Technische Universität Berlin

FACHBEREICH 3
MATHEMATIK

Optimal Routing of Traffic Flows with Length Restrictions in Networks with Congestion

by

Olaf Jahn   Rolf H. Möhring
Andreas S. Schulz

No. 658/1999

Optimal Routing of Traffic Flows with Length Restrictions in Networks with Congestion

Olaf Jahn
Rolf H. Möhring
Fachbereich Mathematik, Technische Universität Berlin
http://www.math.tu-berlin.de/coga/

Andreas S. Schulz
Sloan School of Management, MIT, Cambridge

Summary: When traffic flows are routed through a road network it is desirable to minimize the total road usage. Since a route guidance system can only recommend paths to the drivers, special care has to be taken not to route them over paths they perceive as too long. This leads in a simplified model to a nonlinear multicommodity flow problem with constraints on the available paths. In this article an algorithm for this problem is given, which combines the convex combinations algorithm by Frank and Wolfe with column generation and algorithms for the constrained shortest path problem. Computational results stemming from a cooperation with DaimlerChrysler are presented.

1 Introduction

More and more vehicles get equipped with so-called route guidance systems. They guide the driver by visual and acoustic indicators to the destination, which has been entered at the beginning of the journey. Normally these systems compute their routes based on digital maps, the current position obtained by Global Positioning System (GPS), and possibly up-to-date traffic data, which are broadcast by radio or cellular phone.

A number of surveys and simulations show that route guidance systems decrease the overall road usage since inefficiencies caused by human route choice are mostly eliminated. Unfortunately, many simulations also predict that these benefits will be lost once the number of equipped vehicles exceeds a certain threshold.

This phenomenon can be explained by the fact that the perceivable decrease of the overall journey time under a low percentage of equipped vehicles is merely a fortunate byproduct of the system. Current algorithms try to minimize the individual journey time of each driver separately, without taking into account the effects of their own route recommendation. It is possible that such a system actually causes congestion all by itself. Thus the need for integrated algorithms that actually pay attention to the overall road usage (which can be viewed as the sum of all individual journey times) has been recognized [4].

Many techniques proposed in the literature use feedback loops, which iteratively assign traffic to the network and compute new arc traversal times based on this assignment (and perhaps previous ones). These approaches are often somewhat unsatisfactory from a mathematical point of view because it is not clear what the solution actually converges to, or if it even converges at all since usually "convergence" is established only empirically.

It is certainly desirable from a global point of view to *minimize* the overall road usage, which would allow the existing road network to carry more traffic. Yet special care has to be taken to respect the human factor involved: Only very few drivers would be willing to sacrifice their own short routes for the benefit of others.

For this reason we take a minimization approach but impose on the path assigned to a vehicle the constraint that it is geographically not longer than, say, the shortest possible path (in the uncongested network) plus a few "acceptable" percent. The effects of length constraints are demonstrated in figure 1. This contrasts with the common approach to route drivers along the paths of a so-called *user equilibrium* as introduced by Wardrop [11], where no driver can get a quicker path through the network by unilaterally changing his route. While this should satisfy the drivers, it does not necessarily decrease the overall road usage.

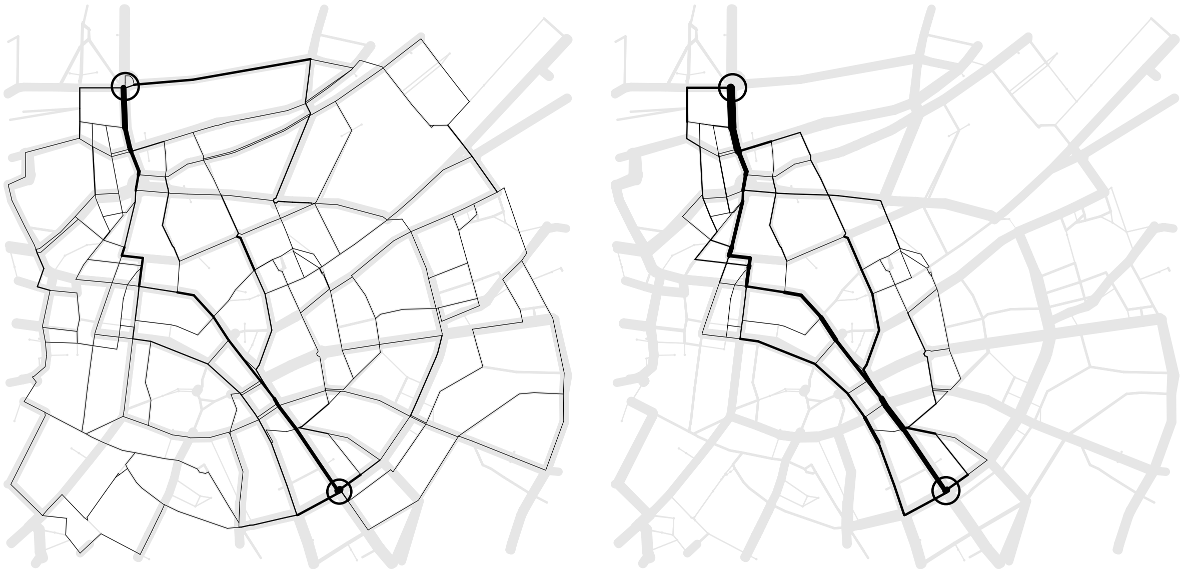

Figure 1: One commodity routed through the road network between the marked nodes. In the left hand side image flow is distributes over the network in order to avoid high arc flows that would incur high arc traversal times. In the right hand image the same amount of flow is routed, but this time with a restriction for the path lengths. Line thickness denotes arc capacity (light gray) and arc usage (black).

In this article we give a simplified model for this problem, an algorithm and computational results. This text is an eclectic summary of one of the author's (Jahn) diploma thesis written at TU Berlin in 1998, under the supervision by the other two authors [7]. It is part of an ongoing cooperation with DaimlerChrysler AG, Berlin.

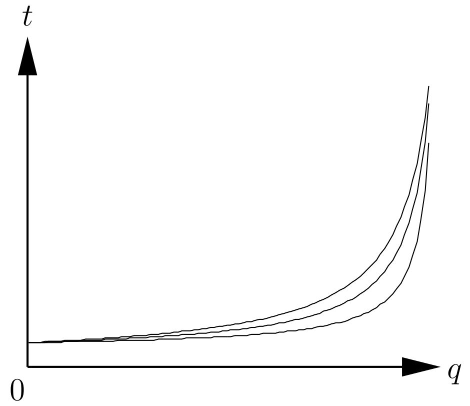

Figure 2: Typical link delay functions. q is the arc flow, and t the travel time.

2 Model and Problem Formulation

We consider a simplified model, in which static flows instead of discrete vehicles are routed through a network with arc traversal times depending nonlinearly on arc usage. This simplification is feasible for time periods when traffic exhibits a flow-like behaviour, e.g. during rush hours. In addition, geographic length restrictions are imposed on all paths.

We use a path based problem formulation with one variable for each feasible path. This has the advantage that it enables us to impose restrictions on paths very easily. As we cannot enumerate all these variables efficiently, we generate them only when they are needed to carry flow.

Let the road network be given by a directed multigraph $G = (V, A)$ with three attributes on each arc $a \in A$: the capacity $u_a \geq 0$ (measured in vehicles per time unit), the link delay function $t_a\colon [0, u_a] \to \mathbf{R}_0^+$ denoting the arc travel time depending on the traffic flow on this arc, and the geographic length $\ell_a \geq 0$ of this arc. We require t_a to be monotonically increasing and twice continuously differentiable. These requirements are naturally met by common link delay functions [9, 10] (fig. 2).

Vehicles with the same source and destination nodes are modeled as commodities. For each commodity k let $(s_k, t_k) \in V \times V, s_k \neq t_k$, denote its source and destination node, and let $b_k > 0$ be the amount of flow to be routed for k (vehicles per time unit). These data together form the so called origin-destination matrix. In addition we want the geographic length of all paths assigned to k not to exceed $L_k > 0$. The set of all commodities is called C. We define path sets $P_k := \{p \,|\, p \text{ is a directed path from } s_k \text{ to } t_k\}$, $P'_k := \{p \in P_k \,|\, \ell(p) \leq L_k\}$ with $\ell(p) := \sum_{a \in p} \ell_a$, and $P' := \bigcup_{k \in C} P'_k$. Without loss of generality we can assume that no two commodities have the same source and destination nodes. This means that each path $p \in P'$ is uniquely related to a commodity.

Let $\Phi \in \{0, 1\}^{|A| \times |P'|}$ be the arc-path incidence matrix, i.e., the vectors $x^A \geq 0$ and $x^P \geq 0$ of arc and path flows are related by $x^A = \Phi x^P$. Similarly, let $\Psi \in \{0, 1\}^{|C| \times |P'|}$

be the commodity-path incidence matrix.

We want to minimize the overall road usage, which can be stated as the sum of all path journey times, weighted with the flow on each path. With the vectors $t^A(x^A) \in \mathbf{R}^{|A|}$, $b \in \mathbf{R}^{|C|}$ and $u \in \mathbf{R}^{|A|}$ composed of the t_a, b_k and u_a, respectively, our problem eventually can be written as

$$\begin{array}{lll} \mathcal{P}: & \min & z^P(x^P) := t^A(\Phi x^P)^\intercal \Phi x^P \\ & \text{s. t.} & \\ & & \Psi x^P = b \\ & & \Phi x^P \leq u \\ & & x^P \geq 0. \end{array}$$

Of course, P' is given only implicitly. Thus $\mathcal{P}$ is a multicommodity flow minimum cost problem with path constraints and a convex nonlinear objective function. Obviously, $\mathcal{P}$ is a generalization of the well known constrained shortest paths problem [6]. Therefore it is NP-hard. Note that the flow variables x^P are not required to be integral.

3 Algorithms

Due to space limitations we can only give the general ideas of the used algorithms and their interaction. Details can be found in [7]. The algorithm we propose for $\mathcal{P}$ needs to solve a linearized version of $\mathcal{P}$ in every iteration, whose algorithm in turn relies heavily on an algorithm for the constrained shortest path problem.

3.1 The Linearized Problem

Let us first consider $\mathcal{P}$ with constant link delay functions t_a. Then $\mathcal{P}$ is a linear program with additional constraints for the paths, which still renders it NP-hard. Since we have got a variable for each possible path we can easily ensure that all paths fulfill the length restriction (or other restrictions we might think of). We cannot store all variables in memory at the same time. Therefore we use the simplex algorithm with column generation to generate new paths when we need to route flow over them (see [1] for this technique applied to the ordinary multicommodity flow problems).

In order to check the reduced costs of the non-basic variables, we have to perform $|C|$ independent computations of constrained shortest paths, which can be done in parallel.

3.2 The Constrained Shortest Path Problem

Two algorithms for the constrained shortest path problem have been implemented for the diploma thesis: A branch-and-bound scheme with Lagrangian relaxation by Beasley and Christofides [3], and a labeling algorithm with lists of labels for each node by Aneja, Aggarwal, and Nair [2].

Computational experience indicates that at least for our problems the latter algorithm is by far superior in its performance. Both algorithms need to compute ordinary shortest paths many times. We use Dijkstra's algorithm for this.

3.3 The Nonlinear Problem

We use the Convex combination algorithms by Frank and Wolfe [5] in the version described by Klessig [8]. It is a feasible direction method for problems with convex objective function and convex domain, especially suitable when the linearized problem to be solved in each iteration decomposes nicely, as is the case with $\mathcal{P}$. It can be shown that this algorithm terminates with an optimal solution or that at least every accumulation point of the generated sequence is optimal if the computed step sizes guarantee a decrease of the objective value.

In this algorithm we have to solve a linearized problem of type $\mathcal{P}$ in each iteration with the objective function $z'(y^P) := \nabla z^P(x_i^P)^\intercal y^P$. We cannot compute $\nabla z^P(x_i^P)$ since enumerating all paths is impractical. Fortunately, $z'(y^P)$ can be expressed in terms of arc flows instead of path flows, which leads to a linearized problem as described in section 3.1 with modified arc costs. This enables us to use column generation for the nonlinear problem as well.

This algorithm usually descends very quickly towards the minimum. Thus computation can be stopped after a handful of iterations, as there is not much to be gained from subsequent iterations.

4 Computational Experience

The algorithms described in the previous section have been implemented in C++. We use the CPLEX callable library to solve the restricted linear programs and the PVM library to facilitate interprocess communication when doing constrained shortest path computations in parallel on several workstations. A number of computations were run on a real street network (or parts of it). The origin-destination matrices partly stem from polls, partly they were generated randomly by us. We compute for each commodity k the geographic length of a shortest path (w.r.t. the travel times) in the empty network and multiply it with a constant $f > 1$ to obtain realistic length constraints L_k. Naturally, as these constraints become tighter the problem tends to get more difficult.

A number of results are shown in table 1. Most of the computation time is spent computing constrained shortest paths, which means that improved algorithms for this subproblem would yield better overall performance.

Further research will investigate the transition to more sophisticated dynamic models, in which single vehicles or platoons of vehicles are considered.

For more information about this project please visit our WWW page at `http://www.math.tu-berlin.de/coga/research/route_guidance/`

$\lvert V \rvert$	$\lvert A \rvert$	$\lvert C \rvert$	f	iterations	objective value	computation time in secs
1616	2476	800	2.00	3	3888793.1	41
				5	3820167.6	69
				8	3796657.5	119
			1.50	3	3883454.6	191
				5	3818633.4	224
				8	3793936.9	310
			1.20	3	3864530.7	1213
				5	3816507.5	1401
				8	3797589.6	1900
			1.10	3	3845923.2	2058
				5	3813543.7	3024
				8	3805159.6	4413
			1.05	5	3853012.0	3946
				8	3847290.2	6419
		1000	2.00	3	4911722.5	99
				5	4819254.9	141
				8	4762865.7	228
			1.50	3	4880698.7	1154
				5	4809073.3	1185
				8	4765599.0	1422
			1.20	3	4834641.3	3240
				5	4788056.3	3539
				8	4756203.8	4435
			1.10	3	4839325.5	4272
				5	4802003.0	5866
				8	4781432.3	8917
			1.05	5	4846630.7	6265
				8	4834892.7	9856
694	2090	3058	2.00	3	5959776.1	490
				5	5853209.0	603
				8	5785542.5	852
			1.50	3	5946063.0	4507
				5	5850489.0	4872
				8	5790798.5	5963
			1.20	3	6895477.8	6820
				8	6151288.0	20687
			1.10	5	6915459.8	10190
				8	6805311.9	15843

Table 1: Some computational results obtained on a standard PC with Intel Pentium II, 266 MHz, running Linux. The graphs are parts of a real city street network. The origin-destination matrices with 800 and 1000 commodities have been generated randomly, the other matrix is obtained from traffic counts (for this centroids have to be added to the graph).

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows*. Prentice-Hall, New Jersey, 1993.

[2] Y. P. Aneja, V. Aggarwal, and K. P. K. Nair. Shortest chain subject to side constraints. *Networks*, 13:295–302, 1983.

[3] J. E. Beasley and N. Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19:379–394, 1989.

[4] G. Beccaria and A. Bolelli. Modelling and assessment of dynamic route guidance: the MARGOT project. In *Vehicle Navigation & Information Systems Conference Proceedings, 1992, Oslo (VNIS '92)*, pages 117–126. IEEE, 1992.

[5] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[6] Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[7] Olaf Jahn. Multicommodity Flow-Modelle und Algorithmen zur dynamischen Lenkung von Verkehrsströmen. Diploma thesis, Fachbereich Mathematik, Technische Universität Berlin, 1998.

[8] R. W. Klessig. An algorithm for nonlinear multicommodity flow problems. *Networks*, 4:343–355, 1974.

[9] Markos Papageorgiou, editor. *Concise Encyclopedia of Traffic & Transportation Systems*. Pergamon Press, Oxford, 1991.

[10] Yosef Sheffi. *Urban Transportation Networks*. Prentice-Hall, New Jersey, 1985.

[11] John Glen Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(2):325–362, 1952.

Reports from the group

"Combinatorial Optimization and Graph Algorithms"

of the Department of Mathematics, TU Berlin

658/1999 *Olaf Jahn, Rolf H. Möhring, and Andreas S. Schulz:* Optimal Routing of Traffic Flows with Length Restrictions in Networks with Congestion

655/1999 *Michel X. Goemans and Martin Skutella:* Cooperative facility location games

654/1999 *Michel X. Goemans, Maurice Queyranne, Andreas S. Schulz, Martin Skutella, and Yaoguang Wang:* Single Machine Scheduling with Release Dates

653/1999 *Andreas S. Schulz and Martin Skutella:* Scheduling unrelated machines by randomized rounding

646/1999 *Rolf H. Möhring, Martin Skutella, and Frederik Stork:* Forcing Relations for AND/OR Precedence Constraints

640/1999 *Foto Afrati, Evripidis Bampis, Chandra Chekuri, David Karger, Claire Kenyon, Sanjeev Khanna, Ioannis Milis, Maurice Queyranne, Martin Skutella, Cliff Stein, and Maxim Sviridenko:* Approximation Schemes for Minimizing Average Weighted Completion Time with Release Dates

639/1999 *Andreas S. Schulz and Martin Skutella:* The Power of α-Points in Preemptive Single Machine Scheduling

634/1999 *Karsten Weihe, Ulrik Brandes, Annegret Liebers, Matthias Müller–Hannemann, Dorothea Wagner and Thomas Willhalm:* Empirical Design of Geometric Algorithms

633/1999 *Matthias Müller–Hannemann and Karsten Weihe:* On the Discrete Core of Quadrilateral Mesh Refinement

632/1999 *Matthias Müller–Hannemann:* Shelling Hexahedral Complexes for Mesh Generation in CAD

631/1999 *Matthias Müller–Hannemann and Alexander Schwartz:* Implementing Weighted b-Matching Algorithms: Insights from a Computational Study

629/1999 *Martin Skutella:* Convex Quadratic Programming Relaxations for Network Scheduling Problems

628/1999 *Martin Skutella and Gerhard J. Woeginger:* A PTAS for minimizing the total weighted completion time on identical parallel machines

627/1998 *Jens Gustedt:* Specifying Characteristics of Digital Filters with FilterPro

620/1998 *Rolf H. Möhring, Andreas S. Schulz, Frederik Stork, and Marc Uetz:* Resource Constrained Project Scheduling: Computing Lower Bounds by Solving Minimum Cut Problems

619/1998 *Rolf H. Möhring, Martin Oellrich, and Andreas S. Schulz:* Efficient Algorithms for the Minimum-Cost Embedding of Reliable Virtual Private Networks into Telecommunication Networks

618/1998 *Friedrich Eisenbrand and Andreas S. Schulz:* Bounds on the Chvátal Rank of Polytopes in the 0/1-Cube

617/1998 *Andreas S. Schulz and Robert Weismantel:* An Oracle-Polynomial Time Augmentation Algorithm for Integer Proramming

616/1998 *Alexander Bockmayr, Friedrich Eisenbrand, Mark Hartmann, and Andreas S. Schulz:* On the Chvátal Rank of Polytopes in the 0/1 Cube

615/1998 *Ekkehard Köhler and Matthias Kriesell:* Edge-Dominating Trails in AT-free Graphs

613/1998 *Frederik Stork:* A branch and bound algorithm for minimizing expected makespan in stochastic project networks with resource constraints

612/1998 *Rolf H. Möhring and Frederik Stork:* Linear preselective policies for stochastic project scheduling

609/1998 *Arfst Ludwig, Rolf H. Möhring, and Frederik Stork:* A computational study on bounding the makespan distribution in stochastic project networks

605/1998 *Friedrich Eisenbrand:* A Note on the Membership Problem for the Elementary Closure of a Polyhedron

596/1998 *Andreas Fest, Rolf H. Möhring, Frederik Stork, and Marc Uetz:* Resource Constrained Project Scheduling with Time Windows: A Branching Scheme Based on Dynamic Release Dates

595/1998 *Rolf H. Möhring Andreas S. Schulz, and Marc Uetz:* Approximation in Stochastic Scheduling: The Power of LP-based Priority Policies

591/1998 *Matthias Müller–Hannemann and Alexander Schwartz:* Implementing Weighted b-Matching Algorithms: Towards a Flexible Software Design

590/1998 *Stefan Felsner and Jens Gustedt and Michel Morvan:* Interval Reductions and Extensions of Orders: Bijections to Chains in Lattices

584/1998 *Alix Munier, Maurice Queyranne, and Andreas S. Schulz:* Approximation Bounds for a General Class of Precedence Constrained Parallel Machine Scheduling Problems

577/1998 *Martin Skutella:* Semidefinite Relaxations for Parallel Machine Scheduling

Reports may be requested from:

S. Marcus
Fachbereich Mathematik, MA 6–1
TU Berlin
Straße des 17. Juni 136
D-10623 Berlin – Germany

e-mail: Marcus@math.TU-Berlin.DE

Reports are also available in various formats from

`http://www.math.tu-berlin.de/coga/publications/techreports/`

and via anonymous ftp as

`ftp://ftp.math.tu-berlin.de/pub/Preprints/combi/Report-`*number*`-`*year*`.ps`