

# A Combinatorial Approximation Algorithm for the Multicommodity Flow Problem

David Coudert, Hervé Rivano, Xavier Roche

► **To cite this version:**

David Coudert, Hervé Rivano, Xavier Roche. A Combinatorial Approximation Algorithm for the Multicommodity Flow Problem. International Workshop on Approximation and Online Algorithms (WAOA'03), 2003, Budapest, Hungary. Springer, 2009, pp.193-230, 2003, Lecture Notes in Computer Science. <10.1007/b95598>. <inria-00429176>

**HAL Id: inria-00429176**

**<https://hal.inria.fr/inria-00429176>**

Submitted on 1 Nov 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A combinatorial approximation algorithm for the multicommodity flow problem<sup>\*</sup>

David Coudert   Hervé Rivano   Xavier Roche

CNRS-I3S-INRIA MASCOTTE – France Telecom R&D – ÉNS Lyon

**Abstract.** This work is motivated by the need for approximation algorithms for the integral multicommodity flow problem which arise in numerous optimization scenarios, including the design of telecommunication networks. We improve on one of the most efficient known combinatorial approximation algorithm for fractional multicommodity flow by using an incremental approach. This approach is validated by experimental results, which show a significant speed-up.

**keywords:** Multicommodity flow, combinatorial approximation, dynamic shortest paths, incremental algorithm.

The multicommodity flow problem is useful in numerous applications, especially when the issue is to compute paths for entities that are concurrent for some resources. In these cases, the resources are modeled by capacities on the edges of the graph bearing all possible routings.

If the multicommodity flow is constrained to be integer, the problem is  $\mathcal{NP}$ -hard. Nevertheless, Raghavan [Rag94] proposed a randomized algorithm constructing a multicommodity flow based on the randomized rounding of the linear relaxation of the problem, namely the fractional multicommodity flow. Recently an improvement of this algorithm has been proposed [CR02]. This algorithm computes a better approximation of the integer multicommodity flow, but at the cost of solving a fractional multicommodity flow for each commodity in an iterative process. Solving the fractional multicommodity flow efficiently is therefore a crucial issue.

Given that the fractional multicommodity flows is used to compute an approximated integral solution, there is no real need for the use of optimal fractional flows. Indeed, the approximation yielded by the randomized rounding process is an additive gap of order the square root of the fractional capacities [Rag94,CR02]. If these fractional capacities are  $(1 + \epsilon)$ -approximations of the optimal capacities, the magnitude of the final gap will not change dramatically. It is therefore natural to exploit this freedom in order to consequently speed up the integer multicommodity flow approximation algorithm. A first article [GK98] proposes a combinatorial algorithm computing a fractional multicommodity flow  $(1 + \epsilon)$ -approximation in time  $O(\epsilon^{-2}km^2n^2)$ , where  $k$  is the number of commodities,  $n$  and  $m$  are the number of vertices and edges of the flow network. It is then possible to avoid the linear program solvers, much more costly to run. This

---

<sup>\*</sup> Work funded by European project FET CRESCCO and action COLOR DYNAMIC.

algorithm has been improved by Fleischer [Fle00], decreasing the complexity to  $O(\epsilon^{-2}m^2 \log n)$ . Both algorithms are mainly based on sequences of shortest path computations. In a previous paper [BCL<sup>+</sup>03], we gave an improvement on Fleischer’s algorithm using dynamic shortest paths algorithms.

In the following we propose an incremental strategy to improve on the running time of these algorithms. We then investigate on our approach through practical tests on different flow networks. We first present the algorithm that we improve.

## 1 Combinatorial multicommodity flow approximation

The  $(1+\epsilon)$ -approximation algorithm for fractional multicommodity flow [BCL<sup>+</sup>03] is based on a combinatorial understanding of the dual of the multicommodity flow *edge-path* linear program. This dual builds a length function  $l$  on the edges of the graph such that the length of an edge is related to the amount of flow it bears.

The process starts by assigning to each edge the same initial length  $\delta > 0$ , a constant which depends on the parameters of the algorithm, including the approximation factor  $\epsilon$ , and corresponds to a null amount of flow. Then, the algorithm pushes flow iteratively along single source shortest paths (SSSP) for each commodity while the length of each edge of the paths is increased. The loop ends when every shortest paths from the source of a commodity to its destination are of length more than 1. At this step, the analysis shows that the flow scaled by  $\log_{1+\epsilon} \left( \frac{1+\epsilon}{\delta} \right)$  fits the capacities, and that the worst case complexity is  $O \left( \frac{m \log n}{\epsilon^2} (m + n \log n) \right)$ . One may notice that this complexity is the same as the one given by Fleischer [Fle00], even though the use of dynamic shortest paths algorithms yields an improvement of the running time. As a matter of fact, the speed up provided by such algorithms cannot be precisely evaluated unless the structure of the flow network is known.

In the following, we try to cope with the  $\epsilon^{-2}$  factor in the complexity.

## 2 An incremental approach

In order to ensure a  $(1 + \epsilon)$ -approximation, each iteration of the previous algorithm pushes a very small amount of flow. We propose an incremental approach based on the idea that the precision is only required at the end of the process. Therefore, our algorithm, reported as Algorithm 1, starts by running the previous algorithm with a large  $\epsilon_0$ , which pushes a lot of flow quickly. Instead of scaling the flow to fit the capacities, Algorithm 1 scales it with the length function so that it looks like an intermediary step of previous algorithm with a smaller  $\epsilon_1 = \epsilon_0/2$ . Starting from this intermediary step, the process is finished with  $\epsilon_1$ , scaled again for  $\epsilon_2 = \epsilon_1/2$  and so forth, until Algorithm 1 reaches the required  $\epsilon_f$ .

---

**Algorithm 1** Incremental multicommodity flow  $(1 + \epsilon)$ -approximation
 

---

**Require:**  $G = (V, E)$ ,  $|V| = n$ ,  $|E| = m$ ,  $C = \{(s_i, t_i), i = 1 \dots k\} \subseteq V^2$ ,  $\epsilon > 0$

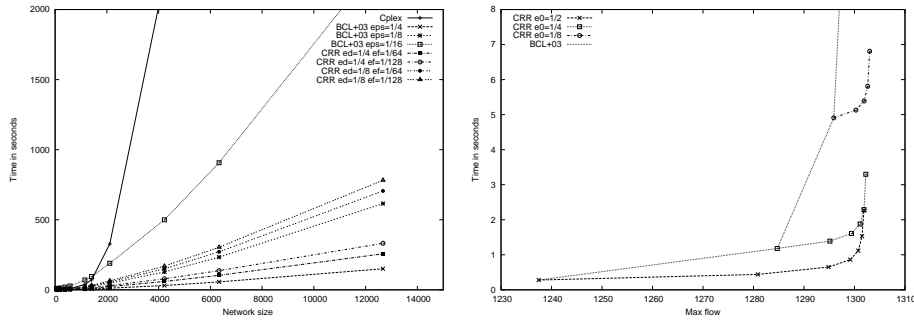
**Ensure:**  $l$  length function over  $E$  s.t. every path  $s_i \rightarrow t_i$  has length  $> 1$ .

**Ensure:**  $f$   $(1 + \epsilon)$ -approximation of max. multicommodity flow for  $C$  on  $G$

- 1: {Initializing}  $t = 4$ ,  $\epsilon(t) = 2^{-t}$ ,  $\delta(t) = e^{\epsilon(t)}(e^{\epsilon(t)}n)^{-\frac{1}{\epsilon(t)}}$ ,  $B(t) = \log_{e^{\epsilon(t)}}\left(\frac{e^{\epsilon(t)}}{\delta(t)}\right)$ ,  
 $\forall e \in E, l(e) = \delta(t)$ ,  $f_i(e) = 0$
  - 2: **while**  $\epsilon(t) \geq \epsilon_0$  **do**
  - 3:   {Lower bound on the length of a SSSP }  $\lambda = \min_{e \in E} l(e)$
  - 4:   **while**  $\lambda \leq 1$  **do**
  - 5:     **for all**  $i = 1 \dots k$  **do**
  - 6:        $P \leftarrow \text{SSSP}(s_i \rightarrow t_i)$
  - 7:       **while**  $l(P) \leq e^{\epsilon(t)}\lambda$  **do**
  - 8:          $c_m \leftarrow \min_{e \in P} c(e)$
  - 9:          $\forall e \in P, f_i(e) \leftarrow f_i(e) + c_m$ ,  $l(e) \leftarrow l(e)e^{\epsilon(t)\frac{c_m}{c(e)}}$
  - 10:         $P \leftarrow \text{USSSP}(s_i \rightarrow t_i)$
  - 11:         $\lambda \leftarrow \lambda e^{\epsilon(t)}$
  - 12:      $t \leftarrow t + 1$
  - 13:    $\forall e \in E, l(e) \leftarrow l(e)\frac{\delta(t)}{\delta(t-1)}e^{\epsilon(t)\frac{f(e)}{c(e)}\left(\frac{B(t)}{B(t-1)} - 2\right)B(t-1)}$
  - 14:    $\forall e \in E, \forall i = 1 \dots k, f_i(e) = \frac{B(t)}{B(t-1)}f_i(e)$
  - 15:  $\forall i = 1 \dots k, \forall e \in E, f_i(e) = \frac{f_i(e)}{B(t)}$
- 

### 3 Experimental results

The analysis of the complexity and approximation performances of Algorithm 1 is a work in progress. Nevertheless, experimental results are encouraging. We report on a set of experiments achieved on variable size multicommodity flow problems, arising in the setting of routing and wavelength assignment for wavelength division multiplexing optical networks [CR02]. The results are depicted as Figure 1.



**Fig. 1.** Running times function of network size (left) or quantity of flow (right).

The left hand side of Figure 1 presents the running time of the optimal resolution of the flow with CPLEX, the algorithm of [BCL<sup>+</sup>03] with  $\epsilon$  varying from 1/4 to 1/16, and Algorithm 1 (named CRR) with  $\epsilon_0 = 1/4$  or 1/8 and  $\epsilon_f = 1/64$  or 1/128. One can see that both combinatorial algorithms are much faster than CPLEX when the size of the problem increases. These results also show that the running time of Algorithm 1 for a given  $\epsilon_0$  equals to the running time of Algorithm [BCL<sup>+</sup>03] for  $\epsilon = \epsilon_0$  plus something small and linear in  $\epsilon_f^{-1}$ .

The right hand side presents the time required by the different combinatorial algorithms to produce a given approximation of the maximum multicommodity flow. These experiments have been achieved on the network of size 500, but the same results are observed with any size. As far as Algorithm 1 for  $\epsilon_0 = \epsilon_f$  is exactly Algorithm [BCL<sup>+</sup>03] for the same parameter, we obtain this *comb graph* like results. One can then realize that Algorithm 1 is able to produce approximated solution much faster than Algorithm [BCL<sup>+</sup>03]. Moreover, the influence of  $\epsilon_0$  on the quality of the solutions of Algorithm 1 is weak, but very strong on the running time.

## 4 Conclusion

In this paper, we have proposed a combinatorial incremental  $(1+\epsilon)$ -approximation algorithm for the fractional multicommodity flow problem, improving on several algorithms of the literature. Practical experiments yield some conjectures on its complexity and approximation performances. A formal analysis of the algorithm is an ongoing work, as well as its integration in an integral multicommodity flow approximation algorithm and further improvements.

## References

- [BCL<sup>+</sup>03] M. Bouklit, D. Coudert, J.-F. Lalande, C. Paul, and H. Rivano. Approximate multicommodity flow for wdm networks design. In *Sirocco 10*, Umea, Sweden, june 2003.
- [BT97] D. Bertimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [CR02] D. Coudert and H. Rivano. Lightpath assignment for multifibers WDM optical networks with wavelength translators. In *IEEE Globecom'02*, Taiwan, November 2002. OPNT-01-5.
- [FF62] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- [Fle00] L. Fleischer. Approximating fractional multicommodity flows independent of the number of commodities. *SIAM J. Discrete Math.*, 13(4):505–520, 2000.
- [GK98] N. Garg and J. Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.
- [Rag94] P. Raghavan. Probabilistic construction of deterministic algorithm: Approximating packing integer programs. *Journal of Computer and Systems Sciences*, 38:683–707, 1994.