Minimizing converters for broadcast traffic in WDM optical networks

R Dutta, P Iyer, C D Savage
Content

- Converters in optical WDM Networks
- The problem statement
- Complexity and algorithms
- Heuristics
- Performance of heuristics
- Future Work
Introduction

- Wavelength division multiplexing (WDM) - Multiple wavelengths used to carry traffic simultaneously over a single physical medium

- Routing traffic in the optical domain => better bandwidth utilization

- Use optical cross connects - OXCs

- Use "converters" to allow wavelength changes directly in the optical domain
Converter placement problem

Traffic requirement - all optic broadcast

- Primarily for control plane functionality
- the available wavelengths in each link is limited by the routing of traffic demands on the network

Goal

- Find a placement of converters on the network to enable all-optical broadcast
- Minimize number of converters
Converter placement
Example
Converter placement
Example
Broadcast Converter Node Selection (BCNS) Problem

Let $G(V,E)$ be an undirected graph and $C = \{1, 2, \ldots, W\}$ be the set of $W$ distinct wavelengths (colors) in the system.

A function $c : E \rightarrow 2^C$ assigns to each edge of $G$ a non-empty subset of $C$.

Converter cost $q$ (for the decision version)
Broadcast Converter Node Selection (BCNS) Problem

Solution

- A spanning tree $T = (V, E_T)$ of $G$
- A wavelength assignment $w(e) \in c(e)$ for each edge of $T$

$$\text{Min. cost } q = | \{ v \mid \exists e, e' \in e_T(v): w(e) \neq w(e') \}|$$

where $e_T(v)$ denotes the set of edges incident on $v$ in the spanning tree $T$. 
NP-Completeness

- Shown to be NP-Complete by Dutta and Savage

- Reduction from Maximum Leaf Spanning Tree (Max-Leaf):

  - Given an undirected graph $G(V,E)$ and integer $p$, is there a spanning tree of $G$ with at least $p$ leaves?
NP-Completeness details

Given Max-Leaf \([G'(V,E), p]\), we create an instance of BCNS\((G, W, c, q)\) by letting

- \(G = G'\),
- \(W = 1+\Delta\) (\(\Delta\) is the max. vertex degree),
- \(q = n-p\) and (\(n\) is the number of vertices),
- \(c(e)\) contains the single color assigned to edge \(e\) in a proper \(W\)-edge coloring of \(G\).
Recursive algorithm for BCNS in tree networks

1. Root the tree $T$ at a node $r$.

2. $\text{opt}(c,v) := \text{Minimum cost for subtree } T_v \text{ at } v$, assuming $c$ is fixed for the parent link at $v$.

3. $\text{opt}(0,v) := \text{Minimum cost for } T_v \text{ assuming no wavelength is fixed for the parent link at } v$.

4. $\text{opt}(0,r)$ gives the minimum converter cost for the entire tree $T$. 
Recursive algorithm for BCNS in tree networks

Base case:

- \( \text{opt}(c,v) = \infty : c > 0 \) and \( c \) is not available on the parent edge
- \( \text{opt}(c,v) = 0 : v \) is a leaf and either \( c=0 \) or \( c \) is available on the parent edge
Recursive algorithm for BCNS in tree networks

For non-leaf vertex v and every x in child(v)
let \( A = 1 + \sum x \text{opt}(0,x) \)
\( B = \sum x \text{opt}(c,x) \) \( c > 0 \)
\( D = \min_c( \sum x \text{opt}(c,x) ) \) c in cc(v)

\text{opt}(c,v) \) is then given by the table

<table>
<thead>
<tr>
<th>c&gt;0</th>
<th>c ∈ cc(v)</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>c ∈ cc(v)</td>
<td>Min(A, B)</td>
<td></td>
</tr>
<tr>
<td>cc(v) = Φ</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>cc(v) ≠ Φ</td>
<td>Min(A, D)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c=0</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc(v) = Φ</td>
<td>A</td>
</tr>
<tr>
<td>cc(v) ≠ Φ</td>
<td>Min(A, D)</td>
</tr>
</tbody>
</table>
Recursive algorithm for BCNS in tree networks

- Dynamic programming implementation
  - Determines $\text{opt}(0,r)$ in $O(nW^2)$
  - Can also determine a placement using $\text{opt}(0,r)$ converters
Heuristics for general networks

- Use the tree algorithm on “good” spanning trees
- How to get a good spanning tree?
  - Color richness
  - Weighted color-induced component size
  - Breadth first search
Heuristics for general networks

- **Breadth first search** - Attempts to increase the number of leaf nodes in the tree
- **Color richness** - Number of wavelengths on a link indicate flexibility in converter placement
- **Weighted color component score** - Links are scored by the sum of the sizes of the color-induced component they belong to
Performance of Heuristics

- Tests based on various classes of random graphs - dense, sparse, large, small
- Wavelength availability randomly selected within specified range
- Heuristics compared with
  - Random spanning tree solution
  - Exhaustive tree search (for small graphs)
Performance of Heuristics

- Tools used
  - Stanford Graph Base (SGB) by D.E. Knuth
  - Spanning tree enumeration
    - $O(1)$ per tree algorithm by Malcolm Smith
    - Implementation in SGB by D.E. Knuth
  - Our implementation of the BCNS algorithm for trees
Performance of Heuristics

N = 100, Average Degree = 3
W = 80-160, W_{av} = 8-16
Performance of Heuristics

$N = 100$, Average Degree = 8
$W = 80-160$, $W_{av} = 8-16$

- Random
- Color Rich
- Weighted Color Component
- Breadth First Tree
Performance of Heuristics

N = 100, Connectivity = 0.4
W = 80-160, W_{av} = 8-16
Performance of Heuristics

N = 50, Connectivity = 0.4
W = 32-80, W_{av} = 8-16

- Random
- Color Rich
- Weighted Color Component
- Breadth First Tree
Performance of Heuristics

\[ N = 25, \text{ Average Degree} = 3 \]
\[ W = 32-80, W_{av} = 8-16 \]
Future Work

- Better heuristics especially for sparse graphs
- Better heuristics for specific topologies or class of graphs
- Converter placement problem – Given a converter placement can we assign wavelengths to enable broadcast?
- Complexity of BCNS when $W \leq 4$ (Not covered by Max-Leaf)
Summary

- Minimize and place converters in WDM Networks for all-optical broadcast
- NP-Complete in general
- P-time algorithm for trees
- Heuristics and their performance
Performance of Heuristics

N = 100, Connectivity = 0.7
W = 80-160, W_{av} = 8-16
Performance of Heuristics

N = 50, Connectivity = 0.7
W = 32-80, W_{av} = 8-16