

Lexicographic Fairness in WDM Optical Cross-connects

Satya R. Mohanty and Laxmi N. Bhuyan

satya,bhuyan@cs.ucr.edu

Abstract— We consider fair allocation of sessions at the outputs of optical cross-connects employing wavelength division multiplexing (WDM). Each session consists of traffic on one or more wavelengths (channels). We identify lexicographic fairness as the most appropriate fairness criterion that is relevant to this setting. Achieving a fair lexicographic solution, commonly referred to as lexicographic optimality (LEX), is trivial and polynomial-time computable when any incoming wavelength can be converted to any outgoing wavelength (full conversion). This is not apparent in the practical and realistic case of limited conversion. We prove that LEX is also polynomial-time computable for the limited conversion case by reducing the problem to a min-cost max-flow optimization objective in network flows. We also motivate, formulate and solve a stronger variant of lexicographic optimality that we refer to as worst-case fair lexicographic optimal (W-LEX). Although our effective setting is an optimization problem in bipartite graphs (the request graph is bipartite), the network-flow based algorithms are applicable to unit capacity graphs in general. Further, we provide fast polynomial-time algorithms that furnish solutions for LEX and the W-LEX optimality problems for arbitrary bipartite graphs (i.e. arbitrary wavelength-conversion rules) and are computationally less expensive than network-flow methods. Finally we report simulation results to validate our findings.

I. INTRODUCTION

Wavelength-Division Multiplexing (WDM) technology exploits high optical fiber bandwidth($50Tb/sec$ compared to few Gb/sec 's in electronics) by multiplexing different channels from end-users onto non-overlapping wavelengths. A wavelength-routed all-optical WDM network consists of wavelength routers called optical cross-connects (OXC) that are connected with one another by high speed optical fiber links. In such networks users communicate via all-optical channels called light-paths. A good introduction to the area of optical networks and relevant issues can be found in [1], [2], [3]. An OXC serves to switch or guide light-paths. An incoming wavelength (channel) on a fiber may be translated to the same or different wavelength in the outgoing fiber. The OXC employs *wavelength conversion* when the incoming and outgoing wavelengths of the light-path are different. Conversion aids in maximizing the network throughput [4], [5]; however, current all-optical converters can convert an incoming wavelength to only a few outgoing wavelengths and are usually referred to as Limited range wavelength converters (LRWCs) ([6] contains a chapter on conversion).

Satya R. Mohanty and Laxmi N. Bhuyan are with the Department of Computer Science and Engineering, University of California at Riverside, Engineering BU2, Riverside, CA 92521. This work was supported in part by NSF grant CCR 0311437.

Usually, an individual channel is associated with an Internet Service Provider (ISP). Demand for bandwidth continues to grow and exciting developments in ultra-dense WDM technology point to a significant increase in the number of channels that can be supported in the fiber; hence, it is common to see ISPs subscribing more channels to satisfy their customers. A session can be thought of a set of channels that are owned by the same ISP. These channels together carry the aggregate traffic of that session. Commercial products like the Cisco ONS 15530, Intel IXF19302, Alcatel 1850 TSS and Ciena ONLINE MetroTM Multiservice DWDM transport platform etc. already offer this “service aggregation” feature.

Several such sessions may compete for channels at the output ports of OXCs. Previous research[7], [8] on Quality of Service (QoS) in OXCs identify the problem as one of maximizing the throughput. Each channel is associated with a weight and the objective is to choose non-conflicting requests such that the overall weight is maximized. This is a maximum weighted bipartite matching problem. Similarly, [9] solves the problem of maximizing throughput and minimizing delay in buffered OXCs by an efficient algorithm referred to as *Scan and Swap Algorithm*. The work in [10] considers OXCs employing LRWCs that are shared per node or per output-fiber. Analysis and simulation studies are done for the loss probability. In all these previous works there is the implicit assumption that a channel is synonymous with a session. On the other hand simultaneously ensuring fairness among sessions and achieving high throughput become more meaningful in this context; rather than, merely maximizing the throughput.

Existing research is inadequate since the algorithms are not session-aware. Intuitively, the carrier ISP (the ISP that has deployed the OXC) would like to (i) serve as many sessions as possible (ii) maximize the minimum allocations of channels to every session and distribute the excess fairly across the sessions and (iii) make sure that the throughput is not affected adversely (to maximize its efficiency and revenue). The first objective is exactly the fairness problem of *lexicographic optimality* on a discrete feasible set. We also show that, for the problem domain under consideration, lexicographic optimality necessarily leads to maximum throughput! We therefore aim at lexicographic optimality in this work.

Our only assumptions are all channels are of same weight and each session is only directed at a single output. To summarize, our work deals with lexicographic optimality (LEX) and a preferred variant, the worst-case fair lexicographic optimality (W-LEX), when sessions share output fibers of the OXCs. We

identify some key properties (inherent in the problem structure) that enable us to provide fast polynomial-time algorithms for determining optimal solutions in the presence of LRWCs with systematic or arbitrary wavelength conversion rules.

Our Contributions We prove that LEX and W-LEX problems for channel-aggregated sessions in OXCs can be computed in polynomial time by reduction from the min-cost flow problem. To the best of our knowledge, our results establish, for the first time, that the discrete-lexicographic allocation problem for unit capacity networks can be solved in polynomial time, even though the problem in general is known to be NP-hard[11]. We also provide simpler and faster algorithms than min-cost flow based approaches that solve the LEX and W-LEX problems in bipartite networks. Finally we illustrate simulation results to validate our proposed algorithms.

II. BACKGROUND, NOTATION AND PRELIMINARIES

Different notions of fairness exist in the literature such as proportional fairness [12], max-min fairness [13] and weighted max-min fairness, utility fairness [14], [15] and several others. Max-min fairness (based on the premise of maximizing the minimum allocations) simply states that increasing the allocation of any entity should not be at the expense of another that received a lesser allocation in the first place. Max-min fairness has conventionally been applied to many networking problems in routing, load balancing, wired and wireless electronic switch scheduling. In these problems it is used to allocate rate or bandwidth to competing entities, the resource in question (i.e. the rate or the bandwidth) is assumed to be a continuously divisible. The max-min fairness is based on the theory of bottlenecks although the existing theory can be generalized to apply indirectly to seemingly different scenarios [16].

Max-min fairness was recently introduced in the electronic switch context by Yim et. al. [17] and Hosaagrahara et. al. [18]. However, in the optical domain, this allocation has to be in discrete amounts, for instance, whether a channel is allocated to a session or not. Thus the problem reduces to one of discrete allocation. To the best of our knowledge there have only been a few attempts to quantify this *discrete max-min* problem. Sarkar and Tassiulas [11] used the discrete max-min problem to solve the multi-rate multicast problem. Independently Ros and Tsai [19] formulated the condition to solve the wavelength assignment problem for optical sub-networks. Although the max-min solution to bandwidth allocation in the discrete case may not exist, two other forms, lexicographic and (a weaker form) maximal fairness do exist. In particular, we discuss lexicographic allocation in this paper. A recent work [20] describes the lexicographic optimization in the context of telecommunication networks but does not consider a discrete set.

A. Formal discussion of various Fairness schemes

We now state formal definitions of relevant fairness notions (borrowed from [11]) in the literature. Under some allocation policy \mathcal{A} , an allocation vector $\vec{\mathcal{A}}$ is defined as the n -tuple $(|\mathcal{A}_1|, |\mathcal{A}_2|, \dots, |\mathcal{A}_n|)$ i.e. $\vec{\mathcal{A}} = (|\mathcal{A}_1|, |\mathcal{A}_2|, \dots, |\mathcal{A}_n|)$ where

the i^{th} entry represents the allocation to entity i . The number of entries in $\vec{\mathcal{A}}$ will be denoted by $|\vec{\mathcal{A}}|$.

Definition 2.1: (Maxmin Fairness) A feasible discrete allocation vector $\vec{\mathcal{A}}$ is maxmin fair if it satisfies the following property with respect to any other feasible discrete allocation vector $\vec{\mathcal{B}}$: if there exists i s.t. $|\mathcal{B}_i| > |\mathcal{A}_i|$ then there exists j s.t. $|\mathcal{A}_j| \leq |\mathcal{A}_i|$ and $|\mathcal{B}_j| \leq |\mathcal{A}_j|$

Definition 2.2: (Lexicographic Comparison) Given a discrete n -allocation vector \vec{V} , define its lexicographically ordered version \hat{V} as follows: $\forall j \in 1, \dots, n, \exists k$ s.t. $|\hat{V}_j| = |\vec{V}_k|$ and $\hat{V}_1 \leq \hat{V}_2 \leq \dots \hat{V}_n$. Thus \hat{V} is a permutation of \vec{V} in ascending order. A discrete allocation vector $\vec{\mathcal{A}}$ is lexicographically greater than another allocation vector $\vec{\mathcal{B}}$ if $|\vec{\mathcal{A}}| > |\vec{\mathcal{B}}|$ or there exist i s.t. $\hat{\mathcal{A}}_i > \hat{\mathcal{B}}_i$ and $\hat{\mathcal{A}}_j = \hat{\mathcal{B}}_j$ if $j < i$.

Definition 2.3: (Lexicographic Optimality) A feasible discrete allocation is lexicographically optimal, iff it is lexicographically greater than or equal to every other feasible allocation.

We illustrate the two fairness criteria with an example.

Example: Sessions from two input ports compete at the same output port of an OXC. f_1 and f_2 are two sessions at the first input port. f_1 consists of a single channel on wavelength λ_1 and f_2 consists of three channels $\lambda_2, \lambda_3, \lambda_4$. Similarly, f_3 and f_4 are two sessions at the second input port. f_3 consists of a single channel on wavelength λ_1 and f_4 consists of four channels on $\lambda_2, \lambda_3, \lambda_4, \lambda_5$ respectively. Suppose the output only provides 6 wavelengths $\lambda_1, \dots, \lambda_6$ and assume, for simplicity, that all wavelengths are fully convertible to one another.

Consider two different allocations at the output \mathcal{A}^i and \mathcal{A}^j with allocation vectors $\vec{\mathcal{A}}^i = (|\mathcal{A}_{f_1}^i|, |\mathcal{A}_{f_2}^i|, |\mathcal{A}_{f_3}^i|, |\mathcal{A}_{f_4}^i|) = (1, 2, 1, 2)$ and $\vec{\mathcal{A}}^j = (|\mathcal{A}_{f_1}^j|, |\mathcal{A}_{f_2}^j|, |\mathcal{A}_{f_3}^j|, |\mathcal{A}_{f_4}^j|) = (1, 1, 1, 3)$. Then \mathcal{A}^i is both a maxmin fair and lexicographically optimal allocation. On the other hand, if the output provides for 7 wavelengths a maxmin fair allocation does not exist. Instead the allocations that result in allocation vectors $(1, 2, 1, 3)$ or $(1, 3, 1, 2)$ can be considered to be lexicographically optimal.

When lexicographic optimality exists but its computation is NP-hard, a weaker notion of fairness, called maximal fairness is sought (fair allocation of discrete bandwidth layers in multicasting [11]). We do not consider this here.

B. OXC Model

The OXC consists of N_s input fibers and M_s output fibers. Each fiber can support Λ unique wavelengths (denoted by $\lambda_j | j = 1 \dots \Lambda$) simultaneously using WDM (assumption of same number of wavelengths in each fiber is not necessary). A channel is distinguished by the 3-tuple (i, j, q) and denotes the incoming request on wavelength λ_j at port i directed at output port q . The i^{th} session, f_i , is some collection of channels directed at the same output i.e. $f_i = \{(j, k, q)\} \exists j, k, q \in I$, and let \mathcal{F}_q be the set of all sessions directed at output q . Let $|f_i|$ and $|\mathcal{F}_q|$ have their usual meanings i.e. the number of elements in these sets. We do not limit that a session consist of connection requests from the same input fiber only.

In the rest of the paper, whenever we refer to input channels or sessions it is *always in relation to the outgoing fiber that they are directed at* (therefore, for instance, we can dispense with the subscript q in \mathcal{F}_q). Time is slotted and scheduling of requests takes place synchronously at these slotted time points. Mathematically, wavelength conversion is a correspondence where an incoming channel (λ_j) can be mapped to a subset of the outgoing wavelengths ($\lambda_{jl}, l = 1 \dots K$), where K (conversion degree) is the maximum number of wavelengths into which λ_j can be converted (including itself) by the LRWC. Common LRWCs include ordered interval wavelength converters or circular symmetrical wavelength converters in which wavelength conversion obeys certain well-defined rules. LRWCs are present at the inputs. The switch model is shown in Figure 1 along with these commonly used wavelength conversions.

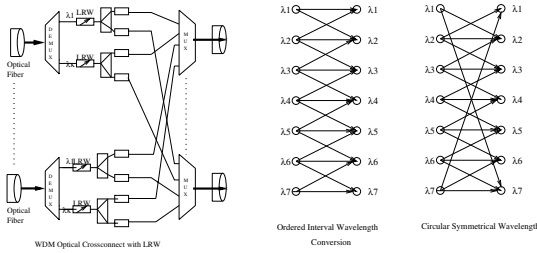


Fig. 1. DWDM Switch Schematic and common wavelength-conversion rules

III. FAIR-SHARE ALGORITHMS BASED ON MAXIMUM MATCHING OF REQUEST GRAPHS

The correspondence between incoming and outgoing wavelengths (due to presence of LRWCs) can be represented as a bipartite *request graph* (A, B, E) ; $A = (a_1, a_2, \dots, a_N)$, N is the number of input channels; $B = (b_1, b_2, \dots, b_m)$, $m \leq N$; $(a_i b_j) \in E$ if incoming wavelength, represented by a_i , can be converted to b_j representing an outgoing wavelength. f_a denote the session of the channel represented by vertex $a \in A$, R_{f_i} denote the number of “requests” (number of channels) of session f_i (i.e. $R_{f_i} = |f_i|$). $A_{f_i} \subseteq A$ denote the set of vertices (channels) of A that belong to session f_i . Vertex a_i and b_j are said to be matched (under some allocation policy) if the wavelength represented by a_i is converted to the wavelength represented by b_j . In this case edge $(a_i b_j)$ is also said to be matched and a_i is said to be allocated. $\mathcal{A}_{f_i}^j$ denotes the set of allocated vertices of session f_i under allocation policy \mathcal{A}^j and $|\mathcal{A}_{f_i}^j|$ the number of these vertices. A path p from vertex x to v in (A, B, E) is some collection of edges in E via which v is reachable from x .

LEX is trivial in the case of full conversion. Arrange all sessions in descending order of the number of channels. Initially all input and outgoing channels are unmatched. Algorithm 1 solves LEX.

A. Lexicographic Optimality under partial conversion

Solutions for LEX and W-LEX optimization problems are non-obvious when LRWCs are present and conversion rules are arbitrary.

Algorithm 1 Solving LEX under full conversion

repeat

 Match one unmatched input channel of each session to an unmatched output channel in round-robin order

until no unmatched channel at either inputs or output

Fact 3.1: An lexicographic optimal fair solution always exists.

Proof: For a given session set, the request graph is unique, and hence the total number of different ways in which channels can be assigned is fixed. The lexicographic fairness property always ensures that any two assignment instances compare affirmatively, either one is more fair than the other, or both are equally fair lexicographically. This implies a total order of assignments with respect to lexicographic optimality. Therefore there exists an assignment (not necessarily unique) that is lexicographic optimal. ■

We now state an important property that applies to any lexicographic optimal assignment.

Lemma 3.2: A Lexicographic optimal Assignment is a maximum matching but not vice-versa.

Proof: We only prove the forward implication since the converse is trivial. We will prove this by contradiction. Suppose that $\Psi_x = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ is a lexicographic optimal solution ($\mathcal{U} \subseteq A, \mathcal{V} \subseteq B, \mathcal{E} \subseteq E$, (A, B, E) is the request graph) but not a maximum matching instance. Then there exists an augmenting path p (alternating path of unmatched and matched edges, starting from an unmatched vertex in A and ending on an unmatched vertex in B). Let π be a permutation of vertices in $\mathcal{U} \cup \mathcal{V}$. s.t. $p = (u_{\pi(1)} u_{\pi(2)} \dots u_{\pi(|p|-1)}, u_{\pi(|p|)})$, where $u_{\pi(j)}$ is the j th vertex on path p .

When we invert the matchings (i.e. by making the matched edges of s unmatched and the unmatched edges matched, a process called augmentation), all internal vertices of $p \in \mathcal{U}$ $u_{\pi(3)}, u_{\pi(5)} \dots u_{\pi(|p|-1)}$ that were matched before still remain matched (although to different vertices). Allocation of none of the sessions reduced due to the augmentation; however, additionally; $u_{\pi(1)}$ that was previously unmatched becomes matched. This means that allocation of one session has increased by one while the rest of the sessions are unaffected. Thus the new allocation, call this Ψ' , is lexicographically more fair than Ψ_x violating the assumption of lexicographic optimality of Ψ_x ; a contradiction. ■

In the next section, we show that the desired lexicographic fairness objective can be cast as an optimization problem in flow networks[21].

IV. NETWORK FLOW APPROACH

Given a graph $G = (V, E)$ with nodes V and edges E , and special nodes source s and sink t , let fl_{uv} be the flow from node u to node v along edge $e_{u,v}$, and κ_{uv} the capacity (maximum flow possible) of e_{uv} . A network flow is a real function $fl : V \times V \rightarrow \mathcal{R}$ with the following three properties for all nodes u and v :

- 1) Skew symmetry: $fl_{uv} = -fl_{vu}$. The net flow from u to v must be the opposite of the net flow from v to u .

- 2) Capacity constraints: $fl_{uv} \leq \kappa_{uv}$. The flow along an edge cannot exceed its capacity.
- 3) Flow conservation: $\sum_{w \in V, w \neq s, t} fl_{uw} = 0$ The net flow to a node is zero, except for the source, which “produces” flow, and the sink, which “consumes” flow.

Notice that fl_{uv} is the net flow from u to v . If the graph represents a physical network, and if there is a real flow of, for example, 4 units from u to v , and a real flow of 3 units from v to u , we have $fl_{uv} = 1$ and $fl_{vu} = -1$. A flow that satisfies the above three properties is also called a “legal” flow. The *maximum flow problem* is finding a legal flow through a flow network of *maximum* value.

In a *min-cost max-flow problem*, each edge e_{uv} has a given unit cost c_{uv} . (A cost is a bijection $\Upsilon(e)_{e \in E} \rightarrow \{\mathcal{R}^+ \cup 0\}$), and the cost of sending the flow fl_{uv} across the edge is $fl_{uv} \cdot c_{uv}$. The objective is to send a given amount of flow from the source to the sink, at the lowest possible cost.

We define the min cost max flow objective formally. Associate with each vertex $k \in V$ a number $b(k) = \sum_i fl_{ki} - \sum_j fl_{jk}$. $b(k)$ denotes the demand or supply of vertex k depending on whether it is negative or positive. Let the maximum flow on this network be f^{max} . The minimum cost flow problem can now be stated.

$$\text{Minimize } z(x) = \sum_{e_{ij} \in E} c_{ij} fl_{ij} \quad (1)$$

subject to

$$\begin{aligned} b(i) &= \sum_{\{j: e_{ij} \in E\}} fl_{ij} - \sum_{\{j: e_{ji} \in E\}} fl_{ji} \\ b(i) &= 0, \forall i \in V - \{s, t\} \\ b(s) &= -b(t) = f^{max} \\ 0 &\leq fl_{ij} \leq \kappa_{ij}, \text{ for all } e_{ij} \in E \end{aligned}$$

A. Reduction of min cost flow problem to Lexicographic Optimality

We add additional vertices and edges to the request graph and assign costs and capacities to all the edges of the resulting graph in an appropriate way (described shortly) and transform it to a flow network. We prove that a min-cost flow solution on this transformed graph also solves LEX. The construction follows.

Construction: Initially we add two vertices, a source vertex s and a sink vertex t to the request graph (A, B, E) . For each session, f_i , in the system, we create a vertex v_i . For all $a_j \in A$ s.t. $f^{a_j} = f_i$, we connect v_i to a_j by unit capacity zero cost directed edge ($c_{v_i a_j} = 0, \kappa_{v_i a_j} = 1$). Similarly, we connect each vertex $b_j \in B$ to the sink t by such a directed edge ($c_{b_j t} = 0, \kappa_{b_j t} = 1$). We also connect s to each v_i by a zero cost directed edge of capacity N . In addition, we consider a set \mathcal{E} of N directed unit capacity edges with edge costs $1, N, \dots, N^{N-1}$. We further add to the graph the set \mathcal{E} , connected between each v_i and t i.e. $|\mathcal{F}|$ such \mathcal{E}' s are inserted. Note that the resulting graph is now a flow-network. Denote this new flow-network by $G = (N_g, E_g)$ that has edge costs \mathcal{C}_{ij} (as just described, $\mathcal{C} : \mathcal{R}^+ \cup 0$) and capacity

1 associated with every edge $e_{ij} \in E_g$ except the edges e_{sv_i} which have capacity N . Call the bipartite sub-network of (N_g, E_g) (that was the original bipartite graph (A, B, E)) (N'_g, E'_g) . Figure 2 illustrates the construction (each edge is labeled with its “capacity, cost”).

Now we push a flow equal to $|\mathcal{F}|N$ from the source s (recall that each edge e_{sv_i} has been assigned a flow that equals N). We explain the intuition behind assigning such capacities and costs. Each unit capacity edge $e_{v_i a_j}$ represents an incoming channel of f_i in the request graph and $e_{b_j t}$ represents a potential outgoing channel. The flow through the bipartite sub-network (N'_g, E'_g) cannot exceed the cardinality ($|\mathcal{M}|$) of a maximum matching in (A, B, E) (\mathcal{M} is one such maximum matching). So the rest of the flow, of size $|\mathcal{F}|N - |\mathcal{M}|$, has

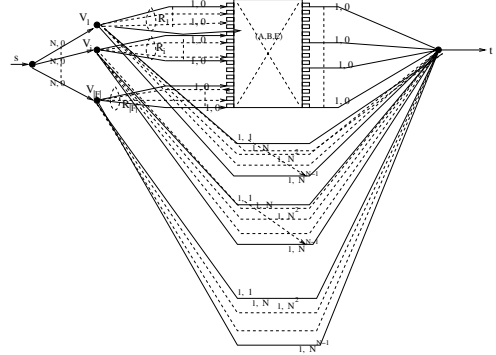


Fig. 2. Construction for proving Lexicographic Optimality

to flow through the \mathcal{E} edges. We can think of flow through (N'_g, E'_g) as the allocations. The cost structure on \mathcal{E} has been defined in such a way that the cost due to an additional rejected flow of a session is more than the total cost incurred due to all the previous rejected flows of that session (i.e. cost incurred on account of the k^{th} rejected flow (same for all sessions) is more than the total cost due to the previous $k - 1$ rejected flows). The idea is to decide which flows to reject and direct through the set of $|\mathcal{F}|$ \mathcal{E} s (equivalently which flows to send through (N'_g, E'_g)) so that the cost is minimum. Intuitively, simultaneously maximizing the number of flows from each session (i.e. lexicographic allocation) through (N'_g, E'_g) leads to minimum cost. We prove this formally.

Theorem 4.1: The solution to the min-cost flow problem with flow-size equal to $|\mathcal{F}|N$ in the flow-network $G = (N_g, E_g)$ (Figure 2) in equation 1 solves LEX.

Proof: Let \mathcal{A}^l be a lexicographic allocation and $\pi_l = (r_{\pi_l(1)}, r_{\pi_l(2)}, \dots, r_{\pi_l(|\mathcal{F}|)})$ be the lexicographic allocation vector, where $r_{\pi_l(i)}$ is the number of channels of the session with the i^{th} lowest allocation i.e. π_l denotes the n -tuple allocation vector in ascending order of allocation. \mathcal{A}^j is another allocation with corresponding allocation vector $\pi_j = (r_{\pi_j(1)}, r_{\pi_j(2)}, \dots, r_{\pi_j(|\mathcal{F}|)})$ (also in ascending order). If \mathcal{A}^l is fairer than \mathcal{A}^j then there exists some k , where $r_{\pi_l(k)} > r_{\pi_j(k)} \wedge \forall i < k, (if i exists) r_{\pi_l(i)} = r_{\pi_j(i)}$. Let $r_{\pi_l(k)} = m$. Let $\mathcal{C}(\mathcal{A})$ denote the total cost under allocation \mathcal{A} . In the flow network (N_g, E_g) , the total cost incurred by sessions whose

allocation is less than m is the same for both the allocations, call this \mathcal{C}_p . Then cost under \mathcal{A}^l is at most

$$\mathcal{C}(\mathcal{A}^l) \leq |\mathcal{F}| \sum_{i=0}^{N-m-1} N^i + \mathcal{C}_p$$

and cost under \mathcal{A}^j is at least

$$\begin{aligned} \mathcal{C}(\mathcal{A}^j) &\geq \sum_{i=0}^{N-m} N^i + \mathcal{C}_p \\ \mathcal{C}(\mathcal{A}^j) - \mathcal{C}(\mathcal{A}^l) &= \sum_{i=0}^{N-m} N^i - |\mathcal{F}| \sum_{i=0}^{N-m-1} N^i \end{aligned}$$

Since $|\mathcal{F}|$ is at most N , we have

$$\mathcal{C}(\mathcal{A}^j) - \mathcal{C}(\mathcal{A}^l) \geq \sum_{i=0}^{N-m} N^i - N \sum_{i=0}^{N-m-1} N^i$$

Hence $\mathcal{C}(\mathcal{A}^j) - \mathcal{C}(\mathcal{A}^l) \geq (1 + N + N^2 \dots N^{N-m}) - (N + N^2 + \dots N^{N-m}) = 1$. Thus $\mathcal{C}(\mathcal{A}^l) < \mathcal{C}(\mathcal{A}^j)$. This concludes the proof. \blacksquare

Note that the lexicographic optimal allocation need not be unique. A natural preference would be for a **LEX** solution in which sessions with higher demands receive higher allocations as far as feasible. Mathematically we formulate this objective as seeking a **LEX** solution (\mathcal{A}^{l_i}) in which the maximum discrepancy between some session's demand and the allocation that it receives as per that solution is still the minimum among all other alternative **LEX** solutions (\mathcal{A}^{l_k}) . Thus,

$$\begin{aligned} &\text{Determine } \mathcal{A}^{l_i} && (2) \\ &\text{subject to} \\ &\sup_j (R_j - |\mathcal{A}_j^{l_i}|) \leq \inf_k \{ \sup_j (R_j - |\mathcal{A}_j^{l_k}|) \} \end{aligned}$$

We refer to the above problem formulation as the **W-LEX** (*worst case fair Lexicographic optimal*) optimization problem. We illustrate that with an appropriate construction on the request graph and careful assignment of costs and capacities to edges, one can reduce the given problem to an equivalent problem of finding the min-cost flow solution in flow networks.

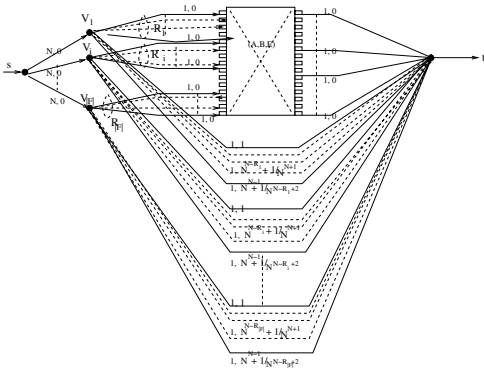


Fig. 3. Construction for proving Worst case fair Lexicographic Optimality

Construction: We perform the same construction as before (Figure 2). However the N edges of the set \mathcal{E} are assigned costs according to following policy

$$\begin{aligned} \mathcal{C}_{e_k} &= N^{k-1}, \quad k \leq N - R_i \\ \mathcal{C}_{e_k} &= N^{k-1} + \frac{1}{N^{N-(k-(N-R_i))+2}}, \quad N - R_i < k \leq N \end{aligned}$$

We refer to the fractional part of the cost (the $\frac{1}{N^{N-(k-(N-R_i))+2}}$'s) as auxiliary costs. The construction with the new costs is shown in Figure 3.

Theorem 4.2: The solution to the min-cost flow problem with flow-size equal to $|\mathcal{F}|N$ in the flow-network $G = (N_g, E_g)$ (Figure 3) in equation 1 also solves **W-LEX**.

Proof: First we provide some intuition regarding the new cost structure. Our goal is to not only achieve lexicographic optimality but also ensure an assignment in which the discrepancy between the request and allocation for any session is the minimum amongst other lexicographic allocations. The auxilliary costs have been chosen in a manner such that any allocation that solves **LEX** and has a maximum discrepancy of at least $d + 1$ results in a higher total auxilliary cost than another allocation that also solves **LEX** and has maximum discrepancy equal to d . Let Δ be the sum of all possible auxilliary costs, under some allocation policy. Then

$$\begin{aligned} \Delta &= \sum_{f_i} \sum_{k=1}^{R_i} \frac{1}{N^{N-k+2}} \\ &\leq |\mathcal{F}| \sum_{k=1}^N \frac{1}{N^{N-k+2}} \\ &\leq N \sum_{k=1}^N \frac{1}{N^{N-k+2}} \\ &< N \frac{1}{N^2} \left(\sum_{k=0}^{\infty} \frac{1}{N^k} \right) \\ &< \frac{1}{N} \left(\frac{1}{1 - \frac{1}{N}} \right) \\ &< 1 \end{aligned}$$

Since the allocation policy was quite general, it follows that the sum-total of all auxilliary costs is at most one irrespective of the allocation policy. First we prove that min-cost flow solution also solves **LEX**. As before, let \mathcal{A}^l be an allocation that solves **LEX** and $\pi_l = (r_{\pi_l(1)}, r_{\pi_l(2)}, \dots, r_{\pi_l(|\mathcal{F}|)})$ be the allocation vector (ascending order) and let \mathcal{A}^j be some other allocation with corresponding allocation vector $\pi_j = (r_{\pi_j(1)}, r_{\pi_j(2)}, \dots, r_{\pi_j(|\mathcal{F}|)})$. Let Δ_l and Δ_j be the auxilliary costs under \mathcal{A}^l and \mathcal{A}^j respectively. As in the proof of Lemma 4.1, If \mathcal{A}^l solves **LEX** and \mathcal{A}^j does not, then there exists some k , where $r_{\pi_l(k)} > r_{\pi_j(k)} \wedge \forall i < k$, (if i exists) $r_{\pi_l(i)} = r_{\pi_j(i)}$. Let $r_{\pi_l(k)} = m$. In the flow network of (N_g, E_g) (Figure 3), the total cost (excluding the auxilliary costs) incurred by sessions whose allocation is less than m is the same for both the allocations, call this \mathcal{C}_p . Then, the cost under \mathcal{A}^l is at most

$$\mathcal{C}(\mathcal{A}^l) \leq |\mathcal{F}| \sum_{i=0}^{N-m-1} N^i + \mathcal{C}_p + \Delta_l$$

and cost under \mathcal{A}^j is at least

$$\mathcal{C}(\mathcal{A}^j) \geq \sum_{i=0}^{N-m} N^i + \mathcal{C}_p + \Delta_j$$

$$\mathcal{C}(\mathcal{A}^j) - \mathcal{C}(\mathcal{A}^l) = \sum_{i=0}^{N-m} N^i - |\mathcal{F}| \sum_{i=0}^{N-m-1} N^i + \Delta_j - \Delta_l$$

Since $|\mathcal{F}|$ is at most N , we have

$$\mathcal{C}(\mathcal{A}^j) - \mathcal{C}(\mathcal{A}^l) \geq \sum_{i=0}^{N-m} N^i - N \sum_{i=0}^{N-m-1} N^i + \Delta_j - \Delta_l$$

Hence $\mathcal{C}(\mathcal{A}^j) - \mathcal{C}(\mathcal{A}^l) \geq (1 + N + N^2 \dots N^{N-m}) - (N + N^2 + \dots N^{N-m}) + \Delta_j - \Delta_l = 1 + \Delta_j - \Delta_l$. Since $0 \leq \Delta_j < 1$, $0 \leq \Delta_l < 1$, $\mathcal{C}(\mathcal{A}^j) > \mathcal{C}(\mathcal{A}^l)$. Hence we prove that any solution to the min-cost flow problem of Figure 3 has to be a lexicographic optimal solution (i.e. solves LEX).

We now prove that the min-cost flow solution not only solves LEX but also W-LEX. Accordingly, let \mathcal{A}^s be a solution for W-LEX and \mathcal{A}^l be another allocation that solves LEX but not W-LEX. Let f_i be the session with the maximum discrepancy between its demand and allocation under allocation policy \mathcal{A}^s , let this discrepancy be denoted as d_i . Similarly let f_j be the corresponding session under allocation policy \mathcal{A}^l , and the resulting discrepancy d_j i.e.

$$d_i = \sup_{f_k, k=1 \dots |\mathcal{F}|} (R_{f_k} - |\mathcal{A}_{f_k}|) \big|_{\mathcal{A}=\mathcal{A}^s}$$

$$d_j = \sup_{f_k, k=1 \dots |\mathcal{F}|} (R_{f_k} - |\mathcal{A}_{f_k}|) \big|_{\mathcal{A}=\mathcal{A}^l}$$

Therefore $d_j \geq d_i + 1$. It is evident that the total cost excluding the auxilliary costs incurred in the flow network due to allocation policies \mathcal{A}^s and \mathcal{A}^l is the same (since both are assumed to be lexicographic optimal). Let us now estimate the auxilliary costs.

$$\begin{aligned} \Delta_s &\leq |\mathcal{F}| \sum_{k=1}^{d_i} \frac{1}{N^{N-k+2}} \\ &\leq N \sum_{k=1}^{d_i} \frac{1}{N^{N-k+2}} \\ &\leq \sum_{k=1}^{d_i+1} \frac{1}{N^{N-k+2}} - \frac{1}{N^{N+1}} \\ &\leq \Delta_l - \frac{1}{N^{N+1}} \quad (\text{since } d_j \geq d_i + 1) \\ &< \Delta_l \end{aligned}$$

Hence \mathcal{A}^s incurs less cost than \mathcal{A}^l in the flow network. Since the discrepancy under \mathcal{A}^s was least compared to other alternative allocations that solved LEX, we see that the min-cost flow solution actually furnishes a solution to W-LEX (In fact, if all W-LEX solutions are arranged according to discrepancies of sessions in descending order, then the min-cost flow solution identifies the one with the lowest discrepancy where the discrepancies first differ). This concludes the proof. ■

The min-cost flow problem can be most efficiently solved by the cost-rescaling approaches as discussed by Goldberg Tarjan [22], Ahuja et. al. [23] and Orlin [24]. The respective time complexities are $O(nm \log(n^2/m) \log(nC))$, $O(nm(\log \log U) \log(nC))$ and $O((m \log n)(m + n \log n))$, where C is the largest cost coefficient, U is the largest of the edge capacities, m is the number of edges and n the number of vertices. Note that the min-cost flow reduction technique can work for any unit capacity network and not necessarily a bipartite graph (the reduction technique used was oblivious to the bipartite nature of (A_g, B_g)). Note also that the benefits of cost scaling notwithstanding, the problem may be computationally expensive owing to the potentially large calculation for C .

In the next section we present fast algorithms that obviate the flow-based approach and instead exploit the bipartite

structure of the request graph to solve the LEX and W-LEX problems.

V. FASTER ALGORITHM FOR BIPARTITE GRAPHS

A. Lexicographic Optimal Solution: The general case (LEX)

The concept of an *improving augmenting path* is key to determining a faster algorithm for the bipartite case.

Definition 5.1: We define an *improving augmenting path* $p = (v_s, v_i, \dots, v_t)$, $v_s, v_t \in A$, as a path in (A, B, E) that starts from some unmatched vertex v_s , consists of alternating unmatched and matched edges and ends in a matched vertex v_t such that $|\mathcal{A}_{f^{v_t}}| - |\mathcal{A}_{f^{v_s}}| \geq 2$.

f^v is said to be an *augmentable* session if v is the source of an *improving augmenting path*. We know that a LEX solution is a maximum matching (Lemma 3.2). Therefore, we perform a maximum matching first. Then, we determine an *augmentable* session and augment along the corresponding *improving augmenting path*. We repeat this process until no *augmentable* sessions remain. An illustration of this procedure is provided through the Example below.

Example: Figure 4(a) shows a request graph with three sessions F_1, F_2, F_3 . The wavelengths are allocated to channels according to the matching shown by bold lines. The initial allocations are $|\mathcal{A}_{F_1}| = 2$, $|\mathcal{A}_{F_2}| = 3$ and $|\mathcal{A}_{F_3}| = 4$. Even though this is a maximum matching, this is not a lexicographic optimal assignment. Note that there exists an *improving augmenting path* $A_3 B_3 A_4 B_4 A_5 B_5 A_6 B_6 A_7$ as shown in bold in Figure 4(b). Augmenting along that path results in Figure 4(c); the new allocations are $|\mathcal{A}_{F_1}| = 3$, $|\mathcal{A}_{F_2}| = 3$ and $|\mathcal{A}_{F_3}| = 3$, resulting in a lexicographic optimal assignment.

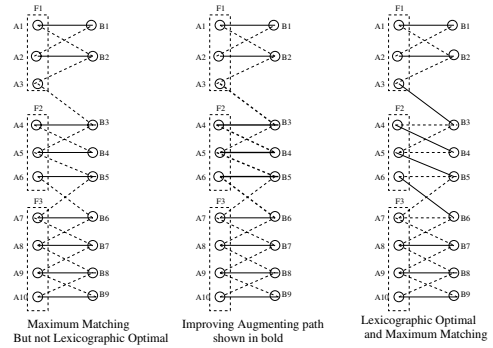


Fig. 4. Improving augmenting path

How many iterations are sufficient? In the simple example above only one iteration was sufficient to achieve lexicographic optimality. However, in general, several iterations (an $O(N^2)$ algorithm is used in the simulations) may be needed to find an unmatched vertex that is the source of an *improving augmenting path*. We would like an algorithm that solves LEX using as few iterations as possible. Such an algorithm is indeed possible. Before stating the algorithm itself, we reveal another interesting property inherent in the problem structure..

Lemma 5.2: Let f_i and f_j be a *non-augmentable* and *augmentable* sessions respectively s.t. $|\mathcal{A}_{f_i}| < |\mathcal{A}_{f_j}|$. Augmenting

f_j has no effect on further allocation to f_i ($|\mathcal{A}_{f_i}|$ remains invariant for all succeeding augmentations of other sessions).

Proof: Assume to the contrary that augmenting f_j along an *improving augmenting path* p results in making f_i augmentable. This implies that p shares at least one matched edge e with an alternating path p' that starts from some unmatched vertex of f_i and ends at e . Let $p \setminus e$ be the extension of p after e till its end-point v_t (i.e. v_t s.t. $|\mathcal{A}_{f_{v_t}}| > |\mathcal{A}_{f_j}| + 1$). Then $p' \oplus p \setminus e$, i.e. p' concatenated with $p \setminus e$ is also an *improving augmenting path* for f_i contradicting that f_i is *non-augmentable*. ■

Lemma 5.2 leads to the following corollary.

Corollary 5.3: At any iteration, if we always check for an *improving augmenting path* from an unmatched vertex of a session with the current lowest allocation, then this vertex need not be considered in future iterations.

Lemma 3.2, Definition 5.1, Lemmma 5.2 and Corollary 5.3 provide the genesis of an algorithm to solve LEX. Initially perform a maximum matching, arrange the unmatched vertices of each session into a linked list and the sessions in the form of a heap (top of the heap contains index of the session with the current lowest allocation). We search for an *improving augmenting path*, call it p , from the vertex extracted from the head of the session's linked list. If such a p exists we add its end-point, v_t , to the linked list of unmatched vertices of f^{v_t} and update the allocations. If no such p exists we extract the next vertex and perform a new search. Once extracted, vertices are never inserted again into their respective lists. If the linked list of a session becomes empty, this session is *non-augmentable* and therefore not to be considered further; otherwise the session is *potentially augmentable* and put into the heap. This process is repeated until all sessions are *non-augmentable*. The algorithm starts with the assumption that all sessions are *potentially augmentable*. The algorithm is formally defined in Algorithm 2. The actual description of the augmentation procedure is summarized in the *Augment()* subroutine (depth first search based), described formally as Algorithm 3.

B. Lexicographic Optimal Solution: The worst case fair case (W-LEX)

Similar to the previous section, we introduce the concept of a *W-Improving Augmenting path*.

Definition 5.4: We define an *W-improving augmenting path* $p = (v_s, v_i, \dots, v_t)$, $v_s, v_t \in A$ as a path in (A, B, E) that starts from an unmatched v_s , consists of alternating unmatched and matched edges and ends in a matched vertex v_t such that $|\mathcal{A}_{f_{v_t}}| - |\mathcal{A}_{f_{v_s}}| \geq 1 \wedge R_{f_{v_s}} > R_{f_{v_t}}$.

If we replace the "if" condition in the statement of line 9 Algorithm 3 by the following condition

if $|\mathcal{A}_{f_{matchR[v]}}| - |\mathcal{A}_{f_{v_s}}| \geq 1 \wedge R_{f_{v_s}} > R_{f_{matchR[v]}}$ **then**

and start with a maximum matching and then apply Algorithm 2 we would get a lexicographic solution that is worst-case fair (In each iteration we consider sessions with lowest allocations and highest discrepancy). Figure 5(a) illustrates

Algorithm 2 Algorithm to solve LEX

Require: Bipartite graph (A, B, E) , Session Set $\{f_j\}$

Ensure: Lexicographic Optimal matching

```

1:  $\mathcal{A}_{f_i} = 0$  : Set Allocations to 0,  $i = 1, \dots, |\mathcal{F}|$ 
2:  $\mathcal{M}$  : Maximum Matching s.t.  $(A_{\mathcal{M}}, B_{\mathcal{M}}, E_{\mathcal{M}})$  is request graph
3:  $matchL[u] = v, matchR[v] = u \quad \forall (uv) \in E_{\mathcal{M}}$ 
4:  $matchL[u] = matchR[v] = -1 \quad \forall$  unmatched  $u, v$ 
5:  $H$  : Heap of sessions as per updated allocations from matching
6:  $P_l^{f_i}$  : Linked list of unmatched vertices of  $f_i, i = 1, \dots, |\mathcal{F}|$ 
7: while  $H \neq \phi$  do
8:    $i \leftarrow \text{Extractmin}([H])$ 
9:    $v \leftarrow \text{Extract}(P_l^{f_i})$  /* Splice out  $v$  from  $f_i$ 's list */
10:  if  $\text{Augment}(v, v) == \text{true}$  then
11:     $|\mathcal{A}_{f^v}| = |\mathcal{A}_{f^v}| + 1$ 
12:     $v_t \rightarrow \text{Addto}(P_l^{f^{v_t}})$ 
13:  else
14:    if  $P_l^{f_i} \neq \text{NULL}$  then
15:      Go to 9
16:    end if
17:  end if
18:  if  $P_l^{f_i} \neq \text{NULL}$  /* potentially augmentable session */ then
19:    AddHeap( $H, i$ )
20:  end if
21: end while

```

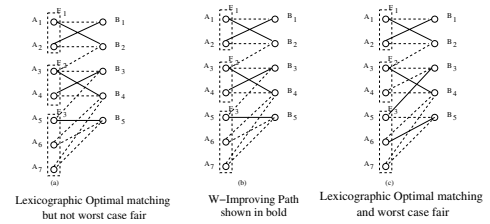


Fig. 5. W-Improving augmenting path

a case where the sessions have been allocated channels in a lexicographic optimal way. However the allocation is not worst-case fair since session F_3 has 3 requests but it has been allocated only one. However a *W-Improving augmenting path* ($A_6B_5A_5B_3A_4$) exists (Figure 5(b)) and the corresponding augmentation results in the worst-case fair lexicographic allocation (Figure 5(c)).

C. Computational Complexity

In Algorithm 2 we only check once whether a vertex is *augmentable* or not for a total of N checks. Determining whether an *improving augmenting path* exists is $O(|E|)$ (whether we do a *BFS* or *DFS* based search). For each vertex, the heap operations involve $O(\log |\mathcal{F}|)$ steps, and the linked list operations are $O(1)$ (we extract and add only at the beginning of the list). Therefore, for each vertex, the total complexity involved is $O(|E| + \log |\mathcal{F}|)$. Since there are at most N vertices the total complexity due to all the iterations is $O(N(|E| + \log |\mathcal{F}|))$. The computational complexity of maximum cardinality matching for general bipartite graphs is $O(\sqrt{NE})$ [25], for the ordered interval graphs (or doubly convex) it is $O(N)$ [7]. Setting up the linked list and the heap can be amortized in the maximum matching computation and therefore do not add to the complexity. $|\mathcal{F}|$ is at most N . Thus

Algorithm 3 Bool Augment(v_s, u)

Require: Bipartite graph (A, B, E) , Vertex $v_s, u \in A$
Ensure: Augment along improving augmenting path

```
1: seen $[v_i]_{i=1\dots N} = \text{false}$  /* whenever Alg. 2 calls Augment() */
2: Bool Augment( $v_s, u$ )
3: for all  $v \in B$  do
4:   if  $(u, v) \in E$  and  $v \neq \text{matchL}[u]$  then
5:     if seen $[v] == \text{true}$  then
6:       Continue
7:     end if
8:     seen $[v] = \text{true}$ 
9:     if  $|\mathcal{A}_{f\text{matchR}[v]}| - |\mathcal{A}_{fv_s}| \geq 2$  then
10:       $|\mathcal{A}_{f\text{matchR}[v]}| = |\mathcal{A}_{f\text{matchR}[v]}| - 1$ 
11:       $\text{matchL}[\text{matchR}[v]] = -1$ 
12:       $\text{matchR}[v] = u$ 
13:       $\text{matchL}[u] = v$ 
14:      return true
15:     else if Augment( $v_s, \text{matchR}[v]$ ) == true then
16:        $\text{matchR}[v] = u$ 
17:        $\text{matchL}[u] = v$ 
18:       return true
19:     end if
20:   end if
21: end for
22: return false
```

for a general bipartite request graph both LEX and W-LEX optimization problems can be computed in $O(N^{2.5} + N^2) \simeq O(N^{2.5})$ steps. However $|E|$ is at most KN for the realistic case of ordered limited conversion (Recall that K is the highest conversion degree of the OXC). K is small compared to N ; hence $|E| = O(N)$ and the complexity is $O(N^2)$ for this case.

VI. SIMULATIONS

Simulation Settings: We generate a variety of bipartite request graphs with arbitrary conversion. We report results of two configurations, the case of 1024 incoming wavelengths (not necessarily unique) and 1024 available outgoing wavelengths and the case of 2048 incoming wavelengths (again not necessarily unique) and 2048 outgoing wavelengths with arbitrary limited conversion degree. In the first configuration we assume

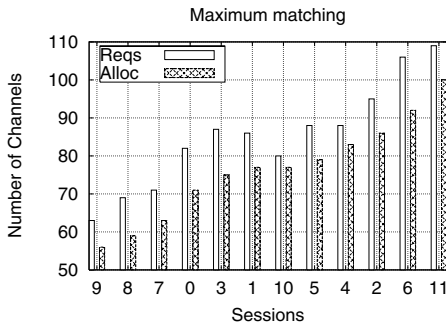


Fig. 6. A Maximum matching

that an incoming wavelength can be converted into a random number (at most 4) of wavelengths. There are 12 sessions and incoming channels are allocated to sessions randomly according to a uniform distribution between $[1, 12]$. For the

second configuration, again we have 12 sessions but the conversion degree is at most 6. Again the actual conversion degree is chosen from a uniform random distribution taken from $[1, 6]$ and a wavelength is allocated to a session according to the uniform distribution $[1, 12]$.

Simulation Results: Figure 6,7,8 show the various session requests and allocations alongside according to a maximum matching, a LEX solution and a W-LEX solution respectively for the 1024×1024 case. Results are shown in ascending order of allocation. There are 944 requests and the matching cardinality of each scheme is same (918 i.e. the maximum matching cardinality). Note that the lowest allocation as per

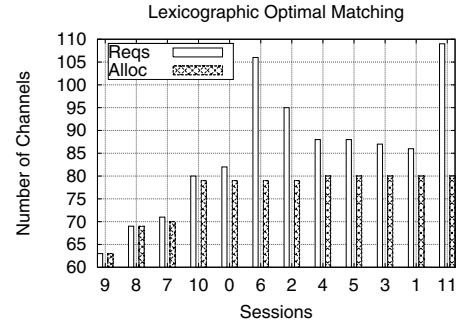


Fig. 7. Lexicographic Optimal matching

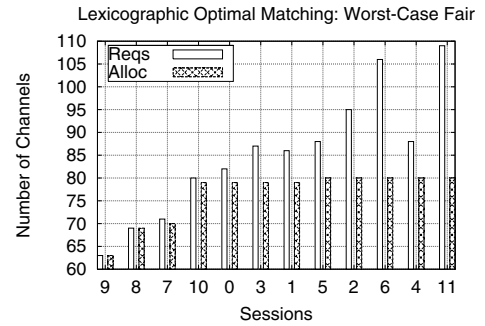


Fig. 8. Lexicographic Optimal matching (worst case fair)

the maximum matching instance is 56 (Figure 6). However in both the lexicographic optimal solutions the lowest allocation is 63. Figure 7 also shows that sessions 2 and 6 with 95 and 106 requests respectively are allocated 79 channels whereas sessions 1 and 3 with 86 and 87 requests are allocated 80 channels. This situation is reversed in Figure 8. Thus although both are solutions for LEX, Figure 8 is a W-LEX solution.

The results of the second configuration (1974 requests, 1953 total allocations) are shown in Figures 9, 10 and 11. Figure 9 shows the allocations according to just a maximum matching instance (143 (session 0) is the lowest allocation). Figure 10 shows a LEX solution (session 0 has the lowest allocation equal to 151). Note that sessions 7, 9 and 2 have lower number of requests than 1, 5 and 10 but are allotted higher number of channels. This is reversed in Figure 11 which shows a W-LEX solution.

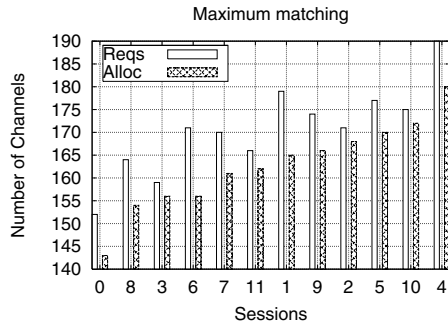


Fig. 9. A maximum matching

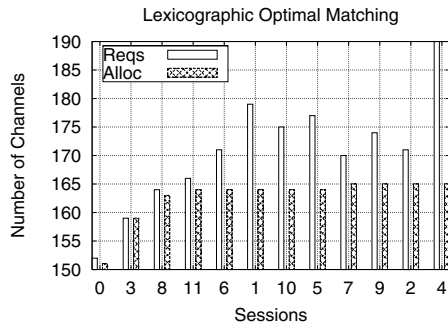


Fig. 10. Lexicographic Optimal matching

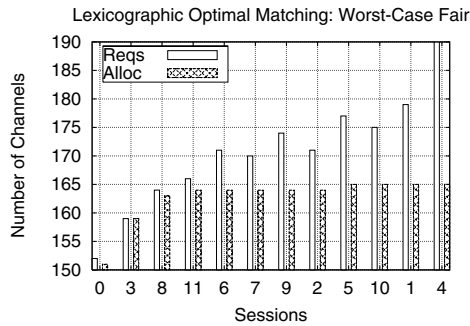


Fig. 11. Lexicographic Optimal matching (worst case fair)

VII. CONCLUSIONS AND FUTURE WORK

We solve the lexicographic optimality fairness problem in WDM systems. Unlike the problem of fair allocation of discrete rates for multicasting, lexicographic optimal allocation can be polynomially computed for session allotment in the case of WDM OXCs, (and more generally for unit capacity networks) by resolving a reduced min-cost flow problem in flow networks. We also provide fast algorithms (based on modified definitions of augmenting paths in bipartite graphs) to solve LEX and W-LEX optimization problems and validate results through simulation. An interesting area for future work can be to explore whether a 1-step approach instead of 2-step (maximum matching followed by augmentation methods) will suffice.

Acknowledgments

The authors wish to thank Neal E. Young for his help in the proof in Lemma 3.2 and Theorem 4.1.

REFERENCES

- [1] R. Ramaswami and K.N. Sivarajan. *Optical networks: A Practical Perspective, first ed.*, 2001.
- [2] George N. Rouskas and Harry G. Perros. A tutorial on optical networks. *Advanced lectures in Networking*, pages 155–193, 2002.
- [3] Biswanath Mukherjee. WDM optical communication networks: Progress and challenges. *IEEE Journal on Selected Areas in Communications*, 18(10), October 2000.
- [4] X. Qin and Y. Yang. Nonblocking WDM switching networks with full and limited wavelength conversion. *IEEE Transactions on Communication*, 50(12):2032–2041, December 2002.
- [5] Rajiv Ramaswami and Galen Sasaki. Multiwavelength optical networks with limited wavelength conversion. *IEEE/ACM Transactions on Networking*, 6(6):744–754, December 1998.
- [6] Byrav Ramamurthy. Design of optical WDM networks: LAN, MAN and WAN architectures. 2001.
- [7] Zhenghao Zhang and Yuanyuan Yang. Optimal scheduling algorithms in WDM optical interconnects with limited range wavelength conversion capability. *IEEE Transactions on Parallel and Distributed Systems*, 15(11):1012–1026, 2004.
- [8] Yuanyuan Yang and Jianchao Wang. Designing WDM optical interconnections with full connectivity by using limited wavelength conversion. *IEEE Transactions on Computers*, 53(12):1547–1556, 2004.
- [9] Zhenghao Zhang and Yuanyuan Yang. Scheduling in buffered WDM packet switching networks with arbitrary wavelength conversion capability. *IEEE Infocom*, pages 1372–1382, March 2004.
- [10] Vincenzo Eramo, Marco Listanti, and Matteo Spaziani. Resource sharing in optical packet switches with limited-range wavelength converters. *Journal of lightwave technology*, 23(2):671–687, February 2005.
- [11] Saswathi Sarkar and Leandros Tassiulas. Fair allocation of discrete bandwidth layers in multicast networks. *IEEE Infocom*, 2003.
- [12] F.P Kelly, A.K. Maulloo, and D.K.H. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* 49, pages 237–252, 1998.
- [13] Dimitri Bertsekas and Robert Gallager. *Data networks*. 1987.
- [14] L. Massoulié and J. Roberts. Bandwidth sharing: Objectives and algorithms. *IEEE/ACM Transactions on Networking*, 10(3):320–328, June 2002.
- [15] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, October 2000.
- [16] B. Radunovic and Le Boudec. A unified framework for max-min and min-max fairness with applications. *Proceedings of Allerton '02*, 2002.
- [17] R. Yim, N. Devroye, V. Tarokh, and H.T. Kung. Achieving fairness in two-dimensional generalized processor sharing. *Proceedings 22nd Biennial Symposium on Communications*, pages 185–187, April 2004.
- [18] Madhusudan Hosaagrahara and Harish Sethu. On achieving max-min fairness in input-queued switches. *Drexel University Technical Report DU-CS-05-11*.
- [19] Jordi Ros and Wei K Tsai. A general theory of constrained max-min rate allocation for multicast networks. *Technical Report*, 2002.
- [20] M Pioro, M Dzida, E Kubilinskas, and P Nilsson. Applications of the max-min fairness principle in telecommunication network design. *NGI*, 2005.
- [21] Ravindra K. Ahuja and Thomas L. Magnanti and James B. Orlin. *Network flows theory, algorithms and applications*. 1993.
- [22] A. V. Goldberg and R. E. Tarjan. Solving minimum cost flow problem by successive approximation. *Proceedings of the 19th ACM Symposium on the Theory of Computing*, pages 7–18, May 1987.
- [23] R. K. Ahuja, A. V. Goldberg, J. B. Orlin, and R. E. Tarjan. Finding minimum cost flows by double scaling. *Mathematical Programming*, 53:243–266, 1992.
- [24] J. B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Proceedings of the 20th ACM symposium on the Theory of Computing*, pages 377–387, May 1988.
- [25] J. E. Hopcroft, , and R. M. Karp. An $o(n^{2.5})$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.