Technische Universität Berlin

FAKULTÄT II
MATHEMATIK UND
NATURWISSENSCHAFTEN
Institut für Mathematik

OPTIMAL ROUTE ASSIGNMENT IN LARGE SCALE MICRO-SIMULATIONS

by

MICHAEL BALMER[1] TORBEN EDELHOFF[2]
ROLF H. MÖHRING[2] HEIKO SCHILLING[2]

No. 2006/014

Optimal Route Assignment in Large Scale Micro-Simulations*

Michael Balmer[1], Torben Edelhoff[2], Rolf H. Möhring[2], and Heiko Schilling[2]

[1] Institute for Transport Planning and Systems, ETH Zurich, Switzerland
[2] Institute of Mathematics, TU Berlin, Germany

Abstract. Traffic management and route guidance are optimization problems by nature. In this article, we consider algorithms for centralized route guidance and discuss fairness aspects for the individual user resulting from optimal route guidance policies. The first part of this article deals with the mathematical aspects of these optimization problems from the viewpoint of network flow theory. We present algorithms which solve the constrained multicommodity minimum cost flow problem (CMCF) to optimality. A feasible routing is given by a flow x, and the cost of flow x is the total travel time spent in the network. The corresponding optimum is a restricted system optimum with a globally controlled constrained or fairness factor $L > 1$. This approach implements a compromise between user equilibrium and system optimum. The goal is to find a route guidance strategy which minimizes global and community criteria with individual needs as constraints. The fairness factor L restricts the set of all feasible routes to the subset of acceptable routes. This might include the avoidance of routes which are much longer than shortest routes, the exclusion of certain streets, preferences for scenic paths, or restrictions on the number of turns to be taken. Most remarkably is that the subset of acceptable routes can also be interpreted as a mental map of routes.

In the second part we apply our CMCF algorithms in a large scale multi-agent transportation simulation toolkit, which is called MATSIM-T. We use as initial routes the ones computed by our CMCF algorithms. This choice of initial routes makes it possible to exploit the optimization potential within the simulation much better then it was done before. The result is a speed up of the iteration process in the simulation. We compare the existing simulation toolkit with the new integration of CMCF to proof our results.

1 Introduction

In view of the steadily growing car traffic and the limited capacity of our street networks, we are facing a situation where methods for better traffic management are becoming more and more important. Studies [34] show that an individual blind choice of routes leads to travel times that are between 6% and 19% longer than necessary. On the other hand, telematics and sensory devices are providing detailed information about the actual traffic flows, thus making data available that is necessary to employ better means of traffic management.

Traffic management and route guidance are optimization problems by nature. We want to utilize the available street network in such a way that the total network *load* is minimized or the *throughput* is maximized. The first part of this article deals with the mathematical aspects of these optimization problems from the viewpoint of network flow theory. This is, in our opinion, currently the most promising method to get structural insights into the behavior of traffic flows in large, real-life networks. It also provides a bridge between variational inequalities [14] and traffic simulation [9,38] or models based on fluid dynamics [40], which are two other common approaches to study traffic problems.

Fluid models and other models based on differential equations are well suited to capture the dynamical behavior of traffic as a continuous quantity, but they can not yet handle large networks. On the other hand, simulation is a powerful tool to evaluate traffic scenarios, but it misses the optimization potential. The second part of this article can be seen as a first step towards the exploitation of the optimization potential within traffic simulations.

* This work was partly supported by the German Research Foundation (DFG grant MO 446/5-2).

Traffic flows have two important features that make them difficult to study mathematically. One is *congestion*, and the other is *time*. Congestion captures the fact that travel times increase with the amount of flow on the streets, while time refers to the movement of cars or agents (in simulations) along a path as a flow over time. The mathematical models of traffic flows become increasingly more difficult as they capture more and more of these two features. The section on static flow problems deals with flows both without and with congestion. The section gives an introduction to the classical static network flow theory and discusses basic results in this area. The presented insights are also at the core of more complex models and algorithms discussed later on in this section when we add congestion to the static flow model. The static flow with congestion becomes a realistic traffic model for rush-hour traffic where the effect of flow changing over time is secondary compared to the immense impact of delays due to congestion. We consider algorithms for centralized route guidance and discuss fairness aspects for the individual user resulting from optimal route guidance policies.

Although this article presents new and innovative approaches in the field, the development of suitable mathematical tools and algorithms which can deal with large real-world traffic networks remains an open problem. The inherent complexity of many traffic flow problems constitutes a major challenge for future research.

2 Static Flows

An exhaustive mathematical treatment of network flow theory started around the middle of the last century with the ground-breaking work of Ford and Fulkerson [17]. Historically, the study of network flows mainly originates from problems related to the transportation of materials or goods, for instance, for instance, see [24,27,33]. For a detailed survey of the history of network flow problems we refer to the work of Schrijver [44].

2.1 Static Flow Problems

Usually, flows are defined on networks (directed graphs) $G = (V, E)$ with *capacities* $u_e \geq 0$ and, in some settings, also *costs* c_e on the arcs $e \in E$. The set of nodes V is partitioned into *source* nodes, *intermediate* nodes, and *sink* nodes. On an intuitive level, flow emerges from the source nodes, travels through the arcs of the network via intermediate nodes to the sinks, where it is finally absorbed. More precisely, a *static flow* x assigns a value $0 \leq x_e \leq u_e$ to every arc $e \in E$ of the network such that for every node $v \in V$

$$\sum_{e \in \delta^-(v)} x_e - \sum_{e \in \delta^+(v)} x_e \begin{cases} \leq 0 & \text{if } v \text{ is a source,} \\ = 0 & \text{if } v \text{ is an intermediate node,} \\ \geq 0 & \text{if } v \text{ is a sink.} \end{cases} \tag{1}$$

Here, $\delta^+(v)$ and $\delta^-(v)$ denote the set of arcs e leaving node v and entering node v, respectively. Thus, the left hand side of (1) is the net amount of flow entering node v. For intermediate nodes this quantity must obviously be zero; this requirement is usually referred to as *flow conservation constraint*. A flow x is said to *satisfy the demands and supplies* d_v, $v \in V$, if the left hand side of (1) is equal to d_v for every node $v \in V$. In this setting, nodes v with negative demand d_v (i.e., positive supply $-d_v$) are sources and nodes with positive demand are sinks. A necessary condition for the existence of such a flow is $\sum_{v \in V} d_v = 0$. Observe that the sum of the left hand side of (1) over all $v \in V$ is always equal to 0.

A flow problem in a network $G = (V, E)$ with several sources $S \subset V$ and several sinks $T \subset V$ can easily be reduced to a problem with a single source and a single sink: introduce a new *super-source* s and a new *super-sink* t. Then add an arc (s, v) of capacity $u_{(s,v)} := -d_v$ for every source $v \in S$ and an arc (w, t) of capacity $u_{(w,t)} := d_w$ for every sink $w \in T$. In the resulting network with node set $V \cup \{s, t\}$ all nodes in V are intermediate nodes and we set and $d_s := \sum_{v \in S} d_v$ and $d_t := \sum_{w \in T} d_w$.

For the case of one single source s and one single sink t, the flow x is called an *s-t-flow* and the left hand side of (1) for $v = t$ is the *s-t-flow value* which we denote by $|x|$. Due to flow conservation constraints, the absolute value of the left hand side of (1) for $v = s$ also equals the flow value. In other words, all flow leaving the source must finally arrive at the sink. Ford and Fulkerson [16,17] and independently Elias, Feinstein, and Shannon [15] show in their so-called *Max-Flow-Min-Cut Theorem* that the maximum s-t-flow value is equal to the minimum capacity of an s-t-cut. An *s-t-cut* $\delta^+(S)$ is given by a subset of vertices $S \subseteq V \setminus \{t\}$, $s \in S$, and defined as the set of arcs going from S to its complement $V \setminus S$, i. e., $\delta^+(S) := (\bigcup_{v \in S} \delta^+(v)) \setminus (\bigcup_{v \in S} \delta^-(v))$. The *capacity* of an s-t-cut is the sum of the capacities of all arcs in the cut. Ford and Fulkerson [17] also observe that the Max-Flow-Min-Cut Theorem can be interpreted as a special case of linear programming duality. Today, a variety of efficient (i.e., polynomial time) algorithms are known for computing an s-t-flow of maximum value and a corresponding minimum capacity cut. We refer to the standard textbook by Ahuja, Magnanti, and Orlin [1] for a detailed account of results and algorithms in this area. A further structural result for s-t-flows we would like to mention here is one by Ford and Fulkerson [17]: any given s-t-flow can be decomposed into the sum of flows of value x_P on certain s-t-paths $P \in \mathcal{P}$ and flows of value x_C on certain cycles $C \in \mathcal{C}$, that is,

$$x_e = \sum_{\substack{P \in \mathcal{P} \\ e \in P}} x_P + \sum_{\substack{C \in \mathcal{C} \\ e \in C}} x_C \qquad \text{for all } e \in E.$$

Moreover, the number of paths and cycles in this decomposition can be bounded by the number of arcs, i.e., $|\mathcal{P}| + |\mathcal{C}| \leq |E|$.

In the setting with costs on the arcs, the cost of a static flow x is defined as $c(x) := \sum_{e \in E} c_e \, x_e$. For given demands and supplies d_v, $v \in V$, the *minimum cost flow problem* asks for a flow x with minimum cost $c(x)$ satisfying the demands and supplies . As for the maximum flow problem discussed above, various efficient algorithms are known for this problem and a variety of structural structural characterizations of optimal solutions has been derived. Again, we refer to [1] for details.

A static flow problem which turns out to be considerably harder than the maximum flow and the minimum cost flow problem is the *multicommodity flow problem*. Every *commodity* $i \in K$ is given by a source-sink pair $s_i, t_i \in V$ and a prescribed flow value d_i. The task is to find an s_i-t_i-flow x_i of value d_i for every commodity $i \in K$ such that the sum of these flows obeys the arc capacities, i.e., $\sum_{i \in K} (x_i)_e \leq u_e$ for all $e \in E$. So far, no combinatorial algorithm has been developed which solves this problem efficiently. On the other hand, polynomial time algorithms exist which, however, rely on a linear programming formulation of the problem and thus on general linear programming techniques such as the ellipsoid method or interior point algorithms, for instance, see [1].

2.2 Static Traffic Flows with Congestion

In the previous section we have considered static flows where transit times on the arcs are fixed. This assumption is no longer true in situations where congestion does occur. Congestion means that the transit time τ_e on an arc e is no longer constant, but a monotonically increasing, convex function

$\tau_e(x_e)$ of the flow value x_e on that arc e. Congestion is inherent to car traffic, but also occurs in evacuation planning, production systems, and communication networks. In all of these applications the amount of time needed to traverse an arc of the underlying network increases as the arc becomes more congested.

Congestion in static traffic networks has been studied for a long time by traffic engineers, for instance, see [45]. It models the rush hour situation in which flow between different origins and destinations is sent over a longer period of time. From a global system-oriented point of view, the usual objective to be optimized is the overall road usage which can be viewed as the sum of all individual travel times.

User Equilibrium and System Optimum From a macroscopic point of view, such a static traffic network with congestion can be modeled by a multicommodity flow problem, in which each commodity $i \in K$ represents two locations in the network between which d_i cars are to be routed per unit of time. The data (s_i, t_i, d_i), $i \in K$, form the so-called *origin-destination matrix*. Feasible routings are given by flows x, and the cost $c(x)$ of flow x is the total travel time spent in the network. On arc e, the flow x then induces a transit time $\tau_e(x_e)$, which is observed by all x_e flow units using that arc e. So the total travel time may be written as $c(x) = \sum_{e \in E} x_e \, \tau_e(x_e)$.

It is usually more convenient to think of feasible routings in terms of *flows on paths* instead of *flows on arcs*. Then, for every commodity i, the d_i amounts of flow sent between s_i and t_i are routed along certain paths P from the set $\mathcal{P}_i$ of possible paths between s_i and t_i. These paths P can be seen as the choice of routes that users take when they want to drive from s_i to t_i. On the other hand, any solution x in path formulation can be seen as a route recommendation to the users of the traffic network, and thus as a route guidance strategy. Therefore, it is reasonable to study properties of flows as route recommendations and investigate their quality within this model.

Current route guidance systems are very simple in this respect and usually they can only recommend shortest or quickest paths without considering the effect on the congestion that their recommendation will have. Simulations show that users of such a simple route guidance system will — due to the congestion caused by the recommendation — experience longer travel times when the percentage of users of such a system gets larger.

Without any guidance and without capacity restrictions, but full information about the traffic situation, users will try to choose a fastest route and achieve a Nash equilibrium, the so-called *user equilibrium*. Such an equilibrium is defined by the property that no driver can get a faster path through the network when everybody else stays with his route. One can show that in such an equilibrium state all flow carrying paths P in each $\mathcal{P}_i$ have the same travel time $\tau_P = \sum_{e \in P} x_e \, \tau_e(x_e)$, and, furthermore, if all τ_e are strictly increasing and twice differentiable, then the value of the user equilibrium is unique. It is thus characterized by certain fairness, since all users of the same origin destination pair have the same travel time. Therefore, it has been widely used as a reasonable solution to the static traffic problem with congestion. Computing the routes of the user equilibrium is usually referred to as *traffic assignment problem*. It was originally introduced by Wardrop [47] in order to model natural driver behavior, and has been studied extensively in the literature; see [45]. For a convex and twice differentiable transit time function τ_e, the traffic assignment problem can be formulated as a convex optimization problem with convex separable objective and linear constraints. It is usually solved with the Frank-Wolfe algorithm [18], which is essentially a feasible direction method.

The advantage of the user equilibrium lies in the fact that it can be achieved without traffic control (it only needs selfish users) and that it represents certain fairness (same travel time on each route). However, it does not necessarily optimize the overall road usage. Roughgarden and Tardos [43] investigate the difference that can occur between the user equilibrium and the system optimum, which is defined as a flow x minimizing the total travel time $\sum_{e \in E} x_e \, \tau_e(x_e)$. In general, the overall road usage in the user equilibrium may be arbitrarily larger than in the system optimum, but it is never more than the road usage incurred by optimally routing twice as much traffic.

Another unfavorable property of the user equilibrium is a non-monotonicity property with respect to network expansion. This is illustrated by the *Braess paradox*, where adding a new road to a network with fixed demands actually *increases* the overall road usage obtained in the updated user equilibrium [10,45,21].

So one may ask what can be achieved by centralized traffic management. It is clear that the system optimum would provide the best possible solution by definition. But it may have long routes for individual users and thus misses fairness, which makes it undesirable as a route guidance policy (unless routes are enforced upon users by pricing and/or observation). To quantify the phenomenon of long routes in the system optimum, Roughgarden [42] introduces a measure of unfairness. The *unfairness* is the maximum ratio between the travel time along a route in the system optimum and that of a route in the user equilibrium. It turns out that even under reasonable assumptions on the transit time functions, the unfairness may be arbitrarily bad.

In computational terms, user equilibrium and system optimum are quite related problems in networks without capacities on the arcs. In fact, the system optimum is the user equilibrium for the modified transit time function $\bar{\tau}_e(x_e) = \tau_e(x_e) + x_e \frac{d\tau_e(x_e)}{dx_e}$; see [8]. It may thus also be computed with the Frank-Wolfe algorithm.

2.3 Restricted System Optimum

Can one overcome the unfairness in the system optimum by introducing constraints on the paths that may be recommended as routes? This problem has been investigated by Jahn, Möhring, and Schulz [26] for static traffic networks with congestion and arc capacities. They consider for each commodity i only a subset $\bar{\mathcal{P}}_i \subseteq \mathcal{P}_i$ of *admissible paths*. In the simplest case, admissibility is defined by geographical length in the following way. Every arc e has a geographical length ℓ_e, and path $P \in \mathcal{P}_i$ is *admissible* if its geographical length $\ell_P := \sum_{e \in P} \ell_e$ does not exceed the geographical length of a shortest path between s_i and t_i by a certain, globally controlled *fairness factor* $L > 1$, e.g., $L = 1.2$. The corresponding optimum is a *restricted system optimum* for the original problem and the total travel time increases with $1/L$. An example from [26] with real data is given in Figure 1. Jahn, Möhring, Schulz, and Stier Moses [22] show that instead of using the shortest –paths one should use the –paths from the user equilibrium to define the set of admissible paths.

This approach implements a compromise between user equilibrium and system optimum, which meets a demand expressed, for instance, by Beccaria and Bolelli [7]: the goal should be to ”find the route guidance strategy which minimizes some global and community criteria with individual needs as constraints”. In principle, also other restrictions leading to acceptable paths are possible. This might include the exclusion of certain streets, preferences for scenic paths, or restrictions on the number of turns to be taken.

Computations with the street network of Zurich show that the system optimum with length restrictions achieves a much better fairness than the unrestricted system optimum. Furthermore, the restricted system optimum avoids the larger road usage and non-monotonicity of the user equilibrium. Here, unfairness is measured as the largest ratio ℓ_{P_1}/ℓ_{P_2} of the geographical path lengths ℓ_{P_j} of two flow carrying paths P_1, P_2, for any origin-destination pair i. The unfairness of the system optimum may go up to 3 and more, while its control via the fairness factor L also implies a strongly corre-

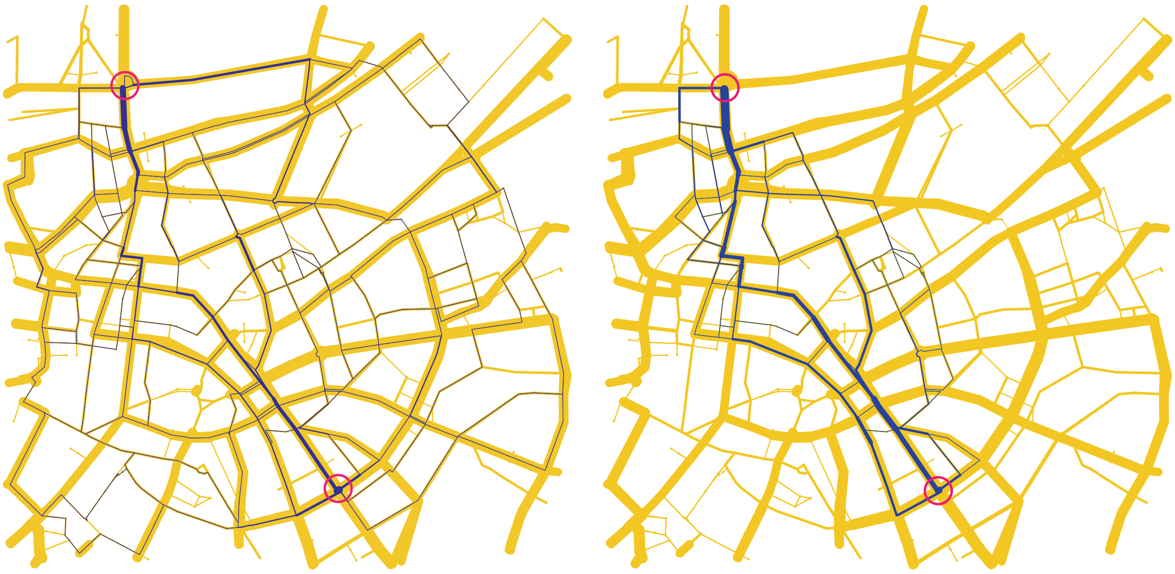


Fig. 1. System optimum with restrictions on path lengths: One commodity routed through the road network between the marked nodes. The left-hand side image displays the system optimum in which flow is distributed over the whole network in order to avoid high arc flows that would incur high arc transit times. In the right-hand side image the same amount of flow gets routed, but this time with a restriction on the geographical path lengths. Line thickness denotes arc capacity (yellow) and arc usage (blue).

lated fairness for other length measures such as actual travel times or travel times in the uncongested network.

2.4 Algorithmic Issues

The computations have been done with an algorithm developed in [26], which is an extension of the Frank-Wolfe algorithm. It has been implemented in a software package called CMCF (Constrained Minimum Cost Flow). It uses a path based flow model, where every potential path $P \in \cup_{i \in K} \bar{\mathcal{P}}_i$ is represented by a path flow variable x_P. However, these paths are only given implicitly by the length constraints $\sum_{e \in P} \ell_e \leq L\, \lambda(s_i, t_i)$ for a path between s_i and t_i, where $\lambda(s_i, t_i)$ is the shortest path length between s_i and t_i.

For *constant* travel times τ_e, this problem can be solved by linear programming with column generation which avoids an explicit generation of all (possibly exponentially many) admissible paths. We refer to [1] for details of this common technique. One maintains a linear program (the *master program*) with a subset of the variables (enough to contain a basis). Checking the master program for optimality then reduces to a *constrained shortest path problem* in the given network: compute a shortest path (where the length is influenced by the dual solution of the master program) such that the geographic length constraint $\sum_{e \in P} \ell_e \leq L\, \lambda(s_i, t_i)$ is preserved. Then either optimality is established and the current solution in the master problem is optimal, or there are paths that are too long. These paths are added to the master problem (usually in exchange for others) and one iterates.

This is the point in the algorithm where also other conditions on the paths could be considered. They can be dealt with in the current algorithm if the shortest path problem resulting from the linear programming optimality conditions can be combined with the restrictions defining the paths.

The constrained shortest path problem is $\mathcal{NP}$-hard in the weak sense, but can be solved in reasonable time by an adaptation of Dijkstra's algorithm for Pareto optimal shortest paths; see [2]. This adaptation requires maintaining a set of labels at every node which express the current Pareto optimal paths from the origin to that node. The algorithm reduces essentially to a repeated application of Dijkstras algorithm and runs polynomial in the maximum number of labels generated at a node.

The non-linear case is dealt with as in the Frank-Wolfe algorithm. One first computes a feasible direction $y - x = \operatorname{argmin}\{\nabla c(x)^T y \mid y \text{ feasible flow}\}$ of the objective function c at the current flow x and then a step size λ by line search. The next solution is then obtained as $x := x + \lambda(y - x)$. At first glance, this natural approach seems to be infeasible since the objective function $c(x)$ involves all of the many path variables x_P. But the scalar product $\nabla c(x)^T y$ can equivalently be expressed in arc variable vectors, thus reducing the number of variables to the number of arcs in the network. Also the line search can be reduced to the number of paths that actually carry flow, which does not get too large.

Figure 2 illustrates the nesting of the different steps in the algorithm. It becomes clear that algorithms for shortest paths and constrained shortest paths are at the core of the algorithm and that their run time largely determines the efficiency of the whole algorithm. We are currently working to make this approach scalable in order to make it applicable to larger networks. To this end, we are implementing and testing several approaches to speed up the calculation of shortest paths and restricted shortest paths, see [30,37] more details on acceleration techniques for (constrained) shortest paths.

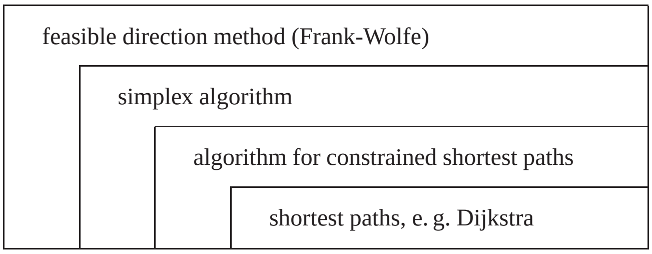

feasible direction method (Frank-Wolfe)
simplex algorithm
algorithm for constrained shortest paths
shortest paths, e. g. Dijkstra

Fig. 2. Steps in computing the restricted system optimum.

3 Large Scale Multi-Agent Transportation Simulation

The main idea of using multi-agent simulations in transportation planning is to combine each single aspect of the *four Step Process* in transportation planning [45], *Activity-Based Demand Generation (ADG)* (discussed by [29]), implemented by [46,39] and others) and *Dynamic Traffic Assignment (DTA)* [28,19] into highly dependent decision making processes of each individual of the given scenario. These decision making processes lead to a complete plan of what a person wants to do: where to perform an activity, in which order different activities should be performed, at which time a trip occurs and which route, and mode the trip will be done.

The detailed outcome will then be executed in a physical environment in which all the individuals actually interact with each other, i.e., producing congestions and spillbacks on a network, see [13]. Because of the interaction of the persons, the outcome of the execution will differ from what has been planned actually by the individuals. These information can therefore be used to adapt the decision making process such that the second outcome does perform better (does have a better utility) then the first try.

This leads to the following feedback algorithm (more information can be found in [41]):

1. Start with an initial guess of a complete plan of each individual of the given scenario.
2. Execute the initial guess in an physical environment.
3. Evaluate the performance (also called utility or fitness) of the executed plans.
4. Re-adjust some (or all) parts of the decision making process based on the outcome of the previous execution.
5. Go to step 2

MATSIM-T [35] is such a multi-agent micro simulation based on the idea described above.

3.1 MATSIM-T—A Large Scale Micro-Simulation

In order to provide a tool to produce the requested daily plans we are developing the multi-agent travel simulation toolkit, MATSIM-T [35], which is an agent-based micro simulation system of daily demand. The agents have daily activity plans that they use to describe how they act in the virtual world. Each agent has the desire to perform optimal according to a utility function that defines what a useful day is. Each agent can change his or her daily decisions to get a higher overall utility. This can be interpreted as learning. When the agents end up in a situation where none is able to improve his or her plan, they are in a user equilibrium and the learning loop ends. Assuming that a user equilibrium is a state of the system that we are looking for we get a set of daily plans of all agents that represent a typical state of the world.

Figure 3 shows the detailed structure of the MATSIM toolkit; see also Balmer et al. [4]. The core lies in the MATSIM-DB which stores all information of a given scenario including the disaggregated description of each individual living in that region. The modular approach of MATSIM-T makes it possible to plug in models and algorithms in any part of the whole demand modeling process. Logically MATSIM-T is split up into different modeling steps:

1. *MATSIM-DB* Parsers are interfaces to fill up the database with description of the scenario.
2. *MATSIM-DATA* combines, consolidates and validates the data.
3. *MATSIM-INI* creates a synthetic population out of the given data points and assigns initial demand for each individual.
4. *MATSIM-EA* optimizes iteratively the individual demand using time- and route-replanning modules until a stable state is reached.
5. *MATSIM-ANALYIS* defines the collection of statistical and visual outcome of any part of MATSIM-T.

More detailed information can be found in Balmer [4] and Balmer [3] and the description of the current state of research of MATSIM-EA and its modules are shown in Charypar [12] and Meister [36]. The focus of this paper is the *MATSIM-IIDM* of MATSIM-INI: the creation of the initial demand of each inhabitant. More precisely, we focus on the assignment of the initial routes for each individual.

4 Combining CMCF with MATSIM-T

While the current simulation system has proved to work, see [4], there is still a substantial computational effort involved in simulating the learning of daily plans for all virtual persons involved in a scenario. This is especially the case when looking at large scale scenarios with 1 million persons or more.

We will focus on how the speed of the overall learning system can be increased by improving the performance of individual modules such as the replanning module (responsible for creating new plans for each agent), the traffic micro simulation and also the modeling of the initial state of each individual.

Compared with Balmer [6], the results presented by Charypar [12] and Meister [36] showed that the calculation time of the system can be decreased from several weeks to about one day by using enhanced replanning modules of MATSIM-EA. Since the iteration process of MATSIM-EA are now reduced to ca. 50 iterations until the stable state is found, it is worthwhile to put more effort into MATSIM-IIDM to produce more realistic initial demand of each individual of the given scenario. In other words, producing more realistic initial demand leads to less iteration until relaxation of MATSIM-EA (saving only 5 iterations means already a speedup of 10%).

Recently, the initial routes choice of each agent was calculated with a time-dependent dynamic Dijkstra router (Balmer et al. [5]). It has the disadvantage that it calculates the shortest routes based on the unloaded network; see [4]. In other words, the dynamic Dijkstra router calculates static shortest paths for each individual of the system. This initialization produces an unrealistic amount of congestion for the first iteration of MATSIM-EA. Therefore we now replace the initial route assigning process step by the CMCF module. The following section presents the experimental setup of different scenarios and analyzes the computational performance of MATSIM-EA with and without CMCF.

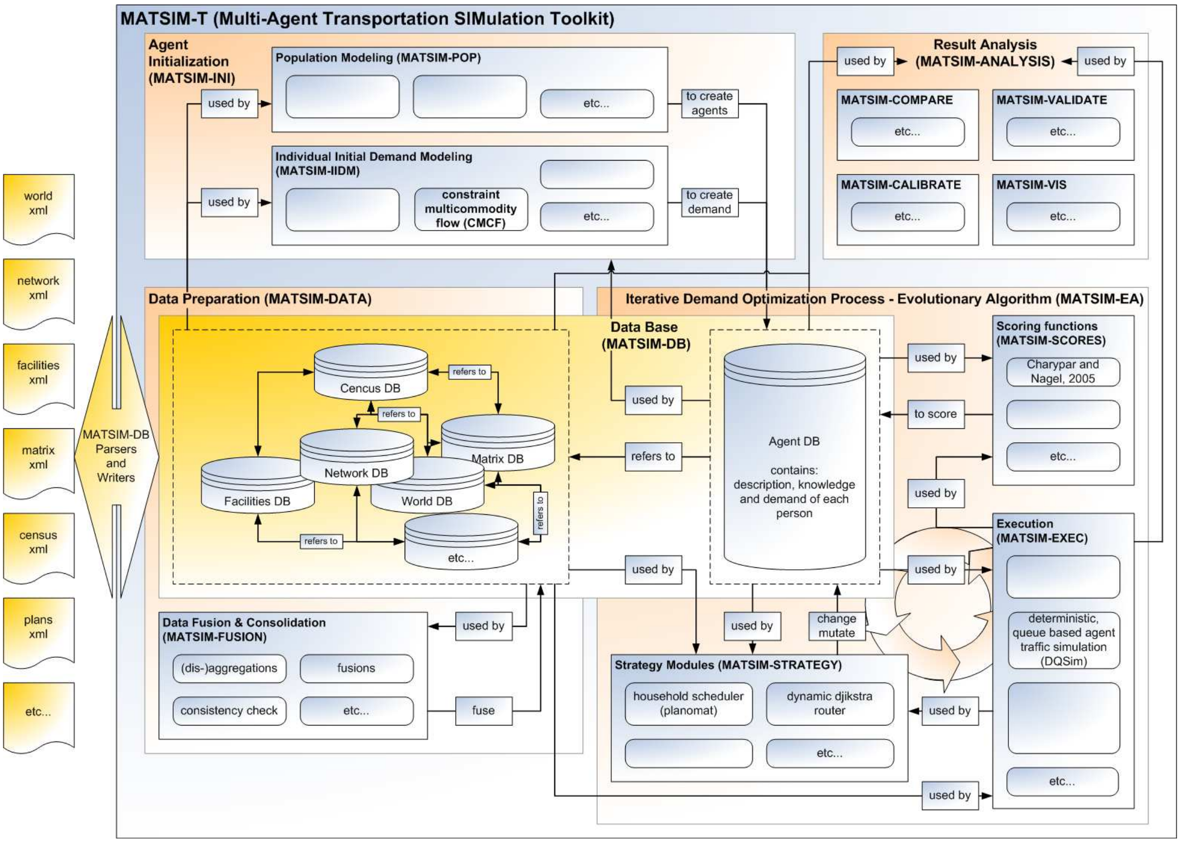

MATSIM-T (Multi-Agent Transportation SIMulation Toolkit)
Agent Initialization (MATSIM-INI)
used by
Population Modeling (MATSIM-POP)
etc...
to create agents
used by
Individual Initial Demand Modeling (MATSIM-IIDM)
constraint multicommodity flow (CMCF)
etc...
to create demand
Result Analysis (MATSIM-ANALYSIS)
used by
used by
MATSIM-COMPARE
etc...
MATSIM-VALIDATE
etc...
MATSIM-CALIBRATE
etc...
MATSIM-VIS
etc...
world xml
network xml
facilities xml
matrix xml
census xml
plans xml
etc...
MATSIM-DB Parsers and Writers
Data Preparation (MATSIM-DATA)
Iterative Demand Optimization Process - Evolutionary Algorithm (MATSIM-EA)
Data Base (MATSIM-DB)
Cencus DB
refers to
refers to
Network DB
Matrix DB
Facilities DB
World DB
refers to
refers to
etc...
used by
refers to
used by
Agent DB
contains: description, knowledge and demand of each person
used by
to score
used by
used by
Scoring functions (MATSIM-SCORES)
Charypar and Nagel, 2005
etc...
Execution (MATSIM-EXEC)
deterministic, queue based agent traffic simulation (DQSim)
etc...
Data Fusion & Consolidation (MATSIM-FUSION)
used by
(dis-)aggregations
fusions
consistency check
etc...
fuse
used by
change mutate
Strategy Modules (MATSIM-STRATEGY)
household scheduler (planomat)
dynamic djikstra router
etc...
used by
used by

Fig. 3. Structure of MATSIM-T (Multi-Agent Transportation SIMulation Toolkit) combined with CMCF (Constrained Multicommodity Flow).

5 Experimental Setup and Computational Results

In this section, we describe each of the experiments conducted and present the computational results.

5.1 Transit Time Function

The CMCF framework is applicable to any transit time function that is monotonically increasing, convex, and twice differentiable. To apply CMCF within the MATSIM-T framework we need a transit time function τ_e associated with each link $e \in E$ such that:

- τ_e is monotonically increasing, convex, and twice differentiable, and
- τ_e approximates the link transit times of the queue model applied in MATSIM-T well enough.

The MATSIM-T framework uses a deterministic queue model (see Cetin [11] for more details) where the link transit time is the sum of free speed travel time and queue waiting time of that link. The free speed travel time of each link is given in the input data. The waiting time in a link queue depends on

- the outflow capacity of that link queue,
- the outflow capacity of neighbouring link queues, i.e., queues of links ending in the same node, and
- the queue capacity of following links.

Therefore, the link transit time in the simulation is an almost linear function that can be approximated by $\tau_0 + \tau(x)$, where τ_0 is the free flow transit time and $\tau(x)$ is defined by using splines of third degree:

$$\tau : \mathbb{R}^+ \to \mathbb{R}^+;\ \tau(x) \mapsto \begin{cases} 3.60 - 6.79x + 15.24(x - 0.2)^3 & \text{if } 0 < x < 0.5, \\ 1.96 - 2.67x + 13.72(x - 0.5)^2 - 26.21(x - 0.5)^3 & \text{if } 0.5 \leq x < 0.7, \\ 0.66 - 0.33x - 2.01(x - 0.7)^2 + 3.34(x - 0.7^3) & \text{if } 0.7 \leq x < 1, \\ 0.87 - 0.63x + 1.00(x - 1)^2 - 0.65(x - 1)^3 & \text{if } 1 \leq x < 1.5, \\ 0.27 - 0.12x + 0.20(x - 1.5)^2 - 0.13(x - 1.5)^3 & \text{if } 1.5 \leq x < 2, \\ 0 & \text{else.} \end{cases}$$

5.2 Scenarios

We run our experiments on two different scenarios:

1. an artificial corridor network with a Manhattan's grid system of links (15 nodes and 24 links) and almost equal capacities on each link
2. parts of the Zurich street network (2210 nodes and 6334 links)

On each of these scenarios, we used different numbers of agents: 100, 500, and 900 agents on Scenario 1 and 2037, 20018, and 50022 agents on Scenario 2. Since our static flow model is more appropriate for rush hour situations, the results produced by CMCF getting better for higher numbers of agents, i.e. more congested networks. For each scenario and each agent setting, we started MATSIM-T twice: one run with the shortest paths in the uncongested network for each agent demand as initial routes and one run with the routes computed by CMCF as initial routes. Each run computed 200 iterations of MATSIM-T. Figure 4 shows the computational results of our tests.

5.3 Fairness Factor

The fairness factor L is set to a very large number except for the experiments concerning the variation of L. The fairness factor L was varied on the Zurich Scenario with 20018 agents from 1.0, 1.05, and 1.2 to one billion. An agent i is restricted to paths shorter than L times the length of the shortest s_i-t_i-path. For sufficiently large numbers of L all s_i-t_i-paths are allowed, but the smaller L is set, the smaller is the number of admissible paths per agent. Therefore reducing L reduces also the size of the mental map per agent.

5.4 Experimental Environment

We implemented the MATSIM-T framework and CMCF in C++ using GNU C++ compiler version 3.3.5 [20] on a Linux 2.6 system (Novell SUSE 10). All computations were done on a 32-bit Intel Pentium IV processor, 2.80GHz, 1 GB memory. CPLEX version 10.0 [25] was used for solving the LPs.

5.5 Computational Running Times

Due to space restrictions in this extended abstract we presents only a few of our computational times. Computing the initial routes for the Corridor Scenario by CMCF takes fractions of a second. On this Scenario with 900 agents 200 iterations with MATSIM-T (SP) take 1 hour and with MATSIM-T (CMCF) this reduces to 56 minutes.

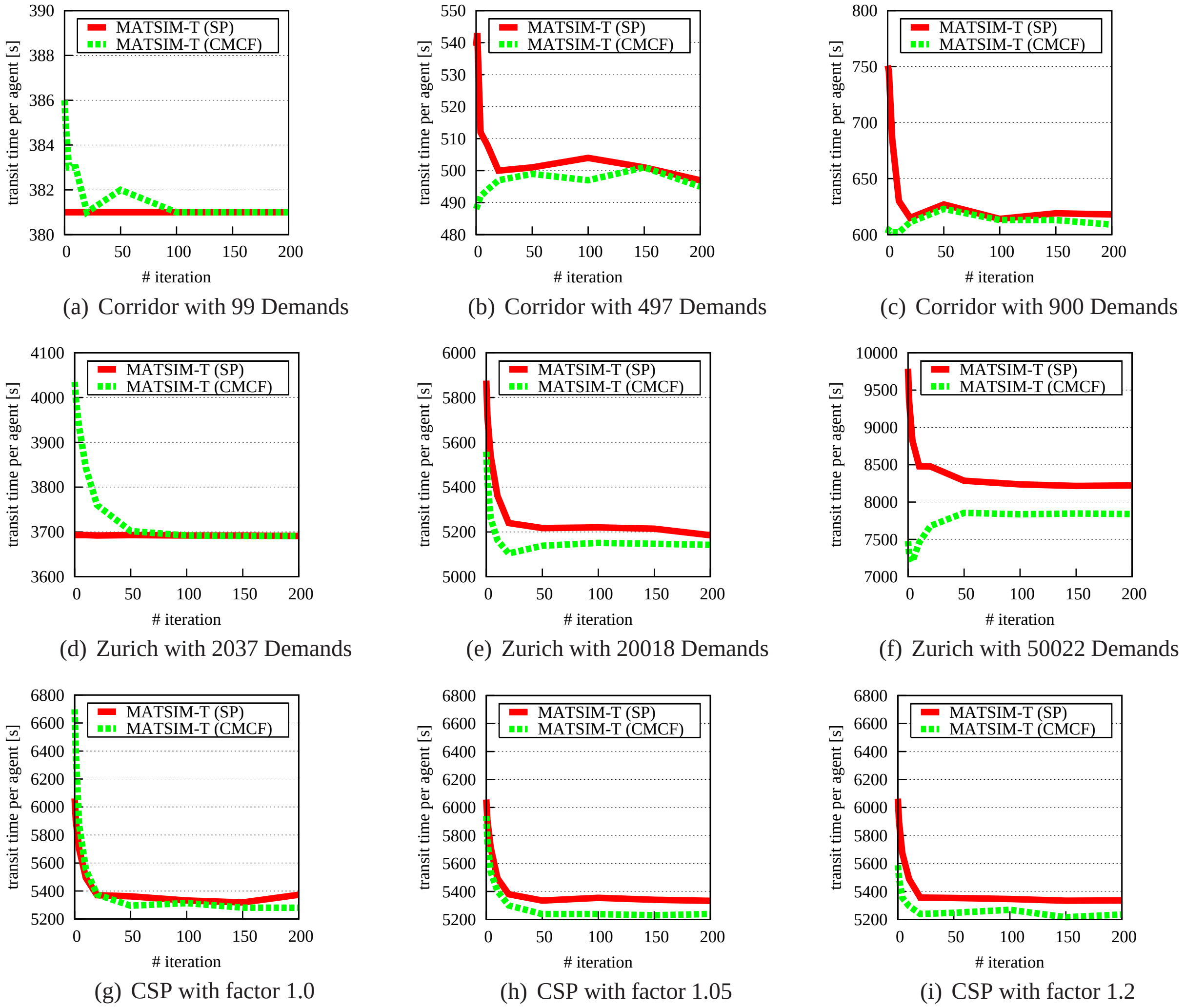
The computation time increases considerably on the real world scenario of Zurich: computing the initial routes by CMCF for 50022 agents on this scenario takes 1:30 hour. Computing 200 iterations with MATSIM-T (SP) take 17:51 hours and with MATSIM-T (CMCF), this reduces to 14:54 hours.

6 Discussion and Outlook

CMCF computes routes for a static flow model with congestion. If we use these routes as initial routes in the simulation, the speed up gets better when the number of congestions in the network increases. An example we discussed was the road network of Zurich: 20018 and 50022 agents on the Zurich network lead to a highly congested network and therefore the speed up of CMCF initial routes is best on this setting. We were able to save up to 1:30 hour of computation time for 200 MATSIM-T iterations and even better we achieved much better travel times per agents on highly congested networks; see Figure 4(e) and 4(f). To gain a good speed up not only for highly congested networks, more realistic network flow models need to be applied, such as the rate-dependent model (Hall and Schilling [23]) or other flow over time models, for instance, see [31,32,22].

CMCF computes an optimal fractional solution which is transformed into an integral solution by rounding up fractional flow values on each path. The integral (rounded) flow is still feasible, but it might not be optimal; see [26]. Because of this, MATSIM-T can then still find a better solution than the integral (rounded) solution found by CMCF.

It is remarkable that on the Zurich network with 50022 agents, Figure 4(f), the average travel time per agent is considerably higher with shortest path initial routes than with CMCF initial routes: almost 500 seconds of travel time per agent. Even after 200 iterations this gap could not be closed by MATSIM-T with shortest paths as initial routes. This indicates that the optimal route assignment approach by CMCF exploits the optimization potential within MATSIM-T much better than the shortest paths approach which was used before. Currently CMCF is appropriate for network sizes of up to several thousand nodes and links. We are working on a more scalable version.



(a) Corridor with 99 Demands

(b) Corridor with 497 Demands

(c) Corridor with 900 Demands

(d) Zurich with 2037 Demands

(e) Zurich with 20018 Demands

(f) Zurich with 50022 Demands

(g) CSP with factor 1.0

(h) CSP with factor 1.05

(i) CSP with factor 1.2

Fig. 4. Computational Results for the Corridor and Zurich Scenario with different numbers of agents and for the Zurich Scenario with 20018 agents and different fairness factors L.

The globally controlled *constrained* or *fairness factor* $L > 1$ allows an easy implementation of a mental map per agent. In principle, all kinds of acceptable routes can be modeled by this. Increasing the fairness factor leads to a larger set of admissible paths where an agent can chose from. A larger selection of admissible paths then leads to smaller travel times per agent: see Figure 4(g),4(h), and 4(i).

CMCF computes a restricted system optimum while MATSIM-T computes a Nash equilibrium. Therefore the overall transit time increases with higher iteration numbers when we use CMCF routes as initial routes in MATSIM-T. This can be seen in Figure 4 in the case of highly congested networks.

As a next step it is highly desirable to replace the dynamic Dijkstra router in MATSIM-EA by a CMCF router. This would reduce the number of iterations to just one. Furthermore, this would implement a mental map in MATSIM-T.

The combination of MATSIM-T with CMCF as a replanning module is a combination of two different interpretations of ”congestion”. While the physical micro simulation produces congestion as ”spillback” CMCF defines it as a mathematical function. This paper gives a number of insights into the interconnections and differences between these two approaches and along these lines deeper research is necessary.

References

1. Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows : Theory, Algorithms, and Applications*. Prentice Hall, New Jersey, NJ, USA, January 1993.
2. Yash Paul Aneja, V. Aggarwal, and Kunhiraman P. K. Nair. Shortest chain subject to side constraints. *Networks*, 13:295–302, 1983.
3. Michael Balmer. *Travel demand modeling for multi-agent traffic simulations: Algorithms and systems*. PhD thesis, ETH Zurich, 2006.
4. Michael Balmer, Kay W. Axhausen, and Kai Nagel. An agent-based demand-modeling framework for large scale micro-simulations. In *in TRB 85th Annual Meeting Compendium of Papers CD-ROM*, Washington, D.C., January 2006. Transportation Research Board.
5. Michael Balmer, Bryan Raney, and Kai Nagel. Adjustments of activity timing and duration in an agent-based traffic flow simulation. In H. J. P. Timmermans, editor, *Progress in Activity-Based Analysis*, chapter 5. Elsevier, Oxford, 2005.
6. Michael Balmer, Marcel Rieser, Arnd Vogel, Kay W. Axhausen, and Kai Nagel. Generating day plans using hourly origin-destination matrices. In C. Laesser T. Bieger and R. Maggi, editors, *Jahrbuch 2004/2005 Schweizerische Verkehrswirtschaft*, pages 5–36. SVWG Schweizerische Verkehrs-wissenschaftliche Gesellschaft, St. Gallen, 2005.
7. G. Beccaria and Alfredo Bolelli. Modelling and assessment of dynamic route guidance: the MARGOT project. In *Vehicle Navigation & Information Systems Conference Proceedings (VNIS ’92)*, pages 117–126. IEEE, 1992.
8. Martin Beckmann, Charles Bartlett McGuire, and Christopher Blake Winsten. Studies in the economics of transportation. Technical report, Yale University Press, 1965.
9. Moshe E. Ben-Akiva, Haris N. Koutsopoulos, Rabi G. Mishalani, and Qi Yang. Simulation laboratory for evaluating dynamic traffic managment systems. *ASCE Journal of Transportation Engineering*, 123(4):283–289, July/August 1997.
10. Dietrich Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.
11. Nurhan Cetin. *Large-scale parallel graph-based simulations*. PhD thesis, ETH Zurich, 2005.
12. David Charypar, Kay W. Axhausen, and Kai Nagel. Implementing activity-based models: Accelerating the replanning process of agents using an evolution strategy. In *Proceedings of the 11th International Conference on Travel Behavior Research (IATBR)*, Kyoto, 2006.
13. Carlos F. Daganzo. Queue spillovers in transportation networks with a route choice. *Transportation Science*, 32(1), 1998.
14. Steven P. Dirkse and Michael C. Ferris. Traffic modeling and variational inequalities using GAMS. In P. L. Toint, M. Labbe, K. Tanczos, and G. Laporte, editors, *Operations Research and Decision Aid Methodologies in Traffic and Transportation Managment*, volume 166 of *NATO ASI Series*, pages 136–163. Springer, Heidelberg, Germany, 1998.
15. Peter Elias, Amiel Feinstein, and Claude Elwood Shannon. A note on the maximum flow through a network. *IRS Trans. Information Theory IT*, IT-2:117–199, 1956.
16. Lester Randolph Ford and Delbert Ray Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 9:399–404, 1956.

17. Lester Randolph Ford and Delbert Ray Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, NJ, USA, 1962.
18. Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quaterly*, 3:95–110, 1956.
19. M. Friedrich, I. Hofsäß, K. Nökel, and P. Vortisch. A dynamic traffic assignment method for planning and telematic applications. In *Proceedings of Seminar K, European Transport Conference*, Cambridge, GB, 2000.
20. version 3.3.5 GNU, C++ compiler. http://www.gcc.gnu.org.
21. Jane N. Hagstrom and Robert A. Abrams. Characterizing Braess's paradox for traffic networks. In *Proceedings of IEEE 2001 Conference on Intelligent Transportation Systems*, pages 837–842, 2001.
22. Alexander Hall, Katharina Langkau, and Martin Skutella. An fptas for quickest multicommodity flows with inflow-dependent transit times. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NY, USA, August 24-26, 2003, Proceedings*, volume 2764 of *Lecture Notes in Computer Science*, pages 71–82, Heidelberg, Germany, 2003. Springer.
23. Alexander Hall and Heiko Schilling. Flows over time: Towards a more realistic and computationally tractable model. In *Proceedings of the 7th Workshop on Algorithm Engineering and Experiments (ALENEX05)*, pages 55–67. SIAM, 2005.
24. F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, 20:224–230, 1941.
25. ILOG CPLEX Optimizer (version 10.0). http://www.ilog.com/products/cplex, 2005.
26. Olaf Jahn, Rolf H. Möhring, and Andres S. Schulz. Optimal routing of traffic flows with length restrictions in networks with congestion. In K. Inderfurth, G. Schwödiauer, W. Domschke, F. Juhnke, P. Kleinschmidt, and G. Wäscher, editors, *Operations Research Proceedings 1999*, pages 437–442. Springer-Verlag, Berlin, 2000.
27. L. V. Kantorovich. On the translocation of masses. *Comptes Rendus (Doklady) de l'Académie des Sciences de l'U.R.S.S.*, 37:199–201, 1942.
28. David E. Kaufman, Karl E. Wunderlich, and Robert L. Smith. An iterative routing/assignment method for anticipatory real-time route guidance. Technical Report 91-02, University of Michigan Department of Industrial and Operations Engineering, Ann Arbor MI 48109, May 1991.
29. Jenny King and David A. Hensher. *The Leading Edge of Travel Behavior Research.* Pergamon, Oxford, 2001.
30. Ekkehard Köhler, Rolf H. Möhring, and Heiko Schilling. Acceleration of shortest path and constrained shortest path computation. In Sotiris E. Nikoletseas, editor, *Experimental and Efficient Algorithms, 4th International Workshop, WEA 2005, Santorini Island, Greece, May 10-13, 2005, Proceedings*, volume 3503 of *Lecture Notes in Computer Science*, pages 126–138, Heidelberg, Germany, 2005. Springer.
31. Ekkehard Köhler, Rolf H. Möhring, and Martin Skutella. Traffic networks and flows over time. In *BMG Festschrift for the opening of the DFG Research Center: Mathematics for Key Technologies*, pages 49–70. Berliner Mathematische Gesellschaft, November 2002.
32. Ekkehard Köhler and Martin Skutella. Flows over time with load-dependent transit times. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 174–183, Philadelphia, PA, USA, 2002. Society for Industrial and Applied Mathematics.
33. T. C. Koopmans. Optimum utilization of the transportation system. *Econometrica*, 17:136–146, 1949.
34. Lösch & Partner GmbH. Verkehrsmanagement — Eine Bestandsaufnahme. Broschüre. Erstellt in Zusammenarbeit mit diversen deutschen Automobilherstellern, 1995.
35. MATSIM-T. A multi agent transportation simulation toolkit software. http://www.matsim.org.
36. Konrad Meister, Michael Balmer, Kay W. Axhausen, and Kai Nagel. Planomat: A comprehensive scheduler for a large-scale multi-agent transportation simulation. In *Proceedings of the 11th International Conference on Travel Behavior Research (IATBR)*, Kyoto, 2006.
37. Rolf H. Möhring, Heiko Schilling, Birk Schütz, Dorothea Wagner, and Thomas Willhalm. Partitioning graphs to speed up dijkstra's algorithm. *ACM Journal of Experimental Algorithms (JEA)*, 12:1–29, 2006. to appear.
38. Kai Nagel, Jorg Esser, and Marcus Rickert. Large-scale traffic simulations for transportation planning. In Dietrich Stauffer, editor, *Annual Review of Computational Physics*, volume VII, pages 151–202, London, UK, 2000. World Scientific Publishing Company.
39. R. M. Pendyala. Phased implementation of a multimodal activity-based travel demand modeling system in florida. accessed 2004. Project number 0510812 (BA496).
40. Ilya Prigogine and Robert Herman. *Kinetic Theory of Vehicular Traffic*. Elsevier Science B. V., Amsterdam, The Netherlands, 1971.
41. Bryan Raney. *Learning framework for large-scale multi-agent simulations*. PhD thesis, ETH Zurich, 2005.

42. Tim Roughgarden. How unfair is optimal routing? In *Proceedings of the 13th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 203–204, San Francisco, CA, 2002.
43. Tim Roughgarden and Éva Tardos. How bad is selfish routing? *Journal of the Association for Computing Machinery*, 49(2):236–259, 2002.
44. Alexander Schrijver. On the history of the transportation and maximum flow problem. *Mathematical Programming*, 91:437–445, 2002.
45. Yosef Sheffi. *Urban Transportation Networks*. Prentice Hall, Englewood Cliffs, NJ, 1985.
46. Peter S. Vovsha, Eric Petersen, and Robert Donnelly. Microsimulation in travel demand modeling: lessons learned from the new york best practice model. In *Transportation Research Record*, volume 1805, pages 68–77, 2002.
47. John Glen Wardrop. Some theoretical aspects of road traffic research. In *Institution of Civil Engineers*, volume 1, pages 325–378, 1952.

Reports from the group

“Combinatorial Optimization and Graph Algorithms”

of the Department of Mathematics, TU Berlin

2006/08 *Nicole Megow and Tjark Vredeveld:* Approximation Results for Preemptive Stochastic Online Scheduling

2006/07 *Ekkehard Köhler and Christian Liebchen and Romeo Rizzi and Gregor Wünsch:* Reducing the Optimality Gap of Strictly Fundamental Cycle Bases in Planar Grids

2006/05 *Georg Baier and Thomas Erlebach and Alexander Hall and Ekkehard Köhler and Heiko Schilling:* Length-Bounded Cuts and Flows

2005/30 *Ronald Koch and Martin Skutella and Ines Spenke :* Maximum k-Splittable Flows

2005/29 *Ronald Koch and Ines Spenke :* Complexity and Approximability of k-Splittable Flows

2005/28 *Stefan Heinz and Sven O. Krumke and Nicole Megow and Jörg Rambau and Andreas Tuchscherer and Tjark Vredeveld:* The Online Target Date Assignment Problem

2005/18 *Christian Liebchen and Romeo Rizzi:* Classes of Cycle Bases

2005/11 *Rolf H. Möhring and Heiko Schilling and Birk Schütz and Dorothea Wagner and Thomas Willhalm:* Partitioning Graphs to Speed Up Dijkstra’s Algorithm.

2005/07 *Gabriele Di Stefano and Stefan Krause and Marco E. Lübbecke and Uwe T.Zimmermann:* On Minimum Monotone and Unimodal Partitions of Permutations

2005/06 *Christian Liebchen:* A Cut-based Heuristic to Produce Almost Feasible Periodic Railway Timetables

2005/03 *Nicole Megow, Marc Uetz, and Tjark Vredeveld:* Models and Algorithms for Stochastic Online Scheduling

2004/37 *Laura Heinrich-Litan and Marco E. Lübbecke:* Rectangle Covers Revisited Computationally

2004/35 *Alex Hall and Heiko Schilling:* Flows over Time: Towards a more Realistic and Computationally Tractable Model

2004/31 *Christian Liebchen and Romeo Rizzi:* A Greedy Approach to Compute a Minimum Cycle Bases of a Directed Graph

2004/27 *Ekkehard Köhler and Rolf H. Möhring and Gregor Wünsch:* Minimizing Total Delay in Fixed-Time Controlled Traffic Networks

2004/26 *Rolf H. Möhring and Ekkehard Köhler and Ewgenij Gawrilow and Björn Stenzel:* Conflict-free Real-time AGV Routing

2004/21 *Christian Liebchen and Mark Proksch and Frank H. Wagner:* Performance of Algorithms for Periodic Timetable Optimization

2004/20 *Christian Liebchen and Rolf H. Möhring:* The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables — and Beyond

2004/19 *Ronald Koch and Ines Spenke:* Complexity and Approximability of k-splittable flow problems

2004/18 *Nicole Megow, Marc Uetz, and Tjark Vredeveld:* Stochastic Online Scheduling on Parallel Machines

2004/09 *Marco E. Lübbecke and Uwe T. Zimmermann:* Shunting Minimal Rail Car Allocation

2004/08 *Marco E. Lübbecke and Jacques Desrosiers:* Selected Topics in Column Generation

2003/050 *Berit Johannes:* On the Complexity of Scheduling Unit-Time Jobs with OR-Precedence Constraints

2003/49 *Christian Liebchen and Rolf H. Möhring:* Information on MIPLIB’s timetab-instances

2003/48 *Jacques Desrosiers and Marco E. Lübbecke:* A Primer in Column Generation

2003/47 *Thomas Erlebach, Vanessa Kääb, and Rolf H. Möhring:* Scheduling AND/OR-Networks on Identical Parallel Machines

2003/43 *Michael R. Bussieck, Thomas Lindner, and Marco E. Lübbecke:* A Fast Algorithm for Near Cost Optimal Line Plans

2003/42 *Marco E. Lübbecke:* Dual Variable Based Fathoming in Dynamic Programs for Column Generation

2003/37 *Sándor P. Fekete, Marco E. Lübbecke, and Henk Meijer:* Minimizing the Stabbing Number of Matchings, Trees, and Triangulations

2003/25 *Daniel Villeneuve, Jacques Desrosiers, Marco E. Lübbecke, and François Soumis:* On Compact Formulations for Integer Programs Solved by Column Generation

2003/24 *Alex Hall, Katharina Langkau, and Martin Skutella:* An FPTAS for Quickest Multicommodity Flows with Inflow-Dependent Transit Times

2003/23 *Sven O. Krumke, Nicole Megow, and Tjark Vredeveld:* How to Whack Moles

2003/22 *Nicole Megow and Andreas S. Schulz:* Scheduling to Minimize Average Completion Time Revisited: Deterministic On-Line Algorithms

2003/16 *Christian Liebchen:* Symmetry for Periodic Railway Timetables

2003/12 *Christian Liebchen:* Finding Short Integral Cycle Bases for Cyclic Timetabling

762/2002 *Ekkehard Köhler and Katharina Langkau and Martin Skutella:* Time-Expanded Graphs for Flow-Dependent Transit Times

761/2002 *Christian Liebchen and Leon Peeters:* On Cyclic Timetabling and Cycles in Graphs

752/2002 *Ekkehard Köhler and Rolf H. Möhring and Martin Skutella:* Traffic Networks and Flows Over Time

739/2002 *Georg Baier and Ekkehard Köhler and Martin Skutella:* On the k-splittable Flow Problem

736/2002 *Christian Liebchen and Rolf H. Möhring:* A Case Study in Periodic Timetabling

Reports may be requested from:

Sekretariat MA 6–1
Fakultt II – Institut fr Mathematik
TU Berlin
Straße des 17. Juni 136
D-10623 Berlin – Germany

e-mail: klink@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`