1992.

[27] A. Dasylva and R. Srikant, "Optimal WDM schedules for optical star networks," *IEEE/ACM Trans. Networking*, vol. 7, pp. 446–456, June 1999.

[28] G. N. Rouskas and V. Sivaraman, "Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies," *IEEE/ACM Trans. Networking*, vol. 5, pp. 359–370, June 1997.

[29] M. Maier, M. Scheutzow, M. Reisslein, and A. Wolisz, "Wavelength reuse for efficient transport of variable-size packets in a metro WDM network," in *Proc. IEEE INFOCOM '02*, New York, NY, June 2002, pp. 1432–1441.

[30] M. C. Chia, D. K. Hunter, I. Andonovic, P. Ball, I. Wright, S. P. Ferguson, K. M. Guild, and M. J. O'Mahony, "Packet loss and delay performance of feedback and feed-forward arrayed-waveguide gratings-based optical packet switches with WDM inputs-outputs," *J. Lightwave Technol.*, vol. 19, pp. 1241–1254, Sept. 2001.

[31] I. Baldine and G. N. Rouskas, "Traffic adaptive WDM networks: A study of reconfiguration issues," *J. Lightwave Technol.*, vol. 19, pp. 433–455, Apr. 2001.

[32] R. Inkret, B. Mikac, and I. Podnar, "A heuristic approach to wavelength assignment in all-optical network," in *Proc. MELECON '98*, vol. 2, 1998, pp. 759–763.

[33] M. Ali, B. Ramamurthy, and J. S. Deogun, "Routing algorithms for all-optical networks with power considerations: The unicast case," in *Proc. 8th IEEE ICCCN '99*, Boston-Natick, MA, Oct. 1999, pp. 237–241.

[34] S. T. Sheu, Y. R. Chuang, Y. J. Cheng, and H. W. Tseng, "A novel optical IP router architecture for WDM networks," in *Proc. 15th Int. Conf. Information Networking*, 2001, pp. 335–340.

[35] M. C. Sinclair, "Minimum cost topology optimization of the COST239 European optical network," in *Proc. Int. Conf. Artificial Neural Networks and Genetic Algorithm*, Apr. 1995, pp. 26–29.

[36] H. Tzeng, J. Chen, and N. Chen, "Traffic grooming in WDM networks using genetic algorithm," in *Proc. IEEE SMC '99*, vol. 1, 1999, pp. 1003–1006.

[37] J. Arabas and S. Kozdrowski, "Applying an evolutionary algorithm to telecommunication network design," *IEEE Trans. Evol. Comput.*, vol. 5, pp. 309–322, Aug. 2001.

[38] D. R. Thompson and G. L. Bilbro, "Comparison of a genetic algorithm with a simulated annealing algorithm for the design of an ATM network," *IEEE Commun. Lett.*, vol. 4, pp. 267–269, Aug. 2000.

[39] H.-S. Yang, M. Maier, M. Reisslein, and W. M. Carlyle, "A Genetic Algorithm Based Methodology for Optimizing Multi-Service Convergence in a Metro WDM Network (Extended Version)," Arizona State University, Telecommunications Research Center, http://www.eas.asu.edu/~re, 2002.

[40] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. New York: Wiley, 2001.

[41] E. Zitzler, "Evolutionary Algorithms for Multi-Objective Optimization: Methods and Application," Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich, 1999.

Hyo-Sik Yang, photograph and biography not available at the time of publication.

Martin Maier, photograph and biography not available at the time of publication.

Martin Reisslein (A'96–S'97–M'98), photograph and biography not available at the time of publication.

W. Matthew Carlyle, photograph and biography not available at the time of publication.