Minimizing congestion with a bounded number of paths

C. Duhamel[†] B. Vatinlen[‡] P. Mahey[†§] and F. Chauvet[‡]

We consider a routing problem where we want to minimize the maximal relative congestion on the arcs of the network with a bounded number of paths. We explain how that minimax problem is linked with the search of approximate solutions obtained by successive shortest-path calculations based on exponential arc distance functions. We then study the complexity of the determination of the minimal number of paths supporting a feasible flow, an NP-hard problem. We show how the Flow deviation method can be adapted to include a control on the number of paths and show some numerical experiments on medium-size networks.

Keywords: routing, flow deviation

1 Congestion problems in broadband communications networks

The problem of minimizing the flow on the most congested link when we not only have to route some traffics in a network, but also have to decidethe topology of the network, is of valuable interest for the design of broadband communication networks (see [2]). The difficulty to implement optimal solutions is the large number of paths they generally present. On the one hand, modern technologies like MPLS (Multiple Protocol Label Switching) allow to use more than one LSP (Label Switched Path) for the same origin-destination pair. On the other hand, from the point of view of the telecommunications provider, it is indeed necessary to operate as few distinct paths as possible in the routing tables. Unfortunately, most routing problems with path constraints are likely to be NP-hard as is the basic routing problem with unsplittable flows. Our objective here is to show that classical descent methods can be adapted to deal with these topology constraints if the maximum number of paths allowed for each commodity is not too small.

We consider a graph $G = (V,E)$ with $|V| = n$ and $|E| = m$, a set of fixed capacities $C_e, e \in E$, and a set given demands b_k for each commodity $k = 1,\ldots,K$ (a commodity stands here for a pair of origin-destination willing to communicate). Let Φ_e be a congestion function defined on arc e (which could be proportional to the average transit delay or to the blocking probability, or to a weighted combination of both). Then, the general routing problem is defined as :

$$(MCF) \qquad \min \quad \sum_e \Phi_e(\sum_k x_e^k)$$

$$s.t. \quad \begin{cases} \sum_k x_e^k \leq C_e & e \in E \\ x^k \in F_k, & k = 1,\ldots,K \\ x \in X \end{cases}$$

where $F_k = \{x^k \in \mathbb{R}^m \,|\, Ax^k = b^k, x^k \geq 0\}$ represents the flow constraints for the k-th commodity and X are some additional topology constraints (like path restrictions in the present case).

[†]Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes, CNRS, Université Blaise Pascal, Campus des Cézeaux, Aubière, France

[‡]Bouygues Telecom Recherche, 10 rue Paul Dautier, 78944 Vélizy

[§]Corresponding author : philippe.mahey@isima.fr

Closely related formulations of (MCF) have been analyzed in the literature, mostly, linear and convex cost multicommodity flow problems without path restrictions (see [1] and [9]), or linear multicommodity flow problems with path constraints like hop constraints [7] or single path routing [3].

To analyze the path structure of the optimal solution of (MCF), we need to come back to the basic minimal congestion problem. That problem, denoted hereafter by (MINCONG), is modelled by the following mathematical program :

$$(MINCONG) \qquad \min \quad z$$
$$s.t. \quad \begin{cases} \sum_k x_e^k - C_e z \leq 0 & e \in E \\ x^k \in F_k, & k = 1, \ldots, K \end{cases}$$

(MINCONG) turns to be a linear program but it is generally considered as a very hard problem. The first reason why this occurs is that the objective function (in the min-max formulation) is convex piecewise linear and not separable with respect to arcs. The second reason is that it produces optimal solutions with a huge number of active paths, being in that sense equivalent to the Maximum Concurrent Flow problem as shown below.

To analyze the path structure of an optimal solution, we will call *active* a path which carries some positive flow and *critical* an arc e such that $x_e = C_e z$. An active path containing critical edges will be called a critical path. Following a classical analysis on the basis structure of the linear program (see [1] for instance), we have :

Proposition 1 *An optimal solution to (MINCONG) will contain at most $K + \sigma$ active paths (and at least K), where σ is the number of critical edges.*

Moreover, if any commodity is routed on a critical path, then all paths used to route that commodity are critical.

(MINCONG) is also strongly related to the Maximum Concurrent Flow problem (MAXTHRU) (see [10]), where one wants to maximize the throughput of the network for a given set of capacities. The throughput of the network is a load factor that multiplies the traffic matrix.

$$(MAXTHRU) \qquad \max Z$$
$$s.t. \quad \begin{cases} \sum_k \tilde{x}_e^k & \leq & C_e & \\ \tilde{x}^k & \in & F_k(Z) & k = 1, \ldots, K \end{cases}$$

where $F_k(Z) = \{\tilde{x}^k \in \mathbb{R}^m \,|\, A\tilde{x}^k = Zb^k, \tilde{x}^k \geq 0\}$.

One can easily verify that the optimal solutions x^* and $\tilde{x}^*$ of (MINCONG) and (MAXTHRU) respectively satisfy $x^* = z^*\tilde{x}^*$ and that $z^* = 1/Z^*$. Both problems have received a lot of attention in the past decade, namely since the seminal paper by Shahrokhi and Matula [10] who first proposed a fully polynomial approximation scheme to solve (MAXTHRU) with uniform capacities. They showed that the minimization of a separable exponential penalty function on the arcs yields a flow with a nearly maximal throughput. They chose the following penalty function :

$$\phi_e(x_e) = \exp(\frac{2m^2}{\varepsilon} x_e)$$

where ε is a positive parameter which defines the approximation. They found a complexity bound of $O(nm^7/\varepsilon^5)$ and showed that the number of active paths is in $O(m^3/\varepsilon^2)$.

The idea we are following here is that the use of a convex increasing congestion function separable on the arcs will have basically the same effect as these potential functions, in other words, it will tend to push flow on a subset of the active paths produced by the minimum congestion solution (or the maximum throughput solution). Bienstock and Raskina [**?**] have proposed an adaptation of the Flow Deviation method [5] based on Kleinrock's delay function for M/M/1 queues of independent Poissonian traffic which yields a fully polynomial approximation scheme for (MINCONG). After a short description of the Flow Deviation

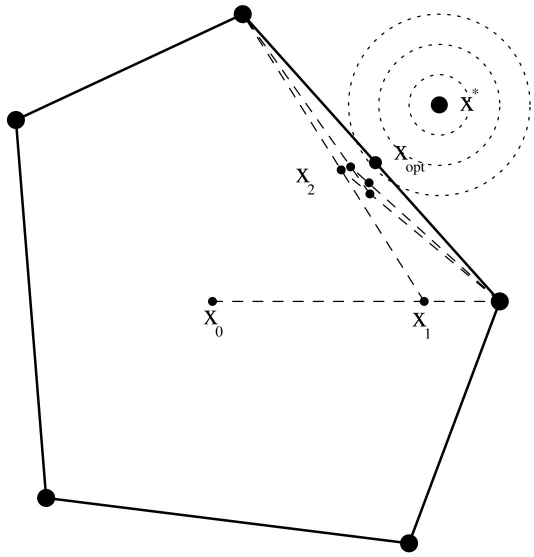

Fig. 1: Zigzagging with Frank-Wolfe's method

method, we will extend these ideas to include an effective control on the number of active paths.In other words, the network is incrementally loaded and the number of active paths is controlled by deviating flows from the less loaded paths to the currently shortest ones. As the path support for a feasible multicommodity flow is not unique, we have added in section 3 a study of the combinatorial difficulty to search for a minimal path support.

2 The flow deviation method with a bounded number of paths

The Flow Deviation method (FD) is an adaptation of the classical linearization algorithm of Frank and Wolfe to solve multicommodity flow problems with convex costs. It has been first proposed by Fratta, Gerla and Kleinrock [5] in the context of design of packet-switched networks and has been widely used by the community of transportation and in telecommunications networks (see [2]).

The method proceeds by successive linearizations solving LP subproblems at each iteration and carrying out a line search on the segment joining the current solution and the extreme point solution of the LP subproblem. Convergence results have been obtained in the strictly convex case but the method suffers from very slow convergence tail which is evidenced by the following fact. The solution of the non linear program is in general not a vertex of the feasible polyhedron so that, as the sequence of feasible solutions tends to a point on a face of the polyhedron, the angle between successive descent directions tends to 180° (see Fig. 1).

On the other hand, the positive aspects of the implementation of Frank-Wolfe's method to multicommodity flow problems are :

- the simplicity of the subproblems resolutions which reduce to shortest-path computation,
- the solution update which consists in deviating flows fairly from the active paths towards the new ones without explicitly computing the individual commodity flows.

Indeed, an *equal* fraction of the flow on the nonshortest paths is shifted to the shortest path. One may observe that this update of path flows will tend to increase monotonically the number of active paths. We will present in the last section a heuristic procedure to carry out(FD) steps with a bounded number of paths. On the other hand, variants of the basic Flow Deviation method can be built by modifying one of the inner steps of the algorithm, i.e.

- either by modifying the search direction, for instance by using the projected gradient direction in the path flow space (see [2])
- or by deviating *non uniform* flow proportions on the newly generated paths.

The modified Flow Deviation method (MFD) presents the following steps (where ε is a small positive tolerance and δ is a positive increment to be tuned by the user) :

1. Initialize with a small throughput Z^0 to get a strictly feasible solution made of shortest paths
2. While $Z^t \leq 1-\varepsilon$, repeat
   (a) Magnify the throughput by loading the existing paths such that $Z^{t+1} = (1+\delta)Z^t$ and identify the critical edges and paths
   (b) Reduce congestion by a fixed number of (FD) steps based on the Kleinrock's congestion function
   (c) Maintain the number of active steps under a given threshold by a heuristic procedure (see the next section)

Observe that Bienstock and Raskina [4] suggested to substitute the shortest-path calculations by min-cost flow subproblems. This comes naturally in the linearization procedure of Frank-Wolfe's algorithm if we add redundant capacity constraints for each commodity, i.e. $x_e^k \leq c_e, \forall e$. The higher cost of the subproblems is compensated by better bounds in the complexity estimates and that trick is also used in other papers (see [8] or [6]).

3 More results on the path number

One characteristic of the path structure associated with a feasible multicommodity flow is its non unicity. It is thus natural to seek a minimal support in term of individual paths. Let (MINPATH) denote the problem to determine the minimum path support for a given $s-t$ flow and let N_{st} be the minimum number of paths. Showing that (MINPATH) can be reduced to the 3-partition problem, we can prove the following fact :

Proposition 2 *The problem to determine a minimal number of paths carrying a given flow vector is NP-hard in the strong sense.*

This remains true even if some set of supporting paths is already given (as is the case during the Flow Deviation iterates). Moreover, computing a lower bound on N_{st} is not easy. Indeed, a valid lower bound is given by the maximal cut obtained in the subgraph associated with the arcs supporting the flow, which is itself a NP-complete problem (but for which efficient heuristics do exist). On the other hand, the number of arcs supporting a positive flow gives an upper bound on N_{st} as the arc-path incidence matrix of the flow support must be full-rank.

We have analyzed the performance of an augmenting path algorithm to get local minimizers for (MINPATH). It iteratively loads the paths associated with the minimal residual capacity in the supporting graph (where the residual capacity of an arc is the initial flow value minus the total path flows of the formerly generated paths which use that arc). As the successive paths saturate at least one arc (the one with the minimal residual capacity), their incidence vectors in the arc-path matrix are linearly independent, which implies that the final solution is a local minimizer for (MINPATH).

Fig.2 illustrates the difficulty to obtain the minimal number of paths on a simple graph with 6 arcs. Arc flows are given with ε a positive number, supposed small with respect to one. The number of paths generated by the augmenting path procedure will in general produce 5 paths ($N_{algo} = 5$) which is better than the upper bound ($m = 6$) but still worse than the minimum ($N_{st} = 4$). We show on Fig. 3 how a slight modification of the arc flows (deviating ε from two less loaded paths towards the most loaded path) can reduce the number of paths to its minimal achievable number (here 3).

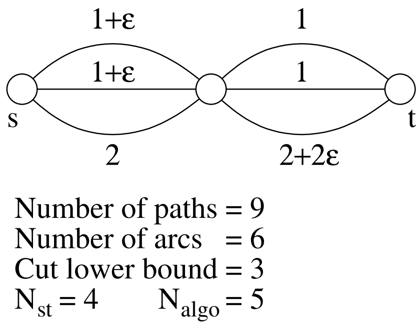

Fig. 2: An example with 4 paths

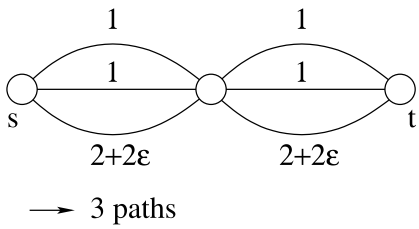

Fig. 3: Deviating flow to reduce the path support

4 Numerical experiments and conclusions

The Modified Flow Deviation method has been tested on a set of medium-size networks. We have compared the number of paths given by the original procedure and the modified algorithm with the following heuristic control of the number of paths :

- Augment the step size in the linesearch to deviate as much flow as possible onto the shortest path
- Drop paths with a negligible quantity of flow by fairly deviating their flow on the remaining paths

The results are computed on the same graph instance with 15 nodes, 38 (undirected) arcs and 49 commodities and are summarized on Fig. 4. The initial point is obtained by a heuristic procedure which iteratively loads the commodities on the shortest paths in a fair distributed way. Then, (MFD) is applied until the improvement drops below a given threshold (see first column). Experiments are compared with and without the path-reduction procedure. The last columns show the total number of paths, and the minimum, average and maximum number of paths per commodity. The algorithm is clearly more efficient when the required accuracy is higher. It is worth mentioning that the path reduction procedure drives further the

accuracy	reduc.	value	gap (%)	time (s)	iter.	paths	min	avg.	max
$1.0e-5$	no	29.775924	$2.69e-2$	0.21	436	77	1	1.57	4
$1.0e-5$	yes	29.775924	$2.69e-2$	0.21	436	77	1	1.57	4
$1.0e-7$	no	29.758062	$3.28e-3$	4.97	10689	77	1	1.57	4
$1.0e-7$	yes	29.757950	$3.17e-3$	5.19	11182	76	1	1.55	4
$1.0e-9$	no	29.755306	$3.23e-4$	58.44	125911	77	1	1.57	4
$1.0e-9$	yes	29.754997	$1.40e-5$	60.11	129475	72	1	1.47	3

Fig. 4: Graph with 15 nodes, 38 arcs, 49 demands : (MFD)with heuristic path control

current solution towards the optimal solution, as it is shown by the relative gap. This procedure has then two advantages: reducing the number of paths and reducing the gap.

We have shown in this paper how to add some control on the number of active paths in the iterative process of minimizing congestion of a communication network. The Flow Deviation method is still the best basic tool to generate new shortest paths, but it must be modified carefully to avoid the generation of too many paths when reaching the optimal face of the multicommodity flow polyhedron. Our limited experimentation on small instances is currently being completed on larger networks where the number of potential paths is very large.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows : Theory, Algorithms and Applications*. Prentice-Hall, Englewood Cliffs, 1993.

[2] D. Bertsekas and R. Gallager. *Data networks*. Prentice-Hall, Englewood Cliffs, 1987.

[3] D. Bienstock and O. Günlük. Computational experience with a difficult mixed-integer multicommodity flow problem. *Mathematical Programming*, 68:213–237, 1995.

[4] D. Bienstock and O. Raskina. Analysis of the flow deviation method for the maximum concurrent flow problem. *Mathematical Programming*, 68:213–237, 2001.

[5] L. Fratta, M. Gerla, and L. Kleinrock. The flow deviation method: an approach to store-and-forward computer-communication network design. *Networks*, 3, 1973.

[6] M. D. Grigoriadis and L. G. Khachiyan. Fast approximation schemes for convex programs with many blocks and coupling constraints. *SIAM Journal on Optimization*, 4(1):86–107, 1994.

[7] L. Leblanc, J. Chifflet, and P. Mahey. Packet routing in telecommunication networks with path and flow restrictions. *INFORMS Journal on Computing*, 11(2):188–197, 1999.

[8] F. T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos, and S. Tragoudas. Fast approximation algorithms for multicommodity flow problems. *Journal of Computer and System Sciences*, 50(2):228–243, 1995.

[9] A. Ouorou, P. Mahey, and J. Vial. A survey of algorihtms for multicommodity flow problems with convex costs. *Management Science*, 46:126–147, 2000.

[10] F. Shahrokhi and D. W. Matula. The maximum concurrent flow problem. *Journal of the ACM*, 37:318–334, 1990.