OPTIMAL CONVERTER-PLACEMENT IN TREE NETWORKS

Sukhamay Kundu and *Tong Yi*

Computer Science Department, Louisiana State University,
Baton Rouge, LA 70803, USA
(kundu, yitong@bit.csc.lsu.edu)

ABSTRACT.

We obtain a linear algorithm for an optimal converter-placement for broadcasting in an optical tree network. We use this to develop a heuristic algorithm for optimal converter-placement in a general network, for which the problem is known to be NP-complete.

Keywords: optical tree network, converter placement.

1. INTRODUCTION

The overall performance of an optical network depends on the complex interaction between the choice of communication paths (routing) and the allocation of wave-lengths to the links in each path using the available converters. An important design problem in optical networks is to minimize the number of converters because of their excessive costs while maintaining a high utilization of the communication links. The placement of a converter at a node v allows a communication path through v use a different wave-length on the incoming link at v than that on the outgoing link from v, which increases the link-utilization by allowing more point-to-point, multicasting, and broadcasting requests to succeed. One way to reduce the need for converters is to have more available wave-lengths for each link in the network so that different wave-lengths can be used to establish new connection paths between one or more node-pairs if we must share some of the links with other previously allocated paths for other node pairs. Another way is to have more links in the network to reduce the link-sharing. But each of these alternatives also add to the cost of the network. In general, one relies on extensive simulations using heuristics methods to determine ways for improving the link utilization and reducing the network cost [1, 2, 5, 7, 8, 9].

In order to get a better understanding of the wave-length allocation to the links to minimize the need for converters, we first consider the simple case of broadcasting in a tree network N. Since there is a unique path connecting any node-pair in a tree network, the choice of communication paths is no longer an issue. We give a linear algorithm for determining the optimal converter-placement P (i.e., $|P|$ is minimum). If we assume that each node in N has a converter, then this amounts to determining the minimum number of places at which the converter will be used for the broadcasting; the converters at other nodes might be then used for other communications. We use this algorithm to develop a heuristic for minimizing the number of converters needed for broadcasting in a general optical network. This later problem is related to the vertex-color cover problem considered in [6].

2. BASIC CONCEPTS

Let $N = (V, E)$ be an optical network, where the vertices $v_i \in V$ represent the set of routing nodes and the links $(v_i, v_j) \in E$ represent bi-directional optical fiber connections between v_i and v_j. For each link (v_i, v_j), we have a set of colors or wave-length channels $c(v_i, v_j)$ that are currently available for direct communication between v_i and v_j. For a subset $E' \subseteq E$, we write $c(E') = \bigcap\{c(v_i, v_j)\colon (v_i, v_j) \in E'\}$ for the set of common colors among the links E'. A *color-path* $\pi(v_i, v_j)$ in the network N is simply a path connecting the nodes v_i and v_j, with a unique color assignment $c_0(v_p, v_q) \in c(v_p, v_q)$ for each link in the path. The path is called *unicolor* if all links on the path are assigned the same color $c_0 = c_0(v_p, v_q)$. □

In the absence of converters, we need a unicolor path $\pi(v_i, v_j)$ for communication between v_i and v_j. However, in the presence of converters, we only need that each segment of $\pi(v_i, v_j)$ between two consecutive converters or between a converter and the end-points v_i or v_j be unicolor.

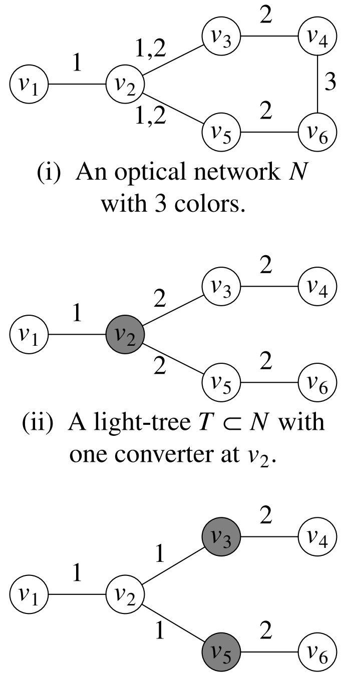
Def. 1. A spanning tree $T \subseteq N$ is called a *light-tree* [4] if each link $(v_i, v_j) \in T$ is assigned a unique color $c_0(v_i, v_j) \in c(v_i, v_j)$. □

To use a light-tree T for broadcasting, it is necessary and sufficient to place a converter at each node v_i for which the tree-links at v_i are not of the same color. The concept of light-tree was introduced in [4] for the more general purpose of multicasting, where the tree need not span all of N.

Converter Placement Problem (CPP): Find a minimum size subset of nodes $P \subseteq V$ such that there is a light-tree T in N which allows broadcasting in N using the converters at P.

There are two sources of difficulty in solving CPP: (1) choosing a spanning tree T of N from all possible spanning trees, which can be exponentially large in n = #(nodes in N), and (2) choosing a suitable color assignment for each link of T to make it a light-tree so that the number of converters needed is minimized. As we observed earlier, given a light-tree, the optimal placement of converters is immediate.

Example 1. Fig. 1(i) shows an optical network N with two of its links having $|c(v_i, v_j)| > 1$. Fig. 1(ii) shows a light-tree $T \subseteq N$ with only one node v_2 at which the tree-links are not of the same color. A converter at v_2 suffices for broadcasting using T. Note that if we use a simple greedy method of successively choosing a color and spreading it as much as possible along the links of N to new nodes that are not covered previously (and keeping the set of nodes covered connected), then it may not give a light-tree with minimum number of converters. This is illustrated in Fig. 1(iii), where we first spread the color 1 along the links $\{(v_1, v_2), (v_2, v_3), (v_2, v_5)\}$ and then spread the color 2 along $\{(v_3, v_4), (v_5, v_6)\}$. If we choose the successive colors on the basis on the number of new nodes that they help to connect, this suffices for the network in Fig. 1(i); we choose the color 2 first and then the color 1, giving the light-tree in Fig. 1(ii). The link (v_4, v_6) with color 3 is not chosen following color 2 because it does not connect new nodes. In general, this method may not give the optimal number of converters. This can be seen by adding two links (v_1, v_7) and (v_7, v_8), each with the color 1, to the network N; now, color 1 is chosen first and this leads to converters at v_3 and v_5. □



(i) An optical network N with 3 colors.

(ii) A light-tree $T \subset N$ with one converter at v_2.

(iii) The result of spreading color 1 first and then color 2; it needs 2 converters.

Figure 1. The notion of a (spanning) light-tree and its optimal placement of converters.

Def. 2. Given a tree T and an edge $(v_i, v_j) \in T$ at the node v_i, let $T_{ij} = \{(v_i, v_j)\} \cup \{$the connected component of $T - v_i$ containing the node $v_j\}$. The trees T_{ij} are disjoint except for the node v_i, their union is the whole tree T, and v_i is a terminal node in each T_{ij}. (If v_i is a terminal node in T, then $T_{ij} = T$.) □

If we know that the node v_i must have a converter (and hence v_i is not a terminal node), then consider the subtree T_{ij} for each link (v_i, v_j) at v_i and let P_j be an optimal converter-placement for T_{ij}. Then, an optimal converter-placement for T is given by $P = \{v_i\} \cup [\bigcup P_j]$, the union over vertices v_j adjacent to v_i.

2.1. Relationship to Vertex-Color Cover Problem

If the set of links of a given color c do not form a connected graph of the (general) network N, then it is easy to see that for the converter-placement problem we may replace the color c for the links within each of those connected graphs by a distinct color $c^{(i)}$. This is because no converter can help to form a bridge between the links of colors c that lie in those different connected parts [6]. For the remainder of this subsection, we assume therefore that for each color c, the links of that color form a connected graph. (For the algorithms in Sections 3 and 4, this assumption is not required.)

Let $c(v_i) = \bigcup\{c(v_i, v_j): v_j$ adjacent to $v_i\}$. A subset of vertices $V' \subseteq V$ is said to be a *vertex-color cover* [6] of the network N if the links containing one or more colors from $c_{V'} = \bigcup\{c(v_i): v_i \in V'\}$ form a connected graph on N and hence contains a light-tree. For the network in Fig. 1(i), $V' = \{v_2\}$ is a minimum size vertex-color-cover as are $\{v_3\}$ and $\{v_4\}$. It is not hard to see that the following inequalities hold for a minimum size vertex-color-cover V' and a minimum size converter-placement set P; the second property motivated the study of vertex-color-cover problem in [6].

$|P| \le 2|V'| - 1$, for tree networks, and
$|V'| \le |P|$, for general (including tree) networks.

The first inequality for tree networks comes from the fact that if we place a converter at each node in V', then we may need at most $(|V'| - 1)$ additional converters for broadcasting in T. To see this, consider a pair of nodes $v_i, v_j \in V'$ such that the unique path π_{ij} in T joining them does not contain any other node in V'; there are exactly $(|V'| - 1)$ such pairs. Let e_i be the link at v_i in the path π_{ij} and similarly let e_j be the link at v_j in π_{ij}. If one of the colors in $c(e_i)$ is common to all links in π_{ij}, then we can select that color for the links in π_{ij} for the broadcasting in T; a similar remark holds for $c(e_j)$. Otherwise, we need at least one converter in between v_i and v_j, and a converter at the node $v_{ij} \in \pi_{ij}$ beyond which none of the colors in $c(e_i)$, say, extends will suffice.

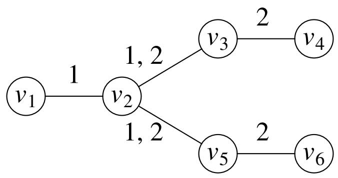
We remark that the second greedy heuristic algorithm in [6] for finding a connected vertex-color-cover V' may not even be minimal, in general; sometimes one can eliminate one or more nodes from the final V' and obtain a smaller connected vertex-color-cover.

3. CONVERTER-PLACEMENT FOR TREE-NETWORKS

In general, we may not find a node v_i such that $v_i \in P$ for every optimal converter-placement P. This can be seen by taking T to be a single path in which the two end links have the unique colors 1 and 2 respectively and all intermediate links have both the colors 1 and 2; here, a converter at any of the non-terminal nodes suffices.

Def. 3. A subtree of the form T_{ij} at v_i is called a *maximal common-color subtree* (*mcc*-subtree, in short) if all links in T_{ij} have one or more common colors (i.e., $c(T_{ij}) \neq \emptyset$) and T_{ij} is maximal in the sense that there is no other subtree T_{km} of this form whose links have a common color and $T_{ij} \subset T_{km}$. □

Note that if $c(T_{ij}) \neq \emptyset$, then T_{ij} is not an *mcc*-subtree if and only if there is an edge (v_i, v_p), $v_p \neq v_j$, at v_i such that $c(T_{pi}) \neq \emptyset$. Fig. 2 shows a tree network T with three *mcc*-subtrees and many more non-maximal common-color subtrees; here, each *mcc*-subtree has exactly one common color for its links. The links in $T_{52} = T - \{v_6\}$ does not have a common color for its links and hence T_{52} is not a common-color subtree. On the other hand, $T - \{v_1\}$ is not of the form T_{ij} although its links have a common color.



The maximal common-color subtrees:
(1) $T_{23} = \{v_2, v_3, v_4\}$ of color 2 at v_2
(2) $T_{25} = \{v_2, v_5, v_6\}$ of color 2 at v_2
(3) $T_{21} = \{v_1, v_2\}$ of color 1 at v_2

Figure 2. A tree network T with three *mcc*-subtree.

Let $d_T(v_i, v_j)$ denote the distance between v_i and v_j in T, i.e., the number of links on the unique path $\pi(v_i, v_j)$ in T. The following two theorems give, respectively, a necessary and a sufficient condition for a converter-placement P to be optimal in T. Intuitively speaking, the condition on P in Theorem 1 amounts to packing the converter-nodes collectively as close to each other as possible. Theorem 2 forms the basis of our algorithm OPTIMAL-PLACEMENT.

Theorem 1. Suppose P is an optimal converter-placement for broadcasting in a tree network T such that $|P| \geq 2$ and $\delta = \sum d_T(v_i, v_j)$, taken over all $v_i, v_j \in P$, is minimum. Let $T(P) \subset T$ be the subtree of T consisting on the nodes P and the paths $\pi(v_i, v_j)$ connecting those nodes. Then, for each $v_i \in T(P)$ and a link $(v_i, v_k) \notin T(P)$, the subtree $T_{ik} \subset T$ is an *mcc*-subtree. □

Theorem 1 is not true for $|P| = 1$. We illustrate Theorem 1 with an example; the proof of Theorem 1 (and of Theorem 2) can be found in [3].

Example 2. Fig. 3 shows a tree network where each of the elliptical regions (together with the node at the narrow tip of the region) gives an *mcc*-subtree. For example, there are three *mcc*-subtrees at the node $v_i = A$ corresponding to $v_k = A_1$, A_2, and A_3; the common color for the links in these subtrees are 1, 1, and 5, respectively. Here, $P = \{A, C, D, E, G\}$ is the only optimal placement of converters and $T(P) = \{(A, B), (B, C), (C, D), (C, F), (D, E), (F, G)\}$. For $v_i = F$, we have $v_k = F_1$ or F_2 and in each case T_{ik} is an *mcc*-subtree; both of these subtrees have color 3 as the common color for their links. Note that if color 1 was also one of the common colors for the links in the *mcc*-subtree at A for $v_k = A_3$, then we still need a converter at A; however, if we additionally had color $1 \in c(A, B)$, then we could move the converter from A to B, reducing the sum δ. In that case, the three previous *mcc*-subtrees at A together with the link (A, B) would form the new *mcc*-subtree for $(v_i, v_k) = (B, A)$. □

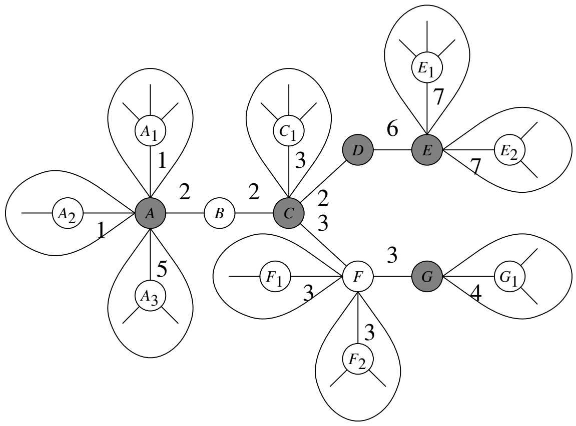


Figure 3. Illustration of Theorem 1; each elliptical shape region is an *mcc*-subtree with the indicated common color for its links.

We notice from Fig. 3 that if we consider T as a rooted tree, with the root node r selected arbitrarily, then the following properties hold for nodes $v_i \in P$, v_i is at the maximum distance from r. In particular, the descendent of v_i do not belong to P; the condition (c1) holds for the descendents of v_i but not the condition (c2).

(c1) For each child v_j of v_i, the links in the tree T_{ij} has a common color, i.e., $c(T_{ij}) \neq \emptyset$.

(c2) $c(v_p, v_i) \cap [\bigcap\{c(T_{ij})\text{: } v_j \text{ a child of } v_i\}] = \emptyset$, where $v_p = \text{parent}(v_i)$.

If the optimal P is of size one and $P = \{r\}$, $r = v_i$, then we need to modify (c2) as follows.

(c2′) $\bigcap\{c(T_{ij})\text{: } v_j \text{ a child of } v_i\} = \emptyset$.

Theorem 2. Consider T as a rooted tree, with the root r selected arbitrarily. If a node $v_i \neq r$ satisfies the properties (c1)-(c2), then there is an optimal converter-placement P for broadcasting in T such that $v_i \in P$. Likewise, if $v_i = r$ satisfies (c1) and (c2′), then $P = \{r\}$ is an optimal converter-placement. □

Theorem 2 suggests the following straightforward algorithm for determining an optimal converter-placement for broadcasting in a tree. Once a root r has been chosen and v_j a child of v_i, let $c'(v_i, v_j) \subseteq c(v_i, v_j)$ denote the current set of colors for the link (v_i, v_j) that are relevant following the determination of converters at nodes belonging to T_i.

Algorithm OPTIMAL-PLACEMENT:

Input: An optical tree network T, with $c(v_i, v_j)$ = the colors for the link (v_j, v_j).

Output: An optimal converter-placement P for broadcasting in T.

1. Choose a terminal node as the root of T (this is done just to avoid the special processing needed when the root has more than one child).
2. Let L = the maximum distance of a terminal node in the rooted tree T and let $P = \emptyset$.
3. For (each level $i = L$ to level $i = 1$ (level(root) being 0) and for (each node x at level i) do the following:
   (a) Let $y = \text{parent}(x)$.
   (b) If (x is a terminal node), then let $c'(y, x) = c(y, x)$.
   (c) Otherwise, do the following:
      (c.1) Let $I = \bigcap\{c'(x, z)\text{: } z \text{ is a child of } x\}$.
      (c.2) [Determine the need for a converter at x.] If ($I = \emptyset$ or $I \cap c(y, x) = \emptyset$), then add x to P and let $c'(y, x) = c(y, x)$.
      (c.3) Otherwise, let $c'(y, x) = I \cap c(y, x)$.

Example 3. To simplify this discussion, assume that each A_j, C_j, E_j, F_j, and G_j in Fig. 3 is a terminal node. Suppose we apply the algorithm to this tree network, with A_1 being the root. Then, the converter-nodes in P are selected in the order: G and E (at level $i = 5$), D (at level $i = 4$), C (at level $i = 3$), and A (at level $i = 1$). On the other hand, if we choose C_1 as the root, then the converter-nodes in P will be selected in the order: A, E, and G (at level 3), D (at level 2), and C (at level 1). □

3.1. Complexity of OPTIMAL-PLACEMENT

If w is the maximum size of the color sets $c(v_i, v_j)$, then the intersection of two such sets can be obtained in $O(w)$ time, assuming that the sets $c(v_i, v_j)$ are ordered. In step (3c), for each node $x \neq$ the root node, we perform the intersection of k color sets, where k is the degree of x, i.e., one plus the number of children of x. (If $I = \emptyset$, we may be able to terminate the intersection computation early.) This takes at most $O(kw)$ time. The total complexity is therefore $O(nw) = O(n)$, for a fixed w, where $n = |T|$, since the computation of levels take $O(n)$ time.

3.2. Verifying a given placement P

To test whether a given placement P suffices for broadcasting in a tree network T, we successively decompose T into smaller trees of the form T_{ij}, v_j adjacent to v_i, by choosing the nodes $v_i \in P$ in some order. The final set of trees does not depend on the order in which we select the nodes v_i. Next, we apply the algorithm OPTIMAL-PLACEMENT to each of the final trees and determine the minimum number of converters for it and an associated placement. This gives the minimum number of converters and their placements needed in addition to P for the broadcasting. The computation needed for decomposition at each node $x \in P$ is $O(d_x)$, where d_x = degree of node x. The total computation for finding the additional converters is therefore $O(n)$, where $n = |T|$.

4. GENERAL CASE

The key problem now is to find a spanning tree $T \subseteq N$ such that if P_T is an optimal converter-placement for broadcasting in T, then $|P_T|$ is as small as possible. Since the number of spanning trees T can be exponentially large in the number of nodes in N, a direct examination of each T is not practical. We give below a simple heuristic where, starting from an arbitrary spanning tree T, we attempt to successively modify T, one link at a time, to reduce the number of converters. As with any heuristic algorithm, we cannot guarantee that this algorithm will find a spanning tree T with the smallest possible $|P_T|$ (see Example 4 below).

Algorithm PLACEMENT-HEURISTIC:

Input: A connected optical network $N = (V, E)$.

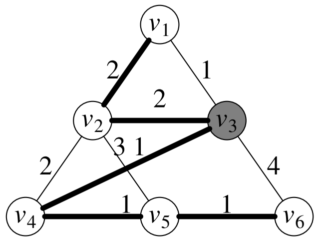
Output: A minimal subset of nodes P for placement of converters for broadcasting in N and an associated spanning light-tree T of N for the broadcasting.

1. Choose an arbitrary spanning tree T of N and compute an optimal placement of converters P_T for T using the algorithm in Section 3. Let $P = P_T$.
2. Choose a link $e = (v_i, v_j) \notin T$ and do the following:
   (a) Choose a link e' in the cycle of $T + e$, consider the spanning tree $T_{ee'} = T + e - e'$ and compute an optimal converter-placement $P_T^{ee'}$ for $T_{ee'}$ using the algorithm in Section 3.
   (b) Find a link e' which gives the minimum $|P_T^{ee'}|$, and let $T_e = T_{ee'}$ and $P_e = P_T^{ee'}$ corresponding to that e'.
   (c) If $|P_e| < |P|$, then replace T by T_e and P by P_e.
3. Repeat step (2) until no more improvement is possible in the number of converters.

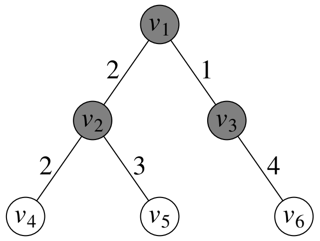
Note that there can be more than one choice of e' in step 2(b) for a given e, and the different choices for e' at any stage may lead to different sizes for the final P. A slight variation of the above algorithm would be to delay the replacement of T by T_e in step (2c) and instead first compute the optimal improvement P_e for each $e \in N - T$, and then replace T by the corresponding T_e (and P by the associated P_e) that gives the best optimal improvement. In this way, we can eliminate the effect on the order in which the links e are selected in step (2), which may affect the final P.

Example 4. Consider the optical network in Fig. 4(i), with each link having a single color; the bold lines show a spanning tree which require only one converter at node v_3, giving an optimal solution. Fig. 1(ii) shows a different starting spanning tree T and an optimal placement of converters P_T. Now, suppose we choose $e = (v_2, v_3)$, giving two choices for $e' = (v_1, v_2)$ and (v_1, v_3); each of these choices leads to a reduction of the number of converters from 3 to 2 for an optimal $P_T^{ee'}$. Fig. 4(iii) shows the case for $e' = (v_1, v_3)$. Now, consider $e = (v_3, v_4)$ giving again two choices for $e' = (v_2, v_3)$ and (v_2, v_4) and neither of these leads to a reduction in the number of converters when we compute an optimal $P_T^{ee'}$. (In the first case, the number of converters needed actually goes up from 2 to 3.) For $e = (v_4, v_5)$ and (v_5, v_6), one cannot improve the number of converters from 2, and the final solution is as shown in Fig. 1(iii). This shows that the heuristic may not always find the optimal solution. Indeed, if we started with T as indicated in Fig. 4(i), we would have obtained the optimal solution. □

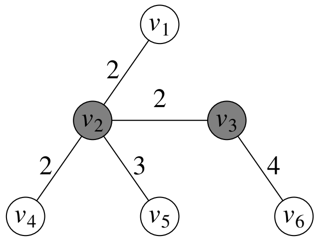
Note that if we modify the above algorithm where we add, say, two links simultaneously to the current T and delete two links to eliminates the cycles, then one could get a better solution than would be possible with one link-at-a-time approach. By adding the links (v_4, v_5) and



(i) An optical network N which needs a convertex at v_3 for broadcasting using the bold links.



(ii) A possible starting spanning tree T; it needs 3 converters at v_1, v_2, v_3.



(iii) Improvement in the number of convertexs for $e = (v_2, v_3)$ and $e' = (v_1, v_3)$.

Figure 4. Illustration of the algorithm PLACEMENT-HEURISTIC.

(v_5, v_6) together and deleting the links (v_2, v_5) and (v_3, v_6), we get a different optimal solution using only one converter at v_4.

4.1. Evaluation of PLACEMENT-HEURISTIC

In addition to an implementation of OPTIMAL-PLACEMENT algorithm and the PLACEMENT-HEURISTIC, we also developed a program to create all possible spanning trees in a connected network. For the network in Fig. 4(i), we get 61 spanning trees $T^{(j)}$. Nine of them give the optimal converter-placement of size 1, 22 give the optimal converter-placement of size 2, 27 give optimal converter-placement of size 3, and finally 3 give optimal converter-placement of size 4. On the other hand, out of $52 = 61 - 9$ trees $T^{(j)}$ for which the optimal placement is > 1, 21 of them require only one edge-exchange to arrive at a tree $T_{ee'}^{(j)}$ with an optimal placement of size 1 (each such $T_{ee'}^{(j)}$ is necessarily one of 9 trees mentioned earlier), and 21 of them require two edge-exchanges to arrive at a tree with an optimal placement of size 1. There are three $T^{(j)}$ which do not allow any improvement; each of them has optimal placement of size 2; there are five $T^{(j)}$ which allow improvement by one edge-exchange and they still require optimal placement of size 2. Finally, there are two $T^{(j)}$ which allow improvement by two edge-exchanges and they also require optimal placement of size 2. In some of these cases, one edge-exchange $T^{(j)}_{ee'}$ had reduced the optimal placement by more than one.

In another experiment, we modified the link colors in the graph in Fig. 4(i) by randomly assigning two colors in the range {1, 2, 3, 4} to each link. The results of the heuristic algorithm for this case are summarized in Table 1 in the first Column labeled "Expr-1". The data-set for other three columns are obtained by successively modifying one color on one edge at a time. This shows that a small change in the color sets $c(v_i, v_j)$ can have a significant impact on the success of the heuristic algorithm.

- #1. Optimal size of converter-placement
- #2. Num of trees $T^{(j)}$ which give optimal converter-placement of size 0.
- #3. Num of trees $T^{(j)}$ which give converter-placement size 0 after 1 edge-exchange.
- #4. Num of trees $T^{(j)}$ which give converter-placement size 0 after 2 edge-exchanges.
- #5. Total number of successful cases of $T^{(j)}$.
- #6. Num of other trees $T^{(j)}$ which do not allow any improvement.
- #7. Num of trees $T^{(j)}$ which allow some improvement but do not reach the optimal size 0.
- #8. Total number of spanning trees.

Table 1.

	Expr-1	Expr-2	Expr-3	Expr-4
#1.	0	0	0	0
#2.	1	1	3	11
#3.	8	9	18	30
#4.	2	8	9	8
#5.	11	18	30	49
#6.	29	32	23	12
#7.	21	11	8	0
#8.	61	61	61	61

5. CONCLUSION

We have presented here a linear algorithm for the optimal converter-placement problem in an optical tree network for broadcasting. We then used it to obtain a heuristic for the general network.

Acknowledgement. The authors express their thanks to E. El-Amawy and S. Pascu for valuable discussions and comments.

6. REFERENCES

1. B. Mukherjee, *Optical communication networks*, McGraw-Hill, 1997.
2. Kleinberg, J. and Kumar, A., Wavelength conversion in optical networks, *Proc. 10th ACM-SIAM Symp. on Discrete Algorithms*, 1999
3. Kundu, S. and Yi, T., Optimal converter-placement in optical tree networks, Tech. Rept #03-02, Louisiana State University, Baton Rouge, LA 70803, USA.
4. Sahasrabuddhe, L.H. and Mukherjee, B., Light-trees: optical multicasting for improved performance in wave-length routed networks, *IEEE Comm. Magazine*, 37(1999 Feb), pp. 67-73.
5. Ramamurthy, B. and Mukherjee, B., Wavelength conversion in WDM networking. *IEEE J. Selected Areas of Communication,* 16(1998), pp. 1061-1073.
6. Ruan, L., Du, D., Hu, X., Jia, X., LI, D., and Sun, Z., Converter placement supporting broadcast in WDM optical networks, *IEEE Transactions on Computers, 50(2001), pp. 750-758.*
7. Auletta, V., Caragiannis, I., Kaklamanis, C., and Persiano, P., Bandwidth allocation algorithms on tree-shaped all optical networks with converters, *Proc. 4th Intern. Coll. Structural Information and Comm. Complexity,* SIROCC0-97, 1997, pp. 13-27.
8. Venugopal, K.R., Sivakumar, M., and Sreenivasakumar, P., A heuristic for placement of limited range wavelength converters in all-optical networks, *Computer Networks,* 35(2001), pp. 143-163.
9. Zang, H., Jue, J., and Mukherjee, B., A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks, *Optical Network Magazine, 1(2000), pp. 47-60.*