arXiv:2608.09472v1 [cs.DS] 10 Aug 2026

Algorithmics for Safe Bicycle Network Design with Bounded Detours in Rural Areas

Till Fluschnik
Humboldt-Universität zu Berlin, Department of Computer Science, Algorithm Engineering Group, Germany

Abstract

We introduce the *Safe Bicycle Network with Bounded Detours* (*SBNBD*) problem, a network-design problem motivated by the upgrade of rural road networks for bicycle traffic. Given an undirected graph whose edges have length and are classified as safe or unsafe with unsafe edges carrying upgrade costs, a set of terminal pairs, an upgrade budget, and a detour factor α, the task is to upgrade unsafe edges so that every terminal pair is connected by a safe path whose length is at most α times its shortest-path distance in the original network.

We study the computational complexity of SBNBD from a parameterized perspective. We prove strong NP-hardness even on highly restricted graph classes, including planar graphs of treewidth two, graphs with feedback vertex set number one, and graphs of maximum degree three. We complement these lower bounds with polynomial-time algorithms for trees and graphs of maximum degree two. For parameterized complexity, we show fixed-parameter tractability for the number of unsafe edges and prove matching lower bounds under SETH, a polynomial-kernel lower bound, and W-hardness for natural parameters. Our main positive structural result is an algorithm that maps any instance to an equivalent instance with $O(\text{fes} + p)$ vertices and edges, where fes is the feedback edge number and p is the number of terminal pairs; this yields fixed-parameter tractability for the combined parameter $\text{fes} + p$.

Finally, we construct and evaluate exact ILP-based algorithms on road networks derived from OpenStreetMap data for small German municipalities and their surrounding rural regions. The experiments show that these networks have small treewidth upper bounds and moderate feedback edge structure. We introduce a preprocessing routine based on the reduction algorithm for the parameter $\text{fes} + p$ and a heuristic cut generation based on tree decompositions. Both improve exact solving performance, in particular for harder instances. Our results for different detour-factor bounds show that a moderate increase in α can substantially reduce the total length of the upgraded network, revealing practical trade-offs between upgrade budget and the maximum allowed relative detour. Our results indicate that structural graph parameters provide a useful algorithmic lens for safe bicycle-network design in rural areas.

2012 ACM Subject Classification Theory of computation → Design & analysis of algorithms; Mathematics of computing → Combinatorial optimization; Social & professional topics → Sustainability

Keywords and phrases Parameterized Algorithms, NP-hardness, Pairwise weighted spanners, Preprocessing, Integer linear programming, Cut generation

Supplementary Material Code and data for the experiments are available at `https://github.com/buhtig-tf/Safe-Bicycle-Network-Design-with-Bounded-Detours`.

Funding *Till Fluschnik*: Funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project PACS (FL 1247/1-1, 522475669).

1 Introduction

Improving bicycle networks for urban and rural areas can help to reduce pollution [35], promote physical activity [37], and mitigate transport poverty [45]. Compared with urban areas, both public transport and bicycle infrastructure are often insufficient in rural regions [46]. Since improving public transport in rural areas is often economically challenging [17] and people use cars even for short-distance rides [2, 23], cost-effective improvement of bicycle networks becomes a central planning task. In Germany, 54% of respondents living in rural areas with less than 20,000 inhabitants report that they would like to use bicycles more frequently in the future [42]. High-quality, connected cycling infrastructure could roughly triple cycling use by 2035 (up to 45% modal share on short trips), with significant untapped potential also in rural areas [12]. Other mobility solutions like bike-sharing for rural areas [38] benefit from such infrastructure. Safety and bounded detour length [20, 40] are key criteria for a network's quality.

This paper treats the problem of cost-optimal infrastructure upgrades in a given network to enable safe bicycle trips for specified terminal pairs with only small detours. Herein, the network's edges are classified either as safe or unsafe, corresponding to low- and high-stress segments [21]. Our model is based on the bicycle network improvement problem due to Lim et al. [32]. They optimize an aggregate objective over weighted origin-destination pairs, while we impose hard relative detour constraints. They study the problem from an OR perspective, while we study the problem from a computational and parameterized complexity perspective, with a focus on graph-theoretical properties derived from the applications on rural areas. Several models similar to ours are already studied in the scientific literature, yet a fine-grained computational and parameterized complexity analysis seems to be missing to date. We complement our complexity analysis with experimental algorithmics.

1.1 Our Model and Decision Problem

We model the street network as an undirected graph G with a set V of vertices (representing junctions or points of interest like schools) and a set $E \subseteq \binom{V}{2}$ of edges (representing street segments), where the edge set E is partitioned into two sets E_s and E_u of *safe* and *unsafe* edges. Each edge is equipped with a (travel) length $\theta\colon E \to \mathbb{N}$, representing the length of or the time needed to traverse the street segment. Moreover, each unsafe edge is equipped with a cost $c\colon E_u \to \mathbb{N}$, representing the cost to make it safe (e.g, by constructing a bike lane on this street segment). For an edge subset $F \subseteq E_u$, we denote by $G\langle F\rangle$ the graph G with edge set partitioned into $E'_s = E_s \cup F$ and $E'_u = E_u \setminus F$.

A terminal pair consists of two distinct vertices from V, representing the endpoints of a requested bicycle trip. For two distinct vertices $s, t \in V$, an s-t path P is a sequence $(v_0, v_1, \ldots, v_\ell)$ of distinct vertices from V with $v_0 = s$ and $v_\ell = t$ such that $\{v_{i-1}, v_i\} \in E$ for all $i \in [\ell]$. The length $\operatorname{len}(P)$ of P is $\sum_{i=1}^{\ell} \theta(\{v_{i-1}, v_i\})$. With $\operatorname{dist}_G(s,t)$ we denote the smallest length of an s-t path in G. We call path P *safe* if $\{v_{i-1}, v_i\} \in E_s$ for all $i \in [\ell]$, and *unsafe* otherwise. We often drop the graph from the notation when it is clear from the context. The detour factor of an s-t path P in G is $\operatorname{len}(P)/\operatorname{dist}(s,t) \in \mathbb{Q}_{\geq 1}$.

▶ **Problem 1** (Safe Bicycle Network with Bounded Detours (SBNBD)). **Given** a network $G = (V, E, c, \theta)$ with $E = E_s \uplus E_u$, a set of $p \in \mathbb{N}$ terminal pairs $\mathcal{P} = \{\{s_i, t_i\} \mid i \in [p]\}$ from V, a budget $k \in \mathbb{N}_0$, and a number $\alpha \in \mathbb{Q}_{\geq 1}$, the **question** is whether there is a solution $F \subseteq E_u$ with $c(F) = \sum_{e \in F} c(e) \leq k$ such that for each $i \in [p]$, there is a safe s_i-t_i path in $G\langle F\rangle$ of detour factor at most α.

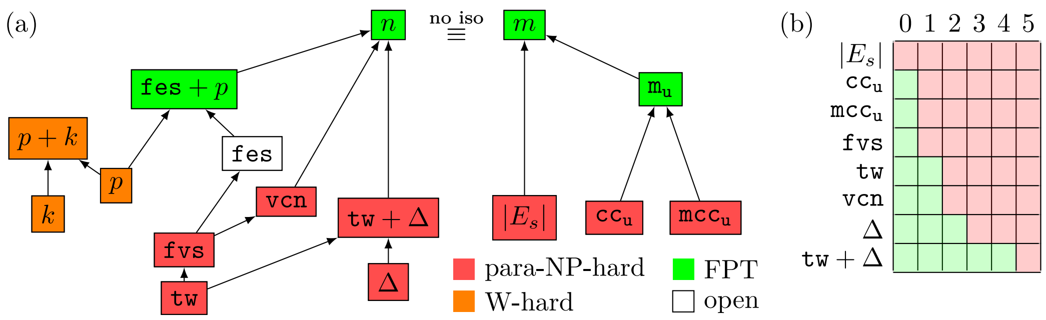

Figure 1 (a) Hasse diagram of our studied parameters with their complexity classification assuming unary encoding. Number n of vertices and number m of edges are parameterized equivalent when there are no isolated vertices. A parameter x points to another parameter y if there is a function f such that $x \leq f(y)$ holds for every instance. Thus, fixed-parameter tractability for x transfers to y, while hardness for y transfers to x. (b) Overview of the polynomial-time solvability (green) versus NP-hardness (red) borders for para-NP-hard parameters. ($\mathtt{m_u}$: number of unsafe edges; $\mathtt{cc_u}$: number of components in the graph induced by all unsafe edges; $\mathtt{mcc_u}$: size of the largest component in the graph induced by all unsafe edges.)

1.2 Our Contributions

Our contributions are two-fold. First, we provide a computational and parameterized complexity analysis for SBNBD. Second, we study exact ILP-based algorithms experimentally, focusing on the impact of reduction rules and treewidth-based cut generation.

Computational and parameterized complexity. Figure 1 arranges our complexity-theoretic results. We prove that SBNBD is NP-hard in very restricted settings, such as vertex cover number two, or treewidth two and maximum degree three. Each of our para-NP-hardness results is tight in the sense that the next smaller parameter value leads to polynomial-time solvability. We prove W[1]-hardness when parameterized by number of terminal pairs combined with the budget. For binary-encoded edge lengths and costs, we prove NP-hardness already for one terminal pair. Finally, we show several reduction rules that transform any instance to an equivalent instance whose combinatorial size is linear in the number of terminal pairs and the feedback edge number, i.e., the size of a minimum feedback edge set. These reduction rules are also effective as preprocessing in our experiments.

Experimental algorithmics. Using OpenStreetMap data, we extracted undirected road networks for 30 German villages and their 3 km surrounding rural areas. These villages were selected from a ranking of over 400 German villages according to their perceived bicycle friendliness. For these networks, we then computed the maximum degree, the size of a minimum feedback edge set, small upper bounds on the treewidth, and checked for planarity. Our main findings are that these networks have small maximum degree ≤ 5, small treewidth upper bound ≤ 14, and roughly 20% of the edges suffice on average as a feedback edge set.

We tested a plain ILP against the ILP with preprocessing via reduction rules, with additional cuts derived from a tree decomposition (TD), and with both. Our results are:

1. Most instances obtained for villages are solvable within seconds by each of the solvers.
2. For the larger instances, preprocessing combined with cut generation performed best.
3. The total length of upgraded unsafe edges decreases by 7.87% on average when α increases from 1.2 to 1.3, illustrating the trade-off between allowed detours and required upgrades.

2 Related Work

Improving bicycle networks. One line of research develops computational optimization models for bicycle-network design and improvement. Lim et al. [32] consider weighted terminal pairs and optimize the weighted detour penalty aggregated over each trip. They formulate their problem as mixed integer program (MIP) and solve it via Benders decomposition on a case-study instance. Duthie and Unnikrishnan [15] study "retrofitting" segments at minimum cost to ensure a minimum safety level with a given bounded detour. Compared to our model, their segments can have several safety levels and they express their length-bound through a function of the shortest path. While structurally, their focus is on roadway quality, ours is on parameters observed in rural areas. Mauttone et al. [34] model user route choices in a multi-commodity-flow MIP solved heuristically, allowing routes to use segments without bicycle infrastructure at higher cost. In contrast, we require each terminal pair to have safe route of bounded relative detour. Natera et al. [36] use a greedy strategy to compute missing links to interconnect components of the bicycle network. Steinacker et al. [43] use greedy network sparsification to design efficient urban bicycle networks, incorporating cycling demand and route choices based on safety preferences. To the best of our knowledge, computational or parameterized complexity analyses are missing for bicycle-network improvement problems with infrastructure upgrades under hard detour constraints. Another line of research analyzes existing bicycle networks and their growth. Szell et al. [44] argue that uncoordinated enhancement of bicycle networks can lead to higher cost and outline the need for careful planning. Schoner and Levinson [40] analyzed 74 bicycle networks from the US and found that connectivity and directness are important factors. Finally, Furth et al. [21] developed a framework to classify street segments as low- to high-stress segments.

Algorithmics of related problems. By assigning zero cost to safe edges and positive cost to unsafe edges, our problem can be viewed as an undirected pairwise weighted spanner or as a distance-constrained Steiner-type problem. Cygan et al. [11] study sparse pairwise spanners in unweighted undirected graphs where the possible increase of a pairwise distance is controlled by a stretch function. They study existential bounds and give a polynomial-time construction for additive stretch functions. Approximation algorithms and hardness results for spanner-type problems have been studied on unweighted graphs [8] and weighted directed graphs [24]. Kobayashi [30] studies the all-pairs t-spanners in unweighted undirected graphs, where each pair's distance is allowed to increase within a factor of t. They prove NP-hardness even in planar bounded-degree graphs. Moreover, they show an FPT algorithm when parameterized by the number of discarded edges (Kobayashi [31] proved such an algorithm to exist also for additive spanners). Regarding parameterized complexity, Feldmann and Lampis [18] study the related problem Steiner Forest with terminal pairs (which has no detour constraints). They focus on structural parameters and prove an EPAS with running time depending on treewidth (the problem is para-NP-hard already for $\mathtt{tw} = 3$ [22]). They also prove an FPT algorithm when parameterized by the size of a feedback edge set. Simonov [41] study the problem of computing pairwise distance preservers in undirected unweighted graphs, where no increase in path length is allowed between any terminal pair. They prove W[1]-hardness when parameterized by the number of terminal pairs and NP-hardness for vertex cover number $\mathtt{vcn} = 3$. Thus, we can view our problem as an undirected common-stretch pairwise weighted spanner problem with designated safe edges of zero cost and unsafe edges of positive cost; to the best of our knowledge, a systematic parameterized complexity study is missing for this variant.

Structural parameters of real-world graphs. Maniu et al. [33] study the treewidth of 25 networks from eight different domains, including infrastructure. They find that many of these networks have treewidth too large for a direct application of treewidth-based algorithms. In contrast, we show that the street networks derived from rural areas admit small treewidth upper bounds. Analyzing large-scale street networks using OSM data has become common [4, 5, 16]. Related to the feedback edge number, the meshedness coefficient of planar real-world street networks is studied [7], which relates the numbers of bounded faces of the network and of a maximally connected planar graph with the same number of vertices. In connected planar graphs, the number of bounded faces equals the feedback edge number. To the best of our knowledge, classic FPT parameters such as treewidth or feedback edge number have not been measured for larger datasets of rural street networks.

3 Preliminaries

We denote by $\mathbb{N}$ and $\mathbb{N}_0$ the natural numbers excluding and including zero, respectively. We denote by $\mathbb{Q}$ and $\mathbb{Q}_{\geq 1}$ the rational numbers and the rational numbers at least one, respectively. We use basic notation from parameterized complexity [10]. We use common graph-theoretic notation and explain only notation that may be non-standard. For an undirected graph $G = (V, E)$, we denote the vertex and edge set of G by $\mathtt{V}(G)$ and $\mathtt{E}(G)$, respectively. For an edge subset $E' \subseteq E$, we denote by $G[E'] = (V', E')$ with $V' = \{v \in V \mid \exists e \in E' : v \in e\}$ the edge-induced graph G induced on the edge set E'. We write $G - E'$ for the graph $(V, E \setminus E')$. For a vertex subset $U \subseteq V$, we denote by $\partial_G(U) = \{\{v, w\} \in E \mid |\{v, w\} \cap U| = 1\}$ the *cut* for U in G, i.e., all edges in G with exactly one endpoint in U.

4 Tight Strong NP-hardness on Simple Graph Classes

▶ **Theorem 1.** *SBNBD is NP-hard even on planar graphs with treewidth* $\mathtt{tw} = 2$*, and*
(a) *feedback vertex set size* $\mathtt{fvs} = 1$*,* $\mathtt{mcc_u} = 1$*, and the graph is bipartite.*
(b) *feedback vertex set size* $\mathtt{fvs} = 1$*, vertex cover number* $\mathtt{vcn} = 2$*, and* $\mathtt{cc_u} = 1$*.*
(c) *maximum degree* $\Delta = 3$*,* $\mathtt{mcc_u} = 1$*, and the graph is bipartite.*
Moreover, unless the ETH breaks, there is no $2^{o(n+m)} \cdot \mathrm{poly}(|I|)$*-time algorithm.*

We give three polynomial-time many-one reductions, each from the NP-hard [28] Vertex Cover problem, where, given an undirected graph $G = (V, E)$ and an integer r, the question is whether there is a subset $W \subseteq V$ with $|W| \leq r$ such that $e \cap W \neq \emptyset$ for every $e \in E$. Vertex Cover admits no $2^{o(|V|+|E|)} \cdot \mathrm{poly}(|I|)$-time algorithm unless the ETH breaks [26, 27].

▶ **Construction 1.** Let $I = (G = (V, E), r)$ be an instance of Vertex Cover with $\eta = |V|$ vertices. In each construction, we construct a graph G' that contains a vertex v^* for each vertex $v \in V$, the set of terminal pairs is $\mathcal{P} = \{\{v^*, w^*\} \mid e = \{v, w\} \in E\}$, and we have $k = r$. See Figure 2 for an illustration to each of the three constructions.

(a) Add a central vertex C. For each vertex $v \in V$, add two vertices v_u, v_s, and add unsafe edge $\{v_u, v^*\}$ of length 1 and cost 1, and safe edge $\{v^*, v_s\}$ of length 4. Make C adjacent to each of v_u, v_s for each $v \in V$ via a safe edge of length 1. Let $\alpha := 2$.

(b) There are two central vertices C_s, C_u (which form the vertex cover of the constructed graph) connected by a safe edge of length 1. For each vertex $v \in V$, connect v^* with C_s via a safe edge of length 4, and with C_u via an unsafe edge of length 1 and cost 1. Let $\alpha := 3$.

(c) We can assume that $\eta := |V| = 2^q$ for some $q \in \mathbb{N}$ and $2\log(\eta) + 4 < \eta$ (by padding isolated vertices). Let G' be initially a binary tree T_η with set $\{\ell_v \mid v \in V\}$ of η leaves.

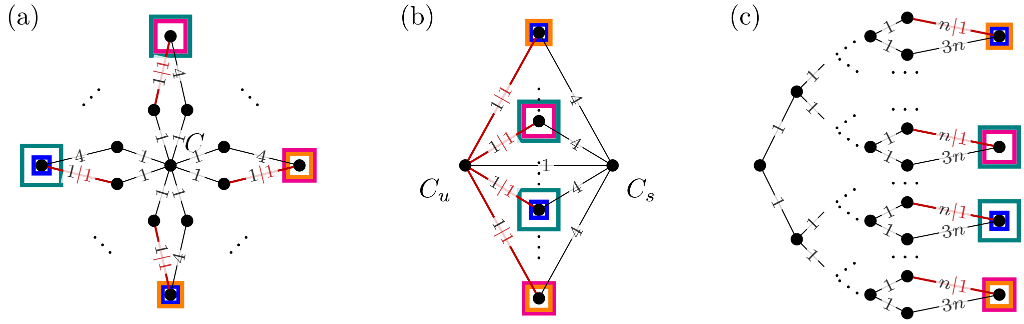

Figure 2 Illustrations for Construction 1(a), (b), and (c). An unsafe edge e is drawn red and marked with its length and cost as $\theta(e)|c(e)$. Frames depict terminal pairs, i.e., two vertices with the same frame (size and color) form a terminal pair.

For each vertex $v \in V$, add two vertices v_u, v_s, and add unsafe edge $\{v_u, v^*\}$ of length η and cost 1, and safe edge $\{v^*, v_s\}$ of length 3η. Make ℓ_v adjacent with each of v_u, v_s for each $v \in V$ via a safe edge of length 1. Let $\alpha \coloneqq 2$. ◇

Each constructed network has treewidth at most two. For (a) and (b), removing C and C_u, respectively, leaves a forest. For (c), start with a tree decomposition of width 1 for T_η. For every $v \in V$, choose a node whose bag contains ℓ_v and append a node with bag $\{\ell_v, v_u, v_s\}$. To this new node, append another node with bag $\{v_u, v_s, v^*\}$.

Proof of Theorem 1. Let $I' = (G' = (V', E', c, \theta), \mathcal{P}, k, \alpha)$ be the instance of SBNBD obtained in polynomial time from an instance $I = (G = (V, E), r)$ of Vertex Cover via Construction 1(a), (b), or (c). By construction, for each terminal pair $\{s, t\}$, every s-t path with detour factor at most α must contain an unsafe edge. For (a), the shortest path is of length 4, and the shortest safe path is of length $10 > 8 = \alpha \cdot 4$. For (b), the shortest path is of length 2, and the shortest safe path is of length $8 > 6 = \alpha \cdot 2$. For (c), the shortest path is of length $2\eta + 2 + x$, where x is the length of the shortest path through the binary tree, and the shortest safe path is of length $6\eta + 2 + x > 4\eta + 4 + 2x = \alpha \cdot (2\eta + 2 + x)$, using $x \leq 2\log(\eta)$ and our assumption $2\log(\eta) + 4 < \eta$. Consequently, for any solution F, every feasible safe s-t path in $G'\langle F\rangle$ must contain an edge of F.

Define the following convention for a subset F of unsafe edges and a set $W \subseteq V$: the unsafe edge incident with v^* is in F if and only if $v \in W$. Since in each of the three constructions (a), (b), and (c), each unsafe edge has cost one, the convention implies that F is of cost at most $k = r$ if and only if W is of size at most r. Now, for every terminal pair $\{v^*, w^*\} \in \mathcal{P}$, let $\{v, w\} \in E$ be the corresponding edge in the input graph. The following equivalences hold. There is a safe v^*-w^* path of detour factor at most α in $G'\langle F\rangle$ if and only if at least one of the two unsafe edges incident with v^* or w^* is in F. By our convention, we have that at least one of the two unsafe edges incident with v^* or w^* is in F if and only if at least one of v, w is in W. Finally, at least one of v, w is in W if and only if the edge $\{v, w\} \in E$ is covered by W. Thus, all terminal pairs admit a feasible safe path in $G'\langle F\rangle$ if and only if W is a vertex cover of G. Together with the cost correspondence, this proves that I is a *yes*-instance if and only if I' is a *yes*-instance.

Finally, note that in construction (b), G' contains $|V| + 2$ vertices and $2|V| + 1$ edges. Hence, the ETH-based lower bound follows. ◀

▶ Remark 2. The graph from Construction 1(b) also has neighborhood-diversity two: C_u, C_s form one type, and $V' \setminus \{C_u, C_s\}$ forms the other type. In fact, adding sufficiently long safe edges between any two non-adjacent vertices turns the graph into clique while preserving correctness, which results in neighborhood diversity one (but unbounded vertex cover number).

▶ Remark 3. None of the reductions has unit lengths. Yet, (a) and (c) can be turned via subdivisions into unit length; this is false for (b) as it would increase the vertex cover number.

In the remainder, we show that decreasing any of the bounds $\texttt{vcn} = 2$, $\texttt{fvs} = 1$, $\texttt{tw} = 2$, or $\Delta = 3$ by one leads to graph classes on which SBNBD is polynomial-time solvable.

▶ **Proposition 4.** *For every instance of SBNBD where the input graph is a tree, we can compute a minimum-cost solution in polynomial time. Thus, SBNBD is polynomial-time solvable on trees.*

The proof of Proposition 4 uses the following two reduction rules.

▶ **Reduction Rule 1** (Irrelevant pair). If a terminal pair is connected by a safe path of detour factor at most α, delete the terminal pair.

▶ **Reduction Rule 2** (Mandatory edge). Let terminal pair $\{s_r, t_r\}$ be not connected by a safe path of detour factor at most α. If an unsafe edge e lies on all s_r-t_r paths of detour factor at most α, then make e safe. If $c(e)$ exceeds the budget, then return a trivial *no*-instance, otherwise decrease the budget by $c(e)$.

The correctness of Reduction Rule 1 is immediate. The correctness of Reduction Rule 2 follows since every feasible solution must upgrade e: otherwise no safe s_r-t_r path of detour factor at most α can exist. Thus, if $c(e) > k$, then the instance is a *no*-instance; otherwise, upgrading e can be fixed in advance and its cost subtracted from the budget.

Proof of Proposition 4. Let I be an instance of SBNBD with G being a tree, budget k, and let κ be the sum of costs of all unsafe edges. Let I' be the instance obtained from I by setting the budget to κ. Apply Reduction Rules 1 and 2 exhaustively to I'. Since each terminal pair has a unique path, each unsafe edge on this path is mandatory and is made safe by Reduction Rule 2. Hence, every edge made safe by Reduction Rule 2 is contained in every feasible solution. Consequently, once all unsafe edges on the unique path of a terminal pair have been made safe, this terminal pair is deleted by Reduction Rule 1. It follows that no terminal pair is left. Let F^* denote the set of all unsafe edges made safe by Reduction Rule 2 during the exhaustive application, and let κ' denote the remaining budget in the final instance. Note that $c(F^*) = \kappa - \kappa'$. Since every edge in F^* is mandatory and upgrading F^* satisfies all terminal pairs, $c(F^*)$ is the minimum cost of any solution. Return *yes* if $c(F^*) \leq k$, and *no* otherwise. ◀

Using Proposition 4, we also handle graphs of maximum degree at most two: path components are immediate, and for each cycle component we either upgrade all unsafe edges or delete one unsafe edge that remains unsafe, reducing the component to a path.

▶ **Observation 5.** *SBNBD is polynomial-time solvable on graphs of degree at most two.*

Proof of Observation 5. Let I be an instance of SBNBD with detour factor α where G has maximum degree at most two. Then, graph G decomposes into paths and cycles. If a terminal pair is not contained in the same connected component, then return *no*. Otherwise, solve each connected component independently. If a connected component is a path, compute a minimum-cost solution using Proposition 4. Now consider a connected component C that is a cycle. First, consider the solution that upgrades all unsafe edges of C. This is feasible for all terminal pairs contained in C. We compare this solution with possible solutions where at least one unsafe edge is not upgraded. Thus, for each unsafe edge e of C, consider the case where e remains unsafe and hence no feasible safe path can use e. Then, every terminal pair $\{s,t\}$ contained in C must be connected along the unique s-t path in $C - \{e\}$. This is feasible only if $\mathrm{dist}_{C-\{e\}}(s,t) \leq \alpha \cdot \mathrm{dist}_G(s,t)$ for every terminal pair $\{s,t\}$ contained in C. If this condition holds, we delete e and compute a minimum-cost solution on the path $C - \{e\}$ using Proposition 4. Among the solution upgrading all unsafe edges and all solutions obtained by deleting one unsafe edge, we keep one of minimum cost. This enumeration is exhaustive since every solution either upgrades all unsafe edges of C or leaves at least one unsafe edge e not upgraded. Finally, the optimal solution for I is composed of the optimal solutions for each of its connected components. If the total cost exceeds the budget, return *no*, otherwise return *yes*. ◀

5 Number of Unsafe Edges

▶ **Observation 6.** *SBNBD can be solved in* $2^{\mathtt{m_u}} \cdot \mathrm{poly}(|I|)$ *time and hence is in FPT when parameterized by the number* $\mathtt{m_u}$ *of unsafe edges.*

Observation 6 corresponds to testing all possible solutions in polynomial-time for feasibility. We show next that, assuming the SETH to hold, we cannot do better than this.

▶ **Theorem 7.** *SBNBD, even if all unsafe edges induce one component (*$\mathtt{cc_u} = 1$*) and the input graph has diameter at most four, admits no* $2^{\varepsilon \cdot \mathtt{m_u}} \cdot \mathrm{poly}(|I|)$*-time algorithm for any* $\varepsilon < 1$ *unless the SETH breaks and no problem kernel of size polynomial in* $\mathtt{m_u}$ *unless NP* $\subseteq$ *coNP/poly. Moreover, it is W[2]-hard when parameterized by the budget* k*.*

We reduce from the Hitting Set problem, where we are given a set X of elements, a set $\mathcal{S}$ of nonempty subsets of X, and an integer $r \in \mathbb{N}$, and ask whether there is a subset $Y \subseteq X$ with $|Y| \leq r$ such that $S \cap Y \neq \emptyset$ for every $S \in \mathcal{S}$. Hitting Set is W[2]-hard when parameterized by the solution size r [14] and admits no problem kernel of size polynomial in $|X|$ unless NP $\subseteq$ coNP/poly [13]. Moreover, unless the SETH breaks, Hitting Set admits no $2^{\varepsilon \cdot |X|} \cdot \mathrm{poly}(|I|)$-time algorithm for any $\varepsilon < 1$ [9].

▶ **Construction 2.** Let $I = (X, \mathcal{S}, r)$ be an instance of Hitting Set. Construct the graph G with vertex set $V \coloneqq \{v_x \mid x \in X\} \cup \{w_S \mid S \in \mathcal{S}\} \cup \{v^*\}$, safe-edge set $E_s \coloneqq \{\{v_x, w_S\} \mid x \in X, S \in \mathcal{S} : x \in S\}$, and unsafe-edge set $E_u \coloneqq \{\{v_x, v^*\} \mid x \in X\}$. The length of each edge is one and the cost of each unsafe edge is one. Let $\mathcal{P} \coloneqq \{\{v^*, w_S\} \mid S \in \mathcal{S}\}$ be the set of terminal pairs. Let $\alpha \coloneqq 1$ and $k \coloneqq r$. ◇

Since every unsafe edge is incident with v^*, the subgraph induced by E_u is connected ($\mathtt{cc_u} = 1$). For every $S \in \mathcal{S}$, since S is nonempty, w_S and v^* have a common neighbor v_x with $x \in S$. Thus, every two vertices are connected by a path of length at most four, and so G has diameter at most four.

Proof of Theorem 7. Let $I' = (G = (V, E, c, \theta), \mathcal{P}, k, \alpha)$ be the instance of SBNBD obtained in polynomial time from an instance $I = (X, \mathcal{S}, r)$ of Hitting Set via Construction 2. For every $S \in \mathcal{S}$, we have $\mathrm{dist}_G(w_S, v^*) = 2$. Moreover, w_S is adjacent to v_x if and only if $x \in S$. Since $\alpha = 1$, for every $F \subseteq E_u$, any safe w_S-v^* path of detour factor 1 in $G\langle F\rangle$ must have length exactly 2. Hence, it contains exactly one inner vertex v_x with $x \in S$, and the edge $\{v_x, v^*\}$ must be made safe by F. We make the following convention for a set $Y \subseteq X$ and a set $F \subseteq E_u$: $x \in Y$ if and only if $\{v^*, v_x\} \in F$. Since every unsafe edge has cost one, the convention implies that $|Y| \leq r$ if and only if $c(F) \leq k$. The following equivalences hold. Set $S \in \mathcal{S}$ is hit by Y if and only if there exists $x \in S \cap Y$. By our convention, for every $x \in S$, $x \in Y$ if and only if $\{v^*, v_x\} \in F$. Moreover, $\{v^*, v_x\} \in F$ for some $x \in S$ if and only if there is a safe w_S-v^* path of detour factor 1 in $G\langle F\rangle$. It follows that every set $S \in \mathcal{S}$ is hit by Y if and only if every terminal pair admits a safe path of detour factor 1 in $G\langle F\rangle$. Together with the cost correspondence, this proves that I is a *yes*-instance if and only if I' is a *yes*-instance.

Finally, note that $\mathtt{m_u} = |X|$. Hence, both the SETH-based lower bound and the kernelization lower bound follow. W[2]-hardness transfers since $k = r$. ◀

6 Number of Terminal Pairs

▶ **Theorem 8.** *SBNBD is weakly NP-hard even if the input graph is planar, of maximum degree at most four, and of treewidth two, all unsafe edges are pairwise disjoint (*$\mathtt{mcc_u} = 1$*), and there is only one terminal pair.*

We reduce from the weakly NP-hard Partition problem, where, given a multiset $X = \{x_1, \dots, x_N\}$ of at least two positive integers with sum $\sum_{x_i \in X} x_i = 2T$, the question is whether there is a partition (X_1, X_2) of X such that $\sum_{x_i \in X_1} x_i = \sum_{x_i \in X_2} x_i = T$.

▶ **Construction 3.** Let $I = (X = \{x_1, \dots, x_N\})$ be an instance of Partition with sum $2T$. Construct the vertex set $V = \{v_i \mid i \in [0, N]\} \cup \{a_i, b_i \mid i \in [N]\}$ and the edge set as follows: For each $i \in [N]$, add the safe edges $\{v_{i-1}, a_i\}$ of length T and $\{v_{i-1}, b_i\}$ of length $2T$, the safe edge $\{b_i, v_i\}$ of length $2Nx_i$, and the unsafe edge $\{a_i, v_i\}$ of length T and cost x_i. The set $\mathcal{P}$ of terminal pairs consists only of one terminal pair $\{v_0, v_N\}$. Let $k := T$ and $\alpha := 2$. ⋄

The graph is outerplanar: an outerplanar embedding is obtained by placing v_0 at $(0, 0)$ and, for each $i \in [N]$, placing v_i at $(2i, 0)$, a_i at $(2i-1, 1)$, and b_i at $(2i-1, -1)$. Since the graph is outerplanar and contains a cycle, it has treewidth two. It has maximum degree four: as $N \geq 2$, each vertex v_i with $i \in [N-1]$ has degree four, whereas each vertex a_i and b_i with $i \in [N]$, as well as v_0 and v_N, has degree two. Finally, the set $\{\{a_i, v_i\} \mid i \in [N]\}$ of unsafe edges consists of pairwise vertex-disjoint edges, and hence $\mathtt{mcc_u} = 1$.

Proof of Theorem 8. Let $I' = (G = (V, E, c, \theta), \mathcal{P}, k, \alpha)$ be the instance of SBNBD obtained in polynomial time from an instance $I = (X = \{x_1, \dots, x_N\})$ of Partition with sum $2T$ via Construction 3. The shortest v_0-v_N path is of length $2NT$ by using the unsafe branch when traversing from v_{i-1} to v_i via a_i for each $i \in [N]$. Hence, since $\alpha = 2$, every feasible path in a solution must be of length at most $4NT$. We show that I is a *yes*-instance if and only if I' is a *yes*-instance.

For the forward direction, let (X_1, X_2) be a solution to I. We claim that $F := \{\{a_i, v_i\} \mid x_i \in X_1\}$ is a solution to I'. We have that $c(F) = \sum_{x_i \in X_1} c(\{a_i, v_i\}) = \sum_{x_i \in X_1} x_i = T \leq k$. Moreover, consider the shortest safe v_0-v_N path in $G\langle F\rangle$. For each $x_i \in X_1$, it uses the branch traversing from v_{i-1} to v_i via a_i, since the unsafe edge $\{a_i, v_i\}$ is upgraded by F; this branch has length $2T$. For each $x_i \in X_2$, it uses the branch traversing from v_{i-1} to v_i via b_i, which has length $2T + 2Nx_i$. Altogether, the path has length $2NT + 2N \sum_{x_i \in X_2} x_i = 4NT$. Thus, F is a solution to I'.

For the backward direction, let F be a solution to I'. We claim that (X_1, X_2), where $X_1 := \{x_i \mid \{a_i, v_i\} \in F\}$ and $X_2 := X \setminus X_1$, is a solution to I. Since F is a solution, the shortest safe v_0-v_N path in $G\langle F\rangle$ is of length $2NT + 2N \sum_{x_i \in X_2} x_i \leq 4NT$. It follows that $\sum_{x_i \in X_2} x_i \leq T$. Moreover, we know that $\sum_{x_i \in X_1} x_i = \sum_{\{a_i, v_i\} \in F} c(\{a_i, v_i\}) = c(F) \leq k = T$. Finally, since $\sum_{x_i \in X_2} x_i + \sum_{x_i \in X_1} x_i = 2T$, (X_1, X_2) is a solution to I. ◀

▶ Remark 9. Note that fvs, and hence fes, is unbounded in the constructed graph. Yet, the graph has cutwidth two, witnessed by an ordering such as $(\dots, v_{i-1}, a_i, b_i, v_i, \dots)$, with exactly two edges between any two consecutive vertices.

Note that in the previous reduction there is only *one* terminal pair. For unary encodings, the reduction no longer applies; however, we obtain W[1]-hardness in this case.

▶ **Theorem 10.** *SBNBD is W[1]-hard when parameterized by the number p of terminals combined with the budget k, even if $\mathtt{cc_u} = 1$, all edges are unsafe, and all shortest terminal paths are of length three.*

We reduced from the Multicolored Clique problem, which asks, given a graph $G = (V, E)$ where $V = V_1 \uplus \cdots \uplus V_r$ is partitioned into r color classes and at least one vertex from V_i is adjacent to a vertex from V_j for every distinct $i, j \in [r]$, whether there is a vertex set W such that W forms a clique in G and $|W \cap V_i| = 1$ for every $i \in [r]$. Multicolored Clique is W[1]-hard when parameterized by r [19].

▶ **Construction 4.** Let $I = (G = (V = V_1 \uplus \cdots \uplus V_r, E))$ be an instance of Multicolored Clique. Let G' be initially a copy of G. Make each edge in G' unsafe with cost 1 and length 1. For each color class add a verifier vertex w_i and make it adjacent by unsafe edges of cost r^3 and length 1 with all vertices from V_i. Let $\mathcal{P} := \{\{w_i, w_j\} \mid 1 \le i < j \le r\}$ be the set of terminal pairs. Let $\alpha := 1$ and $k := r \cdot r^3 + \binom{r}{2}$. ◇

Proof of Theorem 10. Let $I' = (G' = (V', E', c, \theta), \mathcal{P}, k, \alpha)$ be the instance of SBNBD obtained in polynomial time from an instance $I = (G = (V = V_1 \uplus \cdots \uplus V_r, E))$ of Multicolored Clique via Construction 4. We show that I is a *yes*-instance if and only if I' is a *yes*-instance.

For the forward direction, let $W = \{v_1^*, \dots, v_r^*\}$ be solution to I with $v_i^* \in V_i$ for each $i \in [r]$. We claim that $F := \{\{w_i, v_i^*\} \mid i \in [r]\} \cup \{\{v_i^*, v_j^*\} \mid i, j \in [r],\ i < j\}$ is a solution to I'. We have that $c(F) = \sum_{i \in [r]} c(\{w_i, v_i^*\}) + \sum_{i,j \in [r],\, i<j} c(\{v_i^*, v_j^*\}) = r \cdot r^3 + \binom{r}{2} = k$. Moreover, for each $i, j \in [r]$, $i < j$, there is a safe w_i-w_j path in $G'\langle F \rangle$ of length three traversing from w_i to v_i^* to v_j^* to w_j of length $\mathrm{dist}_{G'}(w_i, w_j)$. Hence, F is a solution to I'.

For the backward direction, let F be a solution to I'. By construction, for each color class $i \in [r]$, V_i separates w_i in G' from the rest of the graph. Since w_i appears in at least one terminal pair, at least one edge incident with w_i must be upgraded by F at cost r^3. As there are r color classes, at least $r \cdot r^3$ budget is spent on these upgrades. Thus, the remaining budget is at most $\binom{r}{2} < r^3$, and hence no further edge incident with any w_i can be upgraded. Hence, for every $i \in [r]$, there is exactly one upgraded edge incident with w_i in F. Let this edge be $\{w_i, v_i^*\}$ with $v_i^* \in V_i$. We claim that $W := \{v_1^*, \dots, v_r^*\}$ is a solution to I. Consider distinct $i, j \in [r]$. By the assumption on the input instance, at least one vertex from V_i neighbors a vertex from V_j, and hence $\mathrm{dist}_{G'}(w_i, w_j) = 3$. Since $\alpha = 1$, every feasible safe w_i-w_j path in $G'\langle F \rangle$ must have exactly three edges. Since $\{w_i, v_i^*\}$ is the only upgraded edge incident with w_i and $\{w_j, v_j^*\}$ is the only upgraded edge incident with w_j, the middle edge of such a path must be $\{v_i^*, v_j^*\}$. Thus, $\{v_i^*, v_j^*\} \in E$. Since this holds for every pair of distinct color classes, it follows that W induces a clique. Hence, W is a solution to I.

Since $|\mathcal{P}| = \binom{r}{2}$ and $k = r \cdot r^3 + \binom{r}{2}$ depend only on r, W[1]-hardness for the combined parameter $p + k$ follows. ◀

7 Feedback Edge Number

▶ **Theorem 11.** *Any instance of SBNBD can be mapped in polynomial time to an equivalent instance of SBNBD with $O(\mathtt{fes} + p)$ vertices and edges.*

Note that together with Observation 6, it follows that SBNBD is FPT when parameterized by $\texttt{fes}+p$. Theorem 11 consists of carefully deleting leaves and compressing long paths in the tree $G - X$, where X denotes a minimum feedback edge set. Such an approach has already proved useful (see, e.g., [18, 29]). We call a vertex in $G = (V, E)$ with terminal pairs $\mathcal{P}$ and a feedback edge set $X \subseteq E$ *important* if it is contained in a terminal pair, incident with an edge of X, or is of degree at least three in $G - X$, and *unimportant* otherwise.

▶ **Reduction Rule 3** (Leaf)**.** If a leaf in $G - X$ is unimportant, delete it.

Correctness proof of Reduction Rule 3. Let v be an unimportant leaf of $G - X$. Then, v is neither contained in a terminal pair nor incident with an edge of X. Thus, v is a non-terminal leaf of G and hence not contained in any path between terminals. Deleting v preserves all terminal-pair distances, all feasible safe paths, and the set of feasible solutions. ◀

For terminal pairs $\mathcal{P}$ and a feedback edge set $X \subseteq E$, we call a path P in $G - X$ a *corridor* if it contains at least three vertices, each of its endpoints is important, and all its inner vertices are unimportant and of degree two in $G - X$. We exploit that any s-t path connecting a terminal pair $\{s, t\}$ contains either all edges of such a corridor or none. Hence, we can replace such a corridor by a path with at most two edges thereby preserving all relevant information (lengths and costs) for the instance. However, we need to carefully distinguish the type of the corridor in terms of contained safe and unsafe edges and whether its endpoints are connected by a feedback edge. Note that since P is a subgraph of $G - X$, if the endpoints of P are adjacent in G, then the edge belongs to X. Let v, w be the endpoints of corridor P. We call P *unsafe* if it contains at least one unsafe edge, and *safe* otherwise. We prove next that we can replace corridors with many edges by paths with at most two edges.

▶ **Lemma 12.** *Let I be an instance of SBNBD with graph $G = (V, E)$ and with feedback edge set $X \subseteq E$. Let P be a corridor with endpoints v, w. Let I' be the instance with graph $G' = (V', E')$ obtained from I only by replacing P with a path P' with endpoints v, w such that $\operatorname{len}(P') = \operatorname{len}(P)$, and P' consists of either two edges and one unimportant inner vertex if $\{v, w\} \in E$, or of one edge otherwise. Moreover, P is safe if and only if P' is safe, and if both are unsafe, $\sum_{e \in \mathtt{E}(P) \cap E_u} c(e) = \sum_{e \in \mathtt{E}(P') \cap E'_u} c(e)$, where $E'_u \subseteq E'$ is the set of unsafe edges in G'. Then I is a* yes*-instance if and only if I' is a* yes*-instance.*

Proof. Since every inner vertex of P and of P' is unimportant, none of them is in a terminal pair. Since the inner vertices of P and P' have degree two in G and G', respectively, every path connecting a terminal pair contains either all or none of the edges of P and P'. Since $\operatorname{len}(P) = \operatorname{len}(P')$, it follows that $\operatorname{dist}_G(s, t) = \operatorname{dist}_{G'}(s, t)$ for every terminal pair $\{s, t\} \in \mathcal{P}$. Finally, note that G' is a simple graph and hence a valid input to SBNBD since P' has two edges in the case of $\{v, w\} \in E$.

For the forward direction, let $F \subseteq E$ be a cost-minimal solution to I. Let $F_P \coloneqq F \cap E(P)$. We know that either $F_P = \emptyset$ or $F_P = E(P) \cap E_u$. Let $F_{P'} \coloneqq E(P') \cap E'_u$. Let F' be F if $F_P = \emptyset$, and $(F \setminus F_P) \cup F_{P'}$ otherwise. By the cost equality for P and P', we have $c(F') = c(F) \leq k$. We claim that F' is a solution to I'. Let $\{s, t\} \in \mathcal{P}$ be a terminal pair. Let Q be a safe s-t path in $G\langle F \rangle$ of detour factor at most α. If Q contains no edge from P, then Q is also a safe s-t path in $G'\langle F' \rangle$, since $\operatorname{dist}_G(s, t) = \operatorname{dist}_{G'}(s, t)$. If Q contains an edge from P, then P is a subpath of Q. Let Q' be the path in G' obtained from Q by replacing P by P'. Note that $\operatorname{len}(Q) = \operatorname{len}(Q')$. Moreover, Q' is safe since if all unsafe edges of P are upgraded by F, so are all unsafe edges of P' upgraded by F'. With

$\mathrm{dist}_G(s,t) = \mathrm{dist}_{G'}(s,t)$, it follows that Q' is safe and of detour factor at most α. Since this holds for every terminal pair, F' is a solution to I'.

For the backward direction, let $F' \subseteq E'$ be a cost-minimal solution to I'. Let $F_{P'} := F' \cap E(P')$. We know that either $F_{P'} = \emptyset$ or $F_{P'} = E(P') \cap E'_u$. Let $F_P := E(P) \cap E_u$. Let F be F' if $F_{P'} = \emptyset$, and $(F' \setminus F_{P'}) \cup F_P$ otherwise. By the cost equality for P and P', we have $c(F) = c(F') \leq k$. We claim that F is a solution to I. Let $\{s,t\} \in \mathcal{P}$ be a terminal pair. Let Q' be a safe s-t path in $G'\langle F'\rangle$ of detour factor at most α. If Q' contains no edge from P', then Q' is also a safe s-t path in $G\langle F\rangle$, since $\mathrm{dist}_G(s,t) = \mathrm{dist}_{G'}(s,t)$. If Q' contains an edge from P', then P' is a subpath of Q'. Let Q be the path in G obtained from Q' by replacing P' by P. Note that $\mathrm{len}(Q') = \mathrm{len}(Q)$. Moreover, Q is safe since if all unsafe edges of P' are upgraded by F', so are all unsafe edges of P upgraded by F. With $\mathrm{dist}_G(s,t) = \mathrm{dist}_{G'}(s,t)$, it follows that Q is safe and of detour factor at most α. Since this holds for every terminal pair, F is a solution to I. ◀

If the endpoints of a corridor are connected by a feedback edge, we compare the corridor with this alternative connection. To this end, we introduce the following notation. We call a path R with endpoints v, w and no important inner vertex *irrelevant* if there is another path R' with endpoints v, w in $G - (V(R) \setminus \{v,w\})$ and no important inner vertex such that if R is safe, then R' is safe and $\mathrm{len}(R') \leq \mathrm{len}(R)$; if R is unsafe, then $\mathrm{len}(R') \leq \mathrm{len}(R)$ and $\sum_{e \in \mathtt{E}(R') \cap E_u} c(e) \leq \sum_{e \in \mathtt{E}(R) \cap E_u} c(e)$.

▶ **Lemma 13.** *Let I be an instance of SBNBD with an irrelevant path R. Let I' be the instance obtained from I by deleting all edges and inner vertices of R. Then, I is a* yes*-instance if and only if I' is a* yes*-instance.*

Proof. Let v, w denote the endpoints of R and let R' be a witness such that R is irrelevant. Since no inner vertex of R is a terminal and every inner vertex has degree two, every terminal-to-terminal path contains either all or none of the edges of R. Let G' denote the graph in I'. Since G' is a subgraph of G, we have $\mathrm{dist}_G(s,t) \leq \mathrm{dist}_{G'}(s,t)$. Conversely, every path using R can replace it by R' without increasing its length, as $\mathrm{len}(R') \leq \mathrm{len}(R)$. Thus, $\mathrm{dist}_G(s,t) = \mathrm{dist}_{G'}(s,t)$ for every terminal pair $\{s,t\} \in \mathcal{P}$. Next we show that I is a *yes*-instance if and only if I' is a *yes*-instance. Regarding the backward direction: Since G' is a subgraph of G and distances are preserved, any solution for I' is also feasible for I.

For the forward direction, let F be a solution to I. If R is safe, then every feasible safe path using R can replace R by the safe path R', which is no longer since $\mathrm{len}(R') \leq \mathrm{len}(R)$. All paths not using R remain unchanged. Thus, F is also a solution to I'. Now assume that R is unsafe. If F does not contain all unsafe edges of R, then $F' := F \setminus (F \cap \mathtt{E}(R))$ is also a solution to I' since no safe path in $G\langle F\rangle$ uses an edge from R. Otherwise, let F' be obtained from F by replacing all unsafe edges of R by all unsafe edges of R'. Since $\sum_{e \in \mathtt{E}(R') \cap E_u} c(e) \leq \sum_{e \in \mathtt{E}(R) \cap E_u} c(e)$, the cost of F' is at most the cost of F. Since F is a solution to I, for every terminal pair $\{s,t\} \in \mathcal{P}$, there is a safe path Q in $G\langle F\rangle$ of detour factor at most α. If Q does not contain R, then it is also a safe path in $G'\langle F'\rangle$. If Q contains R, then the s-t path Q' obtained from Q by replacing R by R' is a safe path in $G'\langle F'\rangle$ of length at most $\mathrm{len}(Q)$, since $\mathrm{len}(R') \leq \mathrm{len}(R)$. Hence, F' is also a solution to I'. ◀

We proceed towards our reduction rules. We call a safe corridor P *contractible* if (i) $\{v,w\} \notin E$, (ii) $e = \{v,w\} \in X \cap E_s$, or (iii) $e = \{v,w\} \in X \cap E_u$ and $\mathrm{len}(P) \leq \theta(e)$, and *compressible* if it contains at least four vertices, $e = \{v,w\} \in X \cap E_u$, and $\mathrm{len}(P) > \theta(e)$. A safe corridor can be neither contractible nor compressible, that is, when it has three vertices, $e = \{v,w\} \in X \cap E_u$, and $\mathrm{len}(P) > \theta(e)$.

▶ **Reduction Rule 4** (Safe Corridor Contraction)**.** Let P be a contractible safe corridor with endpoints v, w.

1. If $\{v, w\} \notin E$, then remove all edges and inner vertices from P and add the edge $e = \{v, w\}$ to E_s and set $\theta(e) = \text{len}(P)$.
2. If $e = \{v, w\} \in X \cap E_s$, then remove e and all edges and inner vertices from P, add a new safe edge e' and set $\theta(e') = \min\{\theta(e), \text{len}(P)\}$.
3. If $e = \{v, w\} \in X \cap E_u$ and $\text{len}(P) \leq \theta(e)$, then remove e and all edges and inner vertices from P and add the edge $e' = \{v, w\}$ to E_s and set $\theta(e') = \text{len}(P)$.

Reduction Rule 4 is correct since an unsafe alternative not shorter than the safe corridor is irrelevant, and among two safe options, the longer is irrelevant.

Correctness proof of Reduction Rule 4. The correctness of the first case follows directly from Lemma 12. For the second case, if $\theta(e) \leq \text{len}(P)$, then P is irrelevant and correctness follows from Lemma 13. Otherwise, the single-edge path e is irrelevant. Hence, the correctness follows from Lemma 13 together with Lemma 12. For the third case, since $\text{len}(P) \leq \theta(e)$, the single-edge path e is irrelevant. Hence, the correctness follows from Lemma 13 together with Lemma 12. ◀

The next rule treats a shorter unsafe alternative to the safe corridor. In this case, we have to keep both, but can compress the safe corridor into a path with two edges that preserves the length of the corridor. The correctness follows directly from Lemma 12.

▶ **Reduction Rule 5** (Safe Corridor Compression)**.** Let P be a compressible safe corridor with endpoints v, w and let $e = \{v, w\} \in X \cap E_u$. Then remove all edges and inner vertices from P, add a new vertex $x_{v,w}$ and the edges $e' = \{v, x_{v,w}\}, e'' = \{x_{v,w}, w\}$ to E_s, and set $\theta(e') = \text{len}(P) - 1$ and $\theta(e'') = 1$.

Let P be an unsafe corridor with total cost $M \coloneqq \sum_{e \in \mathtt{E}(P) \cap E_u} c(e)$. We call P *contractible* if (i) $\{v, w\} \notin E$, (ii) $e = \{v, w\} \in X \cap E_s$ and $\text{len}(P) \geq \theta(e)$, or (iii) $e = \{v, w\} \in X \cap E_u$ and

$$(\text{len}(P) \leq \theta(e) \wedge M \leq c(e)) \vee (\text{len}(P) \geq \theta(e) \wedge M \geq c(e)). \tag{1}$$

We call P *compressible* if it contains at least four vertices and either (i) $e = \{v, w\} \in X \cap E_s$ and $\text{len}(P) < \theta(e)$, or (ii) $e = \{v, w\} \in X \cap E_u$ and (1) does not hold. An unsafe corridor can be neither contractible nor compressible, that is, when it has three vertices and the conditions (i) or (ii) for compressible corridors hold.

▶ **Reduction Rule 6** (Unsafe Corridor Contraction)**.** Let P be a contractible unsafe corridor with endpoints v, w. Let $M \coloneqq \sum_{e \in \mathtt{E}(P) \cap E_u} c(e)$ be the total cost of P.

1. If $\{v, w\} \notin E$, then remove all edges and inner vertices from P and add the edge $e = \{v, w\}$ to E_u and set $\theta(e) = \text{len}(P)$ and $c(e) = M$.
2. If $e = \{v, w\} \in X \cap E_s$ and $\text{len}(P) \geq \theta(e)$, then remove e and all edges and inner vertices from P, add the safe edge $e' = \{v, w\}$ and set $\theta(e') = \theta(e)$.
3. If $e = \{v, w\} \in X \cap E_u$ and (1) holds, then remove e and all edges and inner vertices from P, add a new unsafe edge e' and let $c(e') \coloneqq \min\{M, c(e)\}$ and $\theta(e') \coloneqq \min\{\theta(e), \text{len}(P)\}$.

Reduction Rule 6 is correct since when one unsafe connection weakly dominates the other (neither longer nor more costly), the dominated one is irrelevant for both shortest paths and upgrade costs.

Correctness proof of Reduction Rule 6. The correctness of the first case follows directly from Lemma 12. For the second case, since $\mathrm{len}(P) \geq \theta(e)$, P is irrelevant and correctness follows from Lemma 13. Now consider the third case. If $\mathrm{len}(P) \leq \theta(e)$ and $M \leq c(e)$, then the single-edge path e is irrelevant. Hence, the correctness follows from Lemma 13 together with Lemma 12. If $\mathrm{len}(P) \geq \theta(e)$ and $M \geq c(e)$, then P is irrelevant and correctness follows from Lemma 13. These two cases are exactly the two alternatives in (1). ◀

The next rule treats two incomparable unsafe paths with the same endpoints. In this case, we have to keep both, but can compress the unsafe corridor into a path with two edges that preserves the length of the corridor and the cost to upgrade all of its unsafe edges. The correctness follows directly from Lemma 12.

▶ **Reduction Rule 7** (Unsafe Corridor Compression)**.** Let P be a compressible unsafe corridor with endpoints v, w. Let $M := \sum_{e \in \mathtt{E}(P) \cap E_u} c(e)$ be the total cost of P. Then remove all edges and inner vertices from P, add a new vertex $x_{v,w}$ and the edges $e' = \{v, x_{v,w}\}$ to E_u and $e'' = \{x_{v,w}, w\}$ to E_s. Set $\theta(e') = \mathrm{len}(P) - 1$, $c(e') = M$, and $\theta(e'') = 1$.

For easier reference, we summarize the four preceding reduction rules in the following reduction rule. Its correctness follows from the correctness of the four individual rules.

▶ **Reduction Rule 8** (Corridor)**.** Let P be a corridor. If P is safe, then apply Reduction Rule 4 if it is contractible and Reduction Rule 5 if it is compressible. If P is unsafe, then apply Reduction Rule 6 if it is contractible and Reduction Rule 7 if it is compressible.

Each corridor replacement preserves the relevant v-w connections. Safe and unsafe corridors represent length-$\mathrm{len}(P)$ connections that are already safe and that can be made safe at upgrade cost M, respectively. If the endpoints are joined by a feedback edge e, then e contributes a length-$\theta(e)$ connection that is either already safe or that can be made safe at upgrade cost $c(e)$. The contraction rules delete dominated connections, while the compression rules keep two incomparable connections by representing the corridor as a two-edge path. Since every s-t path for terminal pair $\{s, t\}$ uses either the whole corridor or none of it, these replacements preserve feasibility and optimum cost.

Proof of Theorem 11. Let X be a minimum feedback edge set of G with $|X| = \mathtt{fes}$ and let T denote the forest obtained from $G - X$ when Reduction Rule 3 is applied exhaustively. We mark all important vertices and keep these marks fixed throughout the reduction. Now, apply Reduction Rule 8 exhaustively. Whenever a rule deletes an edge of X, we remove it from the current feedback edge set. New edges replacing subpaths of T are treated as forest edges. Let T' denote the forest obtained from T by the above replacements, $X' \subseteq X$ the remaining feedback edges (recall that applying corridor contractions can delete feedback edges as well), and let G' denote the graph T' together with the edges from X'. Let $\mathtt{V}(T') = V_1 \cup V_2^+ \cup V_2^- \cup V_{\geq 3}$, where we denote by V_1 the set of leaves, by V_2^+ the set of important degree-2 vertices, by V_2^- the set of unimportant degree-2 vertices, and by $V_{\geq 3}$ the set of vertices of degree at least three. We know that $|V_1| + |V_2^+| \leq 2\mathtt{fes} + 2p$, and it is well known that $|V_{\geq 3}| \leq |V_1|$. Next we show that $|V_2^-| \leq \mathtt{fes}$. After the execution, every remaining unimportant degree-two vertex is the unique inner vertex of a corridor with three vertices whose endpoints are important and adjacent by a remaining feedback edge. Since T' is a forest, for each feedback edge $\{v, w\} \in X'$ there is at most one v-w path in T'. Hence, this defines an injective mapping from unimportant degree-two vertices to feedback edges in X', and thus $|V_2^-| \leq |X'| \leq \mathtt{fes}$. Hence, we get that $|\mathtt{V}(G')| = |\mathtt{V}(T')| = |V_1| + |V_2^+| + |V_2^-| + |V_{\geq 3}| \leq |V_1| + |V_2^+| + |V_2^-| + |V_1| \leq 5\mathtt{fes} + 4p$. Finally, $|\mathtt{E}(G')| = |\mathtt{E}(T')| + |X'| \leq 6\mathtt{fes} + 4p$. ◀

8 ILP, Preprocessing, and Cut Generation

For every $r \in [p]$, we denote by A_r^α the set of all α-*admissible* arcs, where, for an edge $e = \{v, w\}$ and terminal pair $\{s_r, t_r\}$, arc (v, w) is α-admissible for terminal pair $\{s_r, t_r\}$ if $\mathrm{dist}(s_r, v) + \theta(e) + \mathrm{dist}(w, t_r) \leq \alpha \cdot \mathrm{dist}(s_r, t_r)$. For a vertex $v \in V$, we denote by $\mathrm{out}_r^\alpha(v)$ and $\mathrm{in}_r^\alpha(v)$ the sets of outgoing and incoming arcs at v in A_r^α, respectively. We say that an edge is α-admissible for terminal pair $\{s_r, t_r\}$ if at least one of its two associated arcs is α-admissible for terminal pair $\{s_r, t_r\}$. Let E_r^α denote the set of all α-admissible edges for terminal pair $\{s_r, t_r\}$. Finally, let $G_r^\alpha := G[E_r^\alpha]$ denote the α-admissible graph for terminal pair $\{s_r, t_r\}$. We use the following ILP as our baseline exact formulation.

$$\min \sum_{e \in E_u} c(e) \cdot x_e \tag{2a}$$

$$\text{s.t.} \sum_{a \in \mathrm{out}_r^\alpha(v)} f_a^r - \sum_{a \in \mathrm{in}_r^\alpha(v)} f_a^r = \begin{cases} 1, & \text{if } v = s_r, \\ -1, & \text{if } v = t_r, \\ 0, & \text{otherwise,} \end{cases} \qquad \forall r \in [p],\ \forall v \in V, \tag{2b}$$

$$\sum_{a \in A_r^\alpha \cap \{(v,w),(w,v)\}} f_a^r \leq x_e, \qquad \forall r \in [p],\ \forall \{v, w\} \in E_u, \tag{2c}$$

$$\sum_{a \in A_r^\alpha} \theta(a) \cdot f_a^r \leq \alpha \cdot \mathrm{dist}_G(s_r, t_r), \qquad \forall r \in [p], \tag{2d}$$

$$0 \leq f_a^r \leq 1, \qquad \forall r \in [p],\ \forall a \in A_r^\alpha. \tag{2e}$$

$$x_e \in \{0, 1\}, \qquad \forall e \in E_u, \tag{2f}$$

Constraints (2b)–(2e) ensure that for each terminal pair $\{s_r, t_r\}$, a safe s_r-t_r path exists, that can use upgraded unsafe edges (2c), but must have detour factor at most α (2d).

8.1 FES-based Preprocessing

We apply the reduction rules that we described before together with the following two, where the second is a direct combination of Reduction Rules 1 and 2:

▶ **Reduction Rule 9** (Irrelevant edge)**.** If an edge is not α-admissible for any terminal pair, delete it.

▶ **Reduction Rule 10** (Unique path)**.** If a terminal pair is not connected by a safe path of detour factor at most α but connected by a unique path of detour factor at most α containing at least one unsafe edge, then let C denote the sum of costs of all unsafe edges on this path. If C exceeds the budget, return a trivial *no*-instance, otherwise make all unsafe edges on the path safe, decrease the budget by C, and delete the terminal pair.

We apply each reduction rule exhaustively before moving to the next one. Since one reduction rule can make another rule applicable, their ordering matters. Figure 3 shows how we arrange our reduction rules. Empirically, this two-round execution provides a good trade-off between running time and preprocessing effectiveness compared to an exhaustive application until every reduction rule becomes inapplicable. Before each of Reduction Rules 3 and 8, we recompute a minimum feedback edge set. Notably, after applying the shrink heuristic, the treewidth of the obtained graph is not larger than the treewidth of the input graph, since deletion and edge contraction and compression do not increase the treewidth.

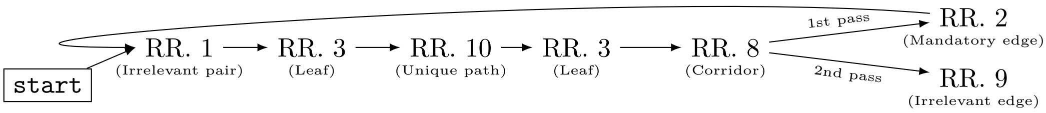

Figure 3 Flow chart of our shrink heuristic.

8.2 TD-separator Cuts

Let $(\mathcal{T}, \{B_i\}_{i\in I})$ with tree $\mathcal{T} = (I, Y)$ be a tree decomposition of G. Our cut generation is based on the following, simple insight about cuts consisting only of unsafe edges.

▶ **Observation 14.** *Let $\{s_r, t_r\} \in \mathcal{P}$ be a terminal pair and $U \subseteq \mathtt{V}(G_r^\alpha)$ such that $|\{s_r, t_r\} \cap U| = 1$. If $\emptyset \neq \partial_{G_r^\alpha}(U) \subseteq E_u$, then $|F \cap \partial_{G_r^\alpha}(U)| \geq 1$ for every solution F.*

Proof. Since $\partial_{G_r^\alpha}(U)$ forms an s_r-t_r cut in G_r^α, it intersects all s_r-t_r paths of detour factor at most α in G. Thus, every feasible s_r-t_r path must traverse an edge from $\partial_{G_r^\alpha}(U)$. Since $\partial_{G_r^\alpha}(U)$ contains only unsafe edges, every solution must upgrade at least one edge from $\partial_{G_r^\alpha}(U)$. ◀

It is well known that every edge $\{i, j\} \in Y$ of $\mathcal{T}$ corresponds to a separator $S_{\{i,j\}} := B_i \cap B_j$ in the following sense. Let $I_i^{\{i,j\}}$ and $I_j^{\{i,j\}}$ denote all nodes of the connected component of $\mathcal{T} - \{\{i, j\}\}$ containing i and j, respectively. Let $B_x^{\{i,j\}} = \bigcup_{y\in I_x^{\{i,j\}}} B_y$ denote the corresponding set of vertices in G for each $x \in \{i, j\}$. Then $B_i^{\{i,j\}} \setminus S_{\{i,j\}}$ is disconnected from $B_j^{\{i,j\}} \setminus S_{\{i,j\}}$ in $G - S_{\{i,j\}}$. Now assume that $\mathcal{T}$ is rooted at some node. For an edge $\{i, j\} \in Y$, let $\min_{\mathcal{T}}(\{i, j\})$ denote the endpoint of larger depth. Every $X \subseteq S_{\{i,j\}}$ defines a *candidate side* $V_{\{i,j\}}^X := \left(B_{\min_{\mathcal{T}}(\{i,j\})}^{\{i,j\}} \setminus S_{\{i,j\}}\right) \cup X$. Altogether, we consider the set $\mathcal{V} = \{V_{\{i,j\}}^X \mid \{i, j\} \in Y, X \subseteq S_{\{i,j\}}\}$ of all candidate sides.

For each terminal pair $\{s_r, t_r\}$ and candidate side $V_{\{i,j\}}^X \in \mathcal{V}$, let $V_{\{i,j\},r}^X := V_{\{i,j\}}^X \cap \mathtt{V}(G_r^\alpha)$ be the candidate side when restricted to the vertices of the α-admissible graph for $\{s_r, t_r\}$. If $|\{s_r, t_r\} \cap V_{\{i,j\},r}^X| = 1$ and $\emptyset \neq \partial_{G_r^\alpha}(V_{\{i,j\},r}^X) \subseteq E_u$, then, due to Observation 14, every feasible solution upgrades at least one edge of this cut. Hence, for this terminal pair and candidate side, we add the valid inequality $\sum_{e\in\partial_{G_r^\alpha}(V_{\{i,j\},r}^X)} x_e \geq 1$ to the ILP.

9 Experiments

Our experiments ran on Intel® Xeon® Silver 4310 CPU at 2.10GHz (12 cores), 125 GB RAM, Ubuntu 24.04.4 LTS (x86_64), using Gurobi(py) 13.0 for the ILPs.

9.1 Instances

We considered 30 village networks and 30 region networks. For each network, we combined three safety models, three terminal-pair fractions, two random samples, and three detour factors. In total, we constructed 1620 village instances and 1620 region instances.

Networks. We select municipalities based on the *ADFC Bicycle Climate Test* [1], a nationwide online cyclist satisfaction survey in which participants rate the perceived bicycle-friendliness of their municipality. From the ranking for towns of at most 20000 inhabitants, we took the subranking when filtered for towns of at most 10000 inhabitants. From this,

Type	Vertices	Edges m	$\mathtt{fes}/m$	$\overline{\mathtt{tw}}$	Δ	Planar
Village	$355.43\ [341^{655}_{61}]$	$443.67\ [412.50^{871}_{76}]$	$0.2\ [0.20^{0.25}_{0.11}]$	$6.00\ [5^{11}_{3}]$	$4.17\ [4^{5}_{4}]$	27/30
Region	$738.63\ [604.50^{1929}_{140}]$	$934.97\ [755^{2623}_{176}]$	$0.2\ [0.20^{0.28}_{0.14}]$	$7.97\ [7^{14}_{4}]$	$4.43\ [4^{5}_{4}]$	20/30

Table 1 Structural statistics for our village and region networks as to the means with median, minimum (lower), and maximum (upper) in parentheses. $\overline{\mathtt{tw}}$ denotes the width of a TD computed via minimum fill-in.

<table>
<tr><th></th><th>Model A</th><th>Model B</th><th>Model C</th></tr>
<tr><td>Safe in all models</td><td colspan="3">Living streets, traffic-calmed roads, and extracted road segments with explicit bicycle infrastructure</td></tr>
<tr><td>Residential roads</td><td rowspan="2">Safe</td><td rowspan="2">Safe if speed is missing or at most 30 km/h</td><td>Safe if speed is missing or at most 30 km/h</td></tr>
<tr><td>Service / unclassified roads</td><td>Safe only if speed is at most 20 km/h</td></tr>
<tr><td>Tertiary roads</td><td colspan="2">Safe only if speed is at most 30 km/h</td><td>Unsafe by default</td></tr>
<tr><td>Paths / pedestrian streets, if present</td><td colspan="2">Safe unless bicycles are explicitly forbidden</td><td>Safe only if bicycles are explicitly allowed</td></tr>
<tr><td rowspan="2">Average unsafe fraction</td><td>$0.22\ [0.21^{0.32}_{0.10}]$</td><td>$0.29\ [0.28^{0.51}_{0.12}]$</td><td>$0.37\ [0.36^{0.58}_{0.15}]$</td></tr>
<tr><td>$0.22\ [0.22^{0.30}_{0.14}]$</td><td>$0.30\ [0.28^{0.56}_{0.15}]$</td><td>$0.36\ [0.35^{0.57}_{0.15}]$</td></tr>
</table>

Table 2 Summary of the three edge safety models. The first row gives rules that make an edge safe in all models; subsequent rows list additional rules specific to road classes. Last two rows report average unsafe-edge fractions, with median, minimum, and maximum in parentheses, for villages (top row) and regions (bottom row).

we selected the top ten, bottom ten, and ten randomly sampled from the remaining towns. From OSM, we extracted the road network using type "drive", made it undirected, kept only the largest connected component, and assigned the length in accordance with the OSM data. Herein, when two vertices are connected by two arcs, we only represent them separately (via paths with two edges) if they differ significantly in length or type. We set the cost of an unsafe edge to its length, assuming the cost to scale with length, thereby neglecting the type of the street segment and any further details. This abstracts from realistic costs, for which we did not find reliable comparable data. Analogously, for each of the selected towns, we computed the network corresponding to the 3 km radius region around the associated town's center, mimicking inter-village networks. Table 1 gives an overview of selected statistics relevant to us for the derived networks. Remarkably, on average, 20% of the edges suffice as a feedback edge set, and the maximum observed treewidth upper bound is 11 for villages and 14 for regions. We also point out that only two thirds of our region networks are planar.

Safety models. For classifying edges as unsafe, we developed three models A, B, and C that are nested in spirit. Model A is an accommodating local-access model where local roads are generally considered bicycle-suitable. Model B is a balanced speed-aware model where local roads require low-speed evidence. Finally, model C is a conservative low-stress model where only clearly low-stress roads are considered safe. See Table 2 for further details.

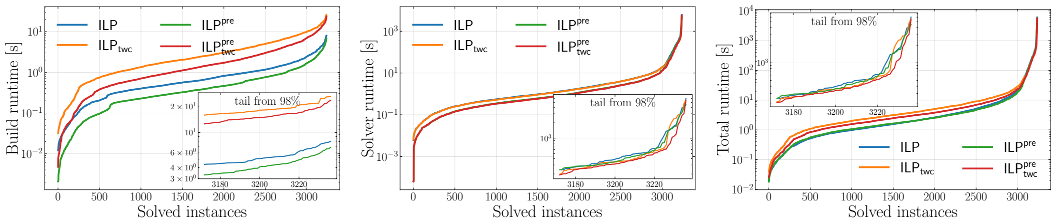

Figure 4 Cactus plots for all instances solved by each of the four solvers regarding building time (left), solver runtime (middle), and total runtime (right). Each inset zooms into the tail of the corresponding cactus plot, i.e., the slowest 2% of the common solved instances.

Terminal pairs and detour-factors. For each network, we randomly sampled $x \cdot n$ many (distinct) terminal pairs for $x \in \{0.25, 0.5, 0.75\}$ for villages and for $x \in \{0.15, 0.3, 0.45\}$ for regions, where n denotes the number of vertices. For each network and such setup, we constructed two instances. Finally, for each of those, we chose detour factors $\alpha \in \{1.2, 1.3, 1.5\}$, where $\alpha = 1.2$ [20, p. 10] and $\alpha = 1.3$ [6] are used in bicycle-network design as target bounds.

9.2 Algorithmic Setup

We compare ILP (cf. (2)) against $\mathsf{ILP}_{\mathsf{twc}}$, which is ILP with TD-separator cuts, $\mathsf{ILP}^{\mathsf{pre}}$, which is ILP where preprocessing is applied, and $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc}}$, the combination of both. To compute a tree decomposition, we use NetworkX's implementation `treewidth_min_fill_in` [25] of the minimum fill-in heuristic [3, 39]. On a high level, the heuristic greedily turns neighborhoods into cliques, thereby creating for the resulting graph a so-called perfect elimination ordering, which yields a tree decomposition. The number of candidate sides for each TD-separator can be as large as $2^{\overline{\mathtt{tw}}+1}$. Each such candidate can be considered for every terminal pair, leading to many candidate cuts. Thus, our implementation limits the process of generating TD-separator cuts as part of the ILP building time in the following three ways.

1. We skip TD-separators of size larger than s^*, since enumerating all 2^{s^*} candidate sets can be expensive. We set $s^* = 14$; recall that the maximum $\overline{\mathtt{tw}}$ in our experiments is at most 14, implying that the limit applies only to boundary cases in our experiments.
2. We only consider the best N^* candidate sides for cut generation according to the following ranking. Let $U \in \mathcal{V}$ be a candidate side with $\partial_G(U) \neq \emptyset$. Define the (negative) fraction $\rho_G(U) = -|\partial_G(U) \cap E_u|/|\partial_G(U)|$ of unsafe edges in the cut. The score of U in G is defined as $\mathrm{score}(U) = (|\partial_G(U)|, \rho_G(U), |U|)$. Each dimension of the score is to minimize, and scores are compared lexicographically. That is, firstly we seek small cuts, secondly high fractions of unsafe edges, and thirdly small sides. Ties are broken arbitrarily (by their time of appearance) for any two candidate sides with the same score. We set $N^* = 5000$.
3. For each terminal pair $\{s_i, t_i\}$, we only consider cuts with at most m^* edges. From these, we only add at most c^* cuts to the ILP. We set $m^* = 10$ and $c^* = 20$.

These limits affect only *which* valid cuts are generated; every added inequality is valid.

9.3 Results

Solver performance. Overall, from our experiments, we can conclude the following regarding the solver performance in terms of runtimes; Figure 4 and Figure 5 accompany our conclusions.

1. Preprocessing significantly decreases the solver runtime; see Figure 4 (middle) and Figure 5. Due to the short runtimes of the preprocessing, both solvers suing preprocessing $\mathsf{ILP}^{\mathsf{pre}}$ and $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc}}$ score best versus ILP; see Figure 5.

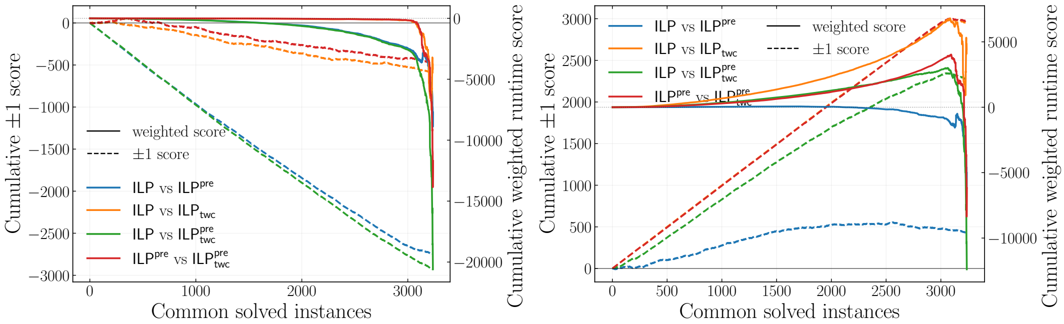

Figure 5 ±1- and weighted scores of solver pairs $\mathcal{S}_1$ versus $\mathcal{S}_2$. The weighted score is $s(\mathcal{S}_1, \mathcal{S}_2) = \text{time}(\mathcal{S}_2) - \text{time}(\mathcal{S}_1)$, where time is either the solver runtime (left) or the total runtime (right). The ±1-score is +1 if $s(\mathcal{S}_1, \mathcal{S}_2) > 0$, and -1 if $s(\mathcal{S}_1, \mathcal{S}_2) < 0$. So, a line going upwards corresponds to $\mathcal{S}_1$ is faster than $\mathcal{S}_2$, and downwards to $\mathcal{S}_1$ is slower than $\mathcal{S}_2$. For each solver pair, the commonly solved instances are sorted separately.

	$1.2 \to 1.3$	$1.3 \to 1.5$	$1.2 \to 1.5$
Cost reduction (in %)	7.87 $[6.59^{37.86}_{0.00}]$	10.76 $[8.97^{39.21}_{0.00}]$	17.63 $[15.42^{57.29}_{0.00}]$

Table 3 Overview of the average relative cost reduction when increasing the detour factor α, with median, minimum (bottom), and maximum (top) in parentheses.

2. The runtime for building the ILP with the TD-separator cuts but without preprocessing is 3.53 s on average and never exceeded 25.64 s; see Figure 4 (left). This indicates that the heuristic makes use of small treewidth, small degree, and sparsity. Preprocessing reduced the building times for both with and without TD-separator cut generation, to roughly 65% for the average time and 90% of the maximum observed time.
3. For instances solved within roughly 100 s, TD-separator cuts do not appear to pay off. For harder instances, however, $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc}}$ dominates; see Figure 4 (right) and Figure 5.
4. While $\mathsf{ILP}_{\mathsf{twc}}$ performs worse than ILP in total, its solver runtime is mostly shorter; see Figure 5 (left). While preprocessing also decreases the solver runtime, finally $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc}}$ scores best against all other solvers, even against $\mathsf{ILP}^{\mathsf{pre}}$ when it comes to solver runtime.

Effect of the preprocessing. Our preprocessing routine (cf. Section 8.1) is fast and effective. Over all of our instances, the routine takes 1.15 seconds on average (median 0.67 second), where the maximum observed time is 12.46 seconds. Yet, the routine significantly reduces the number of unsafe edges and of terminal pairs to 63% and 73% on average, respectively. See Figure 6 for a detailed overview. For our three different safety models A, B, and C, we find only a correlation regarding the reduction of terminal pairs. Comparing safety models A and C, the reduction changes from 50% to 70% for villages, and 62% to 79% for regions.

Cost of detours. Table 3 gives an overview of the costs when the detour factor is increased. Recall that we set the cost of an unsafe edge to its length; hence, our results display the savings of the total upgraded length. We highlight that when moving from $\alpha = 1.2$ to $\alpha = 1.3$, on average 7.87% of the costs are reduced, ranging from no reduction even up to 37.86% in the maximum case. For decision makers, we can often assume that the upper bound on the detour factor lies within a reasonable interval, say $\alpha \in [1.2, 1.3]$. Our results hence indicate

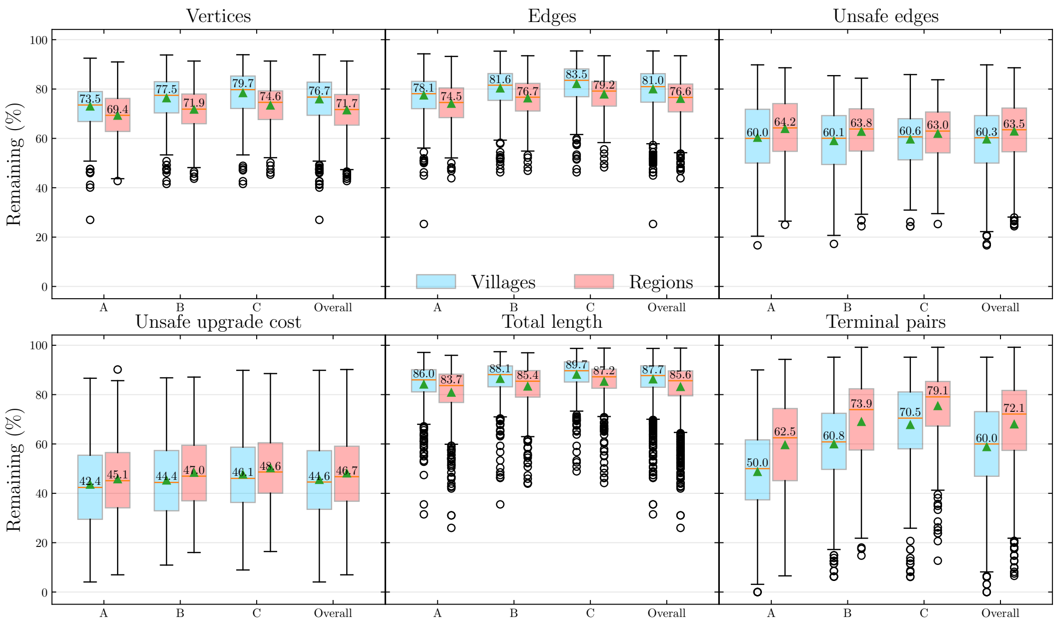

Figure 6 Statistics for the preprocessing routine.

that it is worth computing solutions for several upper bounds on the detour factor in the given interval and compare the trade-offs between detour length and budget.

10 Discussion

On the experimental-modeling side, we randomly sample terminal pairs, whereas real-world demand is likely more structured around hotspots such as schools, supermarkets, or public-transport stops. Similarly, we set the upgrade cost of each street segment to its length. Besides length, however, other factors such as road type or local regulations may affect the construction costs. Incorporating such data would make the experiments more realistic.

On the algorithmic side, our experiments show that preprocessing substantially reduces solver runtimes, making further and improved reduction rules a promising direction. For TD-separator cuts, their effectiveness may depend strongly on the generated candidate sides. Larger heuristic parameters increase model-building time, and adding too many cuts can even slow down the ILP solver (see Section A.1 for details). A better understanding of how to select strong cuts and which candidate sides yield them is an important future direction.

On the theoretical side, it remains open which structural parameters beyond $\mathtt{m_u}$ and $\mathtt{fes} + p$ lead to fixed-parameter tractability, in particular whether $\mathtt{fvs} + p$ or the feedback edge number alone is sufficient.

References

1 Allgemeiner Deutscher Fahrrad-Club e.V. (ADFC). ADFC-Fahrradklima-Test 2024: Rankingliste. https://fahrradklima-test.adfc.de/fileadmin/BV/FKT/Download-Material/Ergebnisse_2024/Download-Element/Rankingliste_FKT_2024.pdf, 2025. Accessed: 2026-06-17.

2 Carolien Beckx, Steven Broekx, Bart Degraeuwe, Bart Beusen, and Luc Int Panis. Limits to active transport substitution of short car trips. *Transportation Research Part D: Transport and Environment*, 22:10–13, 2013. URL: https://www.sciencedirect.com/science/article/pii/S1361920913000424, doi:10.1016/j.trd.2013.03.001.

3 Hans L. Bodlaender and Arie M. C. A. Koster. Treewidth computations i. upper bounds. *Information and Computation*, 208(3):259–275, 2010. doi:10.1016/j.ic.2009.03.008.

4 Geoff Boeing. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017. doi:10.1016/j.compenvurbsys.2017.05.004.

5 Geoff Boeing. A multi-scale analysis of 27,000 urban street networks: Every US city, town, urbanized area, and zillow neighborhood. *Environment and Planning B: Urban Analytics and City Science*, 47(4):590–608, 2020. doi:10.1177/2399808318784595.

6 Yutong Cai, Ghim Ping Ong, and Qiang Meng. Sidewalk-based bicycle path network design incorporating equity in cycling time. *Computer-Aided Civil and Infrastructure Engineering*, 39(20):3063–3082, 2024. doi:10.1111/mice.13240.

7 Alessio Cardillo, Salvatore Scellato, Vito Latora, and Sergio Porta. Structural properties of planar graphs of urban street patterns. *Physical Review E*, 73(6):066107, 2006. doi:10.1103/PhysRevE.73.066107.

8 Eden Chlamtác, Michael Dinitz, Guy Kortsarz, and Bundit Laekhanukit. Approximating spanners and directed steiner forest: Upper and lower bounds. *ACM Transactions on Algorithms*, 16(3):33:1–33:31, 2020. doi:10.1145/3381451.

9 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. *ACM Trans. Algorithms*, 12(3):41:1–41:24, 2016. doi:10.1145/2925416.

10 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

11 Marek Cygan, Fabrizio Grandoni, and Telikepalli Kavitha. On pairwise spanners. In *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 209–220. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.STACS.2013.209.

12 Claus Doll, Clemens Brauer, and Dorien Duffner-Korbee. The potential of cycling for climate protection and livable urban centers and regions: New methods for forecasting supply and demand in germany as a cycling nation up to 2035. Technical report, Fraunhofer Institute for Systems and Innovation Research ISI, Karlsruhe, Germany, May 2024. Prepared on behalf of the Allgemeiner Deutscher Fahrrad-Club (ADFC). URL: https://www.adfc.de/fileadmin/user_upload/Expertenbereich/Politik_und_Verwaltung/Studie_Potenzial_des_Radverkehrs_in_Deutschland/Cycling_Potentials_Germany_Summary_Report.pdf.

13 Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Kernelization lower bounds through colors and ids. *ACM Trans. Algorithms*, 11(2):13:1–13:20, 2014. doi:10.1145/2650261.

14 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, London, 2013. doi:10.1007/978-1-4471-5559-1.

15 Jennifer Duthie and Avinash Unnikrishnan. Optimization framework for bicycle network design. *Journal of Transportation Engineering*, 140(7):04014028, 2014. doi:10.1061/(ASCE)TE.1943-5436.0000690.

16 David Eppstein and Michael T. Goodrich. Studying (non-planar) road networks through an algorithmic lens. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on*

Advances in Geographic Information Systems, pages 16:1–16:10. Association for Computing Machinery, 2008. doi:10.1145/1463434.1463455.

17 European Conference of Ministers of Transport. *Public Transport in Rural Areas.* OECD Publishing, 1984. doi:10.1787/9789282105252-en.

18 Andreas Emil Feldmann and Michael Lampis. Parameterized algorithms for steiner forest in bounded width graphs. *ACM Transactions on Algorithms*, 2025. doi:10.1145/3748724.

19 Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009. doi:10.1016/j.tcs.2008.09.065.

20 Forschungsgesellschaft für Straßen- und Verkehrswesen. *Empfehlungen für Radverkehrsanlagen: ERA*. FGSV 284. FGSV Verlag, Köln, Ausgabe 2010 edition, 2010. See p. 10 for network-level detour-factor recommendations. URL: https://www.fgsv-verlag.de/era.

21 Peter G. Furth, Maaza C. Mekuria, and Hilary Nixon. Network connectivity for low-stress bicycling. *Transportation Research Record: Journal of the Transportation Research Board*, 2587(1):41–49, 2016. doi:10.3141/2587-06.

22 Elisabeth Gassner. The steiner forest problem revisited. *Journal of Discrete Algorithms*, 8(2):154–163, 2010. Selected papers from the 3rd Algorithms and Complexity in Durham Workshop ACiD 2007. URL: https://www.sciencedirect.com/science/article/pii/S1570866709000628, doi:10.1016/j.jda.2009.05.002.

23 Hannah Gorges and Giuliano Mingardo. The potential of active modes to reduce short car trips. a data-driven approach. *Transport Policy*, 168:1–14, 2025. URL: https://www.sciencedirect.com/science/article/pii/S0967070X25001106, doi:10.1016/j.tranpol.2025.03.015.

24 Elena Grigorescu, Nithin Kumar, and Young-San Lin. Approximation algorithms for directed weighted spanners. *Algorithmica*, 88(3):38, 2026. doi:10.1007/s00453-026-01384-6.

25 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11–15, Pasadena, CA USA, 2008.

26 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

27 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

28 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-3-540-68279-0_8.

29 Leon Kellerhals and Tomohiro Koana. Parameterized complexity of geodetic set. *Journal of Graph Algorithms and Applications*, 26(4):401–419, 2022. doi:10.7155/jgaa.00601.

30 Yusuke Kobayashi. NP-hardness and fixed-parameter tractability of the minimum spanner problem. *Theoretical Computer Science*, 746:88–97, 2018. doi:10.1016/j.tcs.2018.06.031.

31 Yusuke Kobayashi. An FPT algorithm for minimum additive spanner problem. In *Proceedings of the 37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*, volume 154 of *LIPIcs*, pages 11:1–11:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. URL: https://doi.org/10.4230/LIPIcs.STACS.2020.11, doi:10.4230/LIPICS.STACS.2020.11.

32 Jisoon Lim, Kevin Dalmeijer, Subhrajit Guhathakurta, and Pascal Van Hentenryck. The bicycle network improvement problem. *Journal of Transportation Engineering, Part A: Systems*, 148(11):04022095, 2022. URL: https://ascelibrary.org/doi/abs/10.1061/JTEPBS.0000742, arXiv:https://ascelibrary.org/doi/pdf/10.1061/JTEPBS.0000742, doi:10.1061/JTEPBS.0000742.

33 Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data. In *22nd International Conference on Database Theory, ICDT 2019,*

Lisbon, Portugal, March 26-28, 2019, LIPIcs, pages 12:1–12:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: `https://doi.org/10.4230/LIPIcs.ICDT.2019.12`, `doi:10.4230/LIPICS.ICDT.2019.12`.

34 Antonio Mauttone, Gonzalo Mercadante, María Rabaza, and Fernanda Toledo. Bicycle network design: Model and solution algorithm. *Transportation Research Procedia*, 27:969–976, 2017. `doi:10.1016/j.trpro.2017.12.119`.

35 Natalie Mueller, David Rojas-Rueda, Tom Cole-Hunter, Audrey de Nazelle, Evi Dons, Regine Gerike, Thomas Götschi, Luc Int Panis, Sonja Kahlmeier, and Mark Nieuwenhuijsen. Health impact assessment of active transportation: A systematic review. *Preventive Medicine*, 76:103–114, 2015. URL: `https://www.sciencedirect.com/science/article/pii/S0091743515001164`, `doi:10.1016/j.ypmed.2015.04.010`.

36 Luis Guillermo Natera Orozco, Federico Battiston, Gerardo Iñiguez, and Michael Szell. Data-driven strategies for optimal bicycle network growth. *Royal Society Open Science*, 7(12):201130, 2020. `doi:10.1098/rsos.201130`.

37 Pekka Oja, Sylvia Titze, Adrian Bauman, Bas de Geus, Patrick Krenn, Bill Reger-Nash, and Thomas Kohlberger. Health benefits of cycling: A systematic review. *Scandinavian Journal of Medicine & Science in Sports*, 21(4):496–509, August 2011. Epub 2011 Apr 18. `doi:10.1111/j.1600-0838.2011.01299.x`.

38 Helen Poltimäe, Merlin Rehema, Janika Raun, and Age Poom. In search of sustainable and inclusive mobility solutions for rural areas. *European Transport Research Review*, 14(1):13, 2022. `doi:10.1186/s12544-022-00536-3`.

39 Donald J. Rose, R. Endre Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976. `doi:10.1137/0205021`.

40 Jessica E. Schoner and David M. Levinson. The missing link: Bicycle infrastructure networks and ridership in 74 US cities. *Transportation*, 41(6):1187–1204, 2014. `doi:10.1007/s11116-014-9538-1`.

41 Kirill Simonov, Farehe Soheil, and Shaily Verma. Finding minimum distance preservers: A parameterized study. *arXiv preprint arXiv:2603.21442*, 2026. URL: `https://arxiv.org/abs/2603.21442`, `arXiv:2603.21442`.

42 SINUS-Institut and Bundesministerium für Verkehr. Fahrrad-Monitor Deutschland 2025 – Erwachsenen- und Jugendbefragung (14–69 Jahre). Technical report, Bundesministerium für Verkehr, 2026. URL: `https://www.bmv.de/SharedDocs/DE/Anlage/StV/fahrrad-monitor-deutschland-2025-langfassung-erwachsenen-jugendbefragung.pdf`.

43 Christoph Steinacker, David-Maximilian Storch, Marc Timme, and Malte Schröder. Demand-driven design of bicycle infrastructure networks for improved urban bikeability. *Nature Computational Science*, 2:655–664, 2022. `doi:10.1038/s43588-022-00318-w`.

44 Michael Szell, Sayat Mimar, Tyler Perlman, Gourab Ghoshal, and Roberta Sinatra. Growing urban bicycle networks. *Scientific Reports*, 12:6765, 2022. `doi:10.1038/s41598-022-10783-y`.

45 Christoph van Dülmen, Martin Šimon, and Andreas Klärner. Transport poverty meets car dependency: A gps tracking study of socially disadvantaged groups in european rural peripheries. *Journal of Transport Geography*, 101:103351, 2022. URL: `https://www.sciencedirect.com/science/article/pii/S0966692322000746`, `doi:10.1016/j.jtrangeo.2022.103351`.

46 Ane Rahbek Vierø and Michael Szell. Network analysis of the danish bicycle infrastructure: Bikeability across urban–rural divides. *Geographical Analysis*, 57(4):616–640, May 2025. URL: `http://dx.doi.org/10.1111/gean.70012`, `doi:10.1111/gean.70012`.

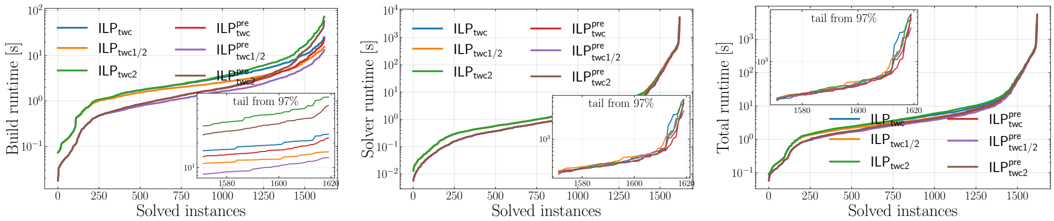

Figure 7 Cactus plots for all region instances solved by each of the three solvers based on different parameters with respect to the TD-separator cut heuristic with and without preprocessing. Presented are building time (left), solver runtime (middle), and total runtime (right). Each inset zooms into the tail of the corresponding cactus plot, i.e., the slowest 3% of the common solved region instances.

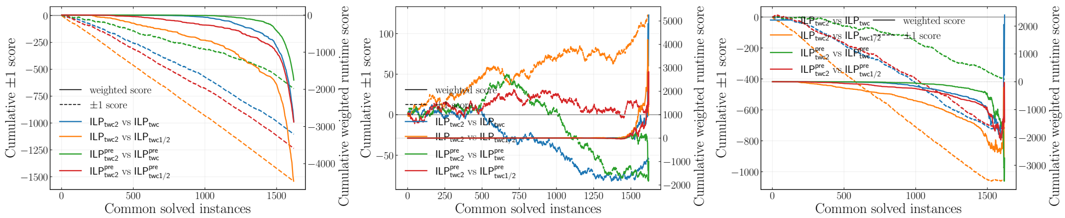

Figure 8 Scores (cf. Figure 5) of all TD-separator cut heuristics for all commonly solved region instances, where the largest setup competes against both the lower two setups, with and without preprocessing. Presented are building time (left), solver runtime (middle), and total runtime (right).

A Appendix

A.1 Further Experimental Results

Recall that for parameters (s^*, N^*, m^*, c^*) of the TD-separator cut heuristic, we selected the setup $\chi_1 = (14, 5000, 10, 20)$ (cf. Section 9.2). We also compared this setup with two other setups $\chi_{1/2} = \frac{1}{2}\cdot\chi_1$ and $\chi_2 = 2\cdot\chi_1$, that is, when halving and doubling the parameters. Denote the corresponding ILP-based algorithms by $\mathsf{ILP}_{\mathsf{twc1/2}}$ and $\mathsf{ILP}_{\mathsf{twc2}}$ without preprocessing, and by $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc1/2}}$ and $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc2}}$ when preprocessing is applied. Figure 7 and Figure 8 show their performances and scores on region instances, respectively. Larger parameters lead to larger model-building times, but does not improve performance overall. Concretely, $\mathsf{ILP}_{\mathsf{twc2}}$ has the largest average model-building time (almost twice the average model-building time of $\mathsf{ILP}_{\mathsf{twc1/2}}$), but the smallest average and median solver runtime (marginally, e.g. regarding the average solver runtime, 28.99 s compared to 29.60 s for $\mathsf{ILP}_{\mathsf{twc1/2}}$) among the commonly solved instances. In particular, larger candidate sets interact nontrivially with preprocessing and can even degrade performance. Concretely, the average total runtime $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc}}$ is the smallest with 28.92 s, yet $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc1/2}}$ achieves the smallest median (2.79 s compared to second place 3.07 s by $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc}}$) and maximum (3373.82 s compared to second place 4780.89 s by $\mathsf{ILP}^{\mathsf{pre}}_{\mathsf{twc2}}$) total runtime among the commonly solved instances.