On the Application of K-Center Algorithms to Hierarchical Traffic Grooming

Bensong Chen, Rudra Dutta, George N. Rouskas
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534
chenbensong@yahoo.com, dutta@csc.ncsu.edu, rouskas@csc.ncsu.edu

***Abstract*— In this paper, we study a clustering technique for the hierarchical traffic grooming approach in WDM mesh networks. The objective is to minimize the cost of electronic ports, as well as the wavelength requirement of the solution. In the hierarchical grooming approach we have presented in previous work, the first phase is to partition a large mesh network into clusters of nodes. The clustering phase is very important for the final grooming result. Various clustering approaches have been considered in literature; however, not all are suitable for traffic grooming application because they do not take grooming goals into account. In this work, we select a suitable existing clustering algorithm, developed for the *K-Center* problem, and study its performance as a clustering algorithm for hierarchical grooming. We then improve the algorithm, adapting it specifically for the traffic grooming problem. Experimental results show that the improved version generally provides better solutions than the original algorithm, on various traffic patterns, for the general topology grooming problem instances.**

I. Introduction

Traffic grooming in general topology networks has gained interest in recent years. In this architecture, WDM and wavelength routing technologies are used to create a *virtual topology* of optical continuous channels or *lightpaths* which can span more than one phsyical link, and end-to-end user demand traffic (typically sub-wavelength in magnitude) is then multiplexed using TDM over a lightpath or sequence of lightpaths. Various objectives have been considered in traffic grooming literature; a practical objective is to minimize the total number of electronic ports required to originate and terminate the set of lightpaths in the logical topology design. This is equivalent to minimizing the total number of lightpaths in the virtual topology. This objective is motivated by the fact that electronic port devices still dominate WDM network equipment cost. Both *static* and *dynamic* traffic models have been considered by various researchers. Early work focused on the ring topologies [1]–[3], which reflected the effort of upgrading SONET with WDM technology. Since more backbone and access optical networks are not, or will not be, constrained to the elemental topologies, recent studies are extending the research to grooming in general mesh networks [4]–[8]. The current literature on traffic grooming has been reviewed and classified by several surveys in the literature, such as [9].

Future interconnection requirements are expected to form even larger (hundreds of nodes) optical networks with complex topology. In recent work, we have shown that a *hierarchical* grooming approach can work well on networks consisting of many nodes [10]. In this approach, the nodes in the network are first partitioned into several *clusters* or groups. This process is called *clustering*. Each cluster is groomed by itself, and then the traffic between nodes in different clusters is groomed. We describe pertinent details of this approach briefly in the next section. Our results show that this approach can provide good grooming characteristics, and its decompositional approach also makes it scalable to large network sizes.

In [10], we primarily focused on the hierarchical grooming mechanism itself, assuming a suitable clustering of the network graph to be available. Our approach is superficially similar to that of [11], but the research presented in [11] pertains to ring networks only, and cannot be easily generalized because it rests on the existence of a set of contiguous or "close" nodes that works for rings but does not extend to general topologies.

In this paper, we focus on a complementary problem to that of [10]. Obviously, arbitrary clustering of the network nodes would not generally provide good grooming performance; thus we need good clustering algorithms that are designed with the hierarchical approach in mind. Such an algorithm design is the focus of this paper. Clustering is widely used in network design, but for the general topology traffic grooming problem, there has not been much related work in the literature. In this work, we study the performance of a clustering algorithm from the study of the *K-Center* problem, and make improvements on the algorithm to make it more suitable for the hierarchical grooming problem.

The rest of the paper is organized as follows. In Section II, we briefly introduce the notation and the hierarchical grooming approach. Section III gives a discussion on clustering methods in general, and the need for clustering algorithm for the traffic grooming problem. In Section IV, we study an algorithm designed for the *K-Center* problem, and revise it to improve its performance on the hierarchical grooming approach. We obtain lower bounds for the number of lightpaths and the number of wavelengths, and present numerical results in a 47-node network with various parameters in Section V, which compares the original and improved clustering algorithms. We conclude the paper in Section VI.

This work was supported by NSF grant ANI-0322107.

II. Overview of Hierarchical Grooming

A. Notation

In this study, we consider the *static* traffic grooming problem in general topology networks. The problem specifies as input a fixed physical network topology as a graph $G(V, E)$, a wavelength limit W on each directed fiber link, a *grooming factor*, or *granularity* C, and an $N \times N$ static traffic demand matrix $T = [t^{(sd)}]$, denoting aggregate or estimated traffic from each source node s to each destination node d (N denotes the number of nodes *i.e.* $N = |V|$). Each link of G consists of a bidirectional pair of fiber links. C defines the *wavelength channel capacity* in multiples of a basic transmission rate (such as OC-3). We allow individual traffic demands $t^{(sd)}$ to be greater than C. We assume the wavelength continuity constraint, that is a lightpath must use the same wavelength over every fiber link it traverses. The output of the traffic grooming problem is a configuration of the network that carries all the traffic demands with a set of lightpaths. Our main objective is to minimize the number of electronic ports in the whole network, which is equivalent to minimizing the number of lightpaths in the configuration. For a more formal definition of the problem expressed in integer linear programming formulation, please see [9].

Note that the wavelength limit W is treated as a constraint rather than part of the optimization objective above, but it remains a useful secondary objective to minimize. To obtain a feasible solution, W must be greater than a certain lower bound, but if there are sufficient number of wavelengths available in the network, we can concentrate on other objectives as long as a solution requires no more wavelengths than the given constraint. On the other hand, if a solution requires fewer number of wavelengths, it is more likely to be feasible for a given W. In our numerical results, we track the performance of the grooming algorithm both in the number of lightpaths and the number of wavelengths required.

B. The Hierarchical Grooming Approach

The traffic grooming problem (and even its subproblems) in general topology networks has long been known to be NP-hard [12]–[14], and heuristic approaches have been advanced in literature. In [10], we have presented a heuristic based on a heirarchical approach. In this section, we describe the broad nature of the approach to provide the context for the clustering algorithm. Briefly, it consists of three phases; the clustering algorithm is needed in Phase 1.

1. Clustering of network nodes. Partition the network into m clusters and designate one node in each cluster as the hub node.

2. Logical topology design and traffic routing. For each cluster, traffic from one node in the cluster to another is designated either to travel directly on a lightpath, or over two lightpaths via the hub where it is electronically groomed with other traffic. Thus each cluster can be treated as a *virtual star* for the purpose of grooming intra-cluster traffic. This requires us to solve as many virtual stars as the number of clusters.

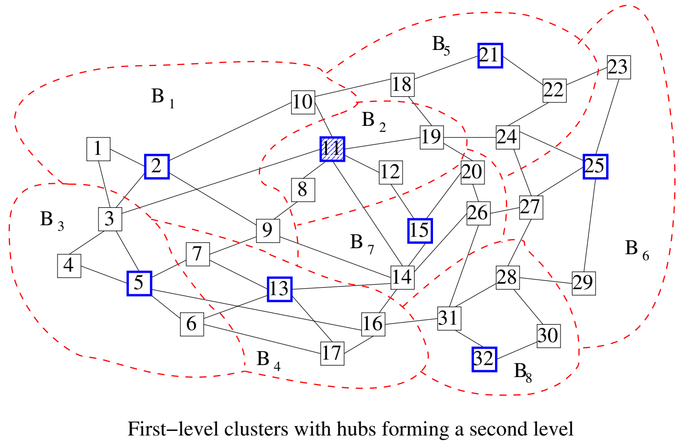

Fig. 1. A 32-node WDM network, its partition into eight first-level clusters $B_1, \cdots, B_8$, and second-level cluster B consisting of the eight first-level hubs

Finally, inter-cluster traffic can be groomed by viewing the set of all hubs as a *single* virtual star again, and designating a second-level hub. Specific enhancements can be made to this approach to address conditions such as some node pairs in different clusters having a large amount of traffic to each other; for details see [10].

3. Lightpath routing and wavelength assignment. Each of the lightpaths created in Phase 2 are assigned a wavelength and path on the underlying physical topology of the original mesh network. Since the RWA problem on arbitrary network topologies has been studied extensively in the literature [6], [7], [12], [15], [16], and RWA is not the focus of our research, we choose not to develop an RWA algorithm of our own for use in this work. Instead, we adopt the LFAP algorithm [15] from recent research, which is fast, conceptually simple, and has been shown to use a number of wavelengths that is close to the lower bounds on different problem inputs. For lack of space, we cannot describe the details of the RWA method, please consult [15] for details.

Figure 1 shows an example of clustering on a 32-node network with 8 clusters, $B_1, \ldots, B_8$, each assigned a hub node shown in bold. These hub nodes form a higher level in the network, where inter-cluster traffic is carried between them. *Note that the RWA in Phase 3 is no longer constrained by the virtual stars;* this is one of the strengths of our approach. For instance, in cluster B_8, the virtual star solution might dictate that a lightpath be formed between nodes 28 and 30. In a physical star, such a lightpath would pass through the hub node 32, but in our approach the RWA also has the option to route it over the direct link between nodes 28 and 30. A similar approach is followed at the second (inter-cluster) level.

From the point of view of the clustering algorithm however, these enhancements are incidental; the focus is on generating a clustering of a given graph, grouping topologically close nodes together. In the next section, we will discuss the existing clustering methods in the literature and how they may relate to the traffic grooming problem.

III. Network Clustering Methods

Clustering is a common approach in network design and organization. A classic text [17] defines clustering as "grouping of similar objects", and discusses many mainstream clustering algorithms. The algorithms were summarized into two categories: *minimum cut* and *spanning tree*. The inputs were generally a set of nodes (objects) and edge weights (relationships) between them, while the outputs are an objective function minimized or maximized. Most studies in the literature also define clustering along these general lines.

In the hierarchical grooming context, we want to find good clustering as the first phase of the whole approach, where the situations are more complicated. For instance, the inputs of our problem consist of not only a physical network topology, but also a traffic demand matrix and several constraints; the outputs also do not have an objective function clearly related to the network topology graph such as physical cut size or inter-cluster traffic. For that reason, most of the existing clustering techniques are not directly applicable in our study.

For instance, some study considers only the communication between nodes. The work in [18] introduced an algorithm that can group a nearly completely decomposable (NCD) matrix into blocks, so that the weighted arcs between blocks have values not exceeding a given threshold. The algorithm, called TPABLO, was used to group nodes in large Markov Chains. In the traffic grooming context, we want the traffic demands within a cluster to be 'denser' than inter-cluster traffic, which is similar to what TPABLO wants. However, the algorithm does not consider the given physical topology, so if we directly apply it to the grooming problem, nodes that are far away from each other may be grouped together, resulting in unnecessary long lightpaths even for local traffic, which will require significantly more wavelengths to avoid channel collision.

Some other work concentrated on the physical topology only. The goal was generally to partition the nodes into contiguous clusters containing roughly equal number of nodes, and at the same time minimize the overall cut size. An example of the work can be found in [19], and the authors developed a software package named METIS for solving a set of similar problems. Such a clustering method is still not directly applicable to traffic grooming. First, we do not have the constraint that each cluster must be equal in size; second, an overly small physical cut size will result in bottlenecks for inter-cluster traffic, which is not good for lowering wavelength requirement. In fact, the algorithms are designed to deal with VLSI design, a very different problem, where equality in size and a minimum of cross-layer connections are essential for each module.

There is another set of clustering problems concerned with the physical topology, namely *K-Center, K-Clustering, K-Median* and *Facility Location* problems [20]–[23]. Unlike the partitioning problem in METIS, they do not require the size of each cluster to be roughly equal. Of all the variations, the *K-Center* problem is the most suitable for our needs. Give a graph G, the goal of the *K-Center* problem is to find a set S of K "center" nodes in G, so that the maximum distance from any network node to its nearest center node in S is minimized. Thus, the set S implicitly defines K clusters with corresponding hub nodes in S.

The *K-Center* clustering method can be useful in the traffic grooming problem, because it can avoid long lightpaths within a cluster, which is bad for lowering the wavelength requirement in the routing and wavelength assignment step. Also, this type of clustering tends to avoid forming path-like physical topology for each cluster as well (which are undesirable for us because the virtual star method on such a cluster will result in high congestion on links connected to the hub, see next Section). Moreover, the parameter K defines the number of clusters in the clustering result, thus allows ready tuning for grooming needs.

The related *K-Median* problem is the overall version of the K-Center problem. That is, it minimizes the *sum* of distances between network nodes and their respective centers. There is also a variation of the K-Center problem called the *capacitated* K-Center problem, which adds restriction to the size of each cluster. In the hierarchical grooming problem we consider, we care more about the length of each local lightpath, and do not restrict on the cluster size, so we will focus on the original K-Center problem in this work.

IV. K-Center Algorithms for Hierarchical Grooming

In this section, we first describe a K-Center algorithm that was known to have the best approximate rate, then we make revisions on it according to our needs in a separate subsection.

A. An $O(NK)$ K-Center Algorithm

Previous studies show that the *K-Center* problem is NP-Complete, and is 2-approximable, while 2 is the best approximation rate we can get in polynomial time [24], [25]. We implemented the 2-approximable algorithm proposed by Gonzalez [24] to do *K-Center* clustering, which is then compared with the improved algorithm later in Section V. Note that the algorithm assumes that edge weights obey the triangle inequality. In our problem, this requirement is satisfied because we use the shortest distance between two nodes as the weight. We actually use the version of Gonzalez' algorithm that can be found in [26]; which we describe below for easy reference. In what follows, we assume that all-pair shortest paths have been calculated and recorded as input matrix $dist$.

1) Initialization: Create a single cluster of all nodes, $B_1 = \{v_1, \ldots, v_n\}$, with hub node $h_1 = v_1$.
2) Our goal is to create k clusters, adding one cluster at each iteration. Suppose in the current iteration, there are x existing clusters, and a distance d is the current maximum distance between any node and its hub, *e.g.*, $d = \max\{dist(v_i, h_j)\}, v_i \in B_j$. We find such a node v that the distance between v and its hub is d.
3) Continue at iteration x, Create a new cluster B_{x+1} with $h_{x+1} = v$ as the only node. Then for each $v' \in B_1 \cup$

$\ldots \cup B_x$, suppose $v' \in B_j$, if we find that $dist(v', h_j) > dist(v', v)$, that is, v' is closer to the new cluster hub than it is to its old hub, we move v' from its old cluster B_j to the new cluster B_{x+1}.

4) Iterate the previous two steps for $K-1$ times, and we finally get k clusters with corresponding hub nodes as output.

In the algorithm, Steps 2 and 3 are iterated for $K-1$ times. In Step 2, checking for the minimum node-to-hub distance takes $O(N)$ time, so does the update for the new cluster in Step 3. Therefore, the whole algorithm has polynomial-time complexity $O(NK)$. (Note that if the all-pair shortest paths are not provided as input, we need $O(N^3)$ preprocessing time to calculate them from the given graph, *e.g.* with Floyd-Warshall algorithm).

B. Improving the K-Center Algorithm

As we discussed in Section III, the K-Center problem itself has only the physical topology as input, and the only goal is to minimize the maximum node-to-hub distance. However, in the traffic grooming context, hub capacity and lightpath routing should also be considered.

Suppose we obtain a clustering solution from the original K-Center algorithm, and apply it to the hierarchical grooming approach. It is possible that the hub nodes in the clustering do not have large physical degrees. Since hubs are responsible for originating and terminating lightpaths that are locally groomed, as well as traffic between different clusters, the number of lightpaths that have to go through the physical links connected to each hub is generally large. Since all such lightpaths will have to be routed over physical links connected to the hub node, this will result in a large number of wavelengths required, even if the RWA distributes the lightpaths on the physical links perfectly evenly. For this reason, hub nodes should be selected so that they have more capacity - in this case, larger physical degree. That will potentially allow each fiber to carry fewer number of lightpaths, which results in lowered wavelength requirement. With this idea, we make the following improvement on the K-Center algorithm in Section IV-A:

1) In Step 1, instead of selecting $h_1 = v_1$, we choose the node v that has the maximum physical degree as the hub node;
2) In each iteration of Step 2, instead of selecting a single node v, find the set of nodes $\{v_d\}$, each having max distance d to its corresponding hub;
3) For the corresponding iteration of Step 3, form the new cluster B_{x+1} has hub node $h_{x+1} \in \{v_d\}$, such that h_{x+1} has the *maximum physical degree*.

Note that the revised algorithm does not change the characteristics of the original algorithm with respect to the Min-Max distance, which automatically ensures the same approximability for the K-Center problem. However, it break ties when there are more than one candidate nodes that can be chosen as the new hub. By this improvement, we get a 2-approximable K-Center algorithm in which the hub degrees can be larger.

The tradeoff between the K-Center objective and our requirement for hub degrees deserves further study. For instance, if we expand the candidate hubs $\{v_d\}$ in Step 2 to nodes with hub distance d and $d-1$, the resulting algorithm will not be 2-approximable, but the relaxation can facilitate in choosing hubs with even larger degrees. In this paper, we consider only the simple improvement that keeps the 2-approximation characteristics.

V. Numerical Results

In this section, we present experimental results to demonstrate the performance of the K-Center clustering algorithm and the improved version.

The traffic matrix $T = [t^{(sd)}]$ of each problem instance we consider is generated by drawing $N(N-1)$ random numbers (rounded to the nearest integer) from a Gaussian distribution with a given mean t and standard deviation σ that depend on the traffic pattern. We consider two traffic patterns here:

1) *Random pattern.* We have found that random patterns are often challenging in the context of traffic grooming, since the matrix does not have any particular structure that can be exploited by a grooming algorithm. To generate a traffic matrix for a problem instance, we let the standard deviation of the Gaussian distribution be 150% of the mean t. Consequently, the traffic elements $t^{(sd)}$ take values in a wide range around the mean, and the loads of individual links also vary widely. If the random number generator returns a negative value for some traffic element, we set the corresponding $t^{(sd)}$ value to zero.
2) *Falling pattern.* This traffic pattern is designed to capture the traffic locality property that has been observed in some networks. Specifically, if the mean of the Gaussian distribution for node pairs that have shortest distance 1 is t, then the mean for node pairs with shortest distance 2 (respectively, 3) is set to $0.8t$ (respectively, $0.6t$); for all other pairs, the mean is set to $0.2t$. We also let the standard deviation of the Gaussian distribution be 20% of the mean.

We also want to test the performance of the clustering algorithm in the first phase of the hierarchical approach as well. Since there is no existing clustering algorithm specifically for the traffic grooming problem we consider, we implement the known *K-Center* algorithm by Gonzalez described in Section III and compare its performance with ours.

A. Lower Bounds

The actual numerical values of the number of lightpaths and wavlengths in solutions obtained by these algorithms would depend on the specific values of the parameters of different problem instances. Thus a normalized value of these quantities is much more preferable in understanding the performance of the algorithms. Ideally, the values obtained should be normalized to the optimal values; unfortunately, computing the optimal by exhaustive computation or solving the ILP is out of the question due to the size of the problem instances. In this section, we obtain lower bounds on both the number of lightpaths and the number of wavelengths required to carry the traffic matrix T, and then use the values of these *bounds* to normalize the number of lightpaths and wavelengths obtained by our algorithms. These bounds are obtained *independently* of the manner (e.g., hierarchical or otherwise) in which traffic grooming is performed. Therefore, the bounds are useful in characterizing the effectiveness of our entire approach (and hence the quality of the clustering as well).

1) ILP Lower Bound on the Number of Lightpaths: A simple lower bound F^l on the total number of lightpaths (our main objective) can be found by:

$$F^l = \max\left(\sum_s \left\lceil \frac{\sum_d t^{(sd)}}{C} \right\rceil, \sum_d \left\lceil \frac{\sum_s t^{(sd)}}{C} \right\rceil\right) \quad (1)$$

This bound is based on the observation that each node must source and terminate a sufficient number of lightpaths to carry the traffic demands from and to this node, respectively. This bound can be determined directly from the traffic matrix T.

We also found a better lower bound by ILP relaxation. That is, from the ILP formulation for the traffic grooming problem, we remove some variables and some constraints, so that we can obtain a relaxed solution that serves as a lower bound for the original objective. We found that for large network of general topology, we have to remove most of the constraints and variables to get even a lower bound in reasonable time constraint. Details on how to obtain the bounds can be found in our recent work [10].

2) A Lower Bound on the Number of Wavelengths: Consider a bisection cut of the network graph G, and let t be the maximum amount of traffic that needs to be carried on either direction of the links in the cut set. Let x be the number of links in the cut set, and C the capacity of each wavelength. Then, the quantity $\lceil t/xC \rceil$ is a lower bound on the number of wavelengths for carrying the given traffic matrix. This bound does not require any information regarding the logical topology or the routing and wavelength assignment of lightpaths.

We use the METIS software [27] to find a small-cut bisection on the network we experiment on, so that the number of nodes in each side of the bisection is roughly equal. This will potentially generate more traffic that traverses the cut, resulting in tighter (higher) lower bounds for the wavelength requirements.

B. Results on a 47-node Network

We experiment on a mid-sized network that is obtained by slightly adapting a topology that appeared in a historical paper on network design [28]. The network has a relatively balanced physical topology, with 47 nodes and 96 links, and the nodes have high connectivity, which offers more options for lightpath routing. As a result, there is no significant bottleneck bisection cuts.

For the falling traffic pattern, we first generate thirty problem instances, *i.e.*, thirty traffic demand matrices using the Falling traffic pattern, then we calculate the lightpath and wavelength results obtained from the hierarchical grooming method. For each problem instance, we calculate the resulting lightpath count and wavelength requirement using different cluster sizes (the parameter K in K-Center clustering), with both the original and the improved clustering algorithms.

Figures 2 and 3 show the results on the 47-node network with falling traffic pattern for the number of lightpaths, from the original and revised K-Center algorithms, respectively. We use the number of clusters $K = 2, 4, 6$ to generate different clusters. Results show that the number of lightpaths from the two methods don't have much difference. However, as Figures 4 and 5 illustrate, the revised algorithm outperforms the original algorithm on wavelength requirements for all three values of K.

The 47-node network has a balanced physical topology, which is more likely to generate clusters of approximately equal size for both algorithms. On the other hand, the revised algorithm provides more hub capacity to route wavelengths, so the solutions require much fewer wavelengths than those from the original algorithm.

C. Results on a 128-node Network

We now consider a 128-node, 321-link network which corresponds to the worldwide backbone operated by a large service provider; we obtained the topology information from data documented on CAIDA's web site [29]. This topology is imbalanced, in the sense that there exists a bisection with a small cut size of 5 links that divides it into two parts of 114 and 14 nodes, respectively. We identify this critical cut with the method discussed in Section V-A.2, and use it to calculate the lower bound on the number of wavelengths.

For the 128-node network, because the physical topology is not balanced, results from the two algorithms are more different. From Figures 6 and 7 for the random traffic pattern, we see that the revised algorithm not only outperforms the original algorithm on the wavelength requirement greatly, but also gives around 1% improvement on the lightpath count. In more imbalanced topologies, node degrees may have large difference, so the choice of hubs becomes more important. Remember that in the K-Center algorithm, we choose a node that has maximum distance to its hub node as a new hub. In imbalanced networks, it is very possible that a 'dangling' node will be chosen, that is, a 'leaf' node that has physical degree 1. Our revised algorithm, however, tries to avoid such case from happening.

Although the revised algorithm considers only increasing the hub capacities, interestingly, it is better with respect to lightpath count as well, though the improvement is small. This is mostly because in the revised algorithm, hub nodes have larger connectivity, so the size of each cluster is likely to be more balanced, which facilitates the virtual topology design step for grooming both within and between clusters. The K-Center problem does not have constraints on each cluster size, but for networks with imbalanced topologies, the choice of hubs with imbalanced cluster sizes may generate small but

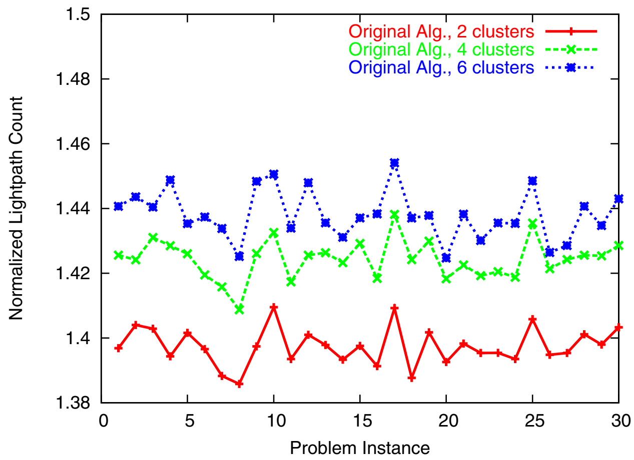

Fig. 2. Lightpath, Falling pattern, Original Algorithm

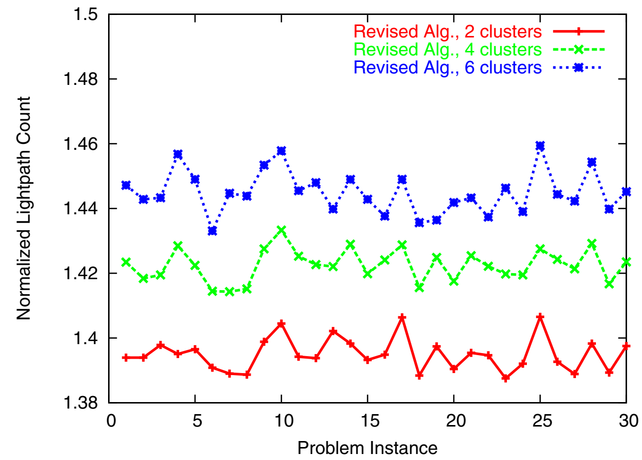

Fig. 3. Lightpath, Falling pattern, Revised Algorithm

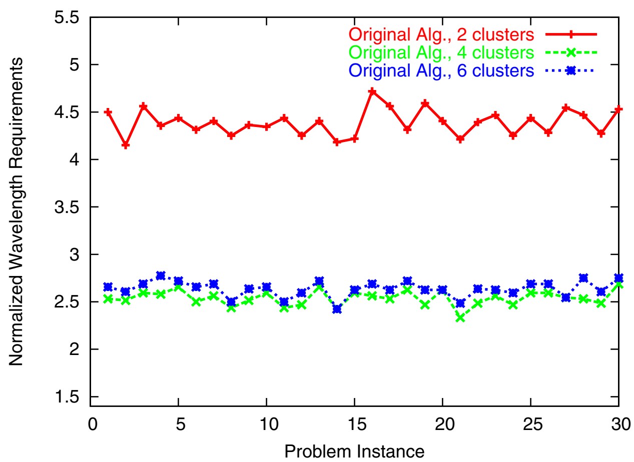

Fig. 4. Wavelength, Falling pattern, Original Algorithm

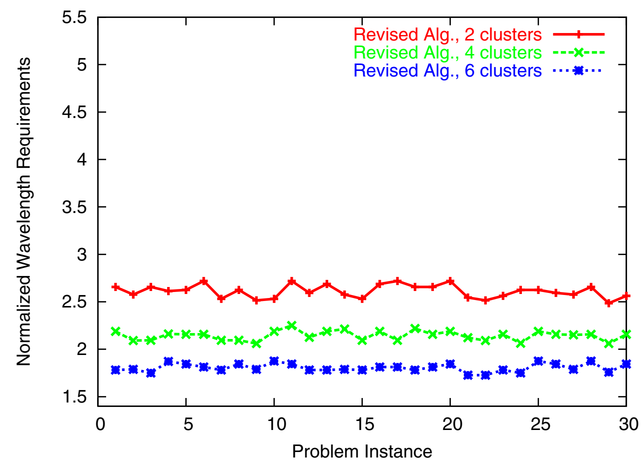

Fig. 5. Wavelength, Falling pattern, Revised Algorithm

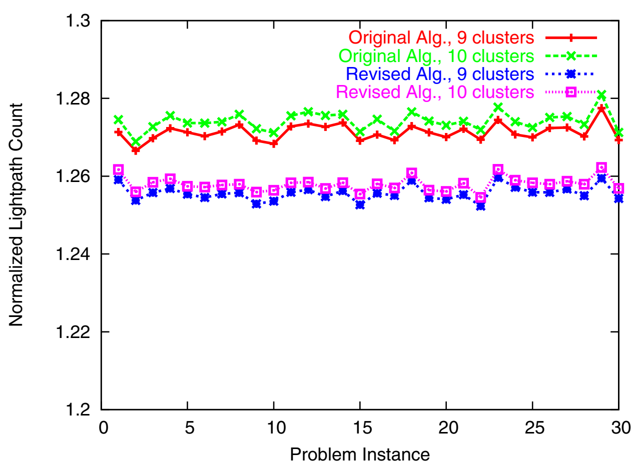

Fig. 6. Lightpath, Random pattern, 128-node network

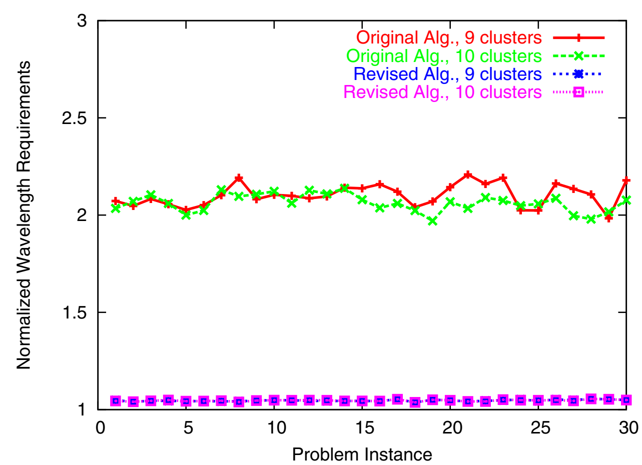

Fig. 7. Wavelength, Random pattern, 128-node network

visible difference in the virtual topology design step. Another reason is that the K-Center algorithm is a 2-approximation algorithm and aims at the Min-Max objective, so the average lightpath length from two algorithms can also be different. This will further impact the results on the distance-dependent traffic patterns.

VI. Concluding Remarks

We have studied the clustering algorithms in the hierarchical approach for efficient and scalable traffic grooming in mesh WDM networks. We implemented an existing clustering algorithm from the K-Center problem that is not dedicated to the traffic grooming problem, then made improvements to make it more suitable to our needs. Experimental results showed that the improvements are effective on large general network topologies with various traffic patterns. Currently, we are continuing our research on comparing and improving clustering algorithms for the general traffic grooming problem.

References

[1] O. Gerstel, R. Ramaswami, and G. Sasaki., "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 618–630, October 2000.

[2] P.-J. Wan, G. Calinescu, L. Liu, and O. Frieder., "Grooming of arbitrary traffic in SONET/WDM BLSRs," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1995–2003, 2000.

[3] R. Dutta and G. N. Rouskas, "On optimal traffic grooming in WDM rings," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 110–121, January 2002.

[4] V. R. Konda and T. Y. Chow, "Algorithm for traffic grooming in optical networks to minimize the number of transceivers," in *IEEE Workshop on High Performance Switching and Routing*, 2001, pp. 218–221.

[5] D.Li, Z.Sun, X.Jia, and K.Makki, "Traffic grooming on general topology WDM networks," *IEE Proc. Commun.*, vol. 150, no. 3, pp. 197–201, June 2003.

[6] J.Q.Hu and B.Leida, "Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks," *Proceedings of the IEEE INFOCOM 2004*, vol. 23, no. 1, pp. 495–501, March 2004.

[7] Z. K. P. Jr. and G. R. Mateus, "A lagrangian-based heuristic for traffic grooming in WDM optical networks," in *IEEE GLOBECOM 2003*, vol. 5,1-5, Dec 2003, pp. 2767–2771.

[8] C. Lee and E. K. Park, "A genetic algorithm for traffic grooming in all-optical mesh networks," in *IEEE SMC 2002*, vol. 7, 6-9, Oct 2002.

[9] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: Past and future," *IEEE Network*, vol. 16, no. 6, pp. 46–56, November/December 2002.

[10] B. Chen, G. N. Rouskas, and R. Dutta, "A framework for hierarchical traffic grooming in wdm networks of general topology," *BroadNets 2005*, 2005.

[11] J. Simmons, E. Goldstein, and A. Saleh, "Quantifying the benefit of wavelength add-drop in WDM rings with distance-independent and dependent traffic," *IEEE/OSA Journal of Lightwave Technology*, pp. 48–57, Jan 1999.

[12] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WANS," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171–1182, July 1992.

[13] B. Chen, G. N. Rouskas, and R. Dutta, "Traffic grooming in wdm ring networks to minimize the maximum electronic port cost," *Optical Switching and Networking*, 2005.

[14] R. Dutta, S. Huang, and G. N. Rouskas, "Traffic grooming in path, star, and tree networks: Complexity, bounds, and algorithms," in *Proceeding of ACM SIGMETRICS*, June 2003, pp. 298–299.

[15] H.Siregar, H.Takagi, and Y.Zhang, "Efficient routing and wavelength assignment in wavelength-routed optical networks," *Proceedings of the 7th Asia-Pacific Network Operations and Management Symposium*, pp. 116–127, October 2003.

[16] S. Baroni and P. Bayvel, "Wavelength requirements in arbitrary connected wavelength-routed optical networks," *IEEE/OSA J. Lightwave Technol*, vol. 15, no. 2, pp. 242–251, Feb. 1997.

[17] J. A. Hartigan, *Clustering ALgorithms*. Wiley, 1975.

[18] H. Choi and D. B. Szyld, "Application of threshold pertitioning of sparse matrices to markov chains," *Proceedings of the IEEE International Computer Performance and Dependability Symposium (IPDS)*, 1996.

[19] K. Schloegel, G. Karypis, and V. Kumar, "A new algorithm for multi-objective graph partitioning," Dept. of Computer Science, Univ. of Minnesota, Tech. Rep. 99-003, September 1999.

[20] J. Bar-Ilan, G. Kortsarz, and D. Peleg, "How to allocate network centers," *J. Algorithms*, vol. 15, pp. 385–415, 1993.

[21] D. Hochbaum and D. Shmoys, "A unified apporach to approximation algorithms for bottleneck problems," *J. ACM*, vol. 33, pp. 533–550, 1986.

[22] D. B. Shmoys, "Approximation algorithms for facility location problems," *Proc. APPROX 2000, LNCS*, vol. 1913, pp. 27–32, 2000.

[23] M. Thorup, "Quick k-median, k-center, and facility location for sparse graphs," *Proc. ICALP 2001, LNCS*, vol. 2076, pp. 249–260, 2001.

[24] T. Gonzalez, "Clustering to minimize the maximum inter-cluster distance," *Theoret. Comput. Sci.*, vol. 38, pp. 293–306, 1985.

[25] D. Hochbaum and D. Shmoys, "A best possible heuristic for the k-center problem," *Math. Oper. Res.*, vol. 10, pp. 180–184, 1985.

[26] Z. Xiang, "Color image quantization by minimizing the maximum intercluster distance," *ACM Transaction on Graphics*, vol. 16, no. 3, pp. 260–276, July 1997.

[27] G. Karypis and V. Kumar, "A fast and highly quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, 1998.

[28] P. Baran, "On distributed communications networks," *IEEE Transactions on Communications*, vol. 12, no. 1, pp. 1–9, March 1964.

[29] http://www.caida.org.