

# A Strategic Routing Framework and Algorithms for Computing Alternative Paths

**Thomas Bläsius**

Hasso Plattner Institute, University of Potsdam,  
Germany  
thomas.blaesius@hpi.de

**Philipp Fischbeck** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
philipp.fischbeck@hpi.de

**Alina Gries** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
alina.gries@student.hpi.de

**Otto Kissig** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
otto.kissig@student.hpi.de

**Louise Molitor** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
louise.molitor@hpi.de

**Armin Wells** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
armin.wells@student.hpi.de

**Maximilian Böther** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
maximilian.boether@student.hpi.de

**Tobias Friedrich** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
tobias.friedrich@hpi.de

**Falk Hüffner**

TomTom Location Technology Germany GmbH,  
Berlin, Germany  
Falk.Hueffner@tomtom.com

**Pascal Lenzner** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
pascal.lenzner@hpi.de

**Leon Schiller** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
leon.schiller@student.hpi.de

**Simon Wietheger** 

Hasso Plattner Institute, University of Potsdam,  
Germany  
simon.wietheger@student.hpi.de

---

## Abstract

Traditional navigation services find the fastest route for a single driver. Though always using the fastest route seems desirable for every individual, selfish behavior can have undesirable effects such as higher energy consumption and avoidable congestion, even leading to higher overall and individual travel times. In contrast, strategic routing aims at optimizing the traffic for all agents regarding a global optimization goal. We introduce a framework to formalize real-world strategic routing scenarios as algorithmic problems and study one of them, which we call *Single Alternative Path (SAP)*, in detail. There, we are given an original route between a single origin–destination pair. The goal is to suggest an alternative route to all agents that optimizes the overall travel time under the assumption that the agents distribute among both routes according to a psychological model, for which we introduce the concept of Pareto-conformity. We show that the SAP problem is NP-complete, even for such models. Nonetheless, assuming Pareto-conformity, we give multiple algorithms for different variants of SAP, using multi-criteria shortest path algorithms as subroutines. Moreover, we prove that several natural models are in fact Pareto-conform. The implementation and evaluation of our algorithms serve as a proof of concept, showing that SAP can be solved in reasonable time even though the algorithms have exponential running time in the worst case.

**2012 ACM Subject Classification** Theory of computation → Routing and network design problems

**Keywords and phrases** Routing, Strategic Routing, Selfish Routing, Route Planning, Network Flow, Algorithm Design

**Digital Object Identifier** 10.4230/OASICS.ATMOS.2020.10

**Related Version** The full version [4] of this paper is available at <https://arxiv.org/abs/2008.10316>.



© Thomas Bläsius, Maximilian Böther, Philipp Fischbeck, Tobias Friedrich, Alina Gries, Falk Hüffner, Otto Kissig, Pascal Lenzner, Louise Molitor, Leon Schiller, Armin Wells, and Simon Wietheger; licensed under Creative Commons License CC-BY

20th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2020).

Editors: Dennis Huisman and Christos D. Zaroliagis; Article No. 10; pp. 10:1–10:14



OpenAccess Series in Informatics

OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

**Supplementary Material** Source code: <https://github.com/MaxiBoether/strategic-routing>

**Acknowledgements** We want to thank TomTom Location Technology Germany GmbH for supplying us with the data necessary for our experiments.

## 1 Introduction

Commuting is part of our daily lives. Street congestion, traffic jams and pollution became an increasingly large issue in the last few decades. In German cities, these effects caused costs of about 3 billion euros in 2019 [11]. Many traffic jams in cities could have been avoided by better route choice. Partly this is because of non-optimal route choices by individuals due to bounded rationality and route preferences other than “fastest” [30]. However, even with individually optimal route choice, average travel time can be substantially worse compared to a system optimum where all routes are centrally assigned [22]. Thus, there is an opportunity for improving traffic via *strategic routing* where (re)routing recommendations are created by traffic authorities and taken into account by the driver’s routing system. More precisely, we speak of strategic routing when two conditions are met:

- (i) One or more routes are calculated to be proposed to *more than one agent*, and
- (ii) the quality of a set of proposed routes is being defined by a *shared scoring* rather than scoring each agent individually.

Recent research indicates that many drivers would accept individually slower routes if this contributes to an overall reduction in traffic [27, 14]; additionally, incentives such as free parking could be granted to those accepting these routes, and future autonomous vehicles may be more amenable to centralized control. Thus, (re)routing recommendations can have a strong impact since they might be followed by a significant fraction of all drivers.

In the ongoing pilot research project *Socrates 2.0*, strategic routing is employed in the area of Amsterdam [25]. For this, experts predefine alternative routes and traffic conditions that trigger their recommendation. This requires extensive work and monitoring, and does not capture well unusual traffic situations where there might be several incidents at once causing delays. Thus, it is desirable to automate this by formalizing strategic routing and finding algorithms that calculate strategic routes.

**Our Contribution.** Strategic routing as defined above is not a single algorithmic problem but rather a concept capturing numerous scenarios leading to different problems. In Section 2, we provide a framework to guide the process of formalizing real-world strategic routing scenarios. We apply it to one specific scenario, namely *Single Alternative Path (SAP)*. This scenario is inspired by the Amsterdam use case mentioned above where congestion can be prevented by suggesting one alternative route to all agents, e.g., via a variable-message sign. We consider different psychological models to determine how many agents follow the suggestion. Moreover, we consider variants of the SAP problem that require the alternative to be more or less disjoint from the original route. See Section 2.2 for a formal definition.

To tackle SAP algorithmically, we introduce the concept of Pareto-conformity of psychological models and, based on this, give various algorithms in Section 3. As they use multi-criteria shortest path algorithms as subroutine, they have an exponential worst-case running time but turn out to be sufficiently efficient in practice; see our evaluation in Section 5. Moreover, in this generality, we cannot hope for better worst-case bounds as SAP is NP-hard, even for Pareto-conform psychological models; see the full version [4] for a proof. In Section 4, we prove the Pareto-conformity of three natural psychological models. Our proofs actually hold for the more general and abstract *Quotient Model* that captures various additional

models. We evaluate our algorithms in Section 5. It serves as a proof of concept that our algorithms have reasonable practical run times and yield promising travel time improvements for instances in the traffic network of Berlin. Missing proofs and some additional evaluation can be found in the full version of this paper [4].

**Related Work.** There has been no unique understanding of strategic routing in research until this point. Van Essen [27] uses a choice-theoretical approach and concludes that individual route choice and travel information that stimulates non-selfish user behavior have a large impact on the network efficiency. Kröller et al. [14] investigate due to what kind of incentives agents would deviate from the shortest-path route. Their results show that certain incentives can increase the drivers' willingness of taking detours. Moreover, they show that there is a high interest in services providing alternative routes, and strategic routing is considered to have the potential of solving traffic issues such as congestion and pollution.

For standard algorithmic techniques in efficient route planning, we refer to the survey of Bast et al. [2]. Köhler et al. [15] deal with finding static and also time-dependent traffic flows minimizing the overall travel time. Also, as stated by Strasser [26], routing with predicted congestion is well-studied, e.g., by Delling and Wagner [8], Demiryurek et al. [9], Delling [6] and Nannicini et al. [19]. Route planning with alternative routes was investigated by Abraham et al. [1] and Paraskevopoulos and Zaroliagis [21]. They propose algorithms that find alternative routes by evaluating properties with regard to an original route.

Lastly, we emphasize that strategic routing is very different from selfish routing as proposed by Roughgarden and Tardos [23]. In contrast to our global optimization approach, in selfish routing individual strategic agents select their routes to optimize their own travel times, given the route choices of other agents. While often static flows are considered in selfish routing, Sering and Skutella [24] analyzed selfish driver behavior for a dynamic flow-over-time model. Another related selfish routing variant is Stackelberg routing [13, 5, 12, 3], where an altruistic central authority controls a fraction of the traffic and first routes it in a way to improve the travel times for all other selfish agents which choose their route afterwards.

## 2 A Framework for Strategic Routing

In the following, we provide a framework that supports the formalization of a given strategic routing scenario. We employ a two-step process. The first step categorizes the scenario by distilling its crucial aspects. The second step transforms it into an algorithmic problem.

### 2.1 Categorization

Categorizing a scenario at hand boils down to answering the following questions.

**What is the goal we aim to achieve?** There are different objectives one can pursue when routing strategically. A city might be interested in reducing particulate matter emission in a certain region. As a routing service provider, the goal could be to minimize the travel time for as many customers as possible. A system of centrally controlled autonomous vehicles might want to achieve a minimum overall travel time.

**How can we influence the agents?** How we recommend routes determines which agents we can influence and whether we can make different suggestions to different agents. A city administration can put up signs to influence all vehicles in a certain area, making the same suggestion to each agent. Navigation providers, on the other hand, can influence only a limited number of vehicles but could make different suggestions to different agents.

**How much control do we have over the agents?** The willingness of users to follow an alternative route depends on the use case. While a navigation provider cannot force its users to use a specific route, and the acceptance of detouring depends heavily on the additional length, there are scenarios where the suggested route will always be accepted or agents end up in an equilibrium or in a system-optimal distribution on the suggested routes.

**What is the starting situation?** We either assume that there is already existing traffic, or that we design traffic from scratch. Although the former is certainly more common, the latter applies to, e.g., the scenario of centrally controlled autonomous vehicles.

**How do the uninfluenced agents react?** If only a fraction of the traffic is routed strategically, the remaining traffic might react with respect to the change. For instance, it is a valid assumption that after some time, all traffic settles in an equilibrium. Another simple assumption is that the other traffic does not change at all.

## 2.2 Problem Formalization

In this section, we first propose a generic formalization whose degrees of freedom can then be filled to reflect a specific scenario. We focus on the *Single Alternative Path* (SAP) scenario, which we study algorithmically in Section 3. We use it as an example how fixing answers to the questions raised in Section 2.1 naturally fills the degrees of freedom.

**Generic Strategic Routing Considerations.** Let  $G = (V, E)$  be a directed graph. For every pair of nodes  $(s, t) \in V^2$ , the *demand*  $d : V^2 \rightarrow \mathbb{Q}$  denotes the amount of traffic flow that has to be routed from  $s$  to  $t$ . For every edge  $e \in E$ , let  $\tau_e : \mathbb{Q}_{\geq 0} \rightarrow \mathbb{Q}_{> 0}$  be a monotonically increasing cost function. For  $x \in \mathbb{Q}_{\geq 0}$ ,  $\tau_e(x)$  describes the costs for a single agent traversing an edge  $e \in E$  while there is a traffic flow of  $x$  vehicles per unit of time on  $e$ .

The solution to a strategic routing problem is a traffic distribution to paths in the network that routes agents according to  $d$ . Let  $\mathcal{P}$  be the set of all simple paths in  $G$ . By  $f : \mathcal{P} \rightarrow \mathbb{Q}_{\geq 0}$  we denote the flow, where  $f(P)$  states the amount of traffic flow using path  $P$ . Extending the notion, let  $f(e) = \sum_{e \in P} f(P)$  be the total traffic flow on an edge  $e$ . For all  $x \in \mathbb{Q}_{\geq 0}$ , let  $\tau_P(x) = \sum_{e \in P} \tau_e(x)$  be the costs per agent on  $P$  assuming that the total traffic on  $P$  is  $x$ .

Paths are denoted as tuples of vertices, i.e.,  $(v_1, \dots, v_k)$  with  $v_i \in V$  is a path if for  $1 < i \leq k$ ,  $(v_{i-1}, v_i) \in E$ . In addition, we consider paths as edge sets and use set operators, which also translates to the notion of cost functions, e.g., for paths  $P$  and  $Q$  let  $\tau_{P \cap Q}(x) = \sum_{e \in P \cap Q} \tau_e(x)$ .

**Means of Influence.** In the SAP problem, we assume that we can influence all agents on a given *original*  $st$ -path  $Q$  and suggest a single *alternative*  $st$ -path  $P$ .

**Starting Situation and Uninfluenced Traffic.** We assume that there is existing traffic that satisfies all demands and that uninfluenced agents stick with their previous routes. Note that this allows us to integrate the uninfluenced traffic into the cost functions. Thus, we can formalize it as if there was no initial traffic and that all demands are equal to 0 except for the traffic on the original route  $Q$  which satisfies the demand  $d(s, t) > 0$ . For brevity, we denote  $d = d(s, t)$ .

**Level of Control.** We assume that agents make their own decisions. Given an original route  $Q$  and alternative  $P$  a *psychological model* determines the amount of flow  $x_P$  on  $P$ . The flow on  $Q$  is then  $d - x_P$ . We consider the following three psychological models; see Section 4 for formal definitions. The *System Optimum* assumes agents distribute optimally with respect to the optimization criterion defined below. In the *User Equilibrium* [28] agents act selfishly leading to an equilibrium where no agent can improve by unilaterally changing their route [23]. In the *Linear Model* we assume that the willingness to choose  $P$  is linearly dependent on the ratio of the costs on  $Q$  and  $P$ .

**Optimization Criterion.** The optimization criterion formalizes the goal to be achieved, which is the *overall travel time* for SAP. Hence, we interpret the cost functions  $\tau_e$  as latency functions, i.e., the time a single agent needs to traverse the edge  $e$ . In the SAP problem, we only consider one alternative  $P$  to an original route  $Q$ . Assume that we have a flow of  $x \in [0, d]$  on  $P$ . Then, the edges in  $P \setminus Q$  have flow  $x$ , the edges of  $Q \setminus P$  have flow  $d - x$  and the edges of  $P \cap Q$  have flow  $d$ . Thus, the overall cost is

$$\mathcal{C}_P(x) = x \cdot \tau_{P \setminus Q}(x) + (d - x) \cdot \tau_{Q \setminus P}(d - x) + d \cdot \tau_{P \cap Q}(d). \quad (1)$$

For the value  $x_P$  determined by the psychological model, the actual cost of an alternative route  $P$  is  $\mathcal{C}_P(x_P)$ , which we abbreviate with  $\mathcal{C}_P$ . Let  $\mathcal{P}$  be a set of alternative paths. Computing the path  $P$  in  $\mathcal{P}$  with optimal  $\mathcal{C}_P$  is called *scoring*  $\mathcal{P}$ .

**Summary and Problem Variants.** To sum up the SAP problem, given a route  $Q$  from  $s$  to  $t$ , a demand  $d$  of agents per unit of time and a psychological model, the SAP problem asks for the optimal alternative route  $P$  such that the overall travel time  $\mathcal{C}_P$  is minimized.

In general  $P$  can have arbitrarily many overlaps with  $Q$ . Additionally, we consider two variants of SAP, where we require the routes to be more or less disjoint. *Disjoint Single Alternative Path (D-SAP)* requires  $P$  and  $Q$  to be completely disjoint. Moreover, *1-Disjoint Single Alternative Path (1D-SAP)* requires  $P \setminus Q$  to be a single connected path, i.e.,  $P$  diverts from  $Q$  at most once but can share the edges at the start and the end with  $Q$ .

### 3 Algorithms for Single Alternative Path

Consider two alternative paths  $P_1$  and  $P_2$  with cost functions  $\tau_{P_1}$  and  $\tau_{P_2}$ , respectively. Assume that for any amount of traffic  $x \in [0, d]$ , the cost of  $P_1$  is not larger than of  $P_2$ , i.e.,  $\tau_{P_1}(x) \leq \tau_{P_2}(x)$ . It seems intuitive that it is never worse to choose  $P_1$  over  $P_2$ . However, this is not quite right for two reasons. First, it does not hold for arbitrary psychological models, which determine the amount of agents (potentially in a somewhat degenerate fashion) who choose  $P_1$  and  $P_2$ , respectively, instead of the original route  $Q$ . Secondly, if the alternative route  $P_1$  shares many edges with the original route  $Q$  it has only little potential to distribute traffic, whereas the seemingly worse alternative  $P_2$  could do better in this regard.

We resolve the first issue by defining a property that we call (weak) Pareto-conformity. Moreover, in Section 4, we show for various psychological models that they are in fact Pareto-conform. To resolve the second issue with shared edges, we introduce a notion of dominance between paths that takes the overlap with  $Q$  into account.

Let  $\tau_1$  and  $\tau_2$  be two cost functions defined on the interval  $[0, d]$  and let  $\tau'_i$  denote the derivative of  $\tau_i$ .<sup>1</sup> For two alternative paths  $P_1$  and  $P_2$ , we say that  $P_1$  *dominates*  $P_2$ , denoted by  $P_1 \preceq P_2$ , if  $\tau_{P_1} \leq \tau_{P_2}$  and  $\tau'_{P_1 \cap Q} \leq \tau'_{P_2 \cap Q}$ . Note that, if  $P_1 \cap Q = P_2 \cap Q$ , then this simplifies to  $\tau_{P_1} \leq \tau_{P_2}$ . With this, we can define the above-mentioned Pareto-conformity.

► **Definition 1.** *A psychological model is Pareto-conform if  $P_1 \preceq P_2$  implies  $\mathcal{C}_{P_1} \leq \mathcal{C}_{P_2}$ . It is weakly Pareto-conform if this holds for paths that have equal intersection with  $Q$ .*

To simplify notation, we assume without loss of generality that there are no two different paths  $P_1$  and  $P_2$  with  $P_1 \preceq P_2$  and  $P_2 \preceq P_1$ . This can, e.g., be achieved by slight perturbation of the cost functions, or by resolving every tie arbitrarily.

In the following we give different algorithms for the SAP, 1D-SAP and D-SAP problems. The algorithms involve solving one or more multi-criteria shortest path problems as subroutine. Algorithms for this problem range from the fundamental examination of the bicriteria case [10] to the usage of speed-up techniques [7, 16] in the multi-criteria case. One such algorithm is the multi-criteria Dijkstra, which has exponential run time in the worst case [17] but is known to be efficient in many practical applications [18].

The algorithms we present first (Sections 3.1–3.3) require solving only a single multi-criteria shortest path problem, with the D-SAP setting requiring fewer criteria than SAP and 1D-SAP. In Sections 3.4 and 3.5, we propose approaches that require multiple such searches. Though the former seems preferable, the latter has some advantages. It requires fewer criteria in the multi-criteria sub-problems, it requires only weak Pareto-conformity for the 1-disjoint setting, and it allows for easy parallelization. Our experiments in Section 5 indicate that the variants requiring fewer criteria are often faster for long routes.

### 3.1 Reduction to Multi-Criteria Shortest Path

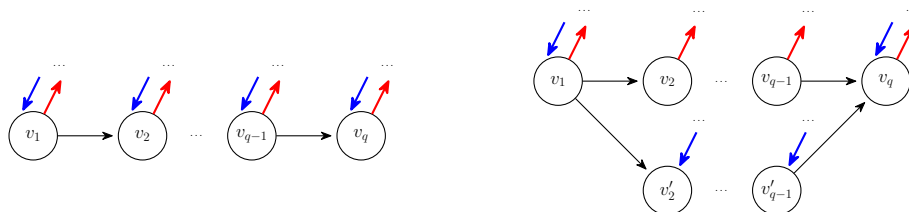
We are now ready to solve SAP. Definition 1 directly yields the following lemma.

► **Lemma 2.** *For any instance of SAP with a Pareto-conform psychological model, there exists an optimal solution that is not dominated by any other alternative.*

Thus, to solve SAP, it suffices to find all alternative paths that are not dominated by other paths, and then choose the best among these potential solutions. We reduce the problem of computing the set of potential solutions to a multi-criteria shortest path problem. In such a problem, each path corresponds to a point  $p \in \mathbb{Q}^k$ , where the entry at the  $i$ -th position of  $p$  is the cost of the path with respect to the  $i$ -th criterion. One then searches for all solutions that are not Pareto dominated by other solutions. For two points  $p_1, p_2 \in \mathbb{Q}^k$ ,  $p_1$  *Pareto dominates*  $p_2$  if  $p_1 \leq p_2$  component-wise. Finding all solutions that are not Pareto dominated is the previously mentioned multi-criteria shortest path problem. How the transformation to a multi-criteria problem exactly works depends on the cost functions.

Assume for now that  $\tau(x) = ax^2 + b$  for positive  $a$  and  $b$ . We call the family of cost functions of this form *canonical cost functions*. It is closed under addition. Thus, the cost function of each path is also a canonical cost function. Note that two different canonical cost functions intersect in at most one point on  $[0, d]$ . Thus, we have  $\tau_1 \leq \tau_2$  if and only if  $\tau_1(0) \leq \tau_2(0)$  and  $\tau_1(d) \leq \tau_2(d)$ . It follows that requiring  $\tau_1 \leq \tau_2$  is equivalent to saying that  $(\tau_1(0), \tau_1(d))$  Pareto dominates  $(\tau_2(0), \tau_2(d))$ . Additionally, the function  $\tau_1 + \tau_2$  can be represented by  $(\tau_1(0) + \tau_2(0), \tau_1(d) + \tau_2(d))$ . Similarly, with  $\tau'_1(x) = 2a_1x$  and  $\tau'_2(x) = 2a_2x$  we have  $\tau'_1 \leq \tau'_2$  if and only if  $a_1 \leq a_2$ . Addition works again as expected.

<sup>1</sup> In the remainder, we implicitly assume all cost functions to be only defined on  $[0, d]$ , e.g.,  $\tau_1 \leq \tau_2$  means  $\tau_1(x) \leq \tau_2(x)$  for all  $x \in [0, d]$ . Also, we implicitly assume functions to be differentiable.



■ **Figure 1** Graph transformation for the 1D-SAP algorithm. Blue edges represent an arbitrary number of incoming edges, red edges an arbitrary number of outgoing edges.

To generalize this concept, consider a class of functions  $\mathcal{T}$  that is closed under addition. We say that  $\mathcal{T}$  has *Pareto dimension*  $k$  if the following holds. There exists a function  $p: \mathcal{T} \rightarrow \mathbb{Q}^k$  such that  $\tau_1$  dominates  $\tau_2$  if and only if  $p(\tau_1)$  Pareto dominates  $p(\tau_2)$ , and such that  $p(\tau_1 + \tau_2) = p(\tau_1) + p(\tau_2)$ . We call  $p$  the *Pareto representation* of  $\mathcal{T}$ . The above canonical cost functions have Pareto dimension 2 and their derivatives have Pareto dimension 1.

With this,  $P_1 \preceq P_2$  reduces to having  $p(\tau_{P_1}) \leq p(\tau_{P_2})$  and  $p'(\tau'_{P_1 \cap Q}) \leq p'(\tau'_{P_2 \cap Q})$ , where  $p'$  is a Pareto representation of the class of all derivatives of functions in  $\mathcal{T}$ . This is equivalent to the concatenation of  $p(\tau_{P_1})$  and  $p'(\tau'_{P_1 \cap Q})$  Pareto dominating the concatenation of  $p(\tau_{P_2})$  and  $p'(\tau'_{P_2 \cap Q})$ . Thus, dominance of paths reduces to Pareto dominance.

► **Theorem 3.** *SAP with Pareto-conform psychological model and cost functions with Pareto dimension  $k$  whose derivatives have Pareto dimension  $\ell$  reduces to solving a multi-criteria shortest path problem with  $k + \ell$  criteria and scoring the result.*

### 3.2 Enforcing 1-Disjoint Routes

1D-SAP can be solved by modifying the graph and then applying the same approach as above. Let  $Q = (v_1, \dots, v_q)$  with  $s = v_1$ ,  $t = v_q$ . We consider the graph  $G'$ , which is a copy of  $G$  where for each  $v_i \in \{v_2, \dots, v_{q-1}\}$  a node  $v'_i$  is added. Moreover,  $v_i$  in  $G'$  has all outgoing edges of  $v_i$  in  $G$ , but only the incoming edge from  $v_{i-1}$ . Similarly,  $v'_i$  in  $G'$  has all incoming edges of  $v_i$  in  $G$ , but only the outgoing edge to  $v'_{i+1}$ ; see Figure 1. With this, computing all non-dominated 1-disjoint paths in  $G$  reduces to computing all non-dominated paths in  $G'$ .

► **Theorem 4.** *Theorem 3 also holds for 1D-SAP.*

### 3.3 Fewer Criteria for Disjoint SAP

We now consider the D-SAP variant whose major advantage is that we can solve it with fewer criteria in the multi-criteria shortest path part of the algorithm. Analogously to Lemma 2, the following lemma follows from Definition 1.

► **Lemma 5.** *For any instance of D-SAP with a weakly Pareto-conform psychological model, there exists an optimal solution that is not dominated by any other alternative disjoint from  $Q$ .*

To guarantee that we only find paths disjoint from  $Q$ , we remove  $Q$  from the graph. For two paths  $P_1$  and  $P_2$  in the resulting graph, the dominance  $P_1 \preceq P_2$  simplifies to  $\tau_{P_1} \leq \tau_{P_2}$ . This observation together with Lemma 5 gives us the following theorem. Note that we only need weak Pareto-conformity here, as all paths have no intersection with  $Q$ .

► **Theorem 6.** *D-SAP with a weakly Pareto-conform psychological model and cost functions with Pareto dimension  $k$  reduces to solving a multi-criteria shortest path problem with  $k$  criteria and scoring the result.*

### 3.4 Fewer Criteria for 1-Disjoint SAP

We start by deleting the edges of  $Q = (v_1, \dots, v_q)$  from the graph. In the resulting graph, for every pair  $1 \leq i < j \leq q$ , we calculate the set  $\mathcal{P}_{i,j}$  of all routes between  $v_i$  and  $v_j$  that are minimal with respect to dominance. We define the corresponding *augmented path* for a path  $P \in \mathcal{P}_{i,j}$  as  $\tilde{P} = (v_1, \dots, v_{i-1}) \cup P \cup (v_{j+1}, \dots, v_q)$ , which is a 1-disjoint path from  $s$  to  $t$ . We denote the set of paths from  $s$  to  $t$  obtained by augmenting all paths in  $\mathcal{P}_{i,j}$  by  $\tilde{\mathcal{P}}_{i,j}$ .

► **Lemma 7.** *For an instance of 1D-SAP with weakly Pareto-conform psychological model, there exists an optimal solution among the paths in the sets  $\tilde{\mathcal{P}}_{i,j}$ .*

From Section 3.3, we know that we can compute all non-dominated paths from  $v_i$  to  $v_j$  by using a multi-criteria shortest path algorithm. Thus, 1D-SAP reduces to solving  $\binom{q}{2} \in \Theta(q^2)$  multi-criteria shortest path problems, one for each pair of vertices  $v_i, v_j \in Q$ . We note that many shortest path algorithms actually solve a more general problem by computing paths from a single start to all other vertices. Thus, instead of  $\Theta(q^2)$  shortest path problems, we can solve  $q$  multi-target shortest path problems, using each vertex in  $Q$  as start once.

► **Theorem 8.** *1D-SAP with weakly Pareto-conform psychological model and cost functions with Pareto dimension  $k$  reduces to solving  $q$  multi-criteria multi-target shortest path problems with  $k$  criteria and scoring the resulting augmented paths.*

### 3.5 Fewer Criteria for SAP

We now provide an algorithm for the SAP problem that requires fewer criteria. We use a dynamic program that combines non-dominated subpaths to obtain the optimal solution. To formalize this, we need the following additional notation. For  $v_i, v_j \in Q$  with  $i < j$ , a path from  $v_i$  to  $v_j$  is called *Q-path* or more specifically  *$Q_{i,j}$ -path*. A set  $A$  of  $Q_{i,j}$ -paths is *reduced* if no path in  $A$  is dominated by another path in  $A$ . Let  $A$  and  $B$  be two reduced sets of  $Q_{i,j}$ -paths. Their *reduced union* is obtained by eliminating from  $A \cup B$  all paths that are dominated by another path in  $A \cup B$ . Moreover, let  $A$  and  $B$  be two reduced sets of  $Q_{i,j}$  and  $Q_{j,k}$ -paths, respectively. Then their *reduced join* is obtained by concatenating every path in  $A$  with every path in  $B$  and eliminating all dominated paths.

We start by applying the algorithm from Section 3.4, computing the sets  $\mathcal{P}_{i,j}$  for all  $1 \leq i < j \leq q$ , which are the reduced sets of all  $Q_{i,j}$ -paths that are disjoint from  $Q$ . Then, we compute sets  $\mathcal{P}_j$  of  $Q_{1,j}$ -paths and one can show that  $\mathcal{P}_j$  is in fact the reduced set of all  $Q_{1,j}$ -paths. We initialize  $\mathcal{P}_1 = \{(v_1)\}$ . Now, assume we have computed  $\mathcal{P}_i$  for all  $i < j$ . We obtain  $\mathcal{P}_j$  as the reduced union of the following sets of  $Q_{1,j}$ -paths: the reduced join of  $\mathcal{P}_i$  and  $\mathcal{P}_{i,j}$  for every  $i < j$ , and the reduced join of  $\mathcal{P}_{j-1}$  and  $\{(v_{j-1}, v_j)\}$ .

► **Lemma 9.** *For an instance of SAP with a Pareto-conform psychological model, there exists an optimal solution among the paths in  $\mathcal{P}_q$ .*

After computing the sets  $\mathcal{P}_{i,j}$  as in Section 3.4, it remains to compute  $\Theta(q^2)$  reduced joins and  $\Theta(q^2)$  reduced unions between reduced sets of  $Q$ -paths. Then, it only remains to score the result  $\mathcal{P}_q$ . The shortest path computations from Section 3.4 use  $k$  criteria where  $k$  is the Pareto-dimension of the cost functions. The reduced joins and unions are with respect to  $k + \ell$  criteria, where  $\ell$  is the Pareto dimension of the derivatives of the cost functions.



► **Theorem 10.** *SAP with Pareto-conform psychological model and cost functions with Pareto dimension  $k$  whose derivatives have Pareto dimension  $\ell$  reduces to solving  $q$  multi-target shortest path problems with  $k$  parameters, executing  $\Theta(q^2)$  reduced join and union operations with respect to  $k + \ell$  criteria between reduced sets of  $Q$ -paths, and scoring the result  $\mathcal{P}_q$ .*

#### 4 Psychological Models and Pareto-Conformity

In this section, we formally define the models mentioned in Section 2.2 and show their Pareto-conformity. After considering the System Optimum Model, we define the Quotient Model, which is a generalization of the User Equilibrium Model and the Linear Model. We give conditions under which a Quotient Model is Pareto-conform and thereby prove that the User Equilibrium Model and the Linear Model are both Pareto-conform.

The *System Optimum Model* assumes that agents distribute optimally, i.e.,  $x_P \in [0, d]$  minimizes  $\mathcal{C}_P(x_P)$ . We get that  $P_1 \preceq P_2$  implies  $\mathcal{C}_{P_1}(x) \leq \mathcal{C}_{P_2}(x)$  for each  $x \in [0, d]$ .

► **Theorem 11.** *The System Optimum Model is Pareto-conform.*

For the *Quotient Model*, let  $c(x)$  be non-decreasing, non-negative on  $[0, d]$  with  $c(d) > 0$ . If

$$\frac{\tau_{Q \setminus P}(d - x) + \tau_{P \cap Q}(d)}{\tau_{P \setminus Q}(x) + \tau_{P \cap Q}(d)} = c(x) \quad (2)$$

has a solution in  $[0, d]$ , it is unique for the following reason. The numerator and denominator are the cost of  $Q$  and  $P$ , which are decreasing and increasing in  $x$ , respectively. Thus, the quotient is decreasing, while  $c(x)$  is non-decreasing, which makes the solution unique. The Quotient Model sets  $x_P$  to this unique solution if it exists. If no solution exists, then the left-hand side is either smaller or larger than  $c(x)$  for every  $x \in [0, d]$ , in which case we set  $x_P = 0$  or  $x_P = d$ , respectively. This is the natural choice, as  $x_P = 0$  and  $x_P = d$  maximizes and minimizes the left-hand side, respectively. We note that  $c$  specifies how conservative the agents are. If  $c(x) = 1$ , the agents distribute on  $Q$  and  $P$  such that both paths have the same cost. If  $c$  is smaller, then agents take the alternative route, if it is not too much longer.

Recall from Equation (1) that the cost function  $\mathcal{C}_P(x)$  is a combination of the three functions  $\tau_{P \setminus Q}$ ,  $\tau_{Q \setminus P}$ , and  $\tau_{P \cap Q}$ . If Equation (2) has a solution  $x_P$ , we know how  $\tau_{P \setminus Q}(x)$  and  $\tau_{Q \setminus P}(x)$  relate to each other at  $x = x_P$ . In other words, solving Equation (2) for  $\tau_{P \setminus Q}(x_P)$  or  $\tau_{Q \setminus P}(x_P)$  and replacing their occurrence in  $\mathcal{C}_P = \mathcal{C}_P(x_P)$  with the result lets us eliminate  $\tau_{Q \setminus P}$  or  $\tau_{P \setminus Q}$ , respectively, from  $\mathcal{C}_P$ . We do this in the following two lemmas, which additionally take the special cases  $x_P = 0$  and  $x_P = d$  into account.

► **Lemma 12.** *Let  $g_P(x) = (d - x) \cdot c(x) + x$ . Then  $\mathcal{C}_P \leq g_P(x_P) \cdot (\tau_{P \setminus Q}(x_P) + \tau_{P \cap Q}(d))$ . If  $x_P > 0$ , then equality holds.*

We note that  $c(x_P) > 0$  holds for the following reason. For  $x_P = d$  this is true by definition. For  $x_P < d$ , the left-hand side of Equation (2) is equal to its right-hand side or less (in which case  $x_P = 0$ ). As the right-hand side is  $c(x_P)$  and the left-hand side is positive, we get  $c(x_P) > 0$ . Thus it is fine to divide by  $c(x_P)$  in the following lemma.

► **Lemma 13.** *Let  $g_Q(x) = d + x/c(x) - x$ . Then  $\mathcal{C}_P \leq g_Q(x_P) \cdot (\tau_{Q \setminus P}(d - x_P) + \tau_{P \cap Q}(d))$ . If  $x_P < d$ , then equality holds.*

The following lemma provides the core inequalities we need when comparing the cost of two alternative paths. Note how the inequalities in parts 1 and 2 of the lemma resemble the representation of the cost  $\mathcal{C}_P$  in Lemma 12 and Lemma 13, respectively.

## 10:10 A Strategic Routing Framework and Algorithms for Computing Alternative Paths

► **Lemma 14.** Let  $P_1$  and  $P_2$  be alternative paths with  $P_1 \preceq P_2$  and let  $x_1, x_2 \in [0, d]$ . Then

1.  $\tau_{P_1 \setminus Q}(x_1) + \tau_{P_1 \cap Q}(d) \leq \tau_{P_2 \setminus Q}(x_2) + \tau_{P_2 \cap Q}(d)$  if  $x_1 \leq x_2$ , and
2.  $\tau_{Q \setminus P_1}(d - x_1) + \tau_{P_1 \cap Q}(d) \leq \tau_{Q \setminus P_2}(d - x_2) + \tau_{P_2 \cap Q}(d)$  if  $x_1 \geq x_2$ .

**Proof sketch.** Recall that  $P_1 \preceq P_2$  means that for all  $x \in [0, d]$ ,  $\tau_{P_1}(x) \leq \tau_{P_2}(x)$  and  $\tau'_{P_1 \cap Q}(x) \leq \tau'_{P_2 \cap Q}(x)$ , where  $\tau'$  denotes the derivative of  $\tau$ . For the first case  $x_1 \leq x_2$ , we get

$$\begin{aligned} \tau_{P_1 \setminus Q}(x_1) + \tau_{P_1 \cap Q}(d) &= \tau_{P_1 \setminus Q}(x_1) + \tau_{P_1 \cap Q}(x_1) - \tau_{P_1 \cap Q}(x_1) + \tau_{P_1 \cap Q}(d) \\ &= \tau_{P_1}(x_1) + \tau_{P_1 \cap Q}(d) - \tau_{P_1 \cap Q}(x_1), \end{aligned}$$

using that  $\tau_{P_1}(x) \leq \tau_{P_2}(x)$  and  $\tau'_{P_1 \cap Q}(x) \leq \tau'_{P_2 \cap Q}(x)$

$$\begin{aligned} &\leq \tau_{P_2}(x_1) + \tau_{P_2 \cap Q}(d) - \tau_{P_2 \cap Q}(x_1) \\ &= \tau_{P_2 \setminus Q}(x_1) + \tau_{P_2 \cap Q}(x_1) + \tau_{P_2 \cap Q}(d) - \tau_{P_2 \cap Q}(x_1) \\ &= \tau_{P_2 \setminus Q}(x_1) + \tau_{P_2 \cap Q}(d). \end{aligned}$$

As  $\tau_{P_2 \setminus Q}$  is an increasing function and  $x_2 \geq x_1$ , we obtain  $\tau_{P_2 \setminus Q}(x_1) \leq \tau_{P_2 \setminus Q}(x_2)$ , which concludes this case. The case  $x_1 \geq x_2$  works very similar. ◀

Applying the previous three lemmas and dealing with the additional functions  $g_P(x)$  and  $g_Q(x)$  in Lemma 12 and Lemma 13, respectively, yields the following.

► **Theorem 15.** The Quotient Model is Pareto-conform if  $c(d) \leq 1$  and, for all  $x \in [0, d]$ ,  $c(x) \cdot (1 - c(x)) - x \cdot c'(x) \leq 0$ .

The User Equilibrium Model is obtained by setting  $c(x) = 1$  in Equation (2). The Linear Model is defined by setting  $c(x)$  to an increasing linear function, i.e.,  $c(x) = c \cdot x/d$  for  $c > 0$ .

► **Corollary 16.** The User Equilibrium and Linear Model with  $c \leq 1$  are Pareto-conform.

The Linear Model with  $c \leq 1$  is less conservative than the User Equilibrium Model, i.e., more agents use the alternative path, in particular if only few agents use it based on its cost.

## 5 Empirical Evaluation

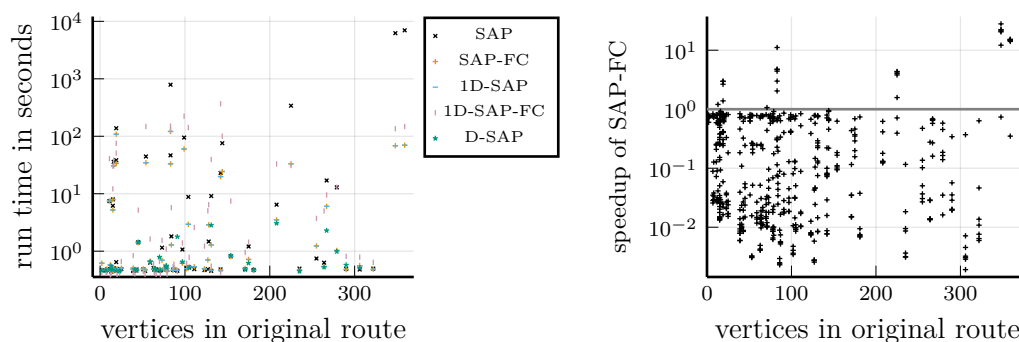
In this section we fix implementation details and evaluate the proposed algorithms. Our evaluation focuses on the following aspects.

**Performance.** Are the algorithms sufficiently efficient for practical problem instances? How do the different algorithms compare in terms of run time?

**Strategic Improvement.** How much does strategic routing improve the overall travel time? How does the requirement of disjoint or 1-disjoint alternatives impact this improvement?



■ **Figure 2** Visualization of example routes: original route (blue), optimal alternative routes with respect to the User Equilibrium for SAP (green), for 1D-SAP (black), and D-SAP (orange).



■ **Figure 3** Left: Absolute run times. Each point represents one OD-pair for demand  $d = 2000$ . For D-SAP, we excluded the OD-pairs that did not have a solution that was disjoint from the original route. Right: Speedup of SAP-FC over SAP, with one point for each OD-pair and each value of  $d$ .

Additional evaluation regarding the psychological models can be found in the full version [4]. For now, we fix the psychological model to be the User Equilibrium.

We model cost functions  $\tau_e$  as proposed by the U.S. Bureau of Public Roads [20], i.e., for parameters  $\alpha, \beta \geq 0$ , we have  $\tau_e(x) = \ell_e/s_e \cdot (1 + \alpha(x/c_e)^\beta)$  where  $s_e$ ,  $c_e$ , and  $\ell_e$  denote free flow speed, capacity and length of  $e$ . We set  $\alpha = 0.15$  and  $\beta = 2$ . Thus, for appropriate  $a$  and  $b$ , we get canonical cost functions of the form  $\tau_e(x) = ax^2 + b$  as defined in Section 3.

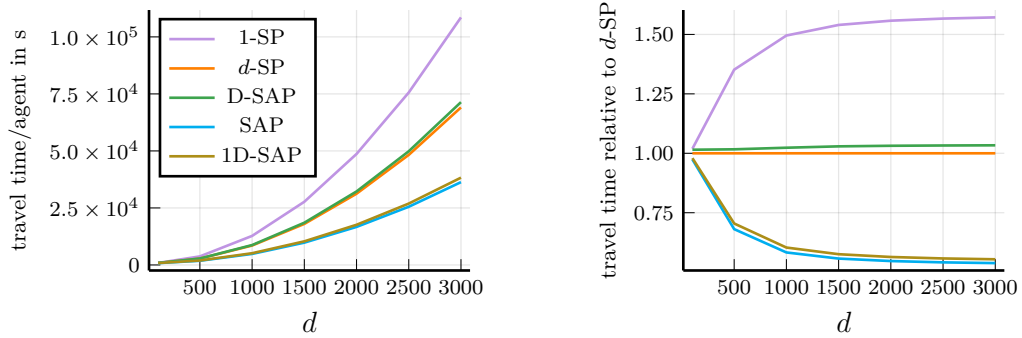
For solving the multi-criteria shortest path problem, we implement a multi-criteria A\* variant [16]. As lower bound, we use the distances in the parameters  $a$  and  $b$  to  $t$ . These distances are calculated using two runs of Dijkstra’s algorithm. We note that A\* solves a multi-target shortest path problem, which we need for two algorithms; see Section 3.4. For calculating the Pareto-frontiers we use the simple cull algorithm [29].

We use the following naming scheme. We abbreviate the algorithms from Sections 3.1 and 3.2 with SAP and 1D-SAP, respectively. We denote the *fewer criteria* (FC) approaches with D-SAP (Section 3.3) 1D-SAP-FC (Section 3.4) and SAP-FC (Section 3.5). To evaluate the strategic improvement, we compare them to the solution of proposing only the shortest path to all agents, assuming either one single agent (1-SP) or  $d$  agents ( $d$ -SP) on every edge.

We test our implementations on the street network of Berlin, Germany with 75 origin–destination pairs (OD-pairs), randomly chosen from real-world OD-pairs. The OD-pairs as well as the network were provided by TomTom. For every OD-pair, we set  $Q$  to the shortest route for a single agent and run all algorithms for our psychological models and demands  $d \in \{100, 500, 1000, 1500, 2000, 2500, 3000\}$ . One unit of demand represents 7–20 vehicles per hour. The imprecision is due to the fact that the exact penetration rate of TomTom devices is unknown and that the map data is given with respect to only TomTom users.

All experiments have been conducted on a machine with two Intel Xeon Gold 5118 (12-core) CPUs with 64GiB of memory. The multi-criteria shortest-path calculations of SAP-FC and 1D-SAP-FC have been parallelized to 20 threads.

**Run Time.** Figure 3 shows the run times of our algorithms, depending on the length of the original route. The main takeaways from Figure 3 (left) are that requiring disjoint routes makes the problem easier and that the algorithms requiring fewer criteria but more multi-criteria shortest path queries are faster for instances with long original routes. Figure 3 (right) shows the speedup of SAP-FC over SAP. One can see that SAP is actually faster than SAP-FC for most instances, sometimes up to two orders of magnitude. However, these are



■ **Figure 4** The plots show the travel time per agent depending on the demand  $d$ , where each data point is averaged over all OD-pairs. Absolute values are shown on the left, relative values with respect to the  $d$ -SP solution are shown on the right.

the instances with short original route, which have low run times anyways. On the other hand, SAP-FC is up to one order of magnitude faster than SAP on some instances with long original path. We note that the multi-criteria shortest path queries in SAP-FC can be parallelized, and we used 20 threads in our experiments. However, this parallelization cannot explain such high speedups. In Figure 3 (left), one can see that SAP-FC actually has rather consistent run times compared to SAP and never exceeded 30 minutes. Thus, our observations show that we can feasibly solve the problems SAP and even more so 1D-SAP in the context of small distance queries, e.g., in city networks, despite the worst-case exponential running time.

**Strategic Improvement.** We assess how much strategic routing gains in terms of travel time with respect to different disjointness. Figure 2 shows solutions for SAP, 1D-SAP and D-SAP routes. The resulting travel times are shown in Figure 4. We see that the larger the number of agents, the more we benefit from strategic routing. The plots show that, in direct comparison to the shortest path assuming  $d$  agents per edge ( $d$ -SP), the SAP algorithms yield results of about 50% reduced travel time for growing values of  $d$ . Constraining the alternative route to be 1-disjoint from the original only has a slight disadvantage (on average 1D-SAP is worse by 2.2%). Thus, taking into account that 1D-SAP can be solved faster, solving 1D-SAP might give a good trade-off between run time and quality of the solution. Demanding full disjointness leads to much worse travel times, as in 62.3% of our test cases, no fully disjoint alternative exists, due to the graph structure. In this case, we assume that all agents use the original route. Restricted to the instances that allow for a fully disjoint solution, the solution to D-SAP on average leads to a 11.4% higher travel time per agent compared to 1D-SAP.

## 6 Conclusion

Besides providing a framework for formalizing strategic routing scenarios, we gave different algorithms solving SAP. Both of these contributions open the door to future research. Concerning SAP, we have seen that different psychological models can lead to different alternative routes, and it would be interesting to study how people actually behave depending on the exact formulation of the suggestion and on potential additional incentives to take a longer route. It is promising to study models that lie in-between the User Equilibrium and the

Linear Model. By setting, e.g.,  $c(x) = \tanh(a \cdot x/d)$  in the Quotient Model (Equation (2)), we obtain a model that behaves like the Linear Model for small  $x$  and approaches the User Equilibrium Model for larger  $x$ , where the constant  $a$  controls how quickly that happens. We note that this choice of  $c(x)$  satisfies the conditions of Theorem 15, implying that the resulting model is Pareto-conform, which makes the algorithms from Section 3 applicable. Concerning algorithmic performance, we have seen that our proof-of-concept implementation yields reasonable run times. Our implementation uses techniques such as A\* to speed up computation. Beyond that, there is still potential for engineering, e.g., by employing preprocessing techniques. Beyond the SAP problem, our framework gives rise to various problems in the context of strategic routing that are worth studying algorithmically.

---

## References

- 1 Ittai Abraham, Daniel Delling, Andrew V. Goldberg, and Renato F. Werneck. Alternative routes in road networks. *Journal of Experimental Algorithmics*, 18, 2013. doi:10.1145/2444016.2444019.
- 2 Hannah Bast, Daniel Delling, Andrew V. Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. Route planning in transportation networks. *Algorithm Engineering*, 2016. doi:10.1007/978-3-319-49487-6\_2.
- 3 Umang Bhaskar, Lisa Fleischer, and Elliot Anshelevich. A Stackelberg strategy for routing flow over time. *Games and Economic Behavior*, 92:232–247, 2015.
- 4 Thomas Bläsius, Maximilian Böther, Philipp Fischbeck, Tobias Friedrich, Alina Gries, Falk Hüffner, Otto Kißig, Pascal Lenzner, Louise Molitor, Leon Schiller, Armin Wells, and Simon Wietheger. A strategic routing framework and algorithms for computing alternative paths, 2020. URL: <https://arxiv.org/abs/2008.10316>.
- 5 Vincenzo Bonifaci, Tobias Harks, and Guido Schäfer. Stackelberg routing in arbitrary networks. *Mathematics of Operations Research*, 35(2):330–346, 2010.
- 6 Daniel Delling. Time-dependent SHARC-routing. *Algorithmica*, 60(1):60–94, 2009. doi:10.1007/s00453-009-9341-0.
- 7 Daniel Delling and Dorothea Wagner. Pareto paths with SHARC. In *Experimental Algorithms*, Lecture Notes in Computer Science, page 125–136. Springer, 2009. doi:10.1007/978-3-642-02011-7\_13.
- 8 Daniel Delling and Dorothea Wagner. Time-dependent route planning. In *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, pages 207–230. Springer, 2009. doi:10.1007/978-3-642-05465-5\_8.
- 9 Ugur Demiryurek, Farnoush Banaei-Kashani, and Cyrus Shahabi. A case for time-dependent shortest path computation in spatial networks. In *Proceedings of SIGSPATIAL 2010*, page 474–477. ACM, 2010. doi:10.1145/1869790.1869865.
- 10 Pierre Hansen. Bicriterion path problems. In *Multiple Criteria Decision Making Theory and Application*, page 109–127. Springer, 1980. doi:10.1007/978-3-642-48782-8\_9.
- 11 INRIX, Inc. INRIX Verkehrsstudie: Stau verursacht Kosten in Milliardenhöhe. *INRIX Press Releases*, 2020. URL: <https://inrix.com/press-releases/2019%2Dtraffic%2Dscorecard%2Dgerman/>.
- 12 George Karakostas and Stavros G Kolliopoulos. Stackelberg strategies for selfish routing in general multicommodity networks. *Algorithmica*, 53(1):132–153, 2009.
- 13 Yannis A. Korilis, Aurel A. Lazar, and Ariel Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking*, 5(1):161–173, 1997. doi:10.1109/90.554730.
- 14 Alexander Kröller, Falk Hüffner, Lukasz Kosma, Katja Kröller, and Mattia Zeni. Driver expectations towards strategic routing. Unpublished Manuscript, 13 Pages.
- 15 Ekkehard Köhler, Rolf H. Möhring, and Martin Skutella. Traffic networks and flows over time. In *Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation*,

- volume 5515 of *Lecture Notes in Computer Science*, page 166–196. Springer, 2009. doi:10.1007/978-3-642-02094-0\_9.
- 16 Lawrence Mandow and José. Luis Pérez De La Cruz. Multiobjective A\* search with consistent heuristics. *Journal of the ACM*, 57(5):27:1–27:25, 2008. doi:10.1145/1754399.1754400.
  - 17 Ernesto Queirós Vieira Martins. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236–245, 1984. doi:10.1016/0377-2217(84)90077-8.
  - 18 Matthias Müller-Hannemann and Karsten Weihe. Pareto shortest paths is often feasible in practice. In *Algorithm Engineering*, page 185–197. Springer, 2001.
  - 19 Giacomo Nannicini, Daniel Delling, Dominik Schultes, and Leo Liberti. Bidirectional A\* search on time-dependent road networks. *Networks*, 59(2):240–251, 2011. doi:10.1002/net.20438.
  - 20 US Bureau of Public Roads. Office of Planning, Urban Planning Division. *Traffic Assignment Manual for Application with a Large, High Speed Computer*. US Department of Commerce, 1964.
  - 21 Andreas Paraskevopoulos and Christos D. Zaroliagis. Improved alternative route planning. In *Proceedings of ATMOS 2013*, pages 108–122. Schloss Dagstuhl, 2013. doi:10.4230/OASICS.ATMOS.2013.108.
  - 22 Tim Roughgarden. *Selfish routing and the price of anarchy*. MIT Press, 2005.
  - 23 Tim Roughgarden and Éva Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, March 2002. doi:10.1145/506147.506153.
  - 24 Leon Sering and Martin Skutella. Multi-source multi-sink Nash flows over time. In *Proceedings of ATMOS 2018*, pages 12:1–12:20, 2018. doi:10.4230/OASICS.ATMOS.2018.12.
  - 25 Socrates2.0. Amsterdam pilot site launched with improved navigation service. Website, December 2019. URL: <https://socrates2.org/news-agenda/amsterdam-pilot-launched-improved-navigation-service-testers-sought>.
  - 26 Ben Strasser. Dynamic time-dependent routing in road networks through sampling. In *Proceedings of ATMOS 2017*, pages 3:1–3:17. Schloss Dagstuhl, 2017. doi:10.4230/OASICS.ATMOS.2017.3.
  - 27 Mariska Alice van Essen. *The potential of social routing advice*. PhD thesis, University of Twente, 2018. doi:10.3990/1.9789055842377.
  - 28 John Glen Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 1952. doi:10.1680/ipeds.1952.11259.
  - 29 Michael A. Yukish. *Algorithms to Identify Pareto Points in Multi-Dimensional Data Sets*. PhD thesis, Mechanical Engineering Dept., The Pennsylvania State University, State College, 2004.
  - 30 Shanjiang Zhu and David Levinson. Do people use the shortest path? An empirical test of Wardrop’s first principle. *PLOS ONE*, 10(8):1–18, 2015. doi:10.1371/journal.pone.0134322.