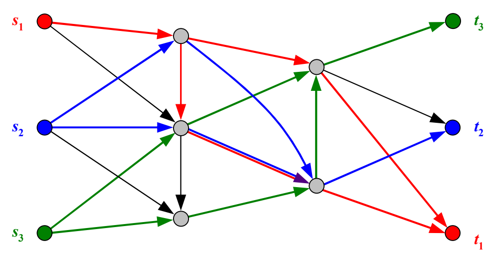
Efficient Multicommodity Flow Algorithms

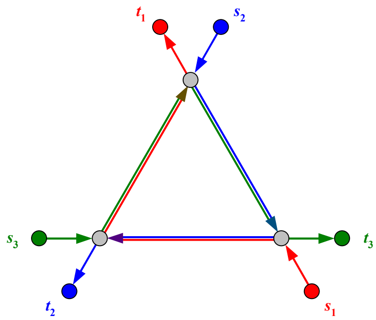
Péter Kovács, PhD student

ELTE CNL, kpeter@inf.elte.hu, supervisor: Zoltán Király

Multicommodity Flows

- Network flow problems with *multiple commodities* (or goods).
- Separate source and sink nodes for each commodity: s_i, t_i $(i = 1, \ldots, k)$.
- The commodities share the edge capacities.
- Various problems with different constraints and object functions.





- Applications:
  - telecommunication network design,
  - traffic engineering,
  - VLSI design,
  - resource planning etc.

Problems

- *Maximum Multicommodity Flow (MMF):* maximize the sum of the flow values of the commodities, i.e. $\max \sum |f_i|$.
  A possible LP formulation (Π denotes the set of all s_i–t_i paths):

$$\textbf{(P)}\quad \max \sum_{P\in\Pi} x(P) \qquad \sum_{P:e\in P} x(P) \le u(e) \quad \forall e \in E \qquad x(P) \ge 0 \quad \forall P \in \Pi$$

$$\textbf{(D)}\quad \min \sum_{e\in E} u(e)l(e) \qquad \sum_{e\in P} l(e) \ge 1 \quad \forall P \in \Pi \qquad l(e) \ge 0 \quad \forall e \in E$$

- *Maximum Weighted Multicommodity Flow (MWMF):* each commodity has a weight w_i, $\max \sum w_i |f_i|$.
- *Maximum Concurrent Flow (MCF):* each commodity has a demand value d_i, maximize the factor λ, for which $|f_i| \ge \lambda d_i$.
- *Minimum Cost Concurrent Flow (MCCF):* maximum concurrent flow with a given upper bound for the total cost.

Solution Methods

- Multicommodity flows yield rather complex optimization problems.
- Exact solution methods are usually too slow for them.
- Developing much faster *approximation algorithms* is worthwhile.
- The framework of Garg and Könemann provides simple and efficient approximation methods for the above problems.
- Lisa K. Fleischer introduced theoretical improvements for them.
- All these methods are based on a *primal–dual* approach (column generation) attained by successive improvements along shortest paths with respect to a varying length function (the dual solution).
- This poster presents the experiments of efficient implementations of these algorithms.

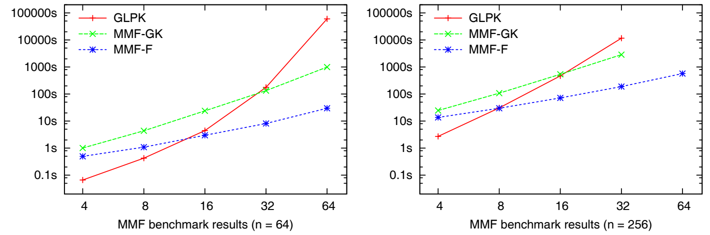
Implementation and Testing

- The **LEMON** C++ graph library was used, http://lemon.cs.elte.hu.
- Validity and benchmark tests were performed on various random instances, which were generated with MNETGEN.
- The results were compared to the exact solutions using GLPK.

LEMON Graph Library

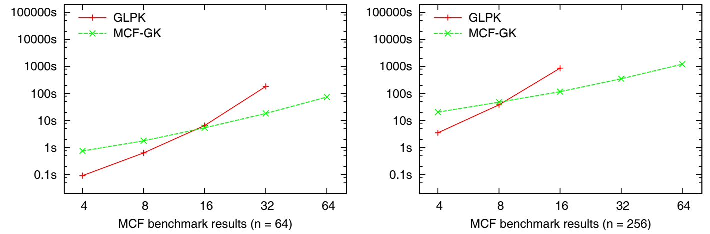
Maximum Multicommodity Flow

- Fleischer reduced the number of shortest path computations by a factor of k for the MMF problem. It results in a much more efficient algorithm.
- Our improvement is a simple *additional heuristic*: after executing the algorithm send flow from s_i to t_i using only the remaining capacities (for each commodity i).
- This heuristic increased the solution value by 1–5% on average. Applying it with approximation factor $r \le 2$, the results were usually equal to the exact optimum.
- The following charts show the running times of our implementations (using $r = 2$) and GLPK as a function of the number of commodities (k).



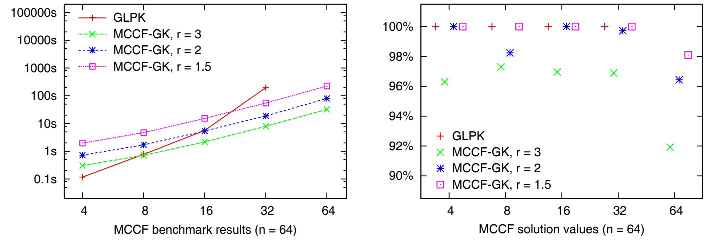
Maximum Concurrent Flow

- Similar heuristic improvement was introduced to the MCF algorithm of Garg and Könemann, which increased the solution value by 1–8%.
- Our implementation (using $r = 2$) is compared to GLPK on the diagrams.



Minimum Cost Concurrent Flow

- Our heuristic was also applied to the MCCF problem with suitable modifications.
- These charts demonstrate the running times and the obtained solution values using our implementation with different approximation factors and GLPK.



Conclusions

- The approximation algorithms are *asymptotically faster* than the exact solution using GLPK.
- For the MMF problem Fleischer's algorithm is also asymptotically faster than the original method and it is much more efficient in practice, as well.
- Our simple heuristic improvements proved to be really effective.