SYSTEM-OPTIMAL ROUTING OF TRAFFIC FLOWS WITH USER CONSTRAINTS IN NETWORKS WITH CONGESTION

OLAF JAHN[†], ROLF H. MÖHRING[‡], ANDREAS S. SCHULZ[*], AND NICOLÁS E. STIER MOSES[**]

ABSTRACT. The design of route guidance systems faces a well-known dilemma. The approach that theoretically yields the system-optimal traffic pattern may discriminate against some users in favor of others. Proposed alternate models, however, do not directly address the system perspective and may result in inferior performance. We propose a novel model and corresponding algorithms to resolve this dilemma. We present computational results on real-world instances and compare the new approach with the well-established traffic assignment model. The essence of this study is that system-optimal routing of traffic flow with explicit integration of user constraints leads to a better performance than the user equilibrium, while simultaneously guaranteeing superior fairness compared to the pure system optimum.

1. INTRODUCTION

Route guidance and information systems are designed to assist drivers in making route decisions. Such devices can provide information (e.g., about conditions drivers are likely to experience) or give recommendations (e.g., "leave the highway at the next exit and turn right"). We will concentrate on in-vehicle route guidance devices that provide recommendations to drivers. Drivers enter their destinations at the beginning of the trip, and the system computes routes based on digital maps, up-to-date traffic data and current vehicle positions determined with the help of the Global Positioning System (Henry, Charbonnier, and Farges 1991). These devices normally use visual and acoustic indicators to aid drivers in following the proposed route.

Currently, many cars are already equipped with simple versions of these devices, and with prices going down many more are likely to have one in the not-so-distant future. For that reason, it is widely hoped that route guidance systems can help to alleviate congestion caused by the still increasing amount of road traffic. Even small improvements can have a significant impact given that the "congestion bill" in the U.S. alone was $67.5 billion in the year 2000, consisting of 3.6 billion hours of delay plus 5.7 billion gallons of gas (Texas Transportation Institute 2002).

Several kinds of in-car navigation systems have been proposed. The simplest devices perform *static* guidance; i.e., they work with information that is infrequently updated. Most of the in-car guidance consoles deployed today are of this type. Their main goal is to provide information to drivers who do not know the area well. From an algorithmic point of view, they are straightforward: they only compute shortest paths (or approximations thereof) to the destinations with respect to

Date: November 2002; revised December 2003.

2000 *Mathematics Subject Classification.* Primary 90C35; 90B20, 90C25, 90C27, 90C90.

Key words and phrases. Intelligent Transportation Systems, Route Guidance, Traffic Flow, System Optimum, User Equilibrium, Multicommodity Flow, Constrained Shortest Path.

[†]Infopark, Berlin, Germany

[‡]Technische Universität Berlin, Fakultät II, Institut für Mathematik, MA 6-1, Strasse des 17. Juni 136, D-10623 Berlin, Germany

[*]Massachusetts Institute of Technology, Sloan School of Management, E53-361, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA

[**]Massachusetts Institute of Technology, Operations Research Center, E40-130, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA

travel time, geographic distance, or other appropriate measures. Computational challenges for these approaches arise "solely" from the huge size of the underlying road networks.

More sophisticated route guidance systems make use of information on current conditions in the traffic network. That knowledge is the basis of *reactive* guidance systems (Papageorgiou 1990; Ben-Akiva, de Palma, and Kaysi 1996). In other words, the recommendation provided to drivers at any given time is based on a snapshot of the traffic at that time. One of the advantages of reactive guidance is that it can respond quickly to demand changes or incidents because no predictions are used.

The most advanced approach, called *anticipatory* guidance, predicts future demands and traffic conditions and gives recommendations accordingly (Chen and Underwood 1991; Kaysi 1992). The issue is how future conditions should be predicted. When market penetration is low, guidance systems can basically ignore their own effect. On the other extreme, when most users are guided and they comply with the guidance, reality is likely to be as predicted. The problematic cases are in between the two extremes. These route guidance systems must predict how users will behave (e.g., follow the recommendation or not) to guide traffic in a way that is consistent with the predictions (Bottom 2000). Otherwise, guidance can fail to achieve the desired objective because recommendations were given making assumptions about the future that may not materialize.

According to Bottom (2000), there is no consensus in the community on which of the latter two approaches should be used in practice. For the present paper, we adopt reactive guidance because it is conceptually simpler.

Regardless of the source of network data, route guidance devices still have to compute concrete routes to be proposed to users. Several systems compute shortest paths, the k shortest paths for some properly chosen parameter k, or pareto-optimal paths (when multiple criteria are considered simultaneously). Some systems perform these computations online while others include them in a preprocessing step. For example, DynaMIT (2002), a simulation-based real-time system to provide travel information, computes shortest paths beforehand with respect to several static impedance functions. Among other measures, it considers free-flow travel times, peak-period travel times, geographic lengths, and the number of signalized intersections.

Another possibility is to assign users to the paths of smallest individual impedance under the current conditions, giving rise to what is commonly known as a *user equilibrium* (or user-optimal solution). Alternatively, one can opt to minimize the total impedance in the system, a solution known as the *system optimum*. Current route guidance systems implement both user and system optimality, although the bias has always been towards user-optimal traffic patterns (e.g., Mahmassani, Hu, Peeta, and Ziliaskopoulos 1994; Ben-Akiva, Bierlaire, Bottom, Koutsopoulos, and Mishalani 1997; Dynasmart 2002). Although system optimality is included in such systems for computing good upper bounds on traffic efficiency, it is not accepted as a realistic option for actual guidance. Indeed, it is well-known that under system-optimal patterns some users may end up traveling longer to allow the system to achieve global efficiency. Of course, it is not likely that many users accept recommendations that are too inefficient with respect to their personal optimal choices. We measure the detriment for users as the ratio of the impedance of the recommended path to that of the shortest possible path the user could have taken. This concept, called *unfairness*, will play a central role in this paper.

For system-optimal solutions, DynaMIT assumes, as we will, that there is a single user-class, 100% market penetration and full user compliance. We remark that its feasible path selection method is similar to what we shall propose. For that reason, the system optimum that DynaMIT computes actually is a "constrained system optimum," the key concept of this paper.

1.1. **Drawbacks of current route guidance systems.** None of the current guidance systems takes directly into account the efficiency of the solution they propose (with the exception of system-optimal solutions, which are not implementable because of their unfairness). Thus, the need for integrated algorithms that actually pay attention to the system-wide performance has been recognized (Henry, Charbonnier, and Farges 1991; Beccaria and Bolelli 1992; Kaysi, Ben-Akiva, and de Palma 1995).

As mentioned earlier, the most popular approach is to route drivers according to a user equilibrium. In that way, drivers are routed along their respective lowest-impedance paths so there are no paths they would prefer to the ones they are given. The resulting flow pattern was originally introduced by Wardrop (1952) in order to model natural driver behavior, and it has been studied extensively in the literature. In fact, transportation engineers have used it to predict network utilization for planning purposes. Magnanti (1984), Sheffi (1985), Patriksson (1994), and Florian and Hearn (1995) provide a comprehensive treatment of mathematical formulations and algorithms for computing the static user equilibrium.

While a user equilibrium should satisfy the drivers, it does not necessarily minimize the total impedance (or latency) in the system, which is defined as the sum of all individual travel times. Roughgarden and Tardos (2002) provide examples that show that the total travel time in equilibrium can be arbitrarily large compared to that of the system optimum, although it is never more than the travel time incurred by optimally routing twice as much traffic.

Another unfavorable property of the user equilibrium is its non-monotonicity with respect to the network's capacity. This is illustrated by the Braess paradox, where adding a new road to a network with fixed demands actually increases the total travel time of the updated user equilibrium (Braess 1968; Sheffi 1985; Hagstrom and Abrams 2001).

Merchant and Nemhauser (1978) recognized that the assumptions of the traffic assignment problem are unrealistic and proposed to consider a *dynamic* model. Since then, there has been significant effort towards the dynamic analysis of traffic networks (e.g., Ben-Akiva 1985; Friesz 1985). Unlike static traffic assignment, where models and solution methods are well established, the dynamic traffic assignment problem has been studied from several different perspectives with no single generally accepted model or methodology. We refer the reader to the article by Mahmassani and Peeta (1995), which provides a discussion of the inherent difficulties and corresponding solution attempts.

1.2. **A different approach.** From a global perspective, e.g., the traffic authority's point of view, it is certainly desirable to explicitly minimize the total travel time (i.e., to compute a system optimum). In particular, the existing road network could then carry more traffic (Lafortune, Sengupta, Kaufman, and Smith 1991; Ferris and Ruszczyński 1997). Yet, users' needs have to be taken into account: directly implemented, this policy could route some drivers on unacceptably long paths in order to use shorter paths for many other drivers. In fact, the length of a route in the system optimum can be higher than in user equilibrium, even in the pathological case of a single origin-destination pair (Roughgarden 2002). This is critical because routes can only be recommended to drivers. It is reasonable to assume that only very few of them would be willing to sacrifice their own short routes for the benefit of the "community". On the other hand, user acceptance of a route guidance system is important if it is supposed to help in reducing traffic congestion. Therefore, Beccaria and Bolelli (1992) have suggested to "find the route guidance strategy which minimizes some global and community criteria with individual needs as constraints."

We adopt a system optimum approach but honor the individual needs by imposing additional constraints to ensure that drivers are assigned to "acceptable" paths only. More precisely, we introduce the concept of the *normal length* of a path, which can be either its traversal time in the uncongested network, its traversal time in user equilibrium, its geographic distance, or any other appropriate measure. The only condition imposed on the normal length of a path is that it may not depend on the actual flow on the path. Equipped with this definition, we look for a *constrained system optimum* in which no path carrying positive flow between a certain origin-destination (OD) pair is allowed to exceed the normal length of a shortest path between the same OD pair by more than a tolerable factor. By doing so, we achieve our primary goal of finding solutions that are *fair* and *efficient* at the same time.

The novelty of our work consists in defining a constrained system optimum with the "right" set of allowable paths. We demonstrate that this model leads to a significantly better utilization of a traffic network than the standard traffic assignment (user equilibrium) and still guarantees fairness similar to that in the user equilibrium. To the best of the authors' knowledge, no other work introduces a constrained system optimum approach that guarantees fairness comparable to that of the ordinary traffic assignment. While this paper studies the method from a computational perspective, Schulz and Stier Moses (2003) analyzed this idea theoretically and provided estimates of the efficiency gain when using constrained system optima instead of user equilibria. In a forthcoming paper, Schulz and Stier Moses (2004) extend this study and present theoretical results on the fairness of constrained system optima.

After specifying the problem and the proposed model in Section 2, we present an algorithm for its solution in Section 3. It is based on a method called Partan, which is a revised version of the Frank-Wolfe algorithm. In Section 4, we give computational results obtained with our implementations. Many of the real-world instances that we used were kindly provided by DaimlerChrysler AG, Berlin. Additional instances were retrieved from an online library called *Transportation Network Test Problems* (Bar-Gera 2002).

2. The Model

We consider a model of reactive route guidance that allows us to work with static flows. While not considering dynamic flows may preclude the direct application to real-world situations, our approach can provide traffic planners with bounds on the total travel time that are more accurate (compared to the ordinary system optimum). Moreover, Sheffi (1985) points out that there are times when traffic exhibits steady-state behavior; e.g., during rush hours. If nothing else, this research is a first step in explicitly incorporating system-wide effects into route guidance systems.

We assume that all drivers use the route guidance system and that they actually follow the recommended routes. Admittedly, this assumption is relatively strong, but this should be considered a first step. Future research will explore the design of *consistent route guidance systems* that optimize efficiency without comprising user acceptance. One way to model a non-perfect market penetration is by considering two classes of users. Some users have access to route guidance devices and follow the recommendations, while the remaining users act selfishly. In this extension, a central question is that of creating a traffic pattern for the guided users that is fair and minimizes the total travel time (for all users, including those without guidance). Along this direction, Roughgarden (2001) studied how to compute an optimal strategy in a network consisting of a set of parallel links.

2.1. **Preliminaries.**

We represent the road network by a directed multigraph $G = (V, A)$ with two attributes on each arc $a \in A$: the normal length $\ell_a \geqslant 0$ serves as an a priori estimate for its traversal time in the solution we seek; the link delay function $\tau_a : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ maps x_a, the volume of traffic on arc a, to its actual traversal time $\tau_a(x_a)$. Normal lengths can be chosen to be any metric for the arcs that is fixed in advance. However, their proper choice will allow us to produce solutions with desirable features; we refer to Section 4 for details.

Link delay functions τ_a measure the impedance of arcs for different congestion levels. We require them to be nondecreasing and differentiable, and $\tau_a(x_a)x_a$ to be convex. These requirements are naturally met by common link delay functions used to reflect congestion effects (Branston 1976; Sheffi 1985; Cohen 1991). Figure 1 illustrates their typical shape: after they reach the practical capacity c_a (Patriksson 1994), they grow very fast. In our computations, we use the function put

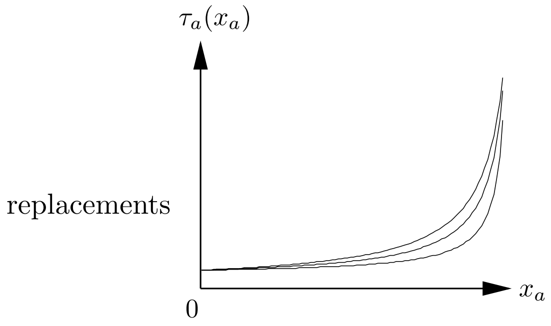

FIGURE 1. Typical link delay functions. Here, x_a is the flow on arc a, and $\tau_a(x_a)$ is the associated travel time.

forward by the U.S. Bureau of Public Roads:

$$\tau_a(x_a) := \tau_a^0 \left(1 + \alpha \left(\frac{x_a}{c_a}\right)^\beta\right) ,$$

where $\tau_a^0 > 0$ is the travel time in the uncongested network (also called free-flow travel time), and $\alpha \geqslant 0$ and $\beta \geqslant 0$ are tuning parameters.

We model vehicles with the same origin and destination as one commodity; K is the set of all commodities. For each commodity $k \in K$, $(s_k, t_k) \in V \times V$ denotes the associated origin-destination (OD) pair. The demand rate $d_k > 0$ for $k \in K$ represents the amount of flow to be routed for commodity k (vehicles per time unit). We denote the set of paths connecting OD pair k by $\mathcal{P}_k := \{P : P \text{ is a directed path from } s_k \text{ to } t_k\}$, and the complete set of paths by $\mathcal{P} := \bigcup_{k \in K} \mathcal{P}_k$. For a given flow x and a path $P \in \mathcal{P}$, its actual traversal time is $\tau_P(x) := \sum_{a \in P} \tau_a(x_a)$, while $\ell_P(x) := \sum_{a \in P} \ell_a$ is its normal length.

We assess the quality of a particular traffic assignment using two criteria. Its (un)fairness is of direct importance to users, while the total travel time in the system matters to the traffic authority. Let us discuss unfairness first.

2.2. **Measures of unfairness.** Without any centralized control, one would expect that different users traveling between the same OD pair experience similar travel times. In fact, if this were not the case, users would have an incentive to switch routes. In a seminal contribution, Wardrop (1952) stated the following principle that formalizes this notion:

> The journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.

A traffic pattern satisfying this principle is commonly called a user equilibrium (Dafermos and Sparrow 1969). It is "fair" in the sense that users between the same OD pair encounter the same delay. However, it is well known that a user equilibrium does in general not minimize the total travel time in the system. Our goal is to select more efficient traffic patterns without loosing the fairness property. To make this more precise, let us introduce several notions of *unfairness* of a solution. For a given flow, we define the unfairness of a particular traveler as follows:

Loaded unfairness: ratio of her experienced travel time to the experienced travel time of the fastest traveler for the same OD pair, where "experienced travel time" means travel time measured in terms of the current congestion level.

Normal unfairness: ratio of the length of her path to the length of the shortest path for the same OD pair, both measured with respect to normal arc lengths.

User equilibrium (UE) unfairness: ratio of her experienced travel time to the travel time for the same OD pair in a user equilibrium (which is the same for all users of that OD pair).

Free-flow unfairness: ratio of her experienced travel time to the length of the fastest path for the same OD pair w.r.t. free-flow travel times.

The respective notion of unfairness for a particular flow is the maximum over all OD pairs of the maximum unfairness of a traveler between that OD pair. More formally, for a given flow x and an equilibrium flow f,

$$\begin{aligned} \text{Loaded unfairness}(x) &:= \max\{\tau_{P_1}(x)/\tau_{P_2}(x) : P_1, P_2 \in \mathcal{P}_k,\, x_{P_1}, x_{P_2} > 0,\, k \in K\}; \\ \text{Normal unfairness}(x) &:= \max\{\ell_{P_1}/\ell_{P_2} : P_1, P_2 \in \mathcal{P}_k,\, x_{P_1} > 0,\, k \in K\}; \\ \text{UE unfairness}(x) &:= \max\{\tau_{P_1}(x)/\tau_{P_2}(f) : P_1, P_2 \in \mathcal{P}_k,\, x_{P_1} > 0,\, f_{P_2} > 0,\, k \in K\}; \\ \text{Free-flow unfairness}(x) &:= \max\{\tau_{P_1}(x)/\tau_{P_2}(0) : P_1, P_2 \in \mathcal{P}_k,\, x_{P_1} > 0,\, k \in K\}. \end{aligned}$$

The notions of loaded and normal unfairness are similar. Both compare, using different metrics, the travel times of users to the shortest travel times they could have had. The UE unfairness, introduced by Roughgarden (2002) in the single-commodity context, indicates how the travel times of the solution relate to those in user equilibrium. In practice though, drivers typically do not know the travel times in equilibrium; it is arguably more important to them how their travel times compare to the actual travel times of others. The free-flow unfairness measures the degradation of performance that users experience due to the prevalence of congestion effects. Note that the normal unfairness and the loaded unfairness are always greater than or equal to 1, while the UE unfairness and the free-flow unfairness can be any nonnegative number.

2.3. **Problem formulation.** As it is difficult to directly control the loaded unfairness, we will instead impose an upper bound on the normal unfairness and show that by doing so the other notions of unfairness will be small as well. In particular, we consider solutions for which the normal length of any used path between OD pair k is not much greater than that of a shortest s_k-t_k-path (with respect to normal lengths), for all $k \in K$. More specifically, we fix a tolerance factor $\varphi \geqslant 1$ and restrict the normal unfairness to be smaller than φ. In other words, a path $P \in \mathcal{P}_k$ is *feasible* if $\ell_P \leqslant \varphi\, L_k$. Here, $L_k := \min_{P \in \mathcal{P}_k} \ell_P$ is the normal length of a shortest path between s_k and t_k. If we let $\mathcal{P}_k^\varphi$ denote the set of all feasible paths for OD pair k, we can define the entire set of feasible paths as $\mathcal{P}^\varphi := \bigcup_{k \in K} \mathcal{P}_k^\varphi$.

Because route guidance systems eventually have to propose paths to the drivers, our formulation is path-based: there is a decision variable x_P for each path $P \in \mathcal{P}^\varphi$. In fact, it is virtually impossible to model the restriction to feasible paths with the help of a formulation based on arc variables only. Moreover, even if one were (somehow) given an arc flow that has a decomposition into feasible paths, it is NP-hard to compute such a decomposition (Correa, Schulz, and Stier Moses 2003, Corollary 4). In contrast, user equilibria and ordinary system optima can be computed using arc-based formulations; any flow decomposition results in path flows with the desired property.

The constrained system optimum that we propose to use in route guidance systems is an optimal solution to the following min-cost multicommodity flow problem with separable convex objective function and path constraints:

$$\begin{aligned} \mathit{CSO}: \quad \min \quad C(x) &:= \sum_{a \in A} \tau_a(x_a) x_a \\ \text{s.t.} \quad \sum_{P \in \mathcal{P}_k^\varphi} x_P &= d_k \qquad k \in K, \\ \sum_{P \in \mathcal{P}^\varphi : a \in P} x_P &= x_a \qquad a \in A, \\ x_P &\geqslant 0 \qquad P \in \mathcal{P}^\varphi. \end{aligned}$$

Note that the flow variables are not required to be integral since they describe abstract flow rates. If paths were not restricted to be feasible (i.e., in $\mathcal{P}^{\varphi}$), an optimal solution to this formulation would coincide with an ordinary system optimum. We denote by CSO^{φ} an optimal solution to the problem with tolerance factor φ.

Figure 2 demonstrates the effect of path constraints on the system optimum. One commodity is routed through the road network between two clearly marked nodes. In the picture on the left, we display the (unconstrained) system optimum. The flow is distributed widely over the network in order to avoid high arc flows, which would incur high arc travel times. In the picture on the right, the same amount of flow is routed, but this time with the restriction that the normal length of any used path is at most 10% longer than that of the shortest path (i.e., $\varphi = 1.1$). In this example, the normal length has been chosen to be the geographic distance. Line thickness reflects arc capacity (light gray) and arc usage (black), respectively.

FIGURE 2. System optimum without and with restrictions on the normal length of paths, resp.

Before we discuss the computational complexity of problem *CSO* and algorithms to find a constrained system optimum, let us emphasize that this model is different from previous traffic assignment formulations with side constraints. The most commonly considered type of side constraints are explicit bounds on arc capacities. In fact, capacity constraints on individual arcs have been used since the work of Charnes and Cooper (1961) to improve the modeling of congestion effects (see also Hearn 1980); some traffic control policies give rise to arc flow capacity constraints as well (Yang and Yagar 1994); arc capacities can also be used to derive tolls for the reduction of flows on overloaded links, we refer to Bernstein and Smith (1994) for references. Moreover, several authors have discussed the algorithmic consequences of modeling arc capacities explicitly (Daganzo 1977a; Daganzo 1977b; Hearn 1980; Hearn and Ribera 1980; Hearn and Ribera 1981; Larsson and Patriksson 1994; Larsson and Patriksson 1995). Larsson and Patriksson (1999) have summarized and extended this work to general convex side constraints on the vector of arc flows.

Nonetheless, such constraints cannot be used to render certain paths infeasible, as we have argued earlier. Still, path-based multicommodity flow models similar to ours with explicit constraints on the set of allowable paths are frequently used in other application areas. A recent example is the work by Holmberg and Yuan (2003), who study routing problems in telecommunication networks and solve the resulting models by column generation. However, nobody has tried to capture aspects of system optimality and user fairness in a network with congestion effects, as we do.

3. Algorithms and Complexity

To solve problem *CSO*, we use a variant of the convex combination algorithm of Frank and Wolfe (1956). As it is well-known that the standard Frank-Wolfe algorithm sometimes shows poor convergence (see, e.g., Sheffi 1985; Patriksson 1994; Florian and Hearn 1995), we consider an improved version called *Partan* that was proposed by LeBlanc, Helgason, and Boyce (1985) and further studied by Florian, Guélat, and Spiess (1987) and Arezki and Van Vliet (1990), among others. As we cannot explicitly work with all variables x_P associated with paths $P \in \mathcal{P}^\varphi$, because there may be exponentially many, we only generate them when needed. For that reason, our algorithm can be considered to be a column generation method. The application of column generation to the computation of system optima and user equilibria was first studied by Gibert (1968) and Leventhal, Nemhauser, and Trotter (1973).

For the sake of completeness, let us briefly describe the Frank-Wolfe method.[1] Given a current solution, the algorithm solves in every iteration a linearized version of *CSO* to determine a feasible descent direction. As the linearization permits the decomposition of the problem by commodities, it is enough to call a subroutine for finding a shortest path in $\mathcal{P}_k^\varphi$ for each commodity $k \in K$. In the subsequent line search, the original nonlinear problem is solved restricted to the line defined by the feasible direction of descent. The algorithm terminates when a certain precision is achieved. To determine when this is the case, the convexity of the objective function is used to derive a lower bound on the value of an optimal solution. It is well known that this algorithm always converges to a global minimum (for convex programs). Partan is based on the same idea, but it performs a more intelligent line search. It determines the descent direction using the results of two consecutive iterations, thereby diminishing the zigzagging effect.

The substep of computing a shortest path in $\mathcal{P}_k^\varphi$ is exactly the so-called *constrained shortest path problem*; see Section 3.1 below. The only difference between the algorithm we just described and the version of Frank-Wolfe (or Partan) employed for computing user equilibria or system optima is the use of constrained shortest paths instead of regular shortest paths in the solution of the linear subproblems.

Note that other methods like *partial linearization algorithms* or *simplicial decomposition* can also be adapted to our problem. Since we want to make the point that constrained system optima are useful, it was not necessary to implement potentially more efficient algorithms as we can solve relatively large instances within acceptable time limits by using Partan. As others concluded before, for our purpose "...the [Frank-Wolfe] algorithm is considered sufficiently good for practical use" (Patriksson 1994). Nevertheless, if one wants to deploy these ideas in a real-time setting, more careful and efficient implementations are needed. We refer the reader to the books by Sheffi (1985) and Patriksson (1994) as well as the chapter by Florian and Hearn (1995) for comprehensive discussions of these algorithms as well as many others.

3.1. **The constrained shortest path problem.** Let us sketch how the computation of constrained shortest paths—the pricing component of our column generation approach—is carried out. In this subproblem, every arc $a \in A$ has two parameters, a traversal time τ_a and a length ℓ_a. Given an origin-destination pair (s, t), the objective is to compute a quickest path from s to t whose length does not exceed a given bound L. That is, one wants to solve the following problem:

$$\min\{\tau_P : P \text{ is a path from } s \text{ to } t \text{ such that } \ell_P \leqslant L\},$$

where $\tau_P := \sum_{a \in P} \tau_a$ and $\ell_P := \sum_{a \in P} \ell_a$. This problem is NP-hard (Garey and Johnson 1979).

For solving this problem, Aneja and Nair (1978) proposed to use Lagrangean relaxation; Ribeiro and Minoux (1986) added a branch-and-bound scheme. Aneja, Aggarwal, and Nair (1983) extended Dijkstra's algorithm to the case of two objective functions, and Climaco and Martins (1982) used path ranking.

Because of its superior computational efficiency, we implemented the label correcting algorithm of Aneja et al. (1983). The algorithm fans out from the start node s and labels each reached node $v \in V$ with labels of the form $(d_\tau(v), d_\ell(v))$. For each path from s to v that has been detected so

[1]For an in-depth description of the implemented algorithms, we refer to Jahn, Möhring, Schulz, and Stier Moses (2002).

far, $d_\tau(v)$ represents its traversal time and $d_\ell(v)$ its distance. During the course of the algorithm, several labels may have to be stored for each node v, namely the pareto-optimal labels of all paths that have reached it. This labeling algorithm can be interpreted as a special kind of branch-and-bound with a search strategy similar to breadth-first search. Starting from a certain label of v, one obtains lower bounds for the remaining paths from v to t by separately computing ordinary shortest path distances from v to t with respect to travel times τ_a and lengths ℓ_a, respectively. If one of these bounds is too large, the label can be dismissed.

3.2. **Computational complexity.** For the sake of completeness, let us also quickly discuss the computational complexity of problem *CSO*. Note that it includes as a specical case the situation in which all link performance functions are constant; i.e., $\tau_a(x_a) = \tau_a$ for all $a \in A$. Moreover, the set of feasible paths is only given implicitly. Hence, the input dimension is $|A| + |K|$. In fact, *CSO* is already NP-hard for $|K| = 1$, as this case amounts to solving a constrained shortest path problem.

4. Computational Study

The computational study is divided into three parts. First, we discuss which normal length should be used in practice. Next, we analyze efficiency vs. fairness of solutions for instances that arise from real-world networks. Finally, we briefly report on the performance of the algorithm itself.

The seven instances we used in this study come from two different sources. Four of them represent different parts of the actual road network of the city of Berlin, Germany, and were provided by DaimlerChrysler AG. Their demand rates stem from origin-destination polls conducted in Berlin. The other three come from the *Transportation Network Test Problems* website (Bar-Gera 2002). Table 1 shows the specifics of each instance. Instances are listed in increasing order of the product of the number of arcs and the number of commodities. This measure of complexity has been used in the literature (e.g., Holmberg and Yuan 2003), and it indeed corresponds to the ordering with respect to solution times. Instances range from rather small ones, which were included because they are standard in the literature, to fairly large ones.

Table 1. Problem instances used in the computational study

Instance Name	Short Name	Source	$\|V\|$	$\|A\|$	$\|K\|$	$\|A\| \cdot \|K\|$
Sioux Falls	SF	TNTP	24	76	528	40K
Friedrichshain	F	DC	224	523	506	265K
Winnipeg	W	TNTP	1,067	2,975	4,344	13M
Neukölln	N	DC	1,890	4,040	3,166	13M
Mitte, Prenzlauerberg & Friedrichshain	MPF	DC	975	2,184	9,801	21M
Chicago Sketch	CS	TNTP	933	2,950	83,113	245M
Berlin	B	DC	12,100	19,570	49,689	972M

The algorithm described in Section 3 was implemented in C++ using the GCC compiler under Linux; the computing platform was a Pentium IV based computer running at 2.4 GHz with 1 GB RAM.

4.1. **Choice of normal length.** We initially considered three possible ways to define the normal length of an arc: geographic distances, free-flow travel times, and travel times when the network is in user equilibrium. Recall that normal lengths can only be static; for instance, it is not possible to consider travel times under the current solution with the methodology described in this paper. The advantage of keeping the model simple is a fast algorithm that still produces solutions with small total travel time and low unfairness. It is important to remark that users do not need to know the normal lengths; they are just an artifact of our algorithm to select solutions that are approximately fair.

Geographic distances and free-flow travel times are highly correlated; therefore, one cannot expect significant differences between solutions resulting from choosing either one as the normal length. For free-flow travel times, Schulz and Stier Moses (2003) showed, and our runs confirm, that the total travel time of user equilibria is smaller than that of constrained system optima when the factor φ is too small. Consequently, to obtain an improvement in the total travel time, bigger factors must be considered. However, this gives rise to relatively high unfairness, which is undesired. As an example, consider instance Neukölln. The graph on the left in Figure 3 shows

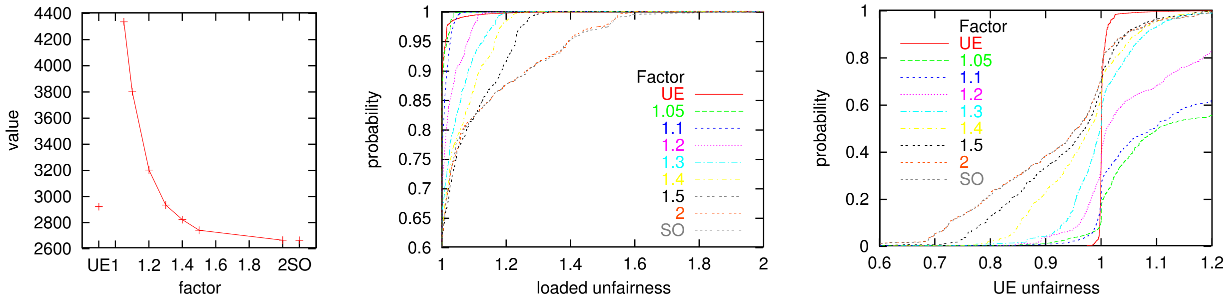

FIGURE 3. Objective values and unfairness distributions for instance Neukölln and normal lengths equal to free-flow travel times

the value of the objective function for different tolerance factors φ, for the user equilibrium (UE), and for the system optimum (SO). Factors smaller than 1.4 are not helpful because the total travel time of the corresponding solutions is greater than the total travel time in user equilibrium. The other two graphs in Figure 3 depict the distribution of unfairness across users for varying tolerance factors; for instance, for factor $\varphi = 1.5$, 80% of all users will experience a loaded unfairness of less than 1.1. This value increases to 1.2 if one considers 90% of all users. For factors greater than 1.5, the distributions are quite similar to that of the system optimum. In the graph on the right, note that for small tolerance factors most users end up traveling longer than they would in user equilibrium. This happens because there are not enough alternative paths between any one OD pair, which explains the poor quality of the solutions under this choice of normal length.

We therefore propose to make use of the travel times in user equilibrium when defining normal arc lengths, which results in high-quality solutions. Indeed, for any factor φ, the user equilibrium itself is a feasible solution to the constrained system optimum problem. Therefore, for all $\varphi \geqslant 0$,

$$C(CSO^{\varphi}) \leqslant C(UE),$$

which guarantees that the optimal solution to problem *CSO* is never worse than the user equilibrium in terms of the total travel time in the system. The advantage of this normal length definition is that it is flow-dependent; it provides a better indication which paths should be selected. Let us repeat that users do not need to know the user equilibrium; it is just an ingredient for the computation of the constrained system optimum.

Figure 4 displays graphs similar to the ones in Figure 3 for this choice of normal length. Most notably, total travel times are distinctively smaller than in equilibrium, while the fraction of users traveling longer than in equilibrium is substantially smaller. We therefore limit our analysis in the sequel to this version of normal length; that is, we assume user equilibrium travel times are used to define normal lengths.

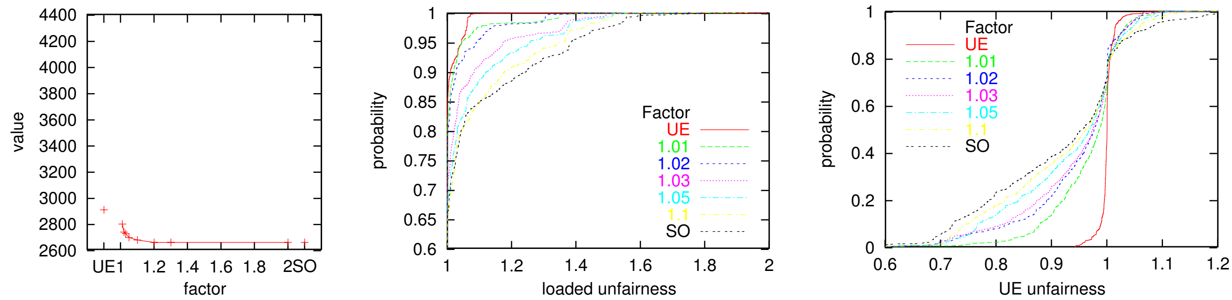

FIGURE 4. Objective values and unfairness distributions for instance Neukölln and normal lengths equal to travel times in user equilibrium

4.2. **Quality of constrained system optima.** Tables 2 and 3 exhibit the output of the algorithm for the instances presented in Table 1 and varying tolerance factors. Every row represents one run for the factor reported in the first column. The column *objective value* is the total travel time of the solution; the column *number of paths* contains the number of paths with positive flow, which is an indication of the complexity of the solution. In addition, the tables include the 99th percentiles of the different *unfairness* distributions, the *number of iterations* (one iteration consists of solving the linearized problem and performing the line search; see Section 3), and the *time* (in seconds) needed to reach the target optimality gap of 0.5%.

For example, the third row for instance Friedrichshain portrays the attributes of the constrained system optimum with tolerance factor $\varphi = 1.02$. The total travel time is 621, and the users between the 506 different OD pairs are assigned to 1,290 different paths. The actual travel time for 99% of all users is not more than 65.7% than that of the fastest route between their OD pair. Compared to the user equilibrium, their individual travel time are at most 11.7% higher. Note that the corresponding quantities for the system optimum (10th row) are 106.3% and 25%, respectively.

Before we interpret the computational results, let us call attention to an apparent anomaly in the rows of Tables 2 and 3 that correspond to user equilibria. In theory, the normal unfairness, the loaded unfairness, and the UE unfairness should be equal to 1; however, in practice they are obviously not. The reason is that each user equilibrium is computed as the optimal solution of an appropriately defined convex optimization problem as per Beckmann, McGuire, and Winsten (1956). As the algorithm terminates as soon as the value of the current solution is within 0.5% of that of an optimal solution, the solution reported here is merely an approximate user equilibrium. In some sense, the normal unfairness, the loaded unfairness, and the UE unfairness give information about its actual deviation from a user equilibrium. Incidentally, in the derivation of the normal arc lengths, we computed the user equilibrium with higher precision, namely a target optimality gap of 0.01% instead of 0.5%. This explains why the 99th percentiles of normal unfairness, loaded unfairness, and UE unfairness of the user equilibrium are not necessarily equal to one another.

Clearly, the larger the tolerance factor φ the closer is the objective function value of an associated constrained system optimum to that of the unconstrained system optimum, and the higher is its unfairness. On the other hand, smaller tolerance factors lead to "fairer" solutions but also result in larger gaps of the total travel time compared to the unconstrained system optimum. However, we will argue that a carefully chosen tolerance factor strikes a good balance between these two conflicting effects. For the sake of argument, let us consider instance Neukölln with $\varphi = 1.02$.

The gap between the total travel time of $CSO^{1.02}$ and that of the system optimum is about a third of the gap between the user equilibrium and the system optimum. In fact, the travel time of the system optimum is 2,653 compared to 2,903 in user equilibrium and 2,732 for $CSO^{1.02}$. Moreover, the travel time of 99% of all users in $CSO^{1.02}$ is at most 30.4% higher than that of any

Table 2. Characteristics of constrained system optima with different tolerance factors, Part I

factor	objective value	number of paths	99th unfairness percentile				number of iterations	runtime (sec.)
			normal	loaded	UE	free-flow		
Sioux Falls								
UE	7448	989	1.001	1.040	1.031	5.098	31	0
1.01	7263	749	1.001	1.282	1.187	4.908	27	0
1.02	7256	754	1.001	1.258	1.184	4.901	38	0
1.03	7251	758	1.001	1.265	1.195	4.789	34	0
1.05	7239	812	1.035	1.290	1.210	4.749	32	0
1.10	7216	893	1.060	1.283	1.178	4.712	56	0
1.20	7207	984	1.078	1.295	1.168	4.573	46	0
1.30	7201	1129	1.092	1.296	1.170	4.598	64	0
SO	7199	1326	1.092	1.295	1.169	4.599	78	0
Friedrichshain								
UE	682	1713	1.011	1.036	1.062	4.382	27	0
1.01	628	1283	1.008	1.657	1.087	4.163	45	1
1.02	621	1290	1.017	1.652	1.117	4.132	30	1
1.03	613	1515	1.029	1.711	1.094	4.124	42	1
1.05	612	1594	1.046	1.733	1.092	4.130	43	1
1.10	594	1598	1.096	1.929	1.109	3.565	40	1
1.20	591	2080	1.170	2.060	1.177	3.932	74	1
1.30	591	2251	1.213	2.058	1.229	3.948	59	1
SO	591	2631	1.213	2.063	1.250	3.947	63	1
Winnipeg								
UE	857	14633	1.029	1.050	1.047	1.503	16	7
1.01	844	10224	1.009	1.119	1.027	1.429	15	8
1.02	842	11901	1.017	1.123	1.019	1.402	16	8
1.03	842	13123	1.027	1.142	1.027	1.389	18	8
1.05	842	15374	1.043	1.164	1.044	1.409	23	10
1.10	841	17846	1.068	1.192	1.054	1.411	30	12
1.20	841	18619	1.075	1.203	1.058	1.429	33	13
1.30	841	18755	1.078	1.210	1.068	1.458	30	12
SO	841	19331	1.076	1.211	1.066	1.449	33	14
Neukölln								
UE	2903	6744	1.025	1.063	1.053	3.806	21	17
1.01	2794	4380	1.008	1.332	1.084	3.182	15	7
1.02	2732	4700	1.015	1.304	1.072	3.054	17	8
1.03	2721	5665	1.028	1.420	1.070	3.079	18	8
1.05	2690	6427	1.045	1.450	1.099	2.987	22	10
1.10	2672	8755	1.091	1.493	1.125	2.944	47	17
1.20	2653	10018	1.168	1.527	1.179	2.292	54	17
1.30	2653	7983	1.183	1.539	1.193	2.327	48	15
SO	2653	8631	1.187	1.555	1.197	2.335	58	48

TABLE 3. Characteristics of constrained system optima with different tolerance factors, Part II

factor	objective value	number of paths	99th unfairness percentile				number of iterations	runtime (sec.)
			normal	loaded	UE	free-flow		
Mitte, Prenzlauerberg & Friedrichshain								
UE	1845	28091	1.015	1.040	1.032	2.236	16	9
1.01	1771	32476	1.008	1.304	1.051	2.086	25	30
1.02	1762	34618	1.017	1.291	1.045	1.993	25	30
1.03	1755	35392	1.026	1.303	1.045	2.008	24	27
1.05	1733	39320	1.046	1.358	1.060	1.808	26	22
1.10	1727	48968	1.086	1.451	1.083	1.881	29	14
1.20	1726	56687	1.122	1.478	1.122	1.918	37	17
1.30	1726	56304	1.123	1.477	1.124	1.910	35	15
SO	1726	64431	1.127	1.471	1.126	1.921	40	24
Chicago Sketch								
UE	18383	194564	1.017	1.039	1.046	1.592	9	46
1.01	18123	119696	1.007	1.101	1.052	1.543	4	27
1.02	18047	155800	1.016	1.123	1.047	1.509	8	46
1.03	18016	192152	1.025	1.148	1.044	1.492	11	57
1.05	17993	242188	1.043	1.193	1.055	1.499	14	69
1.10	17971	289999	1.072	1.211	1.074	1.504	19	89
1.20	17970	334364	1.081	1.227	1.090	1.496	25	118
1.30	17976	344830	1.085	1.224	1.092	1.498	24	118
SO	17981	331146	1.087	1.238	1.093	1.496	25	117
Berlin								
UE	16223	150922	1.038	1.057	1.058	2.400	15	1584
1.01	16254	98271	1.008	1.135	1.906	3.191	9	904
1.02	15806	142944	1.018	1.214	1.112	2.181	14	1274
1.03	15671	171452	1.028	1.247	1.066	2.058	19	1626
1.05	15632	216328	1.045	1.270	1.060	2.003	29	2247
1.10	15587	257707	1.084	1.333	1.083	2.000	39	2689
1.20	15572	295138	1.126	1.372	1.120	2.016	49	3614
1.30	15565	307050	1.137	1.398	1.128	2.022	52	4184
SO	15544	322687	1.148	1.438	1.135	2.066	56	5512

other traveler (between the same OD pair), compared to 55.5% in the system optimum. In other words, the reduction of unfairness amounts roughly to 45%. The numbers are similar for most of the other instances.

Figures 5 and 6 depict the complete unfairness distributions for all instances. Let us again pick Neukölln to highlight typical effects. In $CSO^{1.02}$, the travel time of just 4.5% of all users is more than 10% than that of the fastest paths of their OD pairs. In contrast, this number is 15.3% for the ordinary system optimum; i.e., one sixth of all drivers experience delays that are significantly above and beyond that of their fellow drivers. Moreover, most users (around 80%) spend less time on the road than they would in equilibrium. Actually, for factor 1.02, only 0.3% of the users travel 10% more than in equilibrium. Compare this number to the 4.6% that travel at least 10% longer under the system optimum.

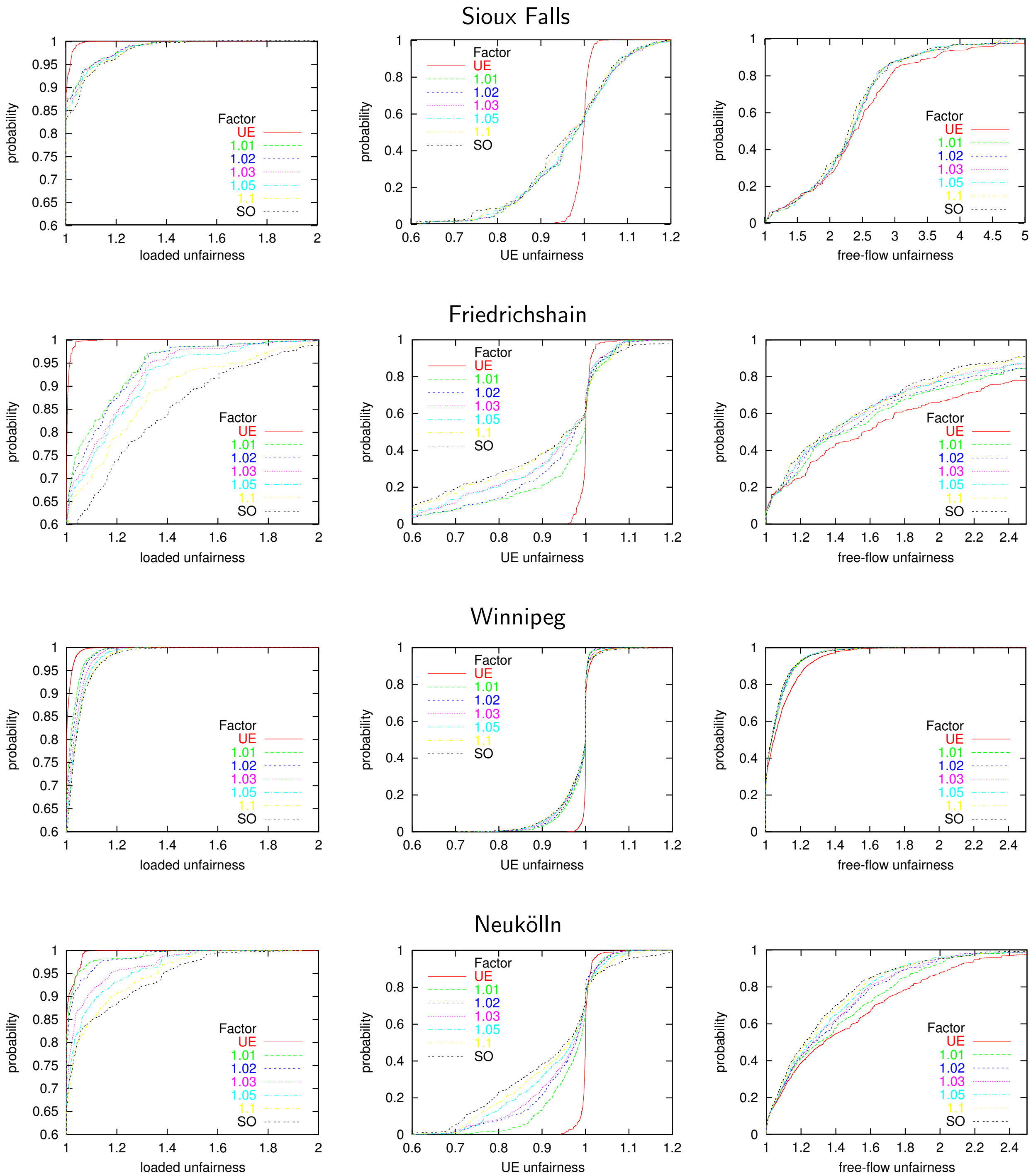

FIGURE 5. Unfairness distributions for various tolerance factors, Part I

To facilitate a comparison of the characteristics of constrained system optima with different tolerance factors, Figures 7–13 plot various percentiles of the different notions of unfairness. The

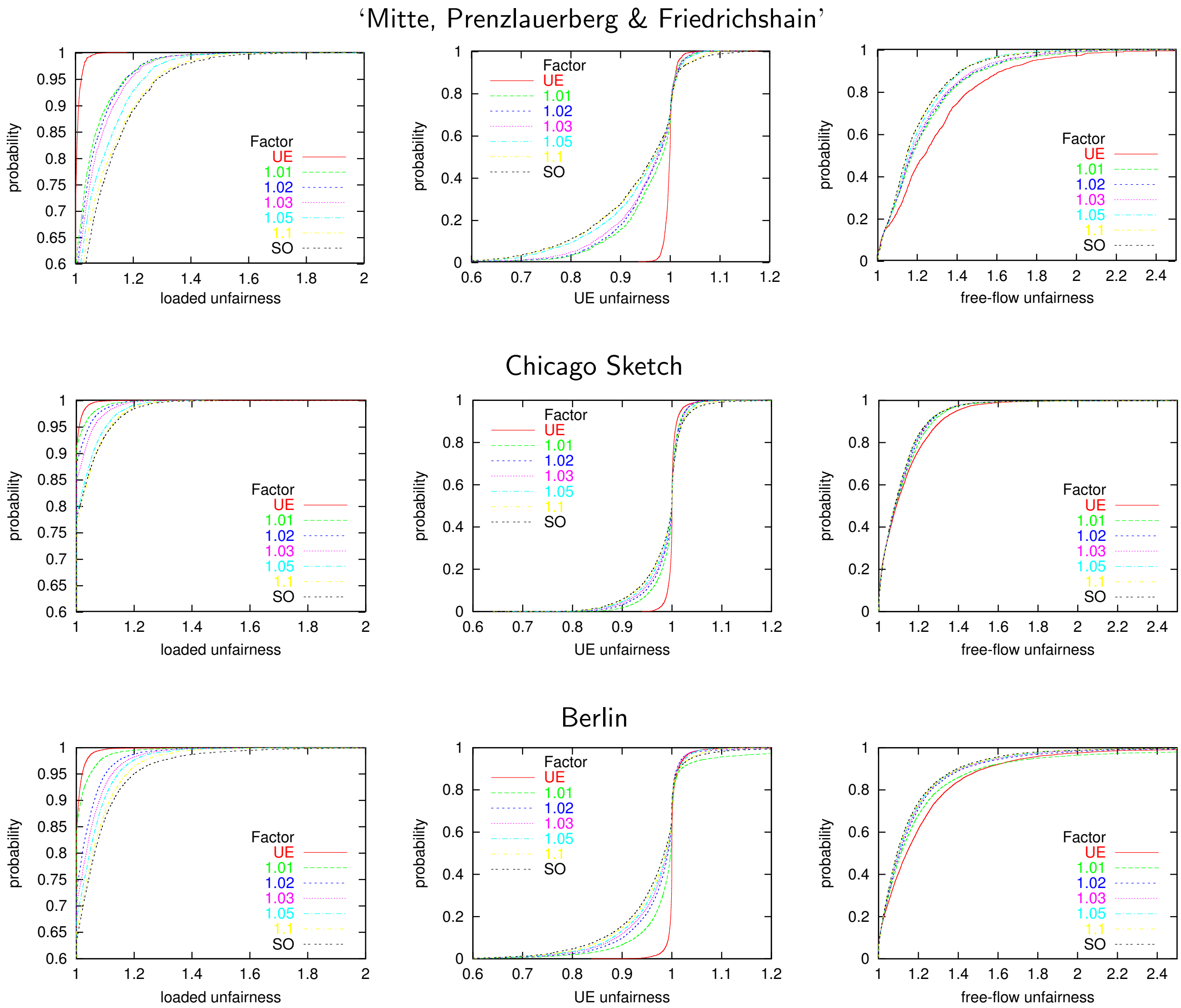

FIGURE 6. Unfairness distributions for various tolerance factors, Part II

two diagrams on top of each figure represent the 95th and 99th percentile, respectively, of the four notions of unfairness. The four remaining graphs correspond to each unfairness definition and show the 95th, 97.5th and 99th percentiles, respectively.

Let us draw attention to some typical effects, and we will once again use instance Neukölln when we need to mention concrete numbers. We first compare the travel times of users in any of the computed route guidance solutions to the length of their shortest paths in the uncongested network (free-flow unfairness). It is remarkable that for virtually all tolerance factors in our study, the increase of travel time due to congestion effects is significantly smaller than the corresponding increase in the (approximate) user equilibrium. For instance, for Neukölln and the 99th percentile, the free-flow unfairness for all constrained system optima is about 3 or lower, while the free-flow unfairness of the user equilibrium is 3.8. The significance of this observation is only reinforced by the fact that at equilibrium all users between the same OD pair experience the same delay, while this is not necessarily the case in a constrained system optimum. The second important observation

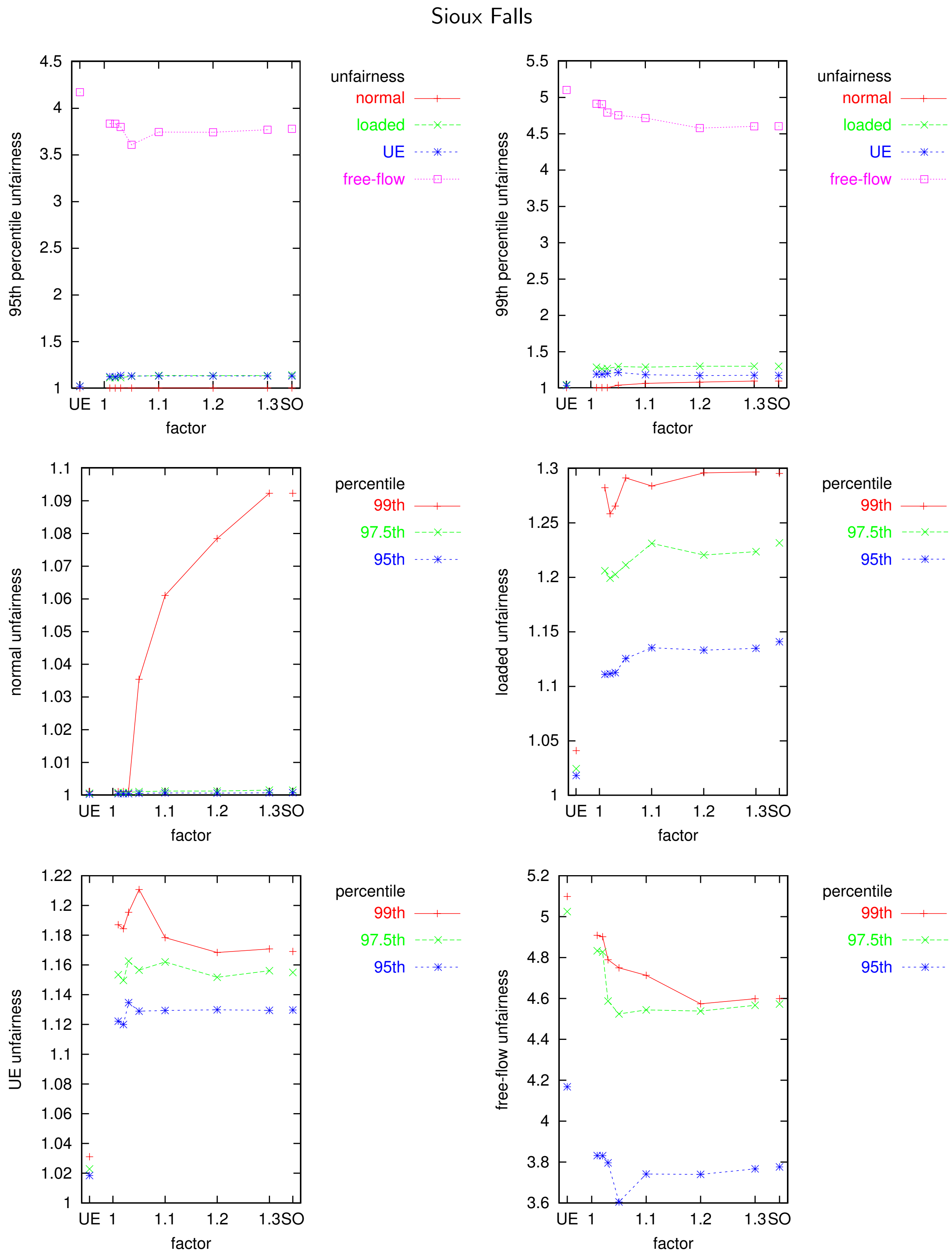

FIGURE 7. Unfairness over the different factors and percentiles for instance Sioux Falls

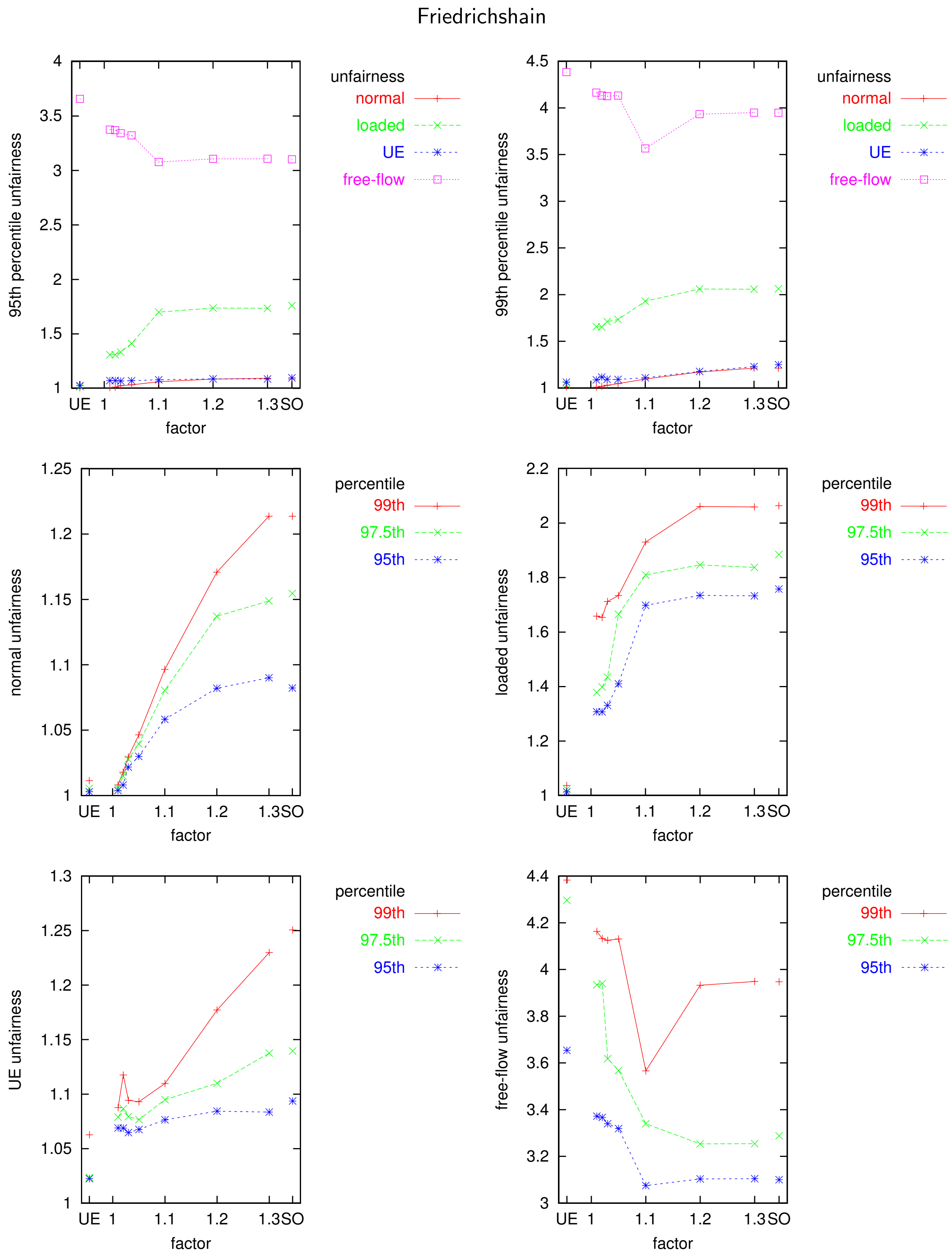

FIGURE 8. Unfairness over the different factors and percentiles for instance Friedrichshain

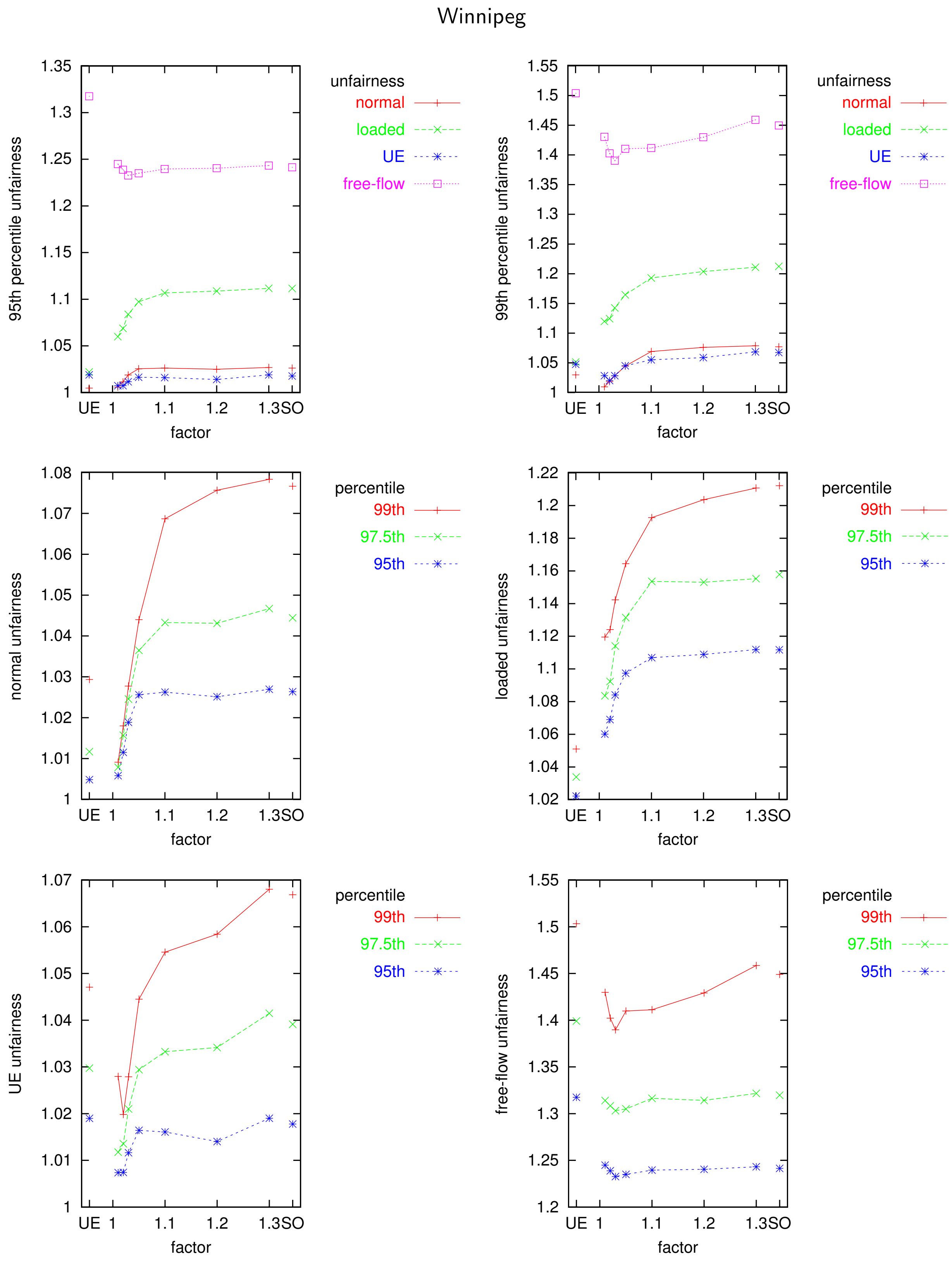

FIGURE 9. Unfairness over the different factors and percentiles for instance Winnipeg

‘Mitte, Prenzlauerberg & Friedrichshain’

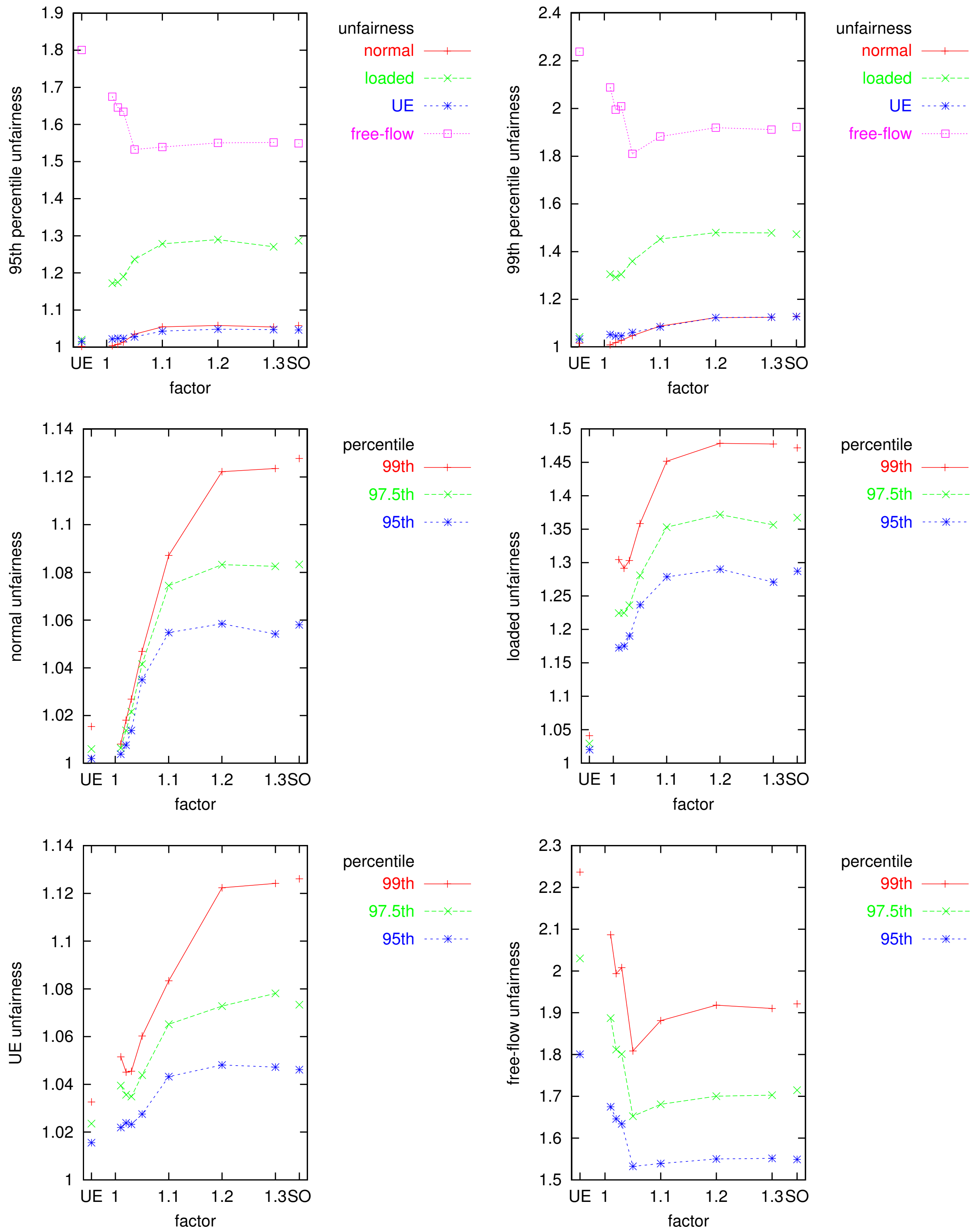

FIGURE 10. Unfairness over the different factors and percentiles for instance ‘Mitte, Prenzlauerberg & Friedrichshain’

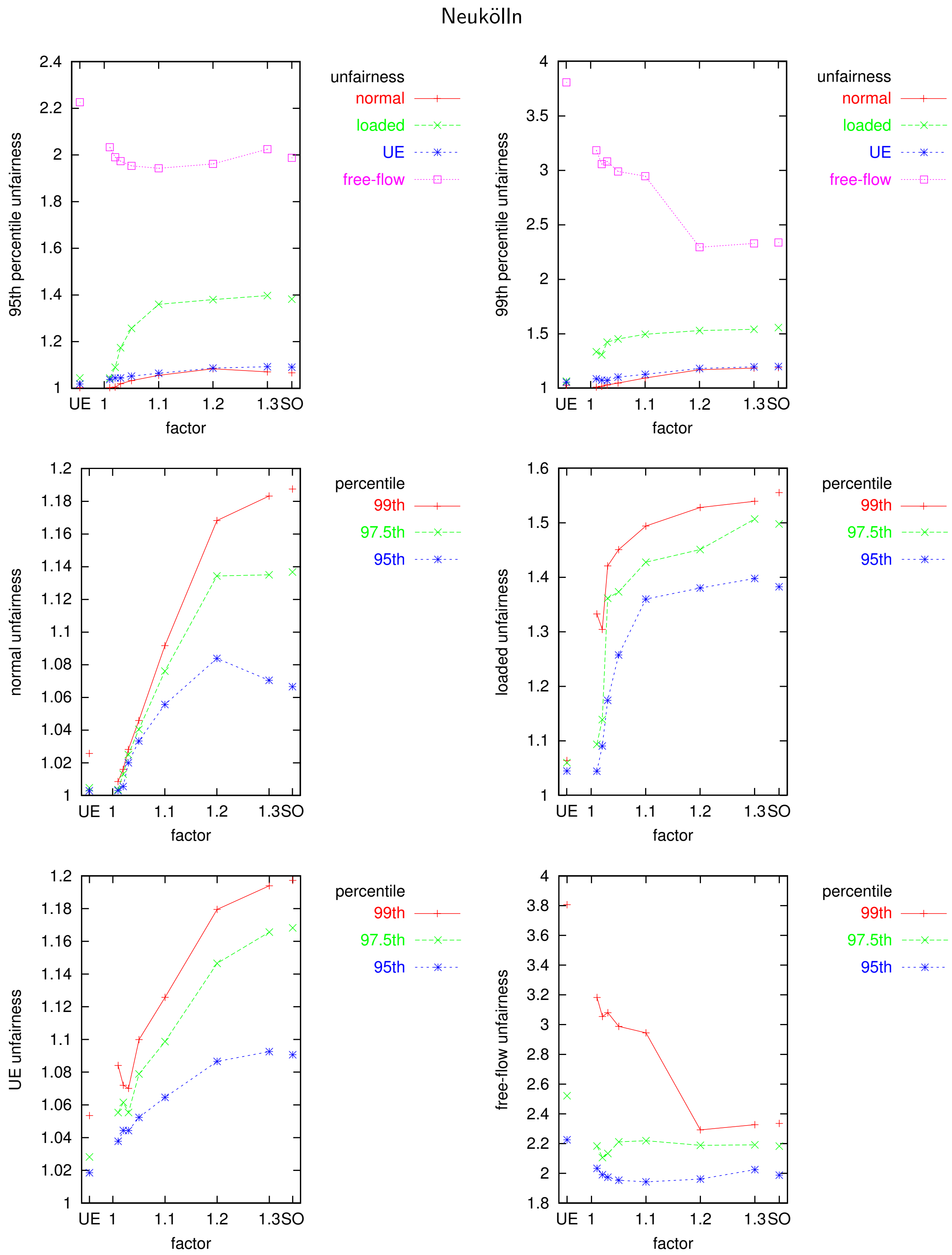

FIGURE 11. Unfairness over the different factors and percentiles for instance Neukölln

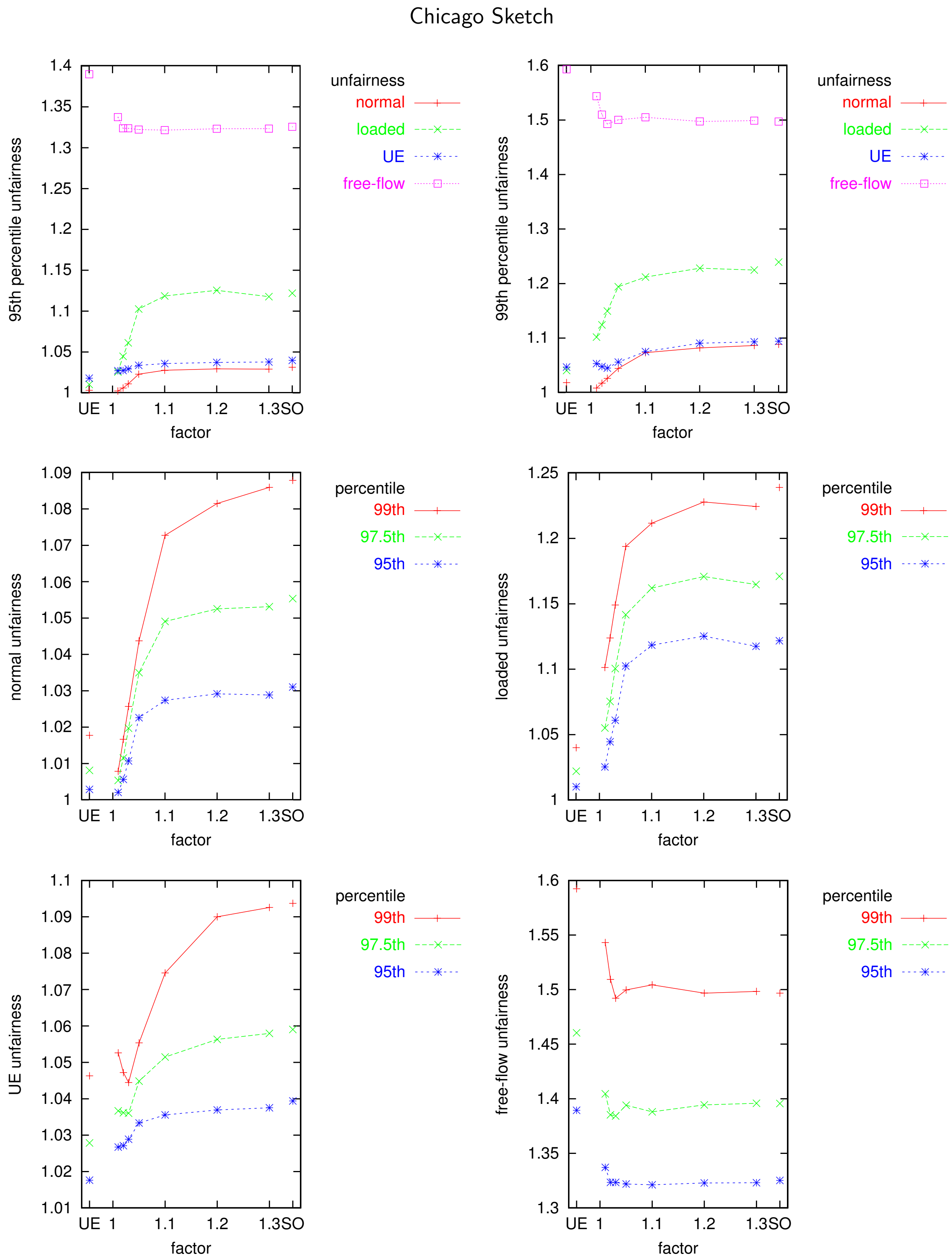

FIGURE 12. Unfairness over the different factors and percentiles for instance Chicago Sketch

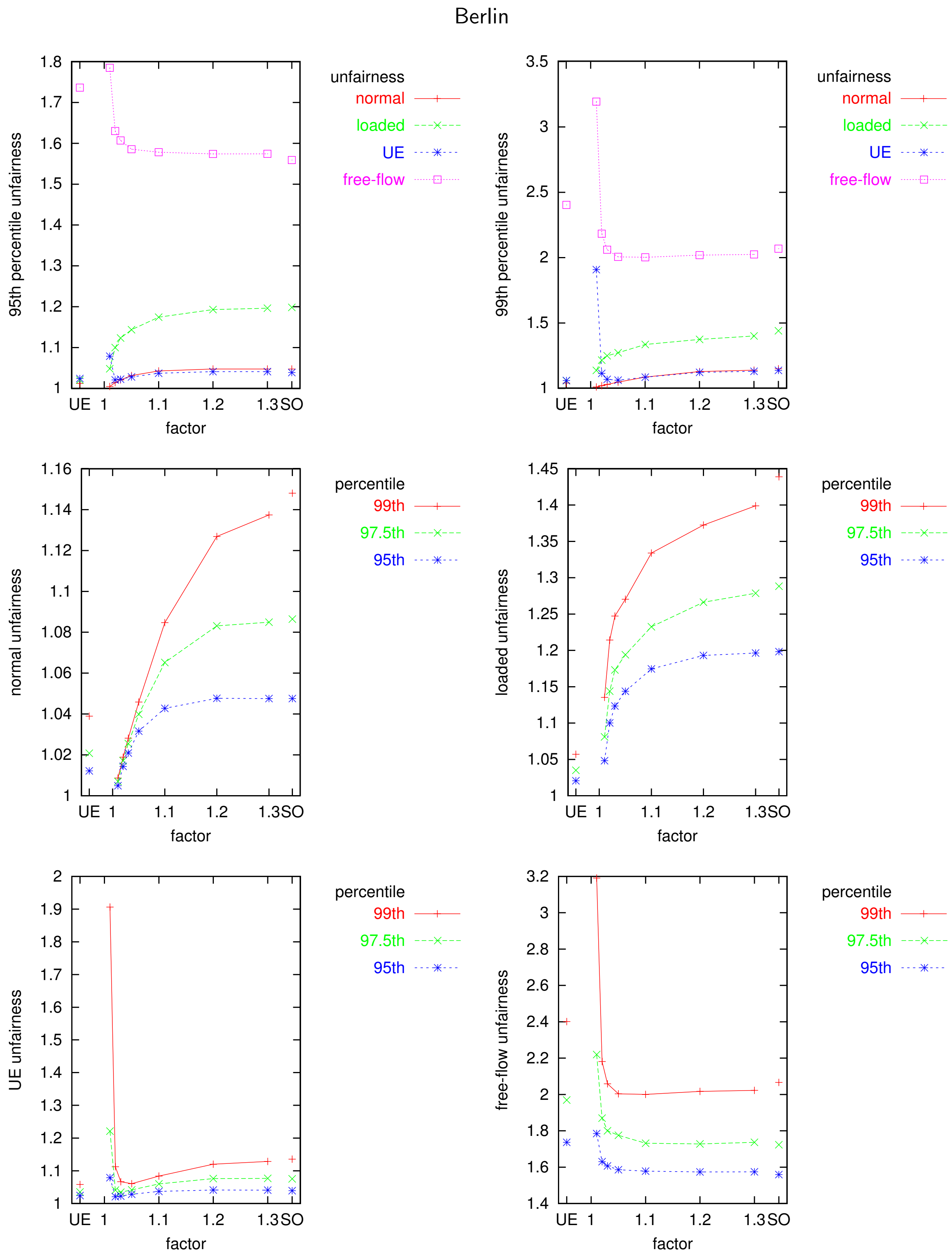

FIGURE 13. Unfairness over the different factors and percentiles for instance Berlin

to be made is the strong correlation between the loaded unfairness and the normal unfairness, which is illustrated by the two diagrams in the middle of each figure. Bounding the normal unfairness (a static measure) results in bounded loaded unfairness (a dynamic measure), which explains why our approach is successful.

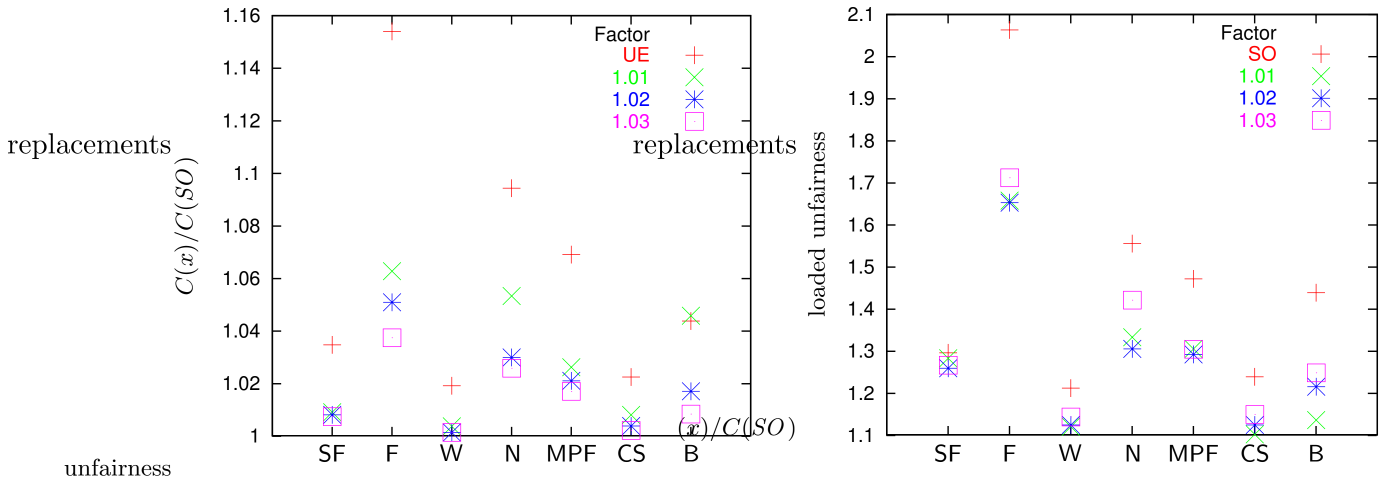

FIGURE 14. Efficiency and loaded unfairness of constrained system optima across all instances. The plot on the left shows the efficiency (the cost of the solution over the cost of the system optimum) of select constrained system optima vs. that of the associated user equilibria; the plot on the right compares the loaded unfairness of the same solutions with that of the corresponding system optima.

Figures 14 and 15 provide conclusive evidence of the benefits of the solutions we propose; constrained system optima with appropriately chosen tolerance factors bring together the favorable attributes of user equilibria and system optima. In Figure 14, we display constrained system optima with tolerance factors close to 1.02 and compare them with the user equilibrium and the unconstrained system optimum, both in terms of efficiency and fairness. Figure 15 illustrates the tradeoff between efficiency and fairness achieved by constrained system optima. The graph shows, for each of the instances we studied, system optima (on the left), user equilibria (at the bottom) and the intermediate solutions represented by constrained system optima (in the center). The circled data-points correspond to $CSO^{1.02}$, for the various instances. In summary, constrained system optima with user equilibrium travel times as normal lengths provide a handle to effectively control the tradeoff between fairness and efficiency.

4.3. **Performance of the algorithm.** Let us briefly discuss our findings with respect to the running time needed by the algorithm described in Section 3. Figure 16 shows a detailed study of the effects of varying the tolerance factor and the target optimality gap. We only present the results for instances Chicago Sketch and Berlin because they are the largest and hence arguably the most difficult ones. For each selected instance, the figure contains a graph describing the objective function value, another one illustrating the number of iterations, and finally one displaying the computation time (in seconds).

Most notably, the time needed by our algorithm to compute a constrained system optimum is typically not larger than that for computing an unconstrained system optimum, and it is only somewhat larger than that for getting a user equilibrium. In fact, the problem of finding a constrained system optimum becomes computationally more costly with increasing values of the tolerance factor φ. The reason is that the number of allowable paths increases. However, the constrained shortest replacements

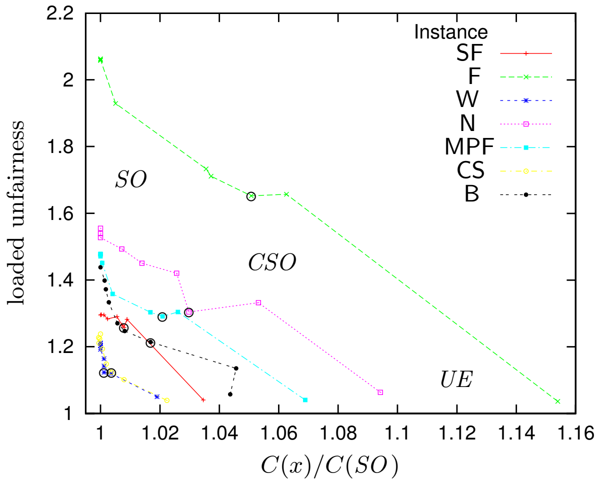

FIGURE 15. Tradeoff between efficiency and unfairness. For all instances, we plot the tradeoff curve between the efficiency (the cost of the solution over the cost of the system optimum) vs. the loaded unfairness. The left area of the graph corresponds to system optima (SO), the lower area corresponds to user equilibria (UE), and the circled data-points (denoted with '$\circ$') correspond to constrained system optima with $\varphi = 1.02$ ($CSO^{1.02}$).

path subproblems become easier because the normal lengths are less binding. In this trade-off situation, the total work and the number of iterations increase, but the work per iteration decreases. Generally, most of the time is spent on computing constrained shortest paths (which implies that improved algorithms for this subproblem would yield greatly improved overall performance).

From our experience, instances with a few thousand nodes, arcs and commodities can be solved on an average PC within minutes. Bigger instances like Berlin take longer but can also be solved without difficulty. Very large instances (e.g., networks with twice as many nodes and arcs as Berlin and with over one million OD pairs) could not be handled mostly due to memory problems resulting from the path-based formulation.

With respect to Partan, we found that the running time is reduced by 30% on average for our target optimality gap of 0.5% when compared to the original version of the Frank-Wolfe method. The reduction is even bigger if just the most difficult instances are considered.

5. Summary and Conclusion

When designing a route guidance system, it is desirable to explicitly aim at reducing the total (and therefore the average) travel time by putting it into the objective function of the underlying optimization problem. Yet, without further constraints, this would include the possibility that some vehicles are assigned to fairly long paths in order to make the shorter paths available to other drivers. Obviously, this phenomenon would render such a system unacceptable for several drivers, jeopardizing the desired effect of improved system performance.

We propose to capture this aspect of human behavior by imposing constraints on paths to eliminate lengthy detours. While it may be ideal to explicitly enforce that travel times of recommended routes between the same origin-destination pair do not deviate significantly from each other, our computational results justify the use of a computationally simpler model, in which the deviation is not measured with respect to the actual flow but with respect to a "normal length". Another plus of the latter tactic is that drivers with different origin-destination pairs can be treated equally.

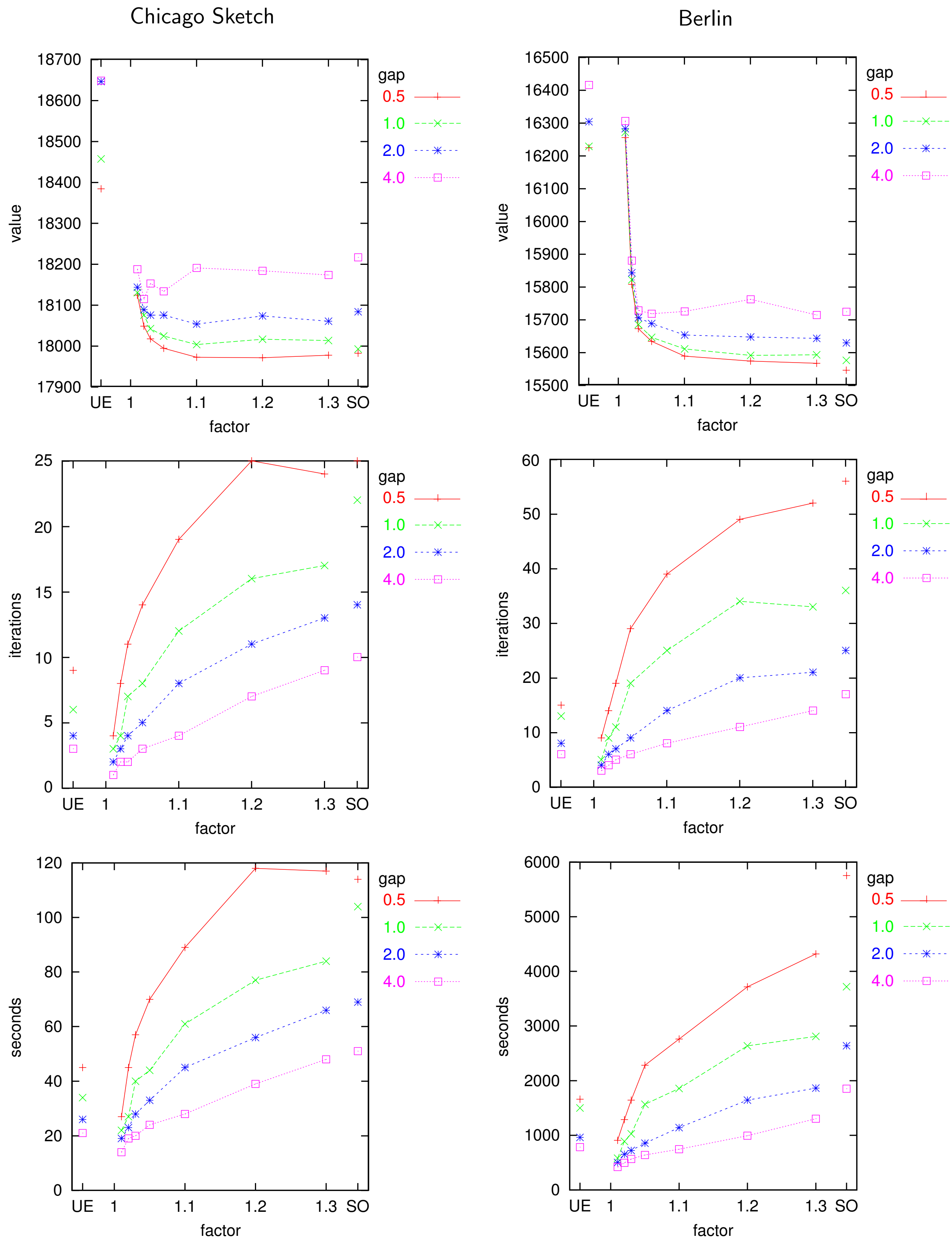

FIGURE 16. Specifics of the algorithm for various optimality gaps and tolerance factors for instances Chicago Sketch and Berlin

Our computational study suggests that the travel time in user equilibrium is an excellent choice for defining the normal length.

In fact, it turns out that this approach offers significant advantages over both the traditionally considered user equilibrium and the system optimum. On the one hand, it guarantees superior fairness for the individual user compared to the system optimum, in which individual travel times between the same origin-destination pair may deviate substantially from each other. On the other hand, the total travel time of a constrained system optimum is still close to that in the (unconstrained) system optimum and thus much better than in user equilibrium. This shows that optimal route guidance with fairness guarantees is in principle feasible.

Apart from the proof of concept, we consider our algorithm practical for problems with several thousand nodes, arcs, and commodities. Future work should incorporate the dynamic aspect of traffic and the behavior of unguided users.

Acknowledgments

The authors are grateful to Dr. Stefan Gnutzmann (DaimlerChrysler AG, Berlin), who excited our work on route guidance, and to Dr. Stefan Gnutzmann and Valeska Naumann (DaimlerChrysler AG, Berlin) for several stimulating discussions and support.

The second author acknowledges support by grant 03-MOM4B1 "Models and Algorithms for Dynamic Route Guidance in Traffic Networks" of the German Ministry for Science and Education (BMBF).

The last two authors gratefully acknowledge support by a General Motors Innovation Grant and the High Performance Computation for Engineered Systems (HPCES) program of the Singapore-MIT Alliance (SMA).

References

Aneja, Y. P., V. Aggarwal, and K. P. K. Nair (1983). Shortest chain subject to side constraints. *Networks 13*, 295–302.

Aneja, Y. P. and K. P. K. Nair (1978). The constrained shortest path problem. *Naval Research Logistics Quarterly 25*, 549–555.

Arezki, Y. and D. Van Vliet (1990). A full analytical implementation of the Partan/Frank-Wolfe algorithm for equilibrium assignment. *Transportation Science 24*, 58–62.

Bar-Gera, H. (2002). Transportation network test problems. `http://www.bgu.ac.il/~bargera/tntp/`.

Beccaria, G. and A. Bolelli (1992). Modelling and assessment of dynamic route guidance: the MARGOT project. In *Proceedings of the IEEE Vehicle Navigation & Information Systems Conference*, Oslo, Norway, pp. 117–126.

Beckmann, M. J., C. B. McGuire, and C. B. Winsten (1956). *Studies in the Economics of Transportation.* Yale University Press, New Haven, CT.

Ben-Akiva, M. E. (1985). Dynamic network equilibrium research. *Transportation Research 19A*, 429–431.

Ben-Akiva, M. E., M. Bierlaire, J. Bottom, H. N. Koutsopoulos, and R. G. Mishalani (1997). Development of a route guidance generation system for real-time application. In M. Papageorgiou and A. Pouliezos (Eds.), *Proceedings of the 8th IFAC Symposium on Transportation Systems*, Chania, Greece, pp. 405–410. Elsevier Science, Oxford.

Ben-Akiva, M. E., A. de Palma, and I. A. Kaysi (1996). The impact of predictive information on guidance efficiency. In L. Bianco and P. Toth (Eds.), *Advanced Methods in Transportation Analysis: An Analytical Approach*, pp. 413–432. Springer, Berlin.

Bernstein, D. and T. E. Smith (1994). Equilibria for networks with lower semicontinuous costs: With an application to congestion pricing. *Transportation Science 28*, 221–235.

Bottom, J. A. (2000). *Consistent Anticipatory Route Guidance*. Ph. D. thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Braess, D. (1968). Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung 12*, 258–268.

Branston, D. (1976). Link capacity functions: A review. *Transportation Research 10*, 223–236.

Charnes, A. and W. W. Cooper (1961). Multicopy traffic network models. In R. Herman (Ed.), *Theory of Traffic Flow*, Proceedings of the Symposium on the Theory of Traffic Flow held at the General Motors Research Laboratories, Warren, MI, 1959. Amsterdam: Elsevier, pp. 85–96.

Chen, K. and S. E. Underwood (1991). Research on anticipatory route guidance. In *Proceedings of the IEEE Vehicle Navigation & Information Systems Conference*, Dearborn, MI, pp. 551–556.

Climaco, J. C. N. and E. Q. V. Martins (1982). A bicriterion shortest path algorithm. *European Journal of Operational Research 11*, 399–404.

Cohen, S. (1991). Flow variables. In M. Papageorgiou (Ed.), *Concise Encyclopedia of Traffic & Transportation Systems*, pp. 139–143. Pergamon Press, Oxford.

Correa, J. R., A. S. Schulz, and N. E. Stier Moses (2003). Computational complexity, fairness, and the price of anarchy of the maximum latency problem. Working Paper No. 4447-03, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.

Dafermos, S. and F. T. Sparrow (1969). The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards 73B*, 91–118.

Daganzo, C. F. (1977a). On the traffic assignment problem with flow dependent costs—I. *Transportation Research 11*, 433–437.

Daganzo, C. F. (1977b). On the traffic assignment problem with flow dependent costs—II. *Transportation Research 11*, 439–441.

DynaMIT (2002). DynaMIT/DynaMIT-P (v 0.9) User's Manual. Intelligent Transportation Systems Program, Massachusetts Institute of Technology.

Dynasmart (2002). Dynasmart-P (v 0.9) User's Guide. Center for Transportation Research, University of Texas at Austin.

Ferris, M. C. and A. Ruszczyński (1997). Robust path choice and vehicle guidance in networks with failures. Technical Report 97-04, Computer Sciences Department, University of Wisconsin.

Florian, M., J. Guélat, and H. Spiess (1987). An efficient implementation of the "Partan" variant of the linear approximation method for the network equilibrium problem. *Networks 17*, 319–339.

Florian, M. and D. W. Hearn (1995). Network equilibrium models and algorithms. In M. O. Ball et al. (Eds.), *Network Routing*, Volume 8 of *Handbooks in Operations Research and Management Science*, Chapter 6, pp. 485–550. Elsevier, New York.

Frank, M. and P. Wolfe (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly 3*, 95–110.

Friesz, T. L. (1985). Transportation network equilibrium, design and aggregation: Key development and research opportunities. *Transportation Research 19A*, 413–427.

Garey, M. R. and D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA.

Gibert, A. (1968). A method for the traffic assignment problem. Technical Report LBS-TNT-95, Transportation Network Theory Unit, London Business School.

Jahn, O., R. H. Möhring, A. S. Schulz, and N. E. Stier Moses (2002). System-optimal routing of traffic flows with user constraints in networks with congestion. Technical Report 754-2002, Institut für Mathematik, Technische Universität Berlin, Germany.

Hagstrom, J. N. and R. A. Abrams (2001). Characterizing Braess's paradox for traffic networks. In *Proceedings of IEEE Conference on Intelligent Transportation Systems*, pp. 837–842. IEEE Computer Society Press, Los Alamitos, CA.

Hearn, D. W. (1980). Bounding flows in traffic assignment models. Technical Report 80-4, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL.

Hearn, D. W. and J. Ribera (1980). Bounded flow equilibrium problems by penalty methods. *IEEE International Conference on Circuits and Computers 1*, 162–166.

Hearn, D. W. and J. Ribera (1981). Convergence of the Frank-Wolfe method for certain bounded variable traffic assignment problems. *Transportation Research 15B*, 437–442.

Henry, J. J., C. Charbonnier, and J. L. Farges (1991). Route guidance. In M. Papageorgiou (Ed.), *Concise Encyclopedia of Traffic & Transportation Systems*, pp. 417–422. Pergamon Press, Oxford.

Holmberg, K. and D. Yuan (2003). A multicommodity network-flow problem with side constraints on paths solved by column generation. *INFORMS Journal on Computing 15*, 42–57.

Kaysi, I. A. (1992). *Framework and Models for the Provision of Real-Time Driver Information.* Ph. D. thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Kaysi, I. A., M. E. Ben-Akiva, and A. de Palma (1995). Design aspects of advanced traveler information systems. In N. H. Gartner and G. Improta (Eds.), *Urban Traffic Networks. Dynamic Flow Modelling and Control*, pp. 59–81. Springer, Berlin.

Lafortune, S., R. Sengupta, D. E. Kaufman, and R. L. Smith (1991). A dynamical system model for traffic assignment in networks. In *Proceedings of the IEEE Vehicle Navigation & Information Systems Conference*, Dearborn, MI, pp. 701–708.

Larsson, T. and M. Patriksson (1994). Equilibrium characterizations of solutions to side constrained asymmetric traffic assignment models. *Le Matematiche 49*, 249–280.

Larsson, T. and M. Patriksson (1995). An augmented Lagrangean dual algorithm for link capacity side constrained traffic assignment problems. *Transportation Research 29B*, 433–455.

Larsson, T. and M. Patriksson (1999). Side constrained traffic equilibrium models—analysis, computation and applications. *Transportation Research 33B*, 233–264.

LeBlanc, L. J., R. V. Helgason, and D. E. Boyce (1985). Improved efficiency of the Frank-Wolfe algorithm for convex network programs. *Transportation Science 19*, 445–462.

Leventhal, T., G. Nemhauser, and L. Trotter (1973). A column generation algorithm for optimal traffic assignment. *Transportation Science 7*, 168–176.

Magnanti, T. L. (1984). Models and algorithms for predicting urban traffic equilibria. In M. Florian (Ed.), *Transportation Planning Models. Proceedings of the Course given at the International Center for Transportation Studies (ICTS), 1982*, Amalfi, Italy, pp. 153–185. North Holland, Amsterdam.

Mahmassani, H. S., T.-Y. Hu, S. Peeta, and A. Ziliaskopoulos (1994). Development and testing of dynamic traffic assignment and simulation procedures for ATIS/ATMS applications. Technical Report DTFH61-90-R-0074-FG, Center for Transportation Research, University of Texas at Austin.

Mahmassani, H. S. and S. Peeta (1995). System optimal dynamic assignment for electronic route guidance in a congested traffic network. In N. H. Gartner and G. Improta (Eds.), *Urban Traffic Networks. Dynamic Flow Modelling and Control*, pp. 3–37. Springer, Berlin.

Merchant, D. K. and G. L. Nemhauser (1978). A model and an algorithm for the dynamic traffic assignment problems. *Transportation Science 12*, 183–199.

Papageorgiou, M. (1990). Dynamic modeling, assignment, and route guidance in traffic networks. *Transportation Research 24B*, 471–496.

Patriksson, M. (1994). *The Traffic Assignment Problem: Models and Methods.* VSP, Utrecht, The Netherlands.

Ribeiro, C. and M. Minoux (1986). Solving hard constrained shortest path problems by Lagrangean relaxation and branch-and-bound algorithms. *Methods of Operations Research 53*, 303–316.

Roughgarden, T. (2001). Stackelberg scheduling strategies. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing (STOC)*, Hersonissos, Greece, pp. 104–113. ACM Press, New York, NY.

Roughgarden, T. (2002). How unfair is optimal routing? In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, San Francisco, CA, pp. 203–204. SIAM, Philadelphia, PA.

Roughgarden, T. and É. Tardos (2002). How bad is selfish routing? *Journal of the ACM 49*, 236–259.

Schulz, A. S. and N. E. Stier Moses (2003). On the performance of user equilibria in traffic networks. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Baltimore, MD, pp. 86–87. SIAM, Philadelphia, PA.

Schulz, A. S. and N. E. Stier Moses (2004). Efficiency and fairness of system-optimal routing with user constraints. In preparation.

Sheffi, Y. (1985). *Urban Transportation Networks.* Prentice-Hall, Englewood, NJ.

Texas Transportation Institute (2002). Urban mobility study. Available at `http://mobility.tamu.edu/ums`.

Wardrop, J. G. (1952). Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers, Part II 1*, 325–378.

Yang, H. and S. Yagar (1994). Traffic assignment and traffic control in general freeway-arterial corridor systems. *Transportation Research 28B*, 463–486.