Cost of Not Splitting in Routing: Characterization and Estimation

Meng Wang, Chee Wei Tan, Weiyu Xu, Ao Tang

***Abstract*—This paper studies the performance difference of joint routing and congestion control when either single-path routes or multipath routes are used. Our performance metric is the total utility achieved by jointly optimizing transmission rates using congestion control and paths using source routing. In general, this performance difference is strictly positive and hard to determine - in fact an NP-hard problem. To better estimate this performance gap, we develop analytical bounds to this "cost of not splitting" in routing. We prove that the number of paths needed for optimal multipath routing differs from that of optimal single-path routing by no more than the number of links in the network. We provide a general bound on the performance loss, which is independent of the number of source-destination pairs when the latter is larger than the number of links in a network. We also propose a vertex projection method and combine it with greedy branch-and-bound algorithm to provide progressively tighter bounds on the performance loss. Numerical examples are used to show the effectiveness of our approximation technique and estimation algorithms.**

I. Introduction

Routing is one of the key network functions in communication networks such as the Internet. It selects paths for traffic to flow from all the sources to their respective destinations. Even though there are proposals to allow flexible multipath routing in the Internet [17], the current Internet Protocol (IP) within an Autonomous System (AS) , e.g., the Open Shortest Path First (OSPF) protocol, primarily uses single-path routing where one user (source-destination pair) uses only one path from the source to the destination, with the exception that traffic may be split evenly among equal-cost paths.

Recently, there has been an interest to consider cross-layer resource allocation, where routing paths and congestion-controlled transmission rates by Transmission Control Protocol (TCP) are jointly optimized ([10], [24], [25], [28]). More specifically, assuming every user has a utility function that depends on its total transmission rate, one seeks to maximize the total utilities of all users subject to the link capacity constraints.[1] This problem is analytically tractable if users can use all available paths because allowing source-based multipath routing makes the problem convex and admits an elegant optimality characterization. When each user optimizes its transmission rate over only one out of all available paths, i.e., using source-based single-path routing, this combinatorial problem is however nonconvex and known to be NP-hard [28].

This paper focuses on a key question: as compared with multipath routing, how is the performance affected in terms of the aggregate utility by restricting to single-path routing? Or what is the "cost of not splitting"? It can guide the decision on whether to support multipath routing with flexible splitting which is more expensive to support, since single-path routing has a smaller overhead. Allowing users to use all possible paths does not necessarily increase the network utility significantly as we show that, for any network topology, the number of paths needed for optimal multipath routing differs from that of optimal single-path routing by no more than the number of links in the network. We also provide analytical bounds as well as algorithms to estimate the performance gap between multipath and single-path routing.

Our problem belongs to the general multicommodity flow category [20]. Though our formulation is closely related to that in [28], there are other formulations in the multicommodity flow category. In particular, there has been recent study on "unsplittable flows" [3], [7], [16], [18], [19], [26]. We remark that even though formulations may differ, the intrinsic difficulty is the same when they are considered as a class of combinatorial optimization problems. Therefore, we expect our work will also be useful to other related multicommodity problems.

The paper is organized as follows. After introducing the model and notations in Section II, we give graph-theoretic characterizations of the existence of the positive performance gap between single-path and multipath routing in Section III. We then derive a general upper bound on the performance loss based on the analysis of the solution set of the multi-path problem in Section IV. The analysis motivates a vertex projection method to find a near optimal single-path solution which is discussed in Section V-A. In Section V-B, we further refine our estimation by combining vertex projection with a greedy branch-and-bound technique. Numerical examples are provided in Section VI to demonstrate the effectiveness of our estimation. We conclude the paper in Section VII.

II. Model and Notations

A network consists of a set of L uni-directional links with finite capacities $c = (c_l, l = 1, ..., L)$, and supports a set of N source-destination pairs or users, indexed by i. There are K^i acyclic paths for user i and is represented by a $L \times K^i$ matrix R^i, where $R^i_{lk} = 1$ if path k of user i uses link l, and $R^i_{lk} = 0$ otherwise. The overall routing matrix is given by

$$R = [\ R^1 \quad R^2 \quad ... \quad R^N\ ].$$

For example, as shown in Fig.1, a seven-link network supports two users, each of which has two possible paths to choose from, and the corresponding routing matrices are

$$R^1 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T,$$

M. Wang, W. Xu and A. Tang are with Cornell University, Ithaca, NY. C. W. Tan is with City University HK.

Partial and preliminary results have appeared in [29].

[1] Other formulations are possible, e.g., minimizing the total link cost while satisfying the traffic demand between the sources and destinations [31].

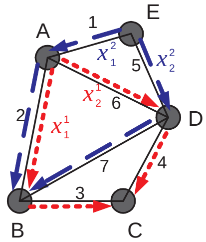


Fig. 1. A network of seven links supporting two users. User one: A to C (red dotted line), and user two: E to B (blue dashed line).

$$R^2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}^T.$$

For every i, define a $K^i \times 1$ vector x^i with the rate x^i_k of path k of user i as the kth entry of x^i. The total rate of user i is $\|x^i\|_1 = \sum_{k=1}^{K^i} x^i_k$. Let a $\sum_i K^i \times 1$ vector x represent the complete bandwidth allocation

$$x = [\ (x^1)^T \quad (x^2)^T \quad ... \quad (x^N)^T\ ]^T.$$

Each user i has a utility function U^i that is a function of its total transmission rate $\|x^i\|_1$. We assume U^i to be strictly increasing and concave, which is the case for most TCP algorithms [21]. Let $U = (U^i, i = 1, ..., N)$. We call (c, R, U) a *network*.

The joint congestion control and multipath routing problem is to maximize the aggregate network utility by allocating the transmission rates for all users over all possible paths subject to link capacity constraints. We assume fairness in the sense that every user receives a positive rate, i.e., $\|x^i\|_1 > 0$ for any user i. It can be formulated as a convex optimization problem:

$$\begin{aligned} \max_{x \geq 0} \quad & \textstyle\sum_i U^i(\|x^i\|_1) \\ \text{s.t.} \quad & Rx \leq c. \end{aligned} \tag{1}$$

Compared with problem (1), there are additional constraints in the joint flow control and single-path routing problem, namely each user only uses one path out of its finite set of possible paths. Let $\|x^i\|_0$ denote the number of nonzero entries of the vector x^i.[2] Then the single-path problem can be formulated as the following optimization problem:

$$\begin{aligned} \max_{x \geq 0} \quad & \textstyle\sum_i U^i(\|x^i\|_1) \\ \text{s.t.} \quad & Rx \leq c, \\ & \|x^i\|_0 = 1 \quad \forall\, i. \end{aligned} \tag{2}$$

Unlike (1), (2) is nonconvex and in fact NP-hard due to the cardinality constraints.

Let opt_M and opt_S denote the values of (1) and (2) respectively. Then, $opt_M - opt_S$ can be interpreted as a measure of performance loss due to the additional single-path routing constraints, or the "cost of not splitting". Moreover, it was shown in [28] that the dual problem of (2) has the same value as (1). Therefore, the duality gap of (2) is precisely the performance loss of utilizing only one out of a finite choice of paths by each user. We will refer to $opt_M - opt_S$ as the performance gap in this paper.

[2]Our notation for the cardinality of a vector is the same as that used in the compressed sensing literature, and is commonly known as the ℓ_0 norm .

Throughout the paper, we assume $\sum_i K^i > N + L$, i.e., the total number of paths that can be used by all users is greater than the sum of the number of users and the number of links. This is the case when the network is large and each user has many available paths. The key notation used is summarized in Table I.

TABLE I
SUMMARY OF KEY NOTATION

Notation	Meaning
$\tilde{x}$	A feasible multipath allocation
x^*	An optimal multipath allocation
$\hat{x}$	A vertex of the solution set Q of the multipath problem
x'	A feasible single-path allocation projected from $\hat{x}$
$\bar{x}$	Optimizer of the fixed routing congestion control problem using the path configuration of x'

III. Sufficient Conditions for Positive Performance Gap

A previous work in [28] focuses on the cases when the performance gap is zero. However, one may expect the performance gap to be positive in the general case. In this section, we quantify this intuition by providing sufficient conditions for the existence of a positive performance gap. For many cases, there exists a positive performance loss between an optimal multipath allocation and an optimal single-path allocation. This motivates the need to estimate this difference in the later sections.

Note that the existence of the positive performance gap implies that every optimal solution to (1) has more than N paths and vice versa. If an optimal solution to (1) uses only N paths, then it is also an optimal solution to (2), therefore the performance gap is zero. Conversely, if every solution to (1) has more than N paths, then the network utility achieved by any single-path allocation is strictly less than the value of (1), therefore the performance gap is positive. Based on this observation, if under certain conditions every optimal solution to (1) has more than N paths, then we can conclude that the performance gap is positive.

To facilitate the presentation of Theorem 1, we order all paths and let P^i be the set of indices of paths of user i. We call two paths **disjoint** if there does not exist a link that belongs to both paths.

Theorem 1. *For a given network (c, R, U), if there exists a set of paths P' with cardinality $|P'| = N + 1$ such that $P' \cap P^i \neq \Phi$ for all user i, and every pair of paths in P' are disjoint, then there exists a positive performance gap between (1) and (2).*

Proof: We prove the statement by arguing that for every single-path allocation, there exists a multipath allocation that has a strictly higher network aggregate utility than that of this single-path allocation. Thus, no single-path allocation could be a solution to (1) , which implies a positive performance gap.

Let x denote a feasible single-path allocation, thus, $\|x^i\|_0 = 1$ for all i, and $Rx \leq c$. Define $\epsilon_1 = \min_{\{l:(Rx)_l < c_l\}}(c_l -$

$(Rx)_l)$, which is the minimum redundant link capacity on links that are not fully utilized under x. If every link is fully utilized under x, i.e., the set $\{l : (Rx)_l < c_l\}$ is empty, we set ϵ_1 to be $+\infty$. Let H be the set of indices of paths used by x, i.e., $x_j > 0$ for every j in H. Let $\epsilon_2 > 0$ be some positive constant that is strictly less than the transmission rate for every user under x, i.e., $\epsilon_2 < \min_{j \in H} x_j$. Let $\epsilon = \min(\epsilon_1, \epsilon_2) > 0$. y is a single-path allocation obtained from x by reducing the rate x_j of every path j in H by ϵ, then for every link l, one can check $c_l - (Ry)_l \geq \epsilon > 0$. Now we increase the rate y_j of every path j in P' by ϵ, and let $\tilde{x}$ denote the resulting rate allocation. Since $|P'| = N+1$, $\tilde{x}$ is a multipath rate allocation. Since every two paths in P' are disjoint, $(R\tilde{x})_l - (Ry)_l \leq \epsilon$ for every link l. Therefore we have $R\tilde{x} \leq c$. Furthermore, from the definition of P', we know that there exists exactly one user, say user i, that has two paths in P', and every other user has only one path in P'. Thus, $\|\tilde{x}^i\|_1 = \|x^i\|_1 + \epsilon$, and $\|\tilde{x}^j\|_1 = \|x^j\|_1$ for every other user j. Thus, the network aggregate utility under $\tilde{x}$ is strictly greater than that under x, which completes the proof. ■

Note that the paths in P' as defined in Theorem 1 are not necessarily used by an optimal multipath allocation, but the existence of such a set of paths guarantees a positive performance gap. Theorem 1 provides a sufficient condition on paths for the existence of a positive performance gap, we next provide another sufficient condition based on the links in the network. We need two definitions to proceed.

Definition 1 ([6]). *Given a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of links, an s-t cut $C = (S, T)$ is a partition of V such that $s \in S$ and $t \in T = V - S$, where s and t are the source and the destination of one user.*

Definition 2. *Given an s-t cut $C = (S, T)$, let $O_S = \{(u, v) \in E \mid u \in S, v \in T\}$ be the set of outgoing links from S, and let $I_S = \{(v, u) \in E \mid u \in S, v \in T\}$ be the set of incoming links to S, the* ***algebraic value*** *$\mathcal{A}(C)$ of cut C is the number of outgoing links of S minus the number of incoming links of S, i.e., $\mathcal{A}(C) = |O_S| - |I_S|$, where $|\cdot|$ denotes the cardinality of a set.*

Theorem 2. *Given a network represented by a directed graph $G = (V, E)$ which supports N users, if there exists a user i (with s_i and t_i as the source and the destination) such that for every s_i-t_i cut $C = (S, T)$, $\mathcal{A}(C) \geq N + 1$ holds, then every optimal multipath allocation has positive flow rates on at least $N + 1$ paths.*

Proof: Suppose there exists an optimal multipath allocation x^* which has positive flow rates on at most N paths. Let E^* be the set of links that x^* transmits positive rates on. Let B be the set of links that are saturated by x^*, i.e. $(Rx^*)_i = c_i$ for every i in B. Let $G' = (V, E - B)$ be the reduced network by removing the links in B from G. Define S to be a subset of nodes in V such that for every node $u \in S$ there exists a path from s_i to u in the reduced network G', and for every node $v \notin S$, there does not exist a path from s_i to v in G'.

We consider two possibilities: $t_i \in S$ and $t_i \notin S$. If $t_i \in S$, then there exists an s_i-t_i path P in G' on which we can send positive flow rate. Thus we can send positive flow rate on P besides the flow allocation x^* and increase the network aggregate utility, which contradicts the fact that x^* is an optimal allocation.

If $t_i \notin S$, $C = (S, V - S)$ is an s_i-t_i cut, thus $\mathcal{A}(C) \geq N+1$ by assumption. Since x^* uses at most N paths, then the number of outgoing links from S that are also in E^* minus the number of incoming links to S that are also in E^* is at most N. Therefore, there exists an outgoing edge $e = (u, v)$ from S such that $u \in S$, $v \in V - S$ and $e \notin E^*$. Since $e \notin E^*$, then $e \in G'$. This contradicts the fact that there is no link from S to $V - S$ on graph G'. ■

Under the condition of Theorem 2, *every* optimal multipath solution uses at least $N + 1$ paths, thus, no single-path allocation could achieve the optimal multipath aggregate utility. Therefore, there always exists a positive performance gap between (1) and (2).

Theorem 2 requires $\mathcal{A}(C) \geq N + 1$ for all s_i-t_i cut C for some user i. For cases where users share the same source or the same destination, the condition of Theorem 2 can be further relaxed as follows.

Theorem 3. *A network $G = (V, E)$ supports N users, and all users have the same source s, and the destination of user i is t_i, $i = 1, 2, ..., N$. If for every set S such that $s \in S$ and $t_i \notin S$ for every user i, the algebraic value of cut $C = (S, V - S)$ satisfies $\mathcal{A}(C) \geq N + 1$, then there always exists a positive performance loss between (1) and (2).*

Similarly, if all users share the same destination t and user i has source s_i, $i = 1, 2, ..., N$, and for every set T such that $t \in T$ and $s_i \notin T$ for every user i, the algebraic value of cut $C = (V - T, T)$ satisfies $\mathcal{A}(C) \geq N + 1$, then there always exists a positive performance loss between (1) and (2).

Proof: We will prove the first half of the theorem and omit the second half, as it is the same as the first half by simply interchanging the source and the destination.

The proof technique is similar to that of Theorem 2. Suppose there exists an optimal multipath allocation x^* which has positive flow rates on at most N paths. Let E^* be the set of links that x^* transmits positive rates on. Let B be the set of links that are saturated by x^*. Let $G' = (V, E - B)$ be the reduced network by deleting the links in B from G. Define S to be a subset of nodes in V such that $\forall u \in S$ there exists a path from s to u in the reduced network G', and $\forall v \notin S$, there is no path from s to v in G'. Then there is no link from S to $V - S$ on graph G'.

Note that $t_i \notin S$ for all i, since otherwise there would exist an s-t_i path P in G' for some i, and we could increase the network aggregate utility of x^* by sending additional positive flow rate on P without violating capacity constraints. Since $t_i \notin S$ for all i, then the cut $C = (S, V - S)$ has an algebraic value $\mathcal{A}(C) \geq N + 1$ from the assumption. Since x^* uses at most N paths, then the number of outgoing links from S that are also in E^* minus the number of incoming links to S that are also in E^* is at most N. Therefore, there exists an outgoing edge $e = (u, v)$ from S such that $u \in S$, $v \in V - S$ and $e \notin E^*$. Since $e \notin E^*$, then $e \in G'$. This contradicts the fact that there is no link from S to $V - S$ on graph G'. ■

We need to mention that one implicit assumption for both Theorem 2 and Theorem 3 is that every user can use every path from its source to its destination as long as the path is on the graph. However, there may be cases when some user is not admitted to a certain path even though the path connects its source to its destination in the graph. For these cases, conditions in Theorem 2 and Theorem 3 need slight modification that we omit here. But Theorem 1 is applicable to all the cases.

Theorem 1, 2 and 3 do not impose any constraint on the link capacities. Note that even if none of the conditions in these theorems hold, the performance gap can still be positive under certain link capacity configurations.

IV. General Bound of Performance Gap

Section III establishes the existence of a positive performance gap. In this section, we obtain a general upper bound of the performance gap, and then show that the average cost of not splitting goes to zero asymptotically as the number of users goes to infinity. We start with analyzing the vertices of the solution set of (1), which turns out to be important for both obtaining a general bound of the performance gap and estimating the performance gap for a given network.

A. Vertices of Optimal Multipath Solution Set

In general, the multipath problem (1) may not admit a unique optimal solution. Among those solutions, we can find one that uses at most $L+N$ paths, as the following result shows, even though the total available paths can potentially be exponential in L or N.

Theorem 4. *Given an L-link network supporting N users, for any multipath allocation $\tilde{x}$, there exists a multipath allocation x using at most $L+N$ paths such that they both achieve the same aggregate utility.*

Proof: Given a multipath allocation $\tilde{x}$, consider the following nonempty and bounded polyhedron:

$$P = \{x \in \mathcal{R}^{\sum_i K^i} \mid x \geq 0,\ \mathbf{1}^T x^i = \mathbf{1}^T \tilde{x}^i\ \forall\, i,\ \text{and}\ Rx \leq c\}.$$

Clearly, P contains at least one vertex [6], denoted by x. Since x is a vertex, there are $\sum_i K^i$ linearly independent constraints that are active at x. Note that we call a constraint active if it holds with equality. Since we already have N active constraints from $\mathbf{1}^T x^i = \mathbf{1}^T \tilde{x}^i$ for every i, and at most L active constraints from $Rx \leq c$, then there are at least $\sum_i K^i - N - L$ constraints from $x \geq 0$ that are active. Therefore, at least $\sum_i K^i - N - L$ entries of x are zero, indicating that x contains at most $N+L$ positive entries. ■

Remark 1. We can obtain a similar result by applying the Shapley-Folkman theorem in [5] such that the optimal multipath utility can be achieved by a multipath allocation using at most $N+L+1$ paths. Theorem 4 is thus a slightly stronger result than a direct application of [5].

Remark 2. Similar arguments hold for a more general class of delay-sensitive utility functions proposed in [27]. The objective function is $\sum_i U^i(\|x^i\|_1) - d^T Rx$ where $d^{L\times 1}$ stores the delay of each link. An additional constraint $d^T Rx = d^T R\tilde{x}$ exists in the set P, making the upper bound on the number of paths $N+L+1$. All later results hold accordingly.

If the multipath allocation x^* is a solution to (1), similarly we can define a polyhedron Q as follows,

$$Q = \{x \in \mathcal{R}^{\sum_i K^i} \mid x \geq 0,\ \mathbf{1}^T x^i = \mathbf{1}^T x^{*i}\ \forall\, i,\ \text{and}\ Rx \leq c\}.$$

Clearly, Q is a convex polyhedron that is bounded, finite and pointed. Every point in Q is a solution to the multipath problem as it has the same optimal network utility. In fact, if U^i is assumed to be strictly concave for all i, one can check that Q also contains all the optimal solutions. Thus we have the following lemma:

Lemma 1. *If U^i is increasing and strictly concave, then Q is the solution set of (1).*

Proof: Suppose there exists an optimal multipath solution y^* that does not belong to Q.

Since x^* is an optimal multipath solution, then the optimal network utility is $\sum_i U^i(\mathbf{1}^T x^{*i})$. Since y^* is an optimal solution, then $Ry^* \leq c$ and $\sum_i U^i(\mathbf{1}^T y^{*i}) = \sum_i U^i(\mathbf{1}^T x^{*i})$. $\forall \lambda \in (0,1)$, let $z = \lambda x^* + (1-\lambda)y^*$. Clearly $Rz \leq c$. Since $y^* \notin Q$, then there exists some i such that $\mathbf{1}^T y^i \neq \mathbf{1}^T x^{*i}$. Then $\mathbf{1}^T z^i \neq \mathbf{1}^T x^{*i}$ and $\mathbf{1}^T z^i \neq \mathbf{1}^T y^{*i}$. Since U^i is strictly concave, then $U^i(\mathbf{1}^T z^i) > \lambda U^i(\mathbf{1}^T x^{*i}) + (1-\lambda)U^i(\mathbf{1}^T y^{*i})$. From concavity, we also know $U^j(\mathbf{1}^T z^j) \geq \lambda U^j(\mathbf{1}^T x^{*j}) + (1-\lambda)U^j(\mathbf{1}^T y^{*j})$ for all j. Thus $\sum_i U^i(\mathbf{1}^T z^i) > \sum_i U^i(\mathbf{1}^T x^{*i})$, which contradicts the fact that x^* is the optimal solution. Therefore, every optimal multipath solution belongs to Q.

Conversely, for all x in Q, since $\mathbf{1}^T x^i = \mathbf{1}^T x^{*i}$, then $\sum_i U^i(\mathbf{1}^T x^i) = \sum_i U^i(\mathbf{1}^T x^{*i})$, thus x is also optimal. ■

Similar to the proof of Theorem 4, one can also argue that every vertex of Q is an optimal multipath allocation that uses at most $L+N$ paths. Since an optimal single-path allocation uses N paths, then the difference of number of paths needed to achieve optimal multipath routing and optimal single-path routing can be reduced to at most L.

Our previous discussions indicate that a vertex of the multipath solution set Q is guaranteed to use at most $L+N$ paths. In fact, there are other useful properties of the vertices of Q that we examine next.

Theorem 5. *Let $P_{\min} = \min_{x \in Q} \|x\|_0$ be the minimum number of paths needed to achieve the optimal multipath aggregate utility, then there exists a vertex $\hat{x}$ of Q such that $\|\hat{x}\|_0 = P_{\min}$.*

To prove Theorem 5, we first state a more general result in Lemma 2. Please refer to the Appendix A for its proof.

Lemma 2. *Let $G = \{x \in \mathbf{R}^n \mid Ax \leq b, x \geq 0\}$ be a nonempty polyhedron and let $P_{\min} = \min_{x \in G} \|x\|_0$, then there exists a vertex $\hat{x}$ of G such that $\|\hat{x}\|_0 = P_{\min}$.*

Proof: (Theorem 5) Since $\mathbf{1}^T x^i = \mathbf{1}^T x^{*i}$ can be represented by two inequalities $\mathbf{1}^T x^i \leq \mathbf{1}^T x^{*i}$ and $-\mathbf{1}^T x^i \leq -\mathbf{1}^T x^{*i}$, then Q can be expressed in the form $\{x \mid Ax \leq b, x \geq 0\}$. The result follows by applying lemma 2. ■

Theorem 5 implies that there exists a vertex of Q using the minimum number of paths among all the optimal multipath solutions. However, we need to clarify that not every optimal multipath solution that uses the minimum number of paths is a vertex of Q. For example, consider a simple network in Fig. 2

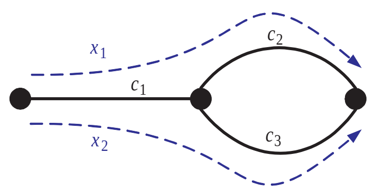
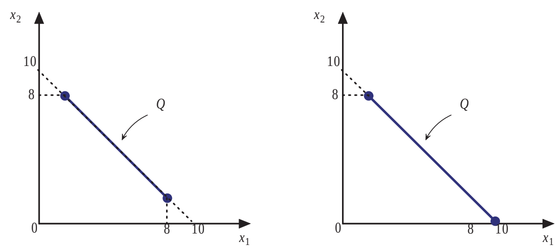


Fig. 2. An example of a simple network supporting one user with two paths



(a) $c_1 = 10$, $c_2 = c_3 = 8$ (b) $c_1 = c_2 = 10$, $c_3 = 8$

Fig. 3. The optimal multipath solution set Q to the network in Fig. 2.

supporting one user with two paths. If $c_1 = 10$, $c_2 = c_3 = 8$, the optimal multipath solution set Q of (1) is

$$Q = \{x_1 + x_2 = 10; 0 \le x_1, x_2 \le 8\},$$

as shown in Fig. 3(a). Now, $\|x\|_0 = 2$ for every x in Q, but Q only has two vertices.

Though in general the statement is not necessarily true that an optimal multipath solution that uses the minimum number of paths is a vertex of Q, it is actually the case if the performance gap is zero. More formally, we have the following theorem.

Theorem 6. *If the performance gap is zero, i.e., there exists a single-path solution in Q, then every optimal single-path solution corresponds to a vertex of Q.*

Proof: Let z^* denote an optimal single-path solution. Then $\mathbf{1}^T z^{*i} = \mathbf{1}^T x^{*i} > 0, \forall\, i = 1, 2, ..., N$ are N constraints that are active at z^*. And z^{*i} has $K^i - 1$ entries that equal to zero. Thus there are at least $\sum_i K^i$ constraints that are active at z^*, and it is easy to check that these $\sum_i K^i$ constraints are linearly independent. Therefore, z^* is a vertex. ■

When the performance gap is zero, an optimal single-path solution must be a vertex of Q. However, in this case not every vertex of Q needs to be an optimal single-path solution. Consider the example in Fig. 2. If $c_1 = c_2 = 10$, $c_3 = 8$, then the optimal solution set Q of (1) is

$$Q = \{x_1 + x_2 = 10; 0 \le x_1 \le 10, 0 \le x_2 \le 8\}.$$

Q has two vertices $(10, 0)^T$ and $(2, 8)^T$, as shown in Fig. 3(b) and the former one is an optimal single-path solution while the latter one is not.

The above three theorems indicate that although many optimal multipath allocations may use a large number of paths, there exist optimal multipath allocations, i.e., the vertices of the solution set Q, which only use at most $L + N$ paths. Moreover, the smallest number of paths used by these vertices is indeed the smallest number of paths needed to achieve optimal multipath routing. In the remaining sections, we will exploit the properties of the vertices of Q to provide a bound to the performance gap in the general case, and propose algorithms to give a good estimate of the duality gap for a given network.

B. Upper bound of the performance gap

Theorem 4 implies that to achieve optimal network utility, at most L users (assuming $L < N$) need to use multiple paths, while every other user only uses one path. We next use this property to upper bound the performance gap.

Theorem 7. *Given an L-link network supporting N users, the performance gap of (1) and (2) is upper bounded by*

$$\min(L, N) \max_i \rho^i, \tag{3}$$

where

$$\rho^i = \max\{U^i(\|x^i\|_1) - U^i(\|x^i\|_\infty) \mid x^i \ge 0, R^i x^i \le c\}. \tag{4}$$

Proof: Let x^* be an optimal solution to (1). Following Theorem 4, there exists an optimal allocation $\hat{x}$ and a set S ($|S| \le \min(L, N)$) of indices such that $\mathbf{1}^T \hat{x}^i = \mathbf{1}^T x^{*i}$ for every i, $R\hat{x} \le c$, $\|\hat{x}^i\|_0 > 1$ for every $i \in S$, and $\|\hat{x}^i\|_0 = 1$ for every $i \notin S$. Note that if S is empty, then the performance gap is zero.

Next, we project $\hat{x}$ to a feasible single-path solution x' by picking the maximum-rate path for each user. More formally, for any $i \in S$, let $d(i)$ be the index of the largest entry of $\hat{x}^i$, i.e.,

$$\hat{x}^i_{d(i)} = \|\hat{x}^i\|_\infty \ \forall\, i \in S.$$

Then, we define a $K^i \times 1$ vector x'^i such that

$$x'^i_{d(i)} = \hat{x}^i_{d(i)}, \text{ and } x'^i_k = 0, \ \forall\, k \ne d(i).$$

For any $i \notin S$, let $x'^i = \hat{x}^i$. Then, one can check that $\|x'^i\|_0 = 1, \ \forall\, i$, and $Rx' \le R\hat{x} \le c$, thus x' is a feasible single-path allocation. Since $\hat{x}$ is an optimal solution to (1), we have

$$\begin{aligned} opt_M - opt_S \le\ & \textstyle\sum_i U^i(\|\hat{x}^i\|_1) - \sum_i U^i(\|x'^i\|_1) \\ =\ & \textstyle\sum_{i \in S} U^i(\|\hat{x}^i\|_1) + \sum_{i \notin S} U^i(\|\hat{x}^i\|_1) \\ & \textstyle - \sum_{i \in S} U^i(\|x'^i\|_1) - \sum_{i \notin S} U^i(\|x'^i\|_1) \\ =\ & \textstyle\sum_{i \in S} \left(U^i(\|\hat{x}^i\|_1) - U^i(\|\hat{x}^i\|_\infty)\right) \\ \le\ & \textstyle\sum_{i \in S} \rho^i \le \min(L, N) \max_i \rho^i, \end{aligned}$$

where ρ^i is given by (4). ■

Note that ρ^i measures the performance loss of user i by restricting itself to single-path routing in our problem setup, and it was first proposed in [1]. Moreover, it can be upper bounded as follows.

Theorem 8.

$$\rho^i \le \max_{y \in [0, M^i]} (U^i(y) - U^i(y/K^i)), \tag{5}$$

where M^i is the maximum total transmission rate of user i that can be supported by a network (c, R, U), i.e.,

$$M^i = \max\{\|x^i\|_1 \ |x^i \ge 0, R^i x^i \le c\}.$$

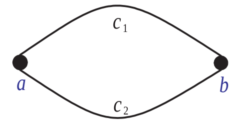


Fig. 4. A network supporting N users

Proof: For any x^i such that $x^i \geq 0, R^i x^i \leq c$, we have $\|x^i\|_\infty \geq \|x^i\|_1/K^i$. Since U^i is strictly increasing, then

$$U(\|x^i\|_1) - U(\|x^i\|_\infty) \leq U(\|x^i\|_1) - U(\|x^i\|_1/K^i).$$

The inequality still holds if we maximize over all feasible x^i, thus

$$\begin{aligned} \rho^i &\leq \max\{U(\|x^i\|_1) - U(\|x^i\|_1/K^i) \mid x^i \geq 0, R^i x^i \leq c\} \\ &= \max_{y\in[0,M^i]} (U(y) - U(y/K^i)). \end{aligned}$$

■

Now, (5) is easy to calculate for a given network. Also, since M^i is always finite, it implies that ρ^i is finite for any user i. For the logarithm utility function, we indeed have an exact formula of ρ^i as follows.

Corollary 1. *If $U^i(\cdot) = \alpha^i \log(\cdot)$ where $\alpha^i > 0$ is some constant, then*

$$\rho^i = \alpha^i \log(K^i). \tag{6}$$

Proof: By Theorem 8, we have $\rho^i \leq \alpha^i \log(K^i)$. Also, there always exists some $\epsilon > 0$ such that $x^i = [\epsilon, \epsilon, ..., \epsilon]^T$ satisfies $R^i x^i \leq c$, thus, from (4), $\rho^i \geq \alpha^i \log(K^i)$. This proves (6). ■

Both the TCP Vegas protocol in [22] and the FAST TCP protocol in [30] implicitly solve (2) using the logarithm utility function. Thus, (6) gives a simple formula of ρ^i, which only depends on the number of paths of user i regardless of the network topology and c.

The significance of Theorem 7 is that when $N \geq L$, the upper bound of the performance gap depends on L and $\max_i \rho^i$, but not on N. Therefore when the number of users N goes to infinity, the performance gap is always upper bounded by $L \max_i \rho^i$, which is finite. Note that opt_M can grow to infinity as N grows even the network topology and c is fixed. For example, consider a network with two nodes a and b and two links connecting them, as show in Fig. 4. Let $c_1 = c_2 = 0.5$. The network supports N users, and each user wants to transmit from a to b. The utility function is $U^i(x) = \sqrt{x}$ for all i. Then the optimal multipath utility is achieved when each user has a total transmission rate $1/N$, and $opt_M = N\sqrt{1/N} = \sqrt{N}$, which goes to infinity as N goes to infinity. In fact, one can check that in this simple network, as long as $U^i(x) = x^p$ for some $p \in (0, 1)$ and for all i, we have $opt_M = N^{1-p}$, which goes to infinity with N. Therefore, even though opt_M can grow to infinity as N goes to infinity, the performance loss $opt_M - opt_S$ is always finite.

V. Estimation of the Performance Gap

A. Vertex projection

In general, (3) may not be a tight upper bound of the performance gap. However, for a given network, if we can find an optimal solution $\hat{x}$ of (1) and its corresponding index set S of the users which use more than one path, a better upper bound to the performance gap than that given by (3) can be obtained as follows.

Suppose we have a vertex $\hat{x}$ of the solution set Q of (1). We will discuss how to obtain such a vertex $\hat{x}$ in a minute. Since $\hat{x}$ is a vertex of Q, then it has positive flow rates on at most $L + N$ paths. Like the arguments in the proof of Theorem 7, we can find a feasible single-path allocation x' by keeping the largest path rate of each $\hat{x}^i$ and setting other path rates to zero. In this case, an improved upper bound to the performance gap is given by

$$opt_M - opt_S \leq \sum_{i\in S} \left(U^i(\|\hat{x}^i\|_1) - U^i(\|\hat{x}^i\|_\infty)\right). \tag{7}$$

However, x' may not achieve the maximum utility for this particular fixed single-path configuration. By solving a fixed routing congestion control problem using the path configuration of x', we obtain a single-path allocation $\bar{x}$ such that $\sum_i U^i(\|\bar{x}^i\|_1) \geq \sum_i U^i(\|x'^i\|_1)$. Thus, a better upper bound of the performance gap than (7) is

$$opt_M - opt_S \leq \sum_i U^i(\|\hat{x}^i\|_1) - \sum_i U^i(\|\bar{x}^i\|_1). \tag{8}$$

Therefore, we first solve (1) to obtain a multipath optimizer x^* (in polynomial time as (1) is convex). We then check whether x^* satisfies $\|x^{*i}\|_0 = 1$ for all $i = 1, ..., N$, or not. If so, (2) has the same optimal value as (1) and the performance gap is zero. Otherwise, we want to find an optimal multipath solution $\hat{x}$ which corresponds to a vertex of Q. Then by the arguments above we obtain an improved upper bound of the performance gap in (8). Then the question is which vertex we should choose if there is more than one vertex in Q. Ideally, we want to find an optimal multipath solution that uses the smallest number of paths, i.e. a solution to the following problem,

$$P_{\min} = \min_{x\in Q} \quad \|x\|_0. \tag{9}$$

There are several reasons why we are interested in solving (9). First, if $P_{\min} = N$, then there is no duality gap, and (9) returns an optimal single-path solution. Second, if $P_{\min} > N$, then performance gap is strictly positive. Since $P_{\min}$ is the minimum number of paths needed to achieve optimal multipath utility, it gives another way to characterize the "cost of not splitting" in terms of the number of paths needed to achieve optimality. Moreover, the solution of (9) can be used to estimate the performance gap via (8). Since it uses the smallest number of paths for all solutions in Q, the number of paths needed to delete in order to obtain a single-path solution is also minimized.

However, (9) is NP-hard [23] and solving it requires a combinatorial search. Since we know (9) has a solution that is a vertex of Q, instead of solving it directly as an optimization problem, we can enumerate all the vertices of Q. Then, the vertex using the minimum number of paths is guaranteed to be a solution to (9). A practical pivot-based algorithm has been proposed in [2] to find v vertices of a polyhedron in $\mathbf{R}^d$ defined by a non-degenerate system of n inequalities in $O(ndv)$ time and $O(nd)$ space. In some cases, the number of vertices of Q can be exponential in the number of users (see Appendix B for an example of such a network), thus vertex enumeration is not practical especially in large network scenarios.

Given a nonconvex problem (9), one naturally considers its convexified problem. Q is a convex set and the nonconvexity of (9) comes from the nonconvex function $f(x)$. Define $f^c_Q(x)$ as the convex envelope [14] of $f(x) = \|x\|_0$ over Q, which is equivalent to the biconjugate function of $\|x\|_0$ as defined by Fenchel [15]. According to the definition of the convex envelope, f^c_Q is a convex function on Q, and $f^c_Q(x) \leq f(x)$ for every x in Q. Moreover, for any convex function g on Q such that $g(x) \leq f(x)$ for every x in Q, we have $g(x) \leq f^c_Q(x)$ for every x in Q. Then the convexified problem of (9) is,

$$\min_{x \in Q} \quad f^c_Q(x). \tag{10}$$

And we have

$$\min_{x \in Q} \ f^c_Q(x) \ \leq \ \min_{x \in Q} \ f(x) \ = \ P_{\min}.$$

In general, the convex envelope of a function is hard to compute analytically. However, notice that $f(x) = \|x\|_0 = \sum_i \mathbf{1}_{\{x_i \neq 0\}}$, where $\mathbf{1}$ is the indicator function. Therefore the objective function $f(x)$ is separable in the decision variables x_i's. Define a $\sum_i K^i \times 1$ vector u such that $|x_k| \leq u_k$ for all k and all x in Q. Define $\underline{Q} := \{x \mid |x_k| \leq u_k, \forall k\}$, clearly $Q \subseteq \underline{Q}$. The convex envelope of $\|x\|_0$ over $\underline{Q}$ in fact is very easy to compute [13][14], which is

$$f^c_{\underline{Q}}(x) = \sum_k \frac{|x_k|}{u_k}. \tag{11}$$

Clearly, for every x in Q, we have

$$f^c_{\underline{Q}}(x) \leq f^c_Q(x) \leq f(x).$$

Thus, (9) can be further relaxed as

$$\min_{x \in Q} \quad f^c_{\underline{Q}}(x), \tag{12}$$

whose value servers as a lower bound of (9). Since here we also have the constraint that x is nonnegative, then (11) is simplified as $f^c_{\underline{Q}}(x) = \sum_k x_k/u_k$, and (12) is simplified as

$$\min_{x \in Q} \quad \textstyle\sum_k x_k/u_k, \tag{13}$$

which is an LP. The solution of (13), denoted by y^*, is a vertex of Q. In general, y^* may not be the optimal solution of (9), however, since y^* is a vertex of Q, the cardinality of y^* is no greater than $L+N$ (cf. Theorem 4). Thus we have

$$N \leq P_{\min} \leq \|y^*\|_0 \leq L + N.$$

Clearly y^* depends on the choice of region $\underline{Q}$. With a good choice of $\underline{Q}$, $\|y^*\|_0$ can be very small, even as small as $P_{\min}$. In other words, y^* could also be the solution of (9) if $\underline{Q}$ is smartly chosen. Then how to choose such $\underline{Q}$?

One simple way to choose $\underline{Q}$ is as follows. As $Rx \leq c$, then $\mathbf{1}^T Rx \leq \mathbf{1}^T c$. Since $x \geq 0$, then we have $x_k \leq \frac{\mathbf{1}^T c}{(R^T \mathbf{1})_k}$ for all k. Let $\underline{Q} = \{x \mid 0 \leq x_k \leq \frac{\mathbf{1}^T c}{(R^T \mathbf{1})_k}, \forall k\}$. We have $f^c_{\underline{Q}}(x) = \frac{1}{\mathbf{1}^T c} \mathbf{1}^T Rx$. Then (13) is equivalent to

$$\min_{x \in Q} \quad \mathbf{1}^T Rx. \tag{14}$$

Algorithm 1 Vertex Projection

1. Solve (2) to obtain an optimal multipath solution x^* and Q.
2. Solve (14) to obtain a vertex $\hat{x}$ of Q using at most $L+N$ paths.
3. Project $\hat{x}$ to a single-path configuration by picking the current largest-rate-path for each user.
4. Maximize the network utility over the chosen single-path configuration.

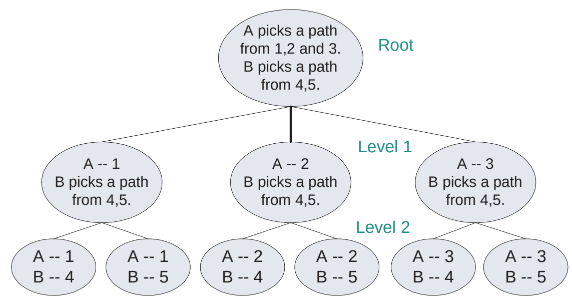


Fig. 5. The feasible path sets of two users (user A has three paths 1,2 and 3, user B has two paths 4 and 5)

Note that $\mathbf{1}^T Rx$ can be interpreted as the total usage of link capacities. Thus by minimizing the total usage of link capacities among all the optimal multipath solutions, (14) returns a multipath solution which is a vertex of Q.

In summary, we first solve (1) to obtain an optimal multipath solution x^* and the solution set Q. We then obtain a vertex of Q through solving (14), project the vertex to a single-path configuration by picking the maximum-rate path for each user, and lastly maximize the network utility over this particular single-path configuration. We refer to this method as the *vertex projection* method as described in Algorithm 1. It gives a lower bound of opt_S and a feasible single-path configuration. As the set S of users that use multiple paths is small for a vertex solution ($|S| \leq \min(L, N)$), we expect the performance loss to be small after the optimal multipath configuration is projected to a single-path configuration. Therefore, the vertex projection method can give a relatively "good" lower bound of opt_S, and also upper-bound the performance gap by (8).

We should remark here that if $\underline{Q}$ is chosen to be $\underline{Q} = \{x \mid 0 \leq x \leq \alpha\}$ for some α, then (13) is equivalent to

$$\min_{x \in Q} \quad \|x\|_1. \tag{15}$$

The objective function of problem (15) is the ℓ_1 norm of a vector, which is closely studied in the area of *compressed sensing* [4], [9], [11], [12] . In fact, both (9) and (15) are closely related to the sparse recovery problem in compressed sensing (we omit the comparison here). However, ℓ_1 norm is not a good candidate for the objective function here, as $\mathbf{1}^T x = \mathbf{1}^T x^*$ is a constant for all x in Q.

B. Refined vertex projection via greedy branch-and-bound

The vertex projection in Section V-A gives a single-path allocation with its corresponding network aggregate utility serving as a lower bound of (2). In this section, we show

how to integrate it with a greedy branch-and-bound algorithm to give a better estimate of opt_S. An N-level tree is introduced to represent a progressively finer partition of the set of paths that each user considers. In particular, the tree has at its root node the original single-path problem (2). The intermediate nodes correspond to problems where some users fix their path choices while every other user can still choose its path from several paths. At each tree node, one user that has not fixed its path choice, say user n, partitions its set of paths producing K^n different subtree nodes. The tree has $\prod_{i=1}^{N} K^i$ leaf nodes, and each leaf node corresponds to a utility maximization problem over a specific single-path configuration. Figure 5 illustrates the feasible path sets in a two-user case.

We first state how to find opt_S via branch-and-bound. The algorithm starts from the root and branches from the current tree node into several subproblems at each step. For each new-found tree node, we find an upper bound and a lower bound of the maximum utility of (2) over the reduced feasible set for this subproblem. The upper bound is given by the value of its dual problem, which is equivalent to the value of its corresponding multipath problem, and the lower bound can be found by the vertex projection in Section V-A. Let m_b be the maximum lower bound that has been found till the current step. If the upper bound for some tree node H is smaller than m_b, then H and all its offspring can be safely pruned from future considering. After pruning, we pick a tree node that has the maximum lower bound among the remaining tree nodes (not including leaf nodes) to branch from for the next step. The algorithm stops when there is no more tree nodes to branch from. It always finds opt_S, which is attained at the leaf node that has the maximum value. However, there is no guarantee that the algorithm will terminate in polynomial time.

To get a polynomial-time approximation algorithm, we propose to do *greedy pruning* at each step. Specifically, at each level of the tree, we only keep the node that has the maximum lower bound among all the nodes at the same level, and delete all its peers. Then, we branch from this node but also only keep one of its offspring, and so on. Thus, we get a path from the root to a leaf node. Though this leaf node may not solve (2) optimally as we greedily prune all but one node at each step, it gives a good lower bound of opt_S. This algorithm terminates in at most N steps as we fix the path choice for one user at each step.

Another issue is which user to branch from at each level. Let W be the set of users that have already fixed their single-path choices until the current step. Let x be a vertex of the optimal sets of the sub multipath problem in the last step. First, we find the user n that solves

$$n = \arg\max_{i \notin W} U^i(\|x^i\|_1) - U^i(\|x^i\|_\infty),$$

and tentatively branch from n. We solve K^n subproblems and find a lower bound m_k of opt_S in the kth subproblem where user n only uses its kth path. Let $m_{k^*} = \max_k m_k$. If $m_{k^*} \geq m_b$, we update m_b with m_{k^*}, make this branch permanent and fix user n's path choice to be its k^*th path. Otherwise, as m_b cannot be improved in this step by branching from user n, we discard this tentative branch, find the user that has the second largest value of $U^i(\|x^i\|_1) - U^i(\|x^i\|_\infty)$ and make another tentative branch from it. The algorithm stops if all users have made their path choices or if we cannot make any tentative branch permanent at some step. Clearly, the algorithm gives a better estimate of opt_S in each step and thus produces a tighter lower bound of opt_S than the vertex projection in Section V-A. The algorithm is summarized in Algorithm 2.

Algorithm 2 Refined vertex projection via greedy branch-and-bound (MP: multipath; SP: single-path)

Initial: $W = \emptyset$ /* users that have fixed SP choice */
    $I_{sp} = 0$, $I_{end} = 0$
Solve the MP problem, find a vertex x of Q, obtain a lower bound m_b of opt_S via vertex projection.
while $|W| < N$, $I_{sp} = 0$ and $I_{end} = 0$ **do**
  if $\|x^i\|_1 = \|x^i\|_\infty, \forall\, i$ **then**
    $I_{sp} = 1$ /* x is also a SP solution. */
  else
    $B = \{1, 2, ...N\} \backslash W$, $I_{pt} = 0$
    while $I_{pt} = 0$, $I_{end} = 0$ **do**
      $n = \arg\max(U^i(\|x^i\|_1) - U^i(\|x^i\|_\infty)), i \in B$
      /* pick a tentative user to branch */
      for each path k of user n **do**
        Solve a sub MP problem and obtain a lower bound m_k of the sub SP problem via vertex projection.
      end for
      if $\max_k m_k \geq m_b$ **then**
        $I_{pt} = 1$, /* make this branch permanent */
        $m_b = \max_k m_k$,
        $k^* = \arg\max m_k$, fix path k^* for user n
        $W \leftarrow W \cup \{n\}$,
        $x \leftarrow$ the vertex found in the k^*th subproblem
      else
        $B \leftarrow B \backslash \{n\}$
        if $B = \emptyset$ **then**
          $I_{end} = 1$
        end if
      end if
    end while
  end if
end while
return x, m_b

C. Extension to convex-cardinality problems

Our techniques in the previous sections and Algorithms 1 and 2 can be extended to other constrained routing problems, similar to that of (2), as a convex-cardinality problem. A convex-cardinality problem is one that would be convex, except for the appearance of the cardinality constraint in the objective or the constraints [8]. For example, the number of paths used by each user can be generalized from a single path to an arbitrarily subset of the possible paths. The utility maximization can then be stated in the following optimization problem:

$$\begin{array}{ll} \max\limits_{x \geq 0} & \sum_i U^i(\|x^i\|_1) \\ \text{s.t.} & Rx \leq c, \\ & \|x^i\|_0 \leq k_i, \quad k_i \in \{1, \ldots, K^i\}, \quad \forall\, i. \end{array} \tag{16}$$

TABLE II
VERTICES OF THE OPTIMAL SOLUTION SET Q

	Vertex 1	**Vertex 2**	**Vertex 3**	**Vertex 4**
x^1	2.7352	0.0000	2.7352	0.0000
	0.0000	0.0000	0.0000	0.0000
	21.8216	***21.8216***	***21.8216***	***21.8216***
	0.0000	0.0000	0.0000	0.0000
	0.0000	2.7352	0.0000	2.7352
x^2	0.0000	0.0000	0.0000	0.0000
	14.3455	14.3455	9.2929	9.2929
	0.0000	0.0000	0.0000	0.0000
	5.0526	5.0526	0.0000	0.0000
	29.8307	***29.8307***	***39.9359***	***39.9359***
x^3	***28.2867***	***28.2867***	***28.2867***	***28.2867***
x^4	0.0000	0.0000	0.0000	0.0000
	32.4964	***32.4964***	22.3912	22.3912
	0.0000	0.0000	0.0000	0.0000
	28.1851	28.1851	***33.2378***	***33.2378***
	18.6865	18.6865	23.7392	23.7392
x^5	***22.7468***	***22.7468***	***22.7468***	***22.7468***
	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000
	5.5335	5.5335	5.5335	5.5335

Obviously, when $k_i = 1$ for all i, then (16) is reduced to the single-path problem (2); when $k_i = K^i$ for all i, (16) is equivalent to the convex problem (1). In general, (16) is hard to solve. As in the previous, (1) can be viewed as a convex relaxation to (16) by removing the cardinality constraints. When viewed as a convex-cardinality heuristic, Algorithm 1 can be adapted to find a feasible solution to (16) by first solving (14) to obtain a vertex, fixing the sparsity pattern of routing by keeping at most k_i paths for user i, and re-solving the utility maximization problem with fixed routes. Likewise, Algorithm 2 can be easily adapted to find a near-optimal solution to (16). Finding the conditions under which (1) solves (16) exactly is an interesting open question.

VI. NUMERICAL EXAMPLES

In this section, we describe our numerical evaluation on a random network of L links supporting N users. Every user can use multiple paths. The link capacities are uniformly chosen from the interval [50,100]. A path uses a link with probability $p = 2\ln(L)/L$. We vary N, L, and the number of paths that each user can use in different simulation setups.

A. Vertices of the optimal solution set of the multipath problem

We fix $N = 5$, $L = 10$, and generate one realization of the network. The five users use 5,5,1,5, and 4 paths respectively. We use the same utility function $U(x) = \ln(1+x)$ for all the users, and opt_M is 18.2982 in this case. We also calculate that opt_S is 17.8609 by exhaustive search. Using a vertex enumeration algorithm [2], we explicitly find all four vertices of the optimal polyhedron Q as listed in Table II.

Though lying in $\mathbf{R}^{20}$, the optimal set Q contains only 4 vertices. Moreover, after projecting the solutions to single-path configurations by choosing the maximum-rate paths, there are only two different configurations, and their maximum utilities are 16.6122 and 16.9176 respectively. Then, the upper bounds of the performance gap are 1.686 and 1.3806 respectively, while the actual performance gap is 0.4374.

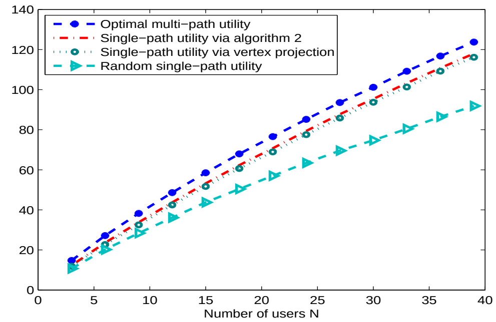


Fig. 6. Comparison of opt_M, two lower bounds of opt_S found by algorithms 1 and 2, and a random single-path utility as N increases

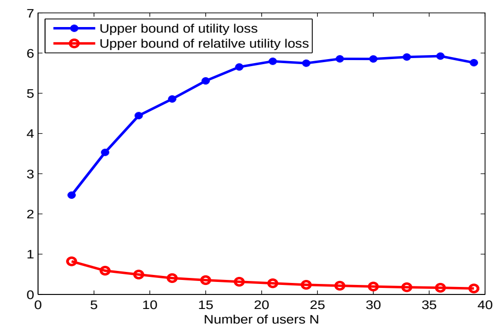


Fig. 7. Total duality and average performance gap as N increases

We observe that the number of paths used by these optimal solutions are 11,11,10 and 10 respectively, while $L+N$ equals 15 in this case. Therefore, the conclusion from Theorem 4 that optimal multipath routing can be achieved with at most $L+N$ paths is a relatively conservative estimate, while it is likely that an optimal multipath configuration requires far fewer paths.

B. Estimation of performance gap via two algorithms

We fix $L = 100$ and let N change from 3 to 40. Each user has 8 available paths. We let the utility function $U(x) = \ln x$ for all the users. All the results are averaged over 100 realizations. For each N, we calculate opt_M and find lower bounds of opt_S by both Algorithm 1, i.e., the vertex projection method, and the improved version, Algorithm 2, that combines vertex projection with greedy branch-and-bound. Note that the difference of opt_M and a lower bound of opt_S serves as an upper bound to the performance gap. We also randomly choose one single-path configuration and calculate its maximum utility. As shown in Fig. 6, all four curves monotonically increase as the number of users increases. The two lower bounds of opt_S are always near opt_M, while the utility of a randomly chosen single-path routing gradually deviates from opt_M. Thus, although opt_S is always near opt_M, and the performance gap or, equivalently, "the cost of not splitting" is not large, the utility of a randomly chosen single-path configuration can be significantly less than opt_M. This demonstrates the need for a routing algorithm to find a near optimal single-path configuration. The single-path configuration found by Algorithm 2 always has a higher utility than that of a single-path configuration found by vertex projection method, as Algorithm 2 can be viewed as a finer sequential application of vertex projection.

Fig. 7 shows an upper bound of the performance gap, which is opt_M minus the lower bound of opt_S obtained from Algorithm 2, and an upper bound of the average cost of not splitting over the number of users. Note that, from Theorem 7 and Lemma 1, the general upper bound of the performance gap for this network is $(3\ln 2)\min(L, N)$, which is a loose upper bound compared with the bound in Fig. 7. Although both curves in Fig. 7 are just upper bounds, they give good estimates of the actual values. So we use them to study the trend of the total and the average cost of not splitting. When the number of users is small ($N < 20$) and the network is partially utilized, users can benefit from using multiple paths, thus the cost of not splitting increases rapidly as the network supports more users. As the number of users increases, users begin to compete for link capacities with one another, and the benefit of multipath routing is no longer apparent. Thus, the cost of not splitting does not increase too much after $N = 20$. Also, note that the average cost of not splitting monotonically decreases and is near 0 when $N = 40$.

VII. Conclusion

We studied the performance difference in utility maximization when either single-path routes or multipath routes are used. We gave a graph-theoretic characterization of the existence of this "cost of not splitting" based on network topology. We showed that the total number of paths needed to achieve optimal multipath utility and that required to achieve the optimal single-path utility differs by no more than the number of links in the network. We provided general bounds of this performance loss which is independent of the number of users. We showed that the performance loss remains finite as the number of users tends to infinity. To provide a good estimate of the performance gap, we proposed a vertex projection method that can also be combined with a branch-and-bound technique to obtain a single-path configuration which can achieve near optimal single-path network utility. The cardinality constraint here appears in many other contexts of combinatorial resource allocation problems. For future work, we plan to use similar techniques to study related convex-cardinality multicommodity flow problems.

Acknowledgments: The authors thank Steven Low of Caltech for his support, wisdom, and encouragement. The authors also thank Jennifer Rexford of Princeton and the anonymous reviewers for helpful comments. The research is supported by NSF under CCF-0835706 and a grant from City University Hong Kong, Project No. CityU 7002698.

References

[1] J. Aubin and I. Ekeland, "Estimates of the duality gap in nonconvex optimization," *Mathematics of Operations Research*, vol. 1, no. 3, pp. 225–245, 1976.

[2] D. Avis and K. Fukuda, "A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra," *Discrete Comput. Geometry*, vol. 8, no. 3, pp. 295–313, 1992.

[3] N. Bansal, Z. Friggstad, R. Khandekar, and M. Salavatipour, "A logarithmic approximation for unsplittable flow on line graphs," in *Proc. 19th ACM-SIAM Symp. Discrete Algorithms*, 2009, pp. 702–709.

[4] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss., "Combining geometry and combinatorics: a unified approach to sparse signal recovery," 2008. [Online]. Available: http://arxiv.org/abs/0804.4666

[5] D. Bertsekas and N. Sandell, "Estimates of the duality gap for large-scale separable nonconvex optimization problems," in *Proc. 21st IEEE Conf. Decision and Control*, vol. 21, Dec. 1982, pp. 782–785.

[6] D. Bertsimas and J. Tsitsiklis, Eds., *Introduction to Linear Optimization*. Belmont, Massachusetts: Athena Scientific, 1997.

[7] A. Bley, "On the approximability of the minimum congestion unsplittable shortest path routing problem," *Book Series Lecture Notes in Computer Science*, vol. 3509, pp. 97–110, 2005.

[8] S. Boyd, "ℓ_1-norm methods for convex cardinality problems," *Lecture Notes for EE364b, Stanford University*, 2007. [Online]. Available: http://www.stanford.edu/class/ee364b

[9] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[10] M. Chiang, S. Low, A. Calderbank, and J. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," vol. 95, no. 1, Jan. 2007, pp. 255 –312.

[11] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[12] D. Donoho and J. Tanner, "Sparse nonnegative solution of underdetermined linear equations by linear programming," in *Proc. Natl. Acad. Sci. U.S.A.*, vol. 102, no. 27, 2005, pp. 9446–9451.

[13] J. Falk, "Lagrange multipliers and nonconvex programs," *SIAM J. Control*, vol. 7, no. 4, pp. 534–545, 1969.

[14] J. Falk and R. Soland, "An algorithm for separable nonconvex programming problems," *Management Science*, vol. 15, no. 9, pp. 550–569, May 1969.

[15] W. Fenchel, Ed., *Convex cones, sets, and functions*. Princeton, New Jersey: Princeton University Press, 1953.

[16] N. Garg and J. Koenemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems." in *Proc. the 39th Annual Symposium on Foundations of Computer Science*, 1998, pp. 300–309.

[17] J. He and J. Rexford, "Toward internet-wide multipath routing," *IEEE Netw.*, vol. 22, no. 2, pp. 16–21, 2008.

[18] J. Kleinberg, "Single-source unsplittable flow," in *Proc. 37th IEEE Symp. Foundations of Computer Science*, 1996.

[19] ——, "Decision algorithms for unsplittable flow and the half-disjoint paths problem," in *Proc. 30th ACM Symp. Theory of Computing*, 1998.

[20] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *J. ACM*, vol. 46, no. 6, pp. 787–832, 1999.

[21] S. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 525–536, Aug. 2003.

[22] S. Low, L. Peterson, and L. Wang, "Understanding TCP Vegas: a duality model," *J. ACM*, vol. 49, no. 2, pp. 207–235, 2002.

[23] B. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, pp. 227–234, April 1995.

[24] F. Paganini and E. Mallada, "A unified approach to congestion control and node-based multipath routing," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 1413–1426, October 2009.

[25] J. Papandriopoulos, S. Dey, and J. Evans, "Optimal and distributed protocols for cross-layer design of physical and transport layers in manets," *IEEE/ACM Trans. Networking*, vol. 16, no. 6, pp. 1392 –1405, Dec. 2008.

[26] S. Plotkin, D. Shmoys, and E. Tardos, "Fast approximation algorithms for fractional packing and covering problems," *Mathematics of Operations Research*, vol. 20, no. 2, pp. 257–301, 1995.

[27] J. Pongsajapan and S. Low, "Reverse engineering TCP/IP-like networks using delay-sensitive utility functions," in *Proc. IEEE INFOCOM*, May 2007, pp. 418–426.

[28] J. Wang, L. Li, S. Low, and J. Doyle, "Cross-layer optimization in TCP/IP networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 3, pp. 582–595, Jun. 2005.

[29] M. Wang, C. Tan, A. Tang, and S. Low, "How bad is single-path routing," in *Proc. of IEEE Globecom*, 2009.

[30] D. Wei, C. Jin, S. Low, and S. Hegde, "FAST TCP: motivation, architecture, algorithms, performance," *IEEE/ACM Trans. Networking*, vol. 14, no. 6, pp. 1246–1259, 2006.

[31] D. Xu, M. Chiang, and J. Rexford, "Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 466–474.

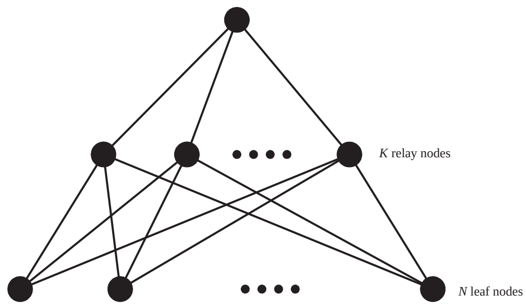


Fig. 8. A relay network with K relay nodes supporting N users

Appendix

A. Proof of Lemma 2

Proof: Since G is nonempty and contains no line, it contains at least one vertex [6]. Let z^1, ..., z^k ($k \geq 1$) denote the vertices of G. Let x^* be a vector in G with the smallest cardinality, i.e. $\|x^*\|_0 = P_{\min} = \min_{x \in G} \|x\|_0$. Since G is the convex hull of z^1,..., z^k, then

$$x^* = \sum_{i=1}^{k} \lambda^i z^i, \tag{17}$$

for some $\lambda^i \geq 0$, $\forall i$ and $\sum_i \lambda^i = 1$. Note that there exists at least one i such that $\lambda^i > 0$. We can assume $\lambda^1 > 0$ without loss of generality.

We claim that for every j such that $x^*_j = 0$, z^1_j is also zero. To see this, note that $\lambda^i \geq 0$ for all i and $z^i_j \geq 0$ for all i, then by (17) we know that $x^*_j = 0$ if and only if $\lambda^i z^i_j = 0$ for all i. Since $\lambda^1 > 0$, then z^1_j must be zero. Thus, the claim follows. Therefore we have $\|z^1\|_0 \leq \|x^*\|_0$.

Since we also have $\|x^*\|_0 = \min_{x \in G} \|x\|_0 \leq \|z^1\|_0$, then we can conclude that $\|z^1\|_0 = \|x^*\|_0 = P_{\min}$, and Lemma 2 follows. ■

B. Exponential number of vertices in Q

Consider a network in Figure 8 with N sources at the root, K relay nodes, and N receives, one at each of the N leaves. Each link has capacity c. All users have the same utility function, $U(\cdot)$, which is increasing and strictly concave. The optimal multipath utility is achieved when each user is allocated with a total rate $\frac{Kc}{N}$, and this optimal utility is $NU(\frac{Kc}{N})$. The performance gap is zero if and only if $\frac{N}{K}$ is an integer. In this case, every vertex of Q corresponds to an optimal single-path solution and vice versa. Q can be represented as follows.

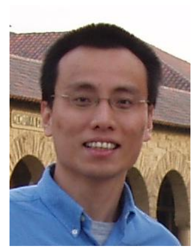
$$Q = \{x \in \mathcal{R}^{KN} \mid x \geq 0, \quad \mathbf{1}^T x^i = \tfrac{Kc}{N} \ \forall i \in \{1, 2, \ldots, N\}, \quad \text{and} \quad \textstyle\sum_{i=1}^{N} x^i_k = c \ \forall k \in \{1, 2, \ldots, K\}\}.$$

The number of vertices of Q is

$$\prod_{i=0}^{\frac{N}{K}-1} \binom{N - \frac{iN}{K}}{\frac{N}{K}} = \binom{N}{\frac{N}{K}} \binom{N - \frac{N}{K}}{\frac{N}{K}} \cdots \binom{\frac{N}{K}}{\frac{N}{K}} = \frac{N!}{(\frac{N}{K}!)^K} \approx K^{\frac{K}{2}} (2\pi N)^{\frac{1-K}{2}} K^N,$$

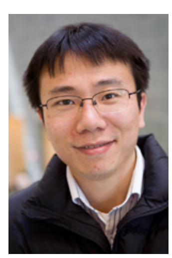
which grows exponentially as N increases.

Meng Wang (S'06) received the B.E. (Hon.) and M.S. (Hon.) in Electrical Engineering from Tsinghua University, Beijing, China, in 2005 and 2007, respectively. She is currently working towards the Ph.D degree in the School of Electrical and Computer Engineering at Cornell University. Her research interests include communication networks, signal processing, and nonconvex optimization and its applications.



Chee Wei Tan (M'08) received the M.A. and Ph.D. degree in Electrical Engineering from Princeton University, Princeton, NJ, in 2006 and 2008, respectively. He is an Assistant Professor at City University Hong Kong. Previously, he was a Postdoctoral Scholar at the California Institute of Technology (Caltech), Pasadena, CA. His research interests are in wireless and broadband communications, signal processing and nonlinear optimization. Dr. Tan was the recipient of the 2008 Princeton University's Gordon Wu Prize for Excellence.

Weiyu Xu received his B.E. degree in Information Engineering from Beijing University of Posts and Telecommunications in 2002, and a M.S. degree in Electronic Engineering from Tsinghua University in 2005. He received his M.S. degree in 2006 and Ph.D. degree in 2009, both in Electrical Engineering, from California Institute of Technology, with a minor in Applied and Computational Mathematics. He was honored as an outstanding graduate of Tsinghua University in 2005, a recipient of the Information Science and Technology Fellowship at California Institute of Technology, a finalist for the Best Student Paper Award at the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), and the recipient of the Charles Wilts doctoral thesis award in 2010. His research interests are in signal processing, compressive sensing, communication networks, information and coding theory.



Ao Tang (S01-M07) received the B.E. (Hon.) in electronics engineering from Tsinghua University, Beijing, China, and the Ph.D. in electrical engineering with a minor in applied and computational mathematics from the California Institute of Technology (Caltech), Pasadena, CA, in 1999 and 2006, respectively.

He is currently an Assistant Professor in the School of Electrical and Computer Engineering at the Cornell University, where his research interests center around various networks including communication networks, power networks and on-chip networks.

Dr. Tang was the recipient of the 2006 George B. Dantzig Best Dissertation Award, the 2007 Charles Wilts best dissertation Prize, and a 2009 IBM Faculty Award.