Modeling non-convex costs in an LP (for traffic engineering on WANs)

Srikanth Kandula, Brendan Lucier, Ishai Menache, Mohit Singh
Microsoft Research

February 19, 2015

1 Traffic Engineering

We describe a Traffic Engineering (TE) solution which can incorporate non-linear link costs. TE decides on the actual assignment of requests' flow to (time, path) pairs.

Formulation. A byte request, indexed by i, has a quantity of data d_i to be routed, and a value per byte v_i. The request indicates the source S_i and target T_i; data must be transmitted along a path from S_i to T_i. Write R_i for the set of admissible paths (or *routes*) for request i. Let X_{irt} denote the number of bytes from request i transmitted along route $r \in R_i$ at time t. The quantities $X = (X_{irt})$ fully describe a schedule of transfers. The objective of TE is to maximize welfare (values minus costs). Formally, the objective is

$$\text{maximize} - C(X) + \sum_i \sum_{t=1}^{T} \sum_{r \in R_i} X_{irt} \cdot v_i \tag{1}$$

$$\text{subject to} \sum_{t=1}^{T} \sum_{r \in R_i} X_{irt} \le d_i \quad \forall i$$

$$\sum_i \sum_{e \in r, r \in R_i} X_{irt} \le c_{e,t} \quad \forall t, e,$$

where $c_{e,t}$ is the available capacity in link e at time t.

The term $C(X)$ is non-convex. It typically corresponds to the sum of all link costs, where a link's cost is linearly proportional to a high-percentile utilization

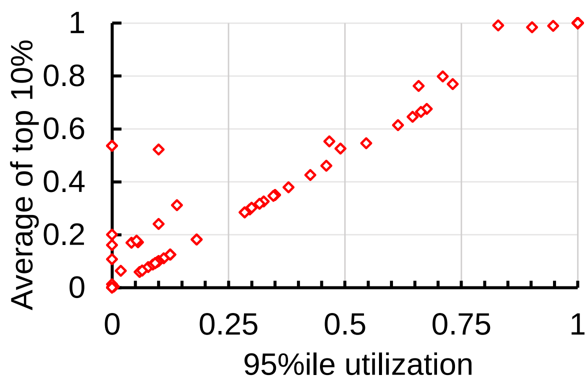

Figure 1: Scatter plot of 95th percentile and average of top 10% utilization values. Each point corresponds to a link.

over time (we use the 95th percentile[1]). This non-linear relation makes the optimization challenging. Formally,

Theorem 1. *Maximizing* (1)*, where* $C(X)$ *is linearly proportional to the sum of 95*th*-percentile utilization in each link, is an NP-hard optimization problem.*

The proof follows by a reduction from the NP-had subset-sum problem [2].

Before describing our solution, we note that we use the above formulation as an exemplar setting for non-convex costs. That is, the techniques we develop here would be useful even for other objective functions/constraints.

Solution. We deal with the above challenge by using an alternative metric, which serves as a proxy for 95th percentile. We use the average of the top k utilization values; e.g., if costs are computed over a time horizon of $T = 100$ time-steps, and we are interested in the 95th percentile, then we choose $k = 10$. Note that if utilization values are uniformly distributed, the two metrics coincide[2]. We have verified experimentally that the two metrics are highly correlated, see Figure 1 for scatter plots of several links.

In principle we can "code" the sum (hence, the average) of the top k by adding a linear term to (LP), and a new set of linear inequalities, which result in a new linear program. Formally, the objective is $\sum_i \sum_t \sum_{r \in R_i} X_{irt} \cdot v_i - \sum_e C_e S_e$, where C_e is the per-unit cost, and S_e is an upper bound on the k largest utilization values

[1] 95-th percentile costs are often used nowadays by operators to lower burst usage.

[2] We may consider choosing different values of k for heavily-skewed distributions. We have not yet fully investigated this direction.

(since we are minimizing $\sum_e C_e S_e$ the upper bound becomes tight at an optimal solution) . The constraints are as follows:

$$S_e \geq \sum_{t \in \mathcal{T}} f_{e,t} \quad \forall \mathcal{T} \subset \{1, \ldots, T\},\ |\mathcal{T}| = k, \tag{2}$$

where $f_{e,t} = \sum_i \sum_{r \in R_i : e \in r} X_{irt}$ is the flow on edge e at time t. A difficulty is that the number of constraints in (2) is exponential in T (more precisely, $\binom{T}{k}$ for each link). The resulting LP is then intractable. We address this by using sorting-network inequalities (see [1] and references therein), which reduces the number of constraints to $O(kT)$ per link, without any loss in accuracy. Formally,

Theorem 2. *There exists a set of $O(kT)$ linear constraints which expresses an upper bound on sum of top k values from the set $f_{e,1}, \ldots f_{e,T}$.*

In a nutshell, the sorting network mimics the operation of the bubble sort algorithm: each iteration i of the algorithm is mapped to a set of equalities/inequalities which bubble up (an upper bound of) the sum of the largest i elements. See appendix for the construction and proof.

References

[1] H. H. Liu, S. Kandula, R. Mahajan, M. Zhang, and D. Gelernter. Traffic Engineering with Forward Fault Correction. In *SIGCOMM*, pages 527–538. ACM, 2014.

[2] R. G. Michael and S. J. David. Computers and Intractability: A Guide to the Theory of NP-completeness. *WH Freeman & Co., San Francisco*, 1979.

A Constructing the Sorting Network

Inspired by the bubble sort algorithm, we construct a set of $O(kT)$ constraints and show that the construction results in an upper bound S_e on the sum of the k largest utilization levels in each link. Since we are maximizing $\sum_i \sum_t \sum_{r : r \in R_i} X_{irt} \cdot p_i - \sum_e c_e S_e$, we are minimizing each S_e and the upper bound becomes tight as required. We will omit subscript e from our notation.We proceed in k iterations:

in the first, we "bubble" the largest element, then the second largest, etc.. Our constraints mimic the bubbling operations – for each two numbers x, y to be compared, we have a linear *comparator*, which is manifested through the following inequalities: $x + y = m + M, m \leq x, m \leq y$. Note this implies $M \geq \max\{x, y\}$ and $m \leq \min\{x, y\}$.

Let f_j^i denote the minimum of the two outputs of the j-th comparator at the i-th iteration, and let F_j^i denote the maximum of the two values. We use the convention $f_j^0 = f_j$ for all $j \in \{1, 2, \dots T\}$. Accordingly, our first comparator at the first iteration is given by $f_1^0 + f_2^0 = f_1^1 + F_1^1$, $f_1^1 \leq f_1^0, f_1^1 \leq f_2^0$. As in bubble sort, the maximum output is pushed to the next comparator, i.e., the rest of the constraints for this iteration have following form: $f_j^0 + F_{j-2}^1 = f_{j-1}^1 + F_{j-1}^1$, $f_{j-1}^1 \leq f_j^0$, $f_{j-1}^1 \leq F_{j-2}^1$, for every $j \in \{3, 4, \dots T\}$. Using all the above constraints, it can be easily shown that

$$F_{T-1}^1 \geq \max\{f_1^0, f_2^0 \dots\ f_T^0\} \tag{3}$$
$$f_1^0 + f_2^0 + \dots + f_T^0 = f_1^1 + f_2^1 + \dots f_{T-1}^1 + F_{T-1}^1 \tag{4}$$

Indeed, (3) follows from a chain of inequalities $F_j^1 \geq f_{j+1}^0, F_j^1 \geq F_{j-1}^1$ for any $2 \leq j \leq T-1$ and $F_1^1 \geq f_1^0, f_2^0$, whereas summing all the above equalities and canceling out equal terms leads to (4).

The next iteration proceeds with variables $f_1^1, f_2^1, \dots f_{T-1}^1$ (one less comparator than previous iteration), which similarly leads to

$$F_{T-2}^2 \geq \max\{f_1^1, f_2^1 \dots\ f_{T-1}^1\}$$
$$f_1^1 + f_2^1 + \dots + f_{T-1}^1 = f_1^2 + f_2^2 + \dots f_{T-2}^2 + F_{T-2}^2$$

It follows from (4) and the last equality that $f_1^0 + f_2^0 + \dots + f_T^0 = f_1^2 + f_2^2 + \dots f_{T-2}^2 + F_{T-2}^2 + F_{T-1}^1$.

Proceeding iteratively, we use $(T - i)$ comparators in the i-th iteration (all outputs of iteration i excluding F_{T-i}^i, are inputs for iteration $i + 1$). Using (4) inductively, we have the following equality after k iterations

$$f_1^0 + \dots + f_T^0 = f_1^k + \dots f_{T-k}^k + F_{T-k}^k + F_{T-k+1}^{k-1} + \dots + F_{T-1}^1. \tag{5}$$

Finally, we add the constraint $S \geq F_{T-k}^k + F_{T-k+1}^{k-1} + \dots + F_{T-1}^1$. Note that we have a total of $O(kT)$ equalities/inequalities.

In order to formally prove that S is not smaller than sum of k largest elements we need the following lemma:

Lemma 1. *For any i and any set of indices $Y_i \subsetneq \{1, 2 \dots T - i\}$ we can find a subset of indices $Y_{i+1} \subseteq \{1, 2 \dots T-i-1\}$ such that $|Y_i| = |Y_{i+1}|$ and $\sum_{j \in Y_i} f_j^i \geq \sum_{j' \in Y_{i+1}} f_{j'}^{i+1}$.*

Proof. The proof follows by a charging argument. Assume that $Y_i = \{a_1, a_2, \dots, a_q\}$ and $a_1 < a_2 < \cdots < a_q$. Let p be a largest index such that Y_i can be represented as $\{1, 2, 3 \dots p\} \cup \{a_{p+1}, \dots a_q\}$. It follows that $a_{p+1} > p + 1$. For Y_{i+1} we take $\{1, 2 \dots p\} \cup \{a_{p+1} - 1, \dots a_q - 1\}$. Inequality $\sum_{j \in Y_i} f_j^i \geq \sum_{j' \in Y_{i+1}} f_{j'}^{i+1}$ follows from $f_j^i \geq f_{j-1}^{i+1}$ and $f_1^i + f_2^i + \cdots + f_p^i = f_1^{i+1} + f_2^{i+1} + \dots f_{p-1}^{i+1} + F_{p-1}^{i+1}$, where $F_{p-1}^{i+1} \geq f_1^i, \dots, f_p^i$. □

We are now ready to prove the theorem. We let Y_0 be the set of indices corresponding to the $T - k$ smallest elements among $f_1^0, f_2^0, \dots f_T^0$. And then consequently construct $Y_1, Y_2 \dots Y_k$. We obtain that $Y_k = \{1, 2, \dots T - k\}$. It means that $f_1^k + f_2^k + \dots f_{T-k}^k$ is not larger than the sum of $T - k$-smallest numbers from $f_1^0, f_2^0, \dots f_T^0$. This together with (5) guarantees that $F_{T-k}^k + F_{T-k+1}^{k-1} + \cdots + F_{T-1}^1$ is greater or equal to sum of k largest elements from $f_1^0, f_2^0, \dots f_T^0$. □