

Fast Heuristic Algorithm for Joint Topology Design and Load Balancing in FSO Networks

In Keun Son and Shiwen Mao

Department of Electrical and Computer Engineering, Auburn University, Auburn, AL

Email: soninkeun@auburn.edu, smao@ieee.org

Abstract—We investigate the challenging problem of joint topology design and load balancing in FSO networks. Important factors such as FSO link characteristics, cost constraints, traffic characteristics, traffic demand, and QoS requirements are considered in the problem formulation, along with objective functions of network-wide average traffic load and delay. We develop a fast heuristic algorithm to provide highly competitive solutions. The heuristic algorithm iteratively perturbs the current topology and computes network flows for the new topology, thus progressively improving the configuration and load balancing of the FSO network. Our simulation results show that the heuristic algorithm can achieve an optimality gap close to that of a branch-and-bound algorithm developed in our prior work, with significantly reduced computation time. The heuristic algorithm is complementary to the branch-and-bound algorithm. Jointly applying the algorithms can make the FSO network dynamically reconfigurable and adaptive to events occurring at both large and small timescales.

I. INTRODUCTION

Free Space Optics (FSO) are wireless communication systems that use open space for optical data transmissions. Compared to radio communications, FSO has many advantages, such as cost effectiveness, large transmission distance, free license, interference immunity, and high-bandwidth, among others [1]. FSO networks provide a promising solution to mitigating the scalability problem of wireless mesh networks [2]. To fully capitalize the potential of this technology, important networking problems, such as topology design, fault tolerance, routing and load balancing, should be carefully addressed.

There have been some interesting work on these issues. In [3], a distributed Minimum Spanning Tree (MST) algorithm was proposed to build degree-bounded tree topologies. In [2], [4], algorithms were developed to maximize network connectivity with a mesh topology. The load balancing problem was addressed in [5], [6], where several topology design heuristics were developed to minimize network-wide average load. To maximize the potential of FSO networks, the unique characteristics of FSO links should be considered, and the problems of topology design and routing of the traffic flows should be jointly considered and optimized [5], [6].

In this paper, we study the problem of joint topology design and load balancing in FSO networks. We consider FSO link characteristics, cost constraints, traffic characteristics, traffic demand, and QoS requirements in the problem formulation, along with objective functions of network-wide average load and delay. This problem is NP-hard in general [6], [7]. In [8], we apply the Reformulation-Linearization Technique (RLT) to obtain linear programming (LP) relaxations of the original complex problem, and incorporate the LP relaxations into a branch-and-bound framework [9]. The branch-and-bound

algorithm can produce highly competitive solutions with performance guarantees in the form of bounded optimality gap.

Although highly appealing, the RLT-based branch-and-bound algorithm has relatively high computational complexity. In this paper, we present a fast heuristic algorithm for the joint topology design and load balancing problem. The heuristic algorithm consists of three components: (i) initial topology design, (ii) multipath routing for load balancing, and (iii) topology perturbation. Starting from an initial topology that is designed to minimize the network-wide average load, the heuristic algorithm iteratively perturbs the current topology and computes network flows for the new topology, thus progressively improving the configuration and load balancing of the FSO network. Our simulation results show that the heuristic algorithm can achieve an optimality gap close to that of the branch-and-bound algorithm, with significantly reduced computation time.

The heuristic algorithm is complementary to the branch-and-bound algorithm. The latter is suitable for optimizing the FSO network design and operation at large timescales with guaranteed optimality. The former is suitable for dynamic reconfiguration of the FSO network in response to small timescale events. We envision that the RLT-based branch-and-bound algorithm will be executed at relatively large time intervals when significant changes occur in the FSO network, while the heuristic algorithm will be kept running to continuously optimize the operation and configuration of the FSO network in response to small timescale events such as bad weather conditions or temporary changes in traffic demand.

The remainder of this paper is organized as follows. In section II, we describe the system model and present the problem statement. The heuristic algorithm is presented in Section III. Our simulation studies are presented in Section IV. Section V concludes this paper.

II. SYSTEM MODEL AND PROBLEM STATEMENT

In this section, we present our system model and assumptions. The problem formulation was also presented in our work [8], but is still included here for completeness.

A. Network and Channel Model

We consider an FSO network consisting of n base stations (BS), which provide mobile users with network access. Each BS could be the head of a cluster consisting of multiple access points [2]. The aggregate traffic will be relayed through wireless optical links. We assume that each BS has multiple sets of wireless optical devices in order to support the aggregate traffic load and provide a rich mesh connectivity. The FSO links are immune to electromagnetic interference and are point-to-point connections with narrow beam divergence [1], [2], [5], [6].

The FSO network can be modeled as a *simple* graph $G(V, E)$, where each vertex $v \in V$ represents a BS and each edge $e \in E$ is an FSO link. Let n and m denote the cardinality of V and E respectively. We assume an $n \times n$ traffic matrix \mathbf{F} that describes the traffic demand (measured, estimated, or projected) for the access network, where each element $f_{sd} = [\mathbf{F}]_{sd}$ represents the mean data rate between each source and destination pair $s-d$.

We characterize each FSO link $e = (i, j) \in E$ with link capacity c_{ij} and link reliability γ_{ij} . As in prior work [6], we assume that each FSO channel is full duplex with symmetric capacity and a nominal data rate c is achievable within a predefined transmission range, i.e., $c_{ij} = c_{ji} = c$ for all $i \neq j$. We also assume symmetric link reliability, i.e., $\gamma_{ij} = \gamma_{ji}$ for all $i \neq j$, due to the line-of-sight transmissions with narrow beam divergence. There is connectivity between two BS's if the link reliability is larger than a threshold γ_{th} .

We adopt the *log-normal* model to characterize FSO link reliability under turbulent atmosphere [10], [11]. The link reliability γ_{ij} is the probability that the intensity of received signal I exceeds a threshold I_{th} , which can be computed using the *error function* $\text{erf}(\cdot)$ as $\gamma_{ij} = \Pr\{I \geq I_{th}\} = \frac{1}{2} - \frac{1}{2} \text{erf}\left(\frac{\ln(I_{th}/I_0)}{2\sqrt{2}\sigma_X}\right)$, where I_0 is the received average intensity without turbulence. The standard deviation σ_X^2 can be approximated as $\sigma_X^2 = 0.30545 (2\pi/\lambda)^{7/6} C_n^2(L) z^{11/6}$, where λ is the wavelength, $C_n^2(L)$ is the *index of refraction structure parameter* with constant altitude L , and z is the distance. We assume the set of edges satisfying $\gamma_{ij} \geq \gamma_{th}$ forms the *candidate link set* \mathcal{E}_c for constructing the FSO network topology.

B. Performance Measures

1) *Network-wide Average Load (L)*: We adopt multipath routing for load balancing, where a flow f_{sd} may be split into multiple subflows. Let f_{ij}^{sd} be the subflow passing through a link (i, j) . We have the *flow-conservation condition*:

$$\sum_{j=1}^n f_{ij}^{sd} - \sum_{j=1}^n f_{ji}^{sd} = \begin{cases} f_{sd}, & i = s, \text{ for all } i \in V \\ -f_{sd}, & i = d, \text{ for all } i \in V \\ 0, & \text{otherwise, for all } i \in V. \end{cases} \quad (1)$$

Considering all the $s-d$ pairs, the average traffic load λ_{ij} and the link utilization ρ_{ij} are $\lambda_{ij} = \sum_{s,d \in V} f_{ij}^{sd}$ and $\rho_{ij} = \lambda_{ij}/c_{ij} < 1$. For the link to be stable, we have $\rho_{ij} < 1$, for all $(i, j) \in E$. We define the *network-wide average load* L as

$$L \cong (1/\lambda) \cdot \sum_{(i,j) \in E} \lambda_{ij}, \quad (2)$$

where $\lambda = \sum_{s,d \in V} f_{sd}$, i.e., the sum of total traffic demands. Note that when a packet is forwarded, it is counted multiple times in L . When all the $s-d$ traffic are transmitted through direct links, L achieves its minimum value 1.

2) *Network-wide Average Delay (T_1, T_2)*: We model each link $(i, j) \in E$ as a general queueing system with average input rate λ_{ij} and service capacity c_{ij} . The average delay incurred at the link depends on the traffic auto-correlation structure. When the traffic constantly exhibits *short-range dependent* (SRD) characteristics (e.g., VoIP traffic), we can model the link queueing delay with an exponential distribution with parameter $c_{ij} - \lambda_{ij}$. Applying Little's formula, the

$$\text{minimize } L/T_1/T_2 \quad (5)$$

subject to:

$$\sum_{i=1}^n \sum_{j=i}^n x_{ij} = m_c, \sum_{i=1}^n \sum_{j=i}^n y_{ij} = m, \text{ for } i, j \in V \quad (6)$$

$$y_{ij} \leq x_{ij}, x_{ij} = x_{ji}, y_{ij} = y_{ji}, \text{ for } i, j \in V \quad (7)$$

$$\theta_i^l \leq \sum_{j=1}^n y_{ij} \leq \theta_i^u, \text{ for } i \in V \quad (8)$$

$$0 \leq f_{ij}^{sd} \leq y_{ij} f_{sd}, \text{ for } i, j, s, d \in V \quad (9)$$

$$\lambda_{ij} = \sum_{s,d \in V} f_{ij}^{sd} \leq c_{ij}, \text{ for } (i, j) \in \mathcal{E}_c \quad (10)$$

$$\text{flow conservation constraint (1)} \quad (11)$$

$$x_{ij} \in \{0, 1\}, y_{ij} \in \{0, 1\}, \text{ for } i, j \in V. \quad (12)$$

network-wide average delay T_1 can be computed as

$$T_1 \cong (1/\lambda) \cdot \sum_{(i,j) \in E} [\lambda_{ij}/(c_{ij} - \lambda_{ij})]. \quad (3)$$

For *long-range-dependent* (LRD) traffic (e.g., data traffic), we can model each link as a *fractional Brownian motion* (fBm) queueing system, where the queue length has a heavy-tailed *Weibull* distribution [12]. Applying Little's formula, the network-wide average delay T_2 is [8]

$$T_2 = \tau \cdot \sum_{(i,j) \in E} [\lambda_{ij}/(c_{ij} - \lambda_{ij})^{2H}]^{\frac{1}{2-2H}}. \quad (4)$$

where $\kappa(H) = H^H(1-H)^{1-H}$, $H \in [0.5, 1)$ is the Hurst parameter, a is the index of dispersion, and $\tau = (1/\lambda) \cdot \Gamma(1 + \frac{1}{2-2H}) [2\kappa^2(H)a]^{\frac{1}{2-2H}}$.

C. Problem Statement

Without loss of generality, we consider the case of fixed BS's. The atmospheric condition can be known through weather forecast and past experiences. Then we can evaluate the edge reliabilities and determine the candidate edge set \mathcal{E}_c . The number of links in the FSO network is upper bounded by $m_c = |\mathcal{E}_c|$. The problem is to select m links from m_c candidate edges to form a mesh topology. In addition, we also determine multipath routing for the $s-d$ flows, such that either the network-wide average load L or the network-wide average delay T_1 or T_2 is minimized. Define the following index variables for each link $(i, j) \in V$ as

$$x_{ij} = \begin{cases} 1, & \text{if } (i, j) \in \mathcal{E}_c \\ 0, & \text{otherwise.} \end{cases} \quad y_{ij} = \begin{cases} 1, & \text{if } (i, j) \text{ is chosen} \\ 0, & \text{otherwise.} \end{cases}$$

We have $x_{ij} \geq y_{ij}$ for all $(i, j) \in V$. The problem of joint topology design and load balancing, denoted as Problem OPT-TDLB, can be formulated as in (5) ~ (12).

D. Discussions

In Problem OPT-TDLB, the optimization variables include binary variables $\mathbf{y} = \{y_{ij} \mid \forall i, j \in V\}$ and continuous variables $\mathbf{f} = \{f_{ij}^{sd} \mid \forall i, j, s, d \in V\}$. With objective function L , Problem OPT-TDLB(L) is a mixed integer linear programming (MILP) problem. With objective function T_1 or T_2 , the corresponding Problems OPT-TDLB(T_1) and OPT-TDLB(T_2) are mixed integer nonlinear programming (MINLP) problems. Constraints (6) ~ (8) are for edge selection and topology design, while constraints (9) ~ (11) are for multipath routing and load balancing. In (8), θ_i^u is the maximum degree (enforced by some cost constraints), and θ_i^l is the minimum

TABLE I
HEURISTIC ALGORITHM FOR PROBLEM OPT-TDLB.

1:	Initialization:
2:	Read \mathcal{N} , \mathbf{F} , and C_n^2 ;
3:	Derive the candidate edge matrix \mathbf{X} ;
4:	Derive the minimum multihop matrix \mathbf{H} ;
5:	Derive the minimum traffic demand matrix \mathbf{F}' ;
6:	Select edges to form adjacency matrix $\mathbf{Y}(\subset \mathbf{X})$;
7:	Choose the largest unselected f_{sd}' ;
8:	Find k multiple shortest paths for $s-d$;
9:	Insert unselected edges into \mathbf{Y} ;
10:	if $\{m$ edges have been selected $\}$ Break with \mathbf{Y} ;
11:	else Goto Step 7;
12:	Solve l -OPT-TDLB($L/T_1/T_2$) for \mathbf{f} with given topology \mathbf{Y} ;
13:	Iteration:
14:	Perturb the current topology:
15:	Execute the branch exchange algorithm to get $\hat{\mathbf{Y}}$;
16:	Delete the link with the smallest load/delay;
17:	Connect the two nodes with the maximum load/delay;
18:	if $\{\text{no further branch exchange}\}$ Terminate with (\mathbf{Y}, \mathbf{f}) ;
19:	Multipath routing for load balancing:
20:	Solve l -OPT-TDLB($L/T_1/T_2$) for $\hat{\mathbf{f}}$ with given topology $\hat{\mathbf{Y}}$;
21:	if $\{(\hat{\mathbf{Y}}, \hat{\mathbf{f}}) < (\mathbf{Y}, \mathbf{f})\}$ $(\mathbf{Y}, \mathbf{f}) = (\hat{\mathbf{Y}}, \hat{\mathbf{f}})$;
22:	if $\{\text{maximum number of iterations has not been reached}\}$
23:	Go to Step 14;
24:	else Terminate with (\mathbf{Y}, \mathbf{f}) ;

degree for BS i , which is required to support the incoming and outgoing traffic from the BS, as

$$\theta_i^l = \max \left\{ \left\lceil \sum_{d=1}^n \lambda_{id} / c \right\rceil, \left\lceil \sum_{s=1}^n \lambda_{si} / c \right\rceil \right\}. \quad (13)$$

In [8], we show how to obtain LP relaxations for Problem OPT-TDLB. We first allow binary variables y_{ij} 's to take real values in $[0, 1]$, thus obtaining an LP relaxation for Problem OPT-TDLB(L), denoted as l -OPT-TDLB(L). Problem OPT-TDLB(T_1) can be linearized by defining substitution variables for the nonlinear objective function and introducing RLT *bound factor constraints* [8], [9], in addition to relaxing y_{ij} 's. The corresponding LP relaxation is called l -OPT-TDLB(T_1). Finally, we define substitution variables and apply RLT *polyhedral outer approximation* [9] to linearize the nonlinear objective function in Problem OPT-TDLB(T_2), and relax y_{ij} 's, to obtain an LP relaxation l -OPT-TDLB(T_2).

These LP relaxations can be solved with an LP solver, but the resulting solutions are usually infeasible due to the relaxed constraints and objective functions. Nevertheless, such infeasible solutions provide useful lower bounds to the original problems OPT-TDLB($L/T_1/T_2$). Note that the relaxations are made for the y_{ij} 's and the objective functions; the original constraints (6) ~ (11) are preserved in the LP relaxations.

III. FAST HEURISTIC ALGORITHM

A. Overview

In this section, we present a fast heuristic algorithm for Problem OPT-TDLB. In [8], we observe that joint optimization with binary variables y_{ij} 's and continuous variables f_{ij}^{sd} 's requires long execution time. To speed up computation, we first determine the network topology to minimize average network load, and then solve the multipath routing problem for load balancing. To further improve the optimality, we iteratively perturb the topology and then compute new network flows \mathbf{f} , until some termination criterion is met.

The pseudo-code of the algorithm is given in Table I. It consists of three parts: (i) initial topology design, (ii) multipath

routing for load balancing, and (iii) topology perturbation. The algorithm iteratively perturbs the current topology by deleting and inserting FSO links, and computes the network flow for the $s-d$ pairs based on the new topology. The network flow is derived by solving the LP relaxations l -OPT-TDLB($L/T_1/T_2$) with fixed y_{ij} values, thus the computation is very fast. If the new topology and network flow produce a better objective value for the original problem, the new solution will replace the existing one. Thus the objective function will be progressively improved over iterations. The algorithm terminates when no further perturbation can be made or when a prescribed maximum number of iterations is reached. The three key components of the algorithm are described in detail in the remainder of this section.

B. Initial Topology Design

As discussed, the proposed algorithm iteratively improves the topology and network flow starting from an initial topology. A properly designed initial topology could speed up the convergence and achieve better solutions. In the following, we show how to choose m links from the candidate set \mathcal{E}_c to form an initial topology. The initial topology is represented by an adjacency matrix $\mathbf{Y} = [y_{ij}]$, a 0-1 $n \times n$ matrix, where $y_{ij} = 1$ if and only if link (i, j) is included in the topology.

1) *Prerequisite Information:* Assume the traffic matrix \mathbf{F} is known. Let \mathcal{N} denote node information, such as node ID and location, and C_n^2 represent FSO channel information. Then we can derive the reliability of the FSO links and the set of candidate links \mathcal{E}_c . The candidate links then form a topology with adjacency matrix $\mathbf{X} = [x_{ij}]$. If link $(i, j) \in \mathcal{E}_c$ (i.e., $\gamma_{ij} \geq \gamma_{th}$), we have $x_{ij} = 1$; otherwise $x_{ij} = 0$. The total number of candidate links is $m_c = |\mathcal{E}_c| = \sum_{i=1}^n \sum_{j=1}^n x_{ij} / 2$. The minimum degree of each node can be derived as in (13) using traffic matrix \mathbf{F} and link capacity c .

2) *Minimum Traffic Matrix \mathbf{F}' :* For a given topology, the traffic matrix \mathbf{F} is usually supported with both single-hop and multihop flows. If nodes s and d are not one-hop neighbors, the flow f_{sd} will be forwarded multiple times through multihop paths. The network traffic load associated with f_{sd} is thus proportional to the hop-counts of the $s-d$ paths. We aim to have small hop-counts for $s-d$ pairs with high traffic demands when determining the initial topology.

We consider both single-hop and multihop traffic patterns by constructing a *minimum traffic matrix*, denoted as \mathbf{F}' . Assume K disjoint paths $\{P_{sd}^k\}_{k=1, \dots, K}$ between nodes s and d . The $s-d$ traffic subflow on path P_{sd}^k is denoted as f_{sd}^k , and the length (or, hop-count) of P_{sd}^k is denoted as l_{sd}^k , for $1 \leq k \leq K$. We have $f_{sd} = \sum_{k=1}^K f_{sd}^k$. For the total traffic load associated with flow f_{sd} , denoted as λ_{sd}' , we have

$$\lambda_{sd}' = \sum_{k=1}^K l_{sd}^k f_{sd}^k \geq \min \{l_{sd}^k \mid k = 1, \dots, K\} f_{sd} \cong f_{sd}'.$$

We next introduce a technique to compute $\min_k \{l_{sd}^k\}$, which is based on the adjacency matrix \mathbf{X} defined earlier. We have the following fact from graph theory.

Fact 1: Let \mathbf{A} be graph G 's adjacency matrix. The number of walks from vertex s to d in G with length l is $[\mathbf{A}^l]_{sd}$.

From Fact 1, the hop-count of the shortest path between nodes s and d can be found by identifying the smallest l , such that $[\mathbf{A}^l]_{sd} > 0$. That is, $\min_k \{l_{sd}^k\} = l$, if $[\mathbf{A}^l]_{sd} > 0$ and $[\mathbf{A}^h]_{sd} = 0$, for all $1 \leq h < l$. Once the hop-counts of the

shortest paths are obtained, we can derive the minimum traffic matrix \mathbf{F}' , with elements $f'_{sd} = \min_k \{l_{sd}^k\} \cdot f_{sd}$.

3) *Link Selection*: When the minimum traffic matrix is obtained, we next construct the initial topology based on \mathbf{F}' and the adjacency matrix \mathbf{X} representing all the candidate links. To minimize the network-wide average load L , we choose the links that are on the shortest paths for all s - d pairs. We examine the elements of \mathbf{F}' in nonincreasing order, starting from the largest one, and choose links from \mathcal{E}_c (or, \mathbf{X}) to insert into \mathbf{Y} until m edges are selected. If two nodes are adjacent according to \mathbf{X} , the direct link will be inserted. In the case of multihop flows, multiple paths will be selected for rich connectivity and load balancing.

C. Multipath Routing for Load Balancing

When the topology \mathbf{Y} is fixed, the binary optimization variables \mathbf{y} are all determined. The LP relaxations l -OPT-TDLB($L/T_1/T_2$) now only have continuous subflow variables \mathbf{f} . We then solve Problem l -OPT-TDLB($L/T_1/T_2$) again with all the y_{ij} 's determined, to obtain multipath routing of the s - d flows \mathbf{f} for this given topology.

To obtain the LP relaxations l -OPT-TDLB($L/T_1/T_2$), the following two types of constraints are relaxed [8]: (i) the binary variables y_{ij} are allowed to take values in $[0, 1]$, and (ii) the nonlinear objective functions T_1 and T_2 are linearized with additional RLT bound factor constraints and polyhedral out approximation constraints [9]. Except for these two, all the other constraints (6) ~ (11) are preserved during the procedure. When y_{ij} 's are fixed, the \mathbf{f} solved from the LP relaxations are also feasible to the original problem OPT-TDLB($L/T_1/T_2$), i.e., solution $\{\mathbf{y}, \mathbf{f}\}$ is feasible to both the LP relaxation and the original problem. Plugging it into problems OPT-TDLB($L/T_1/T_2$) yields an upper bound.

Clearly the optimality of the feasible solution $\{\mathbf{y}, \mathbf{f}\}$ depends on the topology. It is possible to obtain a better solution with a different topology. Thus, we next perturb the current topology \mathbf{Y} to get a new topology $\hat{\mathbf{Y}}$ to further improve the current solution $\{\mathbf{y}, \mathbf{f}\}$, as discussed in the following.

D. Topology Perturbation

We adopt a *branch exchange* algorithm similar to that in [13] for topology perturbation. This algorithm deletes a link from the current topology \mathbf{Y} . It then chooses a remaining link from the candidate set \mathcal{E}_c and inserts it into the reduced topology. When deleting and inserting links, the degree constraints (6) and (8) should be satisfied. The performance of this algorithm depends on the decision rules for link deletion and insertion. We adopt the following strategies in our algorithm.

- Link deletion: for objective function L , we delete the link with the minimum load λ_{ij} . For objective functions T_1 and T_2 , we delete the link with the minimum delay. That is, delete the least used link from the current topology.
- Link insertion: for objective function L , let the sum load at node i be $\lambda_i = \sum_{j=1}^n y_{ij} \cdot \lambda_{ij}$. We insert the link (i, j) with $\arg \max \{x_{ij} \cdot (\lambda_i + \lambda_j) \mid i \neq j, i, j \in V, y_{ij} = 0\}$. For objective functions T_1 and T_2 , let the sum delay at node i be $t_i = \sum_{j=1}^n y_{ij} \cdot t_{ij}$. We insert the link (i, j) with $\arg \max \{x_{ij} \cdot (t_i + t_j) \mid i \neq j, i, j \in V, y_{ij} = 0\}$. If no improvement is achieved by inserting such a link (i, j) , the link that achieves the second largest value will be inserted in the next iteration, and so forth.

The algorithm iteratively perturbs the topology and computes the network flows with the new topology. It terminates with solution $\{\mathbf{y}, \mathbf{f}\}$ when the maximum number of iterations is achieved or when no further perturbation can be made.

E. Discussions

The heuristic algorithm can be executed independently to solve Problem OPT-TDLB for an underlying FSO network, as discussed above. In addition, it is worth noting that it is also complementary to the RLT-based branch-and-bound algorithm developed in [8]. In practice, the branch-and-bound algorithm can be executed at relatively large time intervals (e.g., when significant changes occur), to provide new topology \mathbf{y} and network flows \mathbf{f} with guaranteed optimality. On the other hand, the heuristic algorithm can keep running or be executed at more frequent intervals. It keeps on perturbing the topology generated by the branch-and-bound algorithm and computing new network flows, thus making the FSO network dynamically reconfigurable and adaptive to small timescale changes such as bad weather or fluctuation in the traffic matrix \mathbf{F} .

IV. SIMULATION STUDIES

In this section, we present simulation studies on the performance of the proposed heuristic algorithm. We first examine its optimality performance by comparing with a lower bound [8], and then examine the computational cost. The proposed algorithm was implemented in MATLAB ver 7.4.0 for manipulating matrices and solving the LP relaxations. The codes are executed on a standard PC with a Core Duo 2.20 GHz processor and 2 GB memory.

In the simulations, n BS's are randomly deployed in a rectangular region. We present simulation results for the cases $n = 7$ and $n = 15$. Assume the traffic matrix \mathbf{F} is given, which is randomly generated for the s - d pairs [6]. Each s - d flow f_{sd} ranges from 0 to 40% of the FSO link capacity. The constraints of minimum node degrees are determined from \mathbf{F} as in (13). Link connectivity is determined by the link reliability, which is derived using the FSO channel model given in Section II-A. Then the x_{ij} 's are known and the candidate link set \mathcal{E}_c is found. We use reliability threshold $\gamma_{th} = 99.99\%$ when determining the candidate set. For LRD traffic model, the index of dispersion a is set to a half of the link capacity and Hurst parameter H is chosen to be 0.7.

We first examine the optimality performance of the proposed heuristic algorithm. Since the parameters (such as traffic matrix and node location) are not provided in prior works [5], [6], it is non-trivial to truthfully reproduce their results. Our strategy is to compare the heuristic algorithm with the branch-and-bound algorithm developed in [8]. Specifically, the lower and upper bounds provided by the branch-and-bound algorithm are good indicators of the global optimal for the joint topology design and load balancing problem. The gap between the heuristic solution and the branch-and-bound algorithm solution provides an upper bound on the optimality gap achieved by the heuristic algorithm.

In Fig. 1, we plot the network-wide average traffic load L and average delays T_1/T_2 for the 7-node network. In the figures, the markers correspond to the iterations for the algorithms. When the objective function is L , the branch-and-bound algorithm terminates after 12 iterations, achieving an optimality gap of 0.33% (i.e., the feasible solution it produces

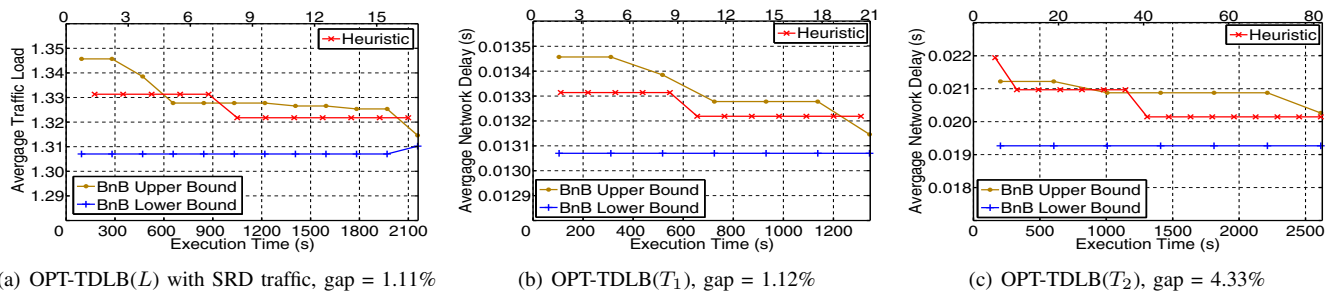


Fig. 1. Performance of the heuristic algorithm for the 7-node network, comparing to the results of the branch-and-bound algorithm [8].

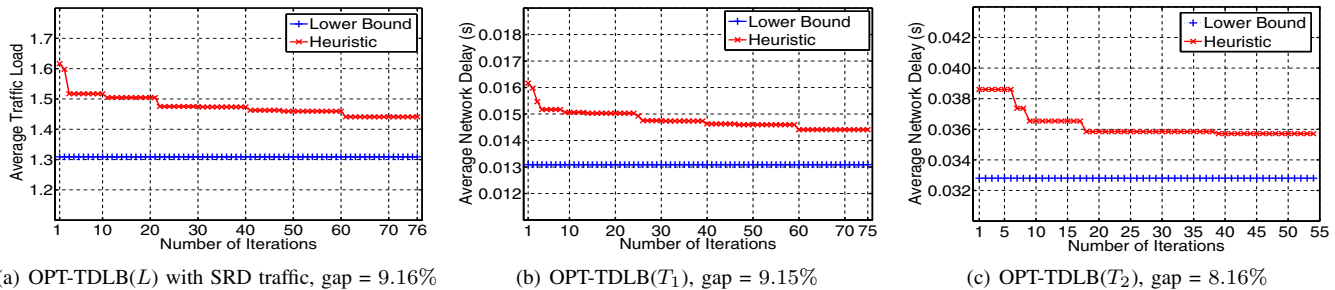


Fig. 2. Performance of the heuristic algorithm for the 15-node network.

is within 0.33% range of the global optimal). The heuristic algorithm also terminates after 12 iterations, achieving an optimality gap of 1.11%. When the objective function is T_1 , the branch-and-bound algorithm achieves optimality gap 0.58% in 7 iterations, while the heuristic achieves optimality gap 1.12% in 12 iterations. When the objective function is T_2 , the branch-and-bound algorithm achieves optimality gap 4.91% in 7 iterations, while the heuristic achieves optimality gap 4.33% in 16 iterations.

The optimality of the heuristic solutions is shown in Fig. 2 for the 15-node network. We also plot the lower bound for comparison purpose. It takes 76, 75, and 55 iterations, respectively, for the heuristic algorithm to achieve optimality gaps of 9.16%, 9.15%, and 8.16%, for problems OPT-TDLB(L), OPT-TDLB(T_1), and OPT-TDLB(T_2), respectively.

We next examine the computation time of the proposed algorithm. In Fig. 1, the x -axis on the top marks the execution time of the iterations of the heuristic algorithm, while the x -axis at the bottom are for the execution time of the iterations of the branch-and-bound algorithm. It can be seen that the heuristic algorithm is computationally efficient compared to the branch-and-bound algorithm. For example, in Fig. 1(a), the heuristic algorithm achieves an optimality gap of 1.11% in about 16 s, while the branch-and-bound algorithm achieves an optimal gap of 0.33% in about 2,100 s. The same observation is made in all our simulation studies.

V. CONCLUSION

In this paper, we presented a fast heuristic algorithm for joint optimization of FSO network topology and load balancing. The proposed algorithm iteratively perturbs the current topology, and solves the optimal multipath routing problem to obtain network flows for the new topology, thus progressively improving the solution performance. Our simulation results show that the proposed algorithm can achieve an optimality

gap close to that achieved by the branch-and-bound algorithm at significantly reduced computation time.

ACKNOWLEDGMENT

This work is supported in part by the NSF under Grants ECCS-0802113, IIP-0738088 and IIP-1032002.

REFERENCES

- [1] V. W. Chan, "Free-space optical communications," *IEEE/OSA J. Lightw. Technol.*, vol. 24, no. 12, pp. 4750–4762, Dec. 2006.
- [2] I. K. Son and S. Mao, "Design and optimization of a tiered wireless access network," in *Proc. IEEE INFOCOM'10*, San Diego, CA, Mar. 2010.
- [3] F. Liu, U. Vishkin, and S. Milner, "Bootstrapping free-space optical networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 12, pp. 13–22, Dec. 2006.
- [4] P. C. Gurumohan and J. Hui, "Topology design for free space optical networks," in *Proc. IEEE ICCN'03*, Oct. 2003, pp. 576–579.
- [5] A. Desai and S. Milner, "Autonomous reconfiguration in free-space optical sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 8, pp. 1556–1563, Aug. 2005.
- [6] A. Kashyap, K. Lee, M. Kalantari, S. Khuller, and M. Shayman, "Integrated topology control and routing in wireless optical mesh networks," *Computer Networks*, vol. 51, no. 15, pp. 4237–4251, 2007.
- [7] D. Mosk-Aoyama, "Maximum algebraic connectivity augmentation is np-hard," *Oper. Res. Lett.*, vol. 36, no. 6, pp. 677–679, 2008.
- [8] I.-K. Son and S. Mao, "A reformulation-linearization technique-based approach to topology design and load balancing for FSO networks," in *Proc. IEEE GLOBECOM 2010*, Miami, FL, Dec. 2010.
- [9] S. Kompella, S. Mao, Y. Hou, and H. Sherali, "On path selection and rate allocation for video in wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 212–224, Feb. 2009.
- [10] X. Zhu and J. M. Kahn, "Free-space optical communication through atmospheric turbulence channels," *IEEE Trans. Commun.*, vol. 50, no. 8, pp. 1293–1300, Mar. 2003.
- [11] M. Al-Habash, L. Andrews, and R. Philips, "Mathematical model for the irradiance probability density function of a laser beam propagating through turbulent media," *Society of Photo-Optical Instrumentation Engineers*, vol. 40, no. 8, pp. 1554–1562, Aug. 2001.
- [12] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 953–962, Aug. 1995.
- [13] E. Miguez, J. Cidras, E. Diaz-Dorado, and J. L. Garcia-Dornelas, "An improved branch-exchange algorithm for large-scale distribution network planning," *IEEE Transactions on Power Systems*, vol. 17, no. 4, pp. 931–936, Nov. 2002.