Stronger ILPs for the Graph Genus Problem

Markus Chimani
Theoretical Computer Science, Osnabrück University, Germany
markus.chimani@uos.de

Tilo Wiedera
Theoretical Computer Science, Osnabrück University, Germany
tilo.wiedera@uos.de

Abstract

The minimum genus of a graph is an important question in graph theory and a key ingredient in several graph algorithms. However, its computation is NP-hard and turns out to be hard even in practice. Only recently, the first non-trivial approach – based on SAT and ILP (integer linear programming) models – has been presented, but it is unable to successfully tackle graphs of genus larger than 1 in practice.

Herein, we show how to improve the ILP formulation. The crucial ingredients are two-fold. First, we show that instead of modeling rotation schemes explicitly, it suffices to optimize over partitions of the (bidirected) arc set A of the graph. Second, we exploit the cycle structure of the graph, explicitly mapping short closed walks on A to faces in the embedding.

Besides the theoretical advantages of our models, we show their practical strength by a thorough experimental evaluation. Contrary to the previous approach, we are able to quickly solve many instances of genus > 1.

2012 ACM Subject Classification Mathematics of computing → Graphs and surfaces; Mathematics of computing → Graph algorithms; Theory of computation → Linear programming

Keywords and phrases algorithm engineering, genus, integer linear programming

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.30

Funding Supported by the German Research Foundation (DFG) grant CH 897/2-2.

1 Introduction

The (orientable) genus of a graph G is the smallest number γ such that G can be embedded on an orientable surface of genus γ. The problem of determining γ is the *graph genus problem*.

Algorithmically exploiting a low graph genus is a vivid research field that spawned many results. It plays a key role for the complexity of certain problems, in particular w.r.t. polynomial-time approximation schemes (PTAS) and fixed parameter tractability (FPT) [8]. Algorithms tailored to achieve faster runtime on planar graphs can often be extended for the bounded genus setting (given a corresponding embedding, see below). For example, a bounded graph genus leads to: linear-time graph isomorphism testing [7]; FPT runtime for dominating set [15]; subexponential FPT runtime for many bidimensional problems, including vertex cover and variants of dominating set [13]; a quasi-PTAS for capacitated vehicle routing [1]; stronger preprocessing for several Steiner problems [29]; and many more.

Apart from such algorithmic uses, the problem is of independent interest in graph theory, where one is concerned with finding the genus of certain graph families (or even single graphs) [3–5, 12, 17, 18, 21, 22, 24, 26, 30–33, 37]. Typically, this involves induction and tedious arguments for the base cases.

Although genus is one of the best-established measures for non-planarity, its NP-hardness was proved relatively late in 1989 by Thomassen [35]. In contrast to other non-planarity measures like crossing number or skewness (or equivalently, maximum planar subgraph), and even compared to the *maximum* graph genus problem, there has been little algorithmic

27th Annual European Symposium on Algorithms (ESA 2019).
Editors: Michael A. Bender, Ola Svensson, and Grzegorz Herman; Article No. 30; pp. 30:1–30:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

progress. In 1988, Thomassen gave an algorithm that computes the genus of a restricted class of graphs in polynomial time [36] (graphs that have so-called LEW-embeddings). The problem is in FPT w.r.t. its natural parameterization due to Robertson and Seymour [34]; Mohar gave a constructive proof for such an FPT-algorithm [25]. Still, the proof spans seven papers of more than 100 pages in total. It was later simplified and extended to graphs with bounded treewidth [19]. It remains open to find a practically feasible FPT-algorithm [27]. No general approximation algorithm is known, although some progress was recently made for restricted variants and closely related problems [6, 20, 23]. The first general exact algorithm – beyond explicit enumeration – was described only recently [2]. It uses ILP/SAT solving to optimize over all rotation systems of the input, but fails to compute genera higher than 1 in practice. Overall, there is no practical, non-trivial algorithm to find low-genus embeddings, not even heuristically. Clearly, this is a main stumbling block for applying any of the genus-based algorithms mentioned above in practice.

Contribution. In this paper, we use the existing ILP as a starting point to develop a practically feasible model. To this end, we establish two novel core ideas. The original approach needs to model the embedding's rotation scheme (see below) explicitly, essentially considering a Hamiltonian cycle problem for each graph node. We show how to only consider partitions of the bidirected arc set to deduce a rotation scheme; among other benefits, this allows us to use fewer variables. The second concept is to explicitly consider the cycle structure of the graph, by examining short, closed walks of the graph. Overall, we obtain both theoretically and practically stronger LP-relaxations. We thus present the first approach to compute the genus of (reasonably sized) graphs in practice, even for genera > 1.

2 Preliminaries

We only need to consider biconnected (since genus is additive over biconnected components), simple graphs where each node has degree ≥ 3: Given an undirected graph, we obtain its *bidirected* counterpart by creating two oppositely directed arcs uv, vu for each edge $\{u, v\}$. Given a graph H, we denote its nodes, edges, and arcs by $V(H)$, $E(H)$, and $A(H)$, respectively. For the input graph G, we may simply write V, E, and A. A *closed walk* c on a bidirected graph G is a set of arcs such that the induced subgraph $G[c]$ is connected and for each node the number of entering arcs is equal to that of leaving arcs. A *cycle* is a closed walk that visits no node more than once. For a node v, we refer to the arcs entering (leaving) v as $\delta_+(v)$ ($\delta_-(v)$, respectively). Let $\mathrm{N}(v)$ be the nodes adjacent to node v. Given two subsets W, U of nodes, we define $W \times_A U := (W \times U) \cap A$ as the arcs from W to U. Given a partition, we refer to its partition sets as *cells*; this term should not be confused with faces of an embedding. For $k \in \mathbb{N}$, let $[k] := \{1, \dots, k\}$.

Graph Embeddings. A *drawing* $\mathcal{D}$ of an undirected graph $G = (V, E)$ on an orientable surface S is a set of points $\mathcal{P}$ and curves $\mathcal{C}$ on S, such that there are bijections $\mathcal{D}_\mathrm{V} \colon V \to \mathcal{P}$ and $\mathcal{D}_\mathrm{E} \colon E \to \mathcal{C}$. We require that for each edge uv the two endpoints of $\mathcal{D}_\mathrm{E}(uv)$ are $\mathcal{D}_\mathrm{V}(u), \mathcal{D}_\mathrm{V}(v)$. A drawing is *crossing free* if for any two edges e, f, $\mathcal{D}_\mathrm{E}(e)$ is disjoint from $\mathcal{D}_\mathrm{E}(f)$ except for common endpoints. We say that a graph is *planar* if it admits a crossing free drawing on the sphere. A crossing free drawing $\mathcal{D}$ induces a cyclic rotation scheme Π of edges around each node, an *embedding*. From a combinatorial perspective, Π fully specifies $\mathcal{D}$ and the genus minimal surface that $\mathcal{D}$ can be drawn on. The regions bounded by edges are the *faces* of Π.

Given an embedding Π, we may identify its faces by tracing them as follows: Consider the bidirected counterpart of G. Starting at a node v, we traverse an arc vw and continue with the cyclically succeeding arc leaving w (w.r.t. the order of the undirected edges in Π). We iterate this until we again arrive at vw, closing the traced face's boundary. Repeating this operation until all edges are traversed exactly once in both directions gives all faces of Π. The traversed arcs of a face form a closed walk. If a node (or edge) appears more than once on the walk, we call this node (or edge) *singular*. The face tracing will allow us to count the number f_Π of faces. Using the Euler characteristic $|V| + f_\Pi - |E| = 2 - 2\gamma$, we are able to determine the lowest-genus (i.e., γ) surface that Π can be drawn on. Algorithmically, we thus ask for an embedding yielding the maximum number of faces.

Integer Linear Programming. A *linear program* (LP) consists of a cost vector $c \in \mathbb{Q}^d$ together with a set of linear inequalities, called *constraints*, that define a polyhedron P in $\mathbb{R}^d$. In polynomial time, we can find a point $x \in P$ that maximizes the *objective value* $c^\intercal x$. Unless P = NP, this is no longer true when restricting x to have integral components; the so-modified problem is an *Integer Linear Program* (ILP). Conversely, the *LP-relaxation* of an ILP is obtained by dropping the integrality constraints on its variables. Typically, there are several ways to reduce a given NP-hard problem to an ILP. These reductions are referred to as *models*. To achieve good practical performance, one aims for a small model where the objective value of the LP-relaxation is close to that of the ILP. This is crucial, as ILP solvers rely on iteratively computing LP-relaxations to obtain dual bounds on the integral objective. When a model has too many constraints to be solved in its entirety, it is often sufficient to use only a reasonably sized constraint subset to achieve a provably optimal solution. Hence, we may dynamically add constraints, during the solving process; this is called *separation*. We say that model A is *at least as strong* as model B, if for all instances, the LP-relaxation's value of model A is no worse than that of B. If there also exists an instance for which A's LP-relaxation yields a tighter value than that of B, we call A *stronger* than B.

Common Foundation and Predecessor Model. As the known model [2], ours will simulate the face tracing algorithm. As such, both models share a common foundation that we borrow from [2]. For the sake of completeness, we repeat its definition below: We use variables x_i that are 1 if and only if face i exists, and variables x_i^a that are 1 if and only if arc a participates in the boundary of face i. Let $\bar{f}$ be an upper bound on the number of faces. We use the shorthands $x(I, A) := \sum_{i \in I, a \in A} x_i^a$ and $x(A) := x([\bar{f}], A)$; thereby, we may omit curly braces when providing sets of cardinality one. Consider the following (by itself insufficient!) model (1a–1e) that we call ILP_{Base}.

$$
\begin{aligned}
\max \quad & \textstyle\sum_{i=1}^{\bar{f}} x_i && && \text{(1a)}\\
\text{s.t.} \quad & 3x_i \le x(i, A) && \forall i \in [\bar{f}] && \text{(1b)}\\
& x(a) = 1 && \forall a \in A && \text{(1c)}\\
& x(i, \delta_-(v)) = x(i, \delta_+(v)) && \forall i \in [\bar{f}],\ v \in V && \text{(1d)}\\
& x_i, x_i^a \in \{0, 1\} && \forall i \in [\bar{f}],\ a \in A && \text{(1e)}
\end{aligned}
$$

Following [2], ILP_{Base} ensures that the faces form a partition of the arc set such that each cell consists of at least three arcs and is a collection of closed walks.

It remains to ensure that the faces are consistent with some rotation scheme of the edges around the nodes. In [2], this is achieved via *predecessor* variables, and we denote the model by ILP_{Pre} in the following. It uses ILP_{Base} and additionally (2a–2d). The idea is to establish a cyclic order of the incident edges of each node by a cut-based sub-ILP known from the traveling salesman problem.

$$x_i^{vw} \geq x_i^{uv} + p^v_{u,w} - 1,\ x_i^{uv} \geq x_i^{vw} + p^v_{u,w} - 1 \qquad \forall i \in [\bar{f}],\ v \in V,\ u,w \in \mathrm{N}(v) : u{\neq}w \tag{2a}$$

$$\textstyle\sum_{\substack{u\in \mathrm{N}(v):\\ u\neq w}} p^v_{w,u} = 1,\ \textstyle\sum_{\substack{u\in \mathrm{N}(v):\\ u\neq w}} p^v_{u,w} = 1 \qquad \forall v \in V,\ \ w \in \mathrm{N}(v) \tag{2b}$$

$$\textstyle\sum_{u\in W, w\in \mathrm{N}(v)\setminus W} p^v_{u,w} \geq 1 \qquad \forall v \in V,\ \ W \subsetneq \mathrm{N}(v) : \varnothing \neq W \tag{2c}$$

$$p^v_{u,w} \in \{0,1\} \qquad \forall v \in V,\ \ u,w \in \mathrm{N}(v) : u \neq w \tag{2d}$$

3 Realizability Model

In contrast to the explicit modeling of an embedding in $\mathrm{ILP}_{\mathrm{Pre}}$, we establish the existence of an embedding *implicitly*. Our *realizability constraints* (see (3) later) require only the variables of $\mathrm{ILP}_{\mathrm{Base}}$. We first need some auxiliary concepts. A graph $G = (V, A)$ is *loopy* if it is directed, connected, each node has at least one incoming arc, and for each arc $uv \in A$ it holds that $\mathcal{K}(uv) := G[\{s : sv \in A\} \cup \{t : ut \in A\}]$ is a complete bipartite graph $K_{k,k}$ for some $k \in \mathbb{N}$, such that each arc is directed from the cell of u to that of v w.r.t. the bipartition.

▶ **Lemma 1.** *Loopy graphs are Hamiltonian, i.e., they contain a cycle traversing all nodes.*

Proof. We first show that any loopy graph allows a cycle cover of pairwise node-disjoint cycles. Assume it does not, consider a collection $\mathcal{C}$ of pairwise node-disjoint cycles covering as many nodes as possible, and let v be an uncovered node. By loopiness, there exists a bipartition that induces two cells, an arc uv, and $\mathcal{K}(uv)$ has a node w (possibly $u = w$) in the cell of u, such that w is not contained in any cycle of $\mathcal{C}$: for any ℓ nodes of one cell in a cycle $c \in \mathcal{C}$, c also contains ℓ nodes of the other cell. As there are only finitely many nodes, we find a new cycle by iterating our argument, i.e., traversing the cycle's arcs in reverse order, thus increasing our cycle cover; a contradiction.

Now, let $\mathcal{C}$ be a node-disjoint cycle cover. For a cycle $c \in \mathcal{C}$, an arc a connecting $V(c)$ with $V \setminus V(c)$ exists by connectivity. Hence, $\mathcal{K}(a)$ contains an arc uv of c and another arc wx of a different cycle $c' \in C$. We join c with c' to a single cycle by replacing uv, wx with ux, wv. Iterating this yields the claim. ◀

▶ **Theorem 2.** *A graph G allows an embedding Π with at least ξ faces if and only if there exists a partition P of $A(G)$, such that*

(a) *P consists of at least ξ cells;*
(b) *every cell of P is a set of pairwise node-disjoint closed walks; and*
(c) *for all subsets $X \subseteq P$, nodes $v \in V(G)$, and non-empty subsets $W \subsetneq \mathrm{N}(v)$, we have $\{wv, vw : w \in W\} \neq \bigcup_{x\in X}\{a \in x : v \text{ incident to } a\}$.*

Before giving the formal proof, let us provide some intuition on property (c): It models that the rotation around each node v is consistent. While in $\mathrm{ILP}_{\mathrm{Pre}}$ constraints (2a–2d) model the rotation explicitly, property (c) ensures the *existence* of a feasible rotation by preventing subcycles. In the rotation around v, any two subsequent arcs must share an edge or a face. Hence, there cannot exist a *proper* subset W of v's neighbors, such that *exactly* the arcs between v and W belong to a subset X of faces. As shown below this is also sufficient.

Proof (of Theorem 2). ($\Longrightarrow$) We obtain P by creating a cell for each face f of Π: it contains exactly the arcs traversed by f. This satisfies (a) and (b). Assume that (c) is not satisfied, i.e., there exist X, v, W (following the above selection rules) such that

$\{wv, vw : w \in W\} = \bigcup_{x \in X}\{a \in x : v \text{ incident to } a\}$. Since W is a proper subset of $\mathrm{N}(v)$, X cannot span all faces incident with v. We choose any face contained (not contained) in X and denote it by f (resp. g). Since Π is an embedding, there exists a sub-sequence of edges incident with v that corresponds to a dual path from f to g. But according to (c), all edges incident with X join two faces in X.

($\Longleftarrow$) We find an embedding Π by forming a face from each component of each cell of P. We establish feasible rotations around each node v in the following way: Let D_v be a directed graph with nodes $\mathrm{N}(v)$ such that $uw \in A(D_v)$ if and only if uv and vw are in the same cell of P (i.e., the arcs could be traversed in that order when tracing the face corresponding to v's component of the cell). A feasible rotation around v corresponds to a Hamiltonian cycle in D_v. We show that D_v is loopy, and hence Hamiltonian by Lemma 1: By construction of D_v, $\mathcal{K}(a)$ is a $K_{k,k}$ for some $k \in \mathbb{N}$, for each $a \in A(D_v)$. By property (b), all nodes of D_v have at least one incoming arc. For disconnected D_v, let W denote the nodes of a single component of D_v, and X the cells that induce $A(D_v[W])$. Then X, v, W contradict property (c). ◀

The above theorem shows that it suffices to optimize over all partitions of arcs into faces. Given a feasible partition (w.r.t. Theorem 2), a corresponding embedding is easily determined in polynomial time following our proof. We can now establish our new model $\mathrm{ILP}_{\mathrm{Real}}$, which extends $\mathrm{ILP}_{\mathrm{Base}}$ with constraints (3). While the former already establishes properties (a) and (b), the latter models property (c): the connectivity of the "local dual graph" D_v around each primal node. Here, index set I corresponds to X from Theorem 2.

$$x(I, v \times_A (\mathrm{N}(v) \setminus W)) \geq 1 + x(I, v \times_A W) - 2|W| \qquad \forall v \in V, I \subseteq [\bar{f}], W \subsetneq \mathrm{N}(v) : W \neq \varnothing \quad (3)$$

Separation. Clearly, it is impractical to add all exponentially many constraints (3) when solving $\mathrm{ILP}_{\mathrm{Real}}$. We use a heuristic separation routine to identify a relevant subset of these constraints. For each LP-feasible solution encountered during the solving process, we proceed as follows: For each node v we check if all variables x_a^i of its incident arcs a are integral. If this holds, but the corresponding D_v is disconnected, we found a new violation of (3).

4 Small Faces

The following approach is inspired by the cycle model for the maximum planar subgraph problem [11]. There, a mapping between small faces and short cycles was used to (only) strengthen the LP-relaxation of another, by itself sufficient, model. Thus, it was possible to mostly disregard longer cycles. In our setting we have to be more careful: On the one hand, we need to consider a far wider range of drawings as we embed on surfaces of higher genera. On the other hand, we have to directly adapt the core model itself; to continue to have a sufficient model, we need to *precisely* encode *all*, even very large, faces. We will model "short" faces by new binary y-variables, one for each specific feasible set of arcs. We continue to use the x-variables for *generic*, i.e., "large" faces. On sparse graphs, this yields a reduction of x-variables, as we may drastically decrease the upper bound on the number of generic faces. Both models, $\mathrm{ILP}_{\mathrm{Pre}}$ and $\mathrm{ILP}_{\mathrm{Real}}$, can be extended in this way.

Let $\mathcal{C}_\chi$ denote the maximal set of closed walks such that each walk's length satisfies property χ. Expanding on $\mathrm{ILP}_{\mathrm{Base}}$, we parameterize our new model $\mathrm{ILP}^D_{\mathrm{Base}}$ by some $D \geq 2$ and obtain (4a–4g) below. We introduce a new decision variable y_c for each $c \in \mathcal{C}_{\leq D}$. Let $y(a) := \sum_{c \in \mathcal{C}_{\leq D} : a \in c} y_c$ and $\bar{f}_{>d}$ any upper bound on the number of faces with length $> d$.

$$
\begin{array}{lllr}
\max & \sum_{i=1}^{\bar{f}_{>D}} x_i + \sum_{c \in \mathcal{C}_{\leq D}} y_c & & \text{(4a)} \\
\text{s.t.} & (D+1)x_i \leq x(i, A) & \forall i \in [\bar{f}_{>D}] & \text{(4b)} \\
& x(a) + y(a) = 1 & \forall a \in A & \text{(4c)} \\
& x(i, \delta_-(v)) = x(i, \delta_+(v)) & \forall i \in [\bar{f}_{>D}], v \in V & \text{(4d)} \\
& \sum_{i=1}^{\bar{f}_{>D}} x_i + \sum_{c \in \mathcal{C}_{\leq D} : |c| > d} y_c \leq \bar{f}_{>d} & \forall d \in \{2, \dots, D\} & \text{(4e)} \\
& x_i \in \{0,1\}, x_i^a \in \{0,1\} & \forall i \in [\bar{f}_{>D}], a \in A & \text{(4f)} \\
& y_c \in \{0,1\} & \forall c \in \mathcal{C}_{\leq D} & \text{(4g)}
\end{array}
$$

Each arc is contained either in one of the generic faces that each form a set of closed walks (as for ILP_{Base}), or in a closed walk c with dedicated variable y_c, see constraints (4c). Generic faces are large, as required by constraints (4b). Constraints (4d) are essentially (1d). Albeit not required for integral solutions, constraints (4e) enforce the previously implicit upper bound on the total number of faces and bound the number of gradually smaller faces.

Predecessor Model. To obtain $\text{ILP}_{\text{Pre}}^{D}$, we add equations (2a–2d) to $\text{ILP}_{\text{Base}}^{D}$, i.e., the same set as for the transition from ILP_{Base} to ILP_{Pre}. Additionally, we require

$$\sum_{c \in \mathcal{C}_{\leq D} : uv, vw \in c} y_c \geq p_{u,w}^v - x([\bar{f}], uv) \qquad \forall v \in V,\ u, w \in \mathrm{N}(v). \tag{5}$$

Similar to (2a), this ensures that if an arc uv is contained in a face modeled by a y-variable, the succeeding arc vw has to be contained in the same face.

Realizability Model. We obtain $\text{ILP}_{\text{Real}}^{D}$ by starting with $\text{ILP}_{\text{Base}}^{D}$ and adding the following constraints to realize property (c) of Theorem 2.

$$
\begin{gathered}
x(I, \overline{A_W^v}) \geq 1 + x(I, A_W^v) + \sum_{c \in \mathcal{C}_{\leq D} : c \cap \overline{A_W^v} = \varnothing} |c \cap A_W^v| y_c - 2|W| \\
\forall v \in V, I \subseteq [\bar{f}], W \subsetneq \mathrm{N}(v) : W \neq \varnothing,\ A_W^v := v \times_A W,\ \overline{A_W^v} := v \times_A (\mathrm{N}(v) \setminus W)
\end{gathered}
\tag{6}
$$

They ensure there is no subset W of arcs at a common node v that is fully assigned to a set (consisting of I and a subset of $\mathcal{C}_{\leq D}$) of face variables that do not have an arc outside of W.

***D*-Hierarchy: Strength of LP-Relaxations.** Clearly $\text{ILP}_{\text{Base}} = \text{ILP}_{\text{Base}}^{2}$. The value of $\bar{f}$ has a strong influence on the dual bounds obtained by LP-relaxations. Hence, we describe how to determine $\bar{f}$ and $\bar{f}_{>D}$ on general graphs. Let $n := |V(G)|$ and $m := |E(G)|$. Let $f_{\text{UB}}(a,b) := \min\{a, b - \mathbb{1}_{a-b=1 \mod 2}\}$, $\bar{f} := f_{\text{UB}}(m-n, \lfloor 2m/3 \rfloor)$, $\bar{f}_{>2} := \bar{f}$, and $\bar{f}_{>d} := \min\{\bar{f}_{>d-1}, \lfloor 2m/(d+1) \rfloor\}$ for $d > 2$. The validity of these bounds follows directly from Eulers formula (assuming non-planar G). We are not aware of any better, general, dual bounds. In the following comparison of LP-relaxations we always assume the above bounds.

▶ **Lemma 3.** *For every graph G, the LP-relaxation of* ILP_{Base} *has objective value $\bar{f}$.*

Proof. The domains (1e) establish $\bar{f}$ as an upper bound. Set $\tilde{x}_a^i = 1/\bar{f}$ and $\tilde{x}^i = 1$ for all $i \in \bar{f}, a \in A$. Clearly $\tilde{x}$ is an LP-feasible solution and achieves the claimed objective. ◀

▶ **Lemma 4.** *For every graph G,* $\text{ILP}_{\text{Base}}^{D}$ *admits an LP-feasible solution with objective value $\bar{f}_{>D}$. If G contains no closed walk of length at most D, this value is optimal.*

Proof. The first claim follows from the LP-feasible solution $\tilde{x}_i^a = 1/\bar{f}_{>D}$ and $\tilde{x}_i = 1$ for all $i \in \bar{f}_{>D}, a \in A$, and $\tilde{y} = 0$. When $\mathcal{C}_{\leq D} = \varnothing$, there are no y variables and the domains (4f) bound the objective from above, yielding the second claim. ◀

▶ **Lemma 5.** *Model* $\mathrm{ILP}_{\mathrm{Base}}^{D+1}$ *is at least as strong as* $\mathrm{ILP}_{\mathrm{Base}}^{D}$ *for any* $D \geq 2$.

Proof. Observe that $\mathrm{ILP}_{\mathrm{Base}}^{D+1}$ generally contains more y- but fewer x-variables than $\mathrm{ILP}_{\mathrm{Base}}^{D}$. Consider an LP-feasible solution $(\hat{x}, \hat{y})$ for $\mathrm{ILP}_{\mathrm{Base}}^{D+1}$. We derive an LP-feasible solution $(\tilde{x}, \tilde{y})$ for $\mathrm{ILP}_{\mathrm{Base}}^{D}$ that achieves no smaller objective value. For notational simplicity, let $\hat{x}_i = \hat{x}_i^a = 0$ for all $i > f_{>D+1}$ and $\beta := \sum_{c \in \mathcal{C}_{=D+1}} \hat{y}_c$. For $\beta = 0$, already $(\hat{x}, \hat{y})$, when interpreted for $\mathrm{ILP}_{\mathrm{Base}}^{D}$, is LP-feasible. Assume $\beta > 0$. Let $\alpha := \bar{f}_{>D} - \sum_{i=1}^{\bar{f}_{>D+1}} \hat{x}_i$ and $\beta_a := \sum_{c \in \mathcal{C}_{=D+1} : a \in c} \hat{y}_c \; \forall a \in A$. From $\alpha < \beta$ it would follow that $\bar{f}_{>D} < \sum_{i=1} \bar{f}_{>D+1} \hat{x}_i + \beta$, a direct contradiction of constraint (4e) for $d = D$ in $\mathrm{ILP}_{\mathrm{Base}}^{D+1}$. Thus, $\alpha \geq \beta$. We define $(\tilde{x}, \tilde{y})$ by $\tilde{x}_i := \hat{x}_i + (1 - \hat{x}_i)\beta/\alpha, \forall i \in [\bar{f}_{>D}]$; $\tilde{x}_i^a := \hat{x}_i^a + (\tilde{x}_i - \hat{x}_i)\beta_a/\beta, \forall i \in [\bar{f}_{>D}], a \in A$; and $\tilde{y}_c := \hat{y}_c, \forall c \in \mathcal{C}_{\leq D}$. The objective value (4a) for $(\tilde{x}, \tilde{y})$ in $\mathrm{ILP}_{\mathrm{Base}}^{D}$ is

$$\begin{aligned}
\textstyle\sum_{i=1}^{\bar{f}_{>D}} \tilde{x}_i + \sum_{c \in \mathcal{C}_{\leq D}} \tilde{y}_c &= \textstyle\sum_{i=1}^{\bar{f}_{>D}} \big(\hat{x}_i + (1 - \hat{x}_i)\beta/\alpha\big) + \sum_{c \in \mathcal{C}_{\leq D}} \hat{y}_c \\
\text{(by } x_i = 0 \;\forall i > \bar{f}_{D+1}\text{)} &= \textstyle\sum_{i=1}^{\bar{f}_{>D+1}} \hat{x}_i + \beta/\alpha \cdot \big(\bar{f}_{>D} - \sum_{i=1}^{\bar{f}_{>D+1}} \hat{x}_i\big) + \sum_{c \in \mathcal{C}_{\leq D}} \hat{y}_c \\
\text{(by def. of } \alpha, \beta\text{)} &= \textstyle\sum_{i=1}^{\bar{f}_{>D+1}} \hat{x}_i + \sum_{c \in \mathcal{C}_{\leq D+1}} \hat{y}_c,
\end{aligned}$$

i.e., equal to that of $(\hat{x}, \hat{y})$ in $\mathrm{ILP}_{\mathrm{Base}}^{D+1}$. Assuming constraint (4b) to be violated, we obtain $(D+1)\tilde{x}_i > (D+1)(\tilde{x}_i - \hat{x}_i) + \sum_{a \in A} \hat{x}_i^a$, since $\sum_{a \in A} \beta_a/\beta = D+1$. This implies $(D+1)\hat{x}_i > \sum_{a \in A} \hat{x}_i^a$, a violation of constraint (4b) already by $(\hat{x}, \hat{y})$. Let us show the feasibility of $(\tilde{x}, \tilde{y})$ w.r.t. constraints (4c) by expanding their left-hand side.

$$\begin{aligned}
\tilde{x}(a) + \tilde{y}(a) &= \textstyle\sum_{i=1}^{f_{>D}} \big(\hat{x}_i^a + (1 - \hat{x}_i)\beta_a/\alpha\big) + \sum_{c \in \mathcal{C}_{\leq D} : a \in c} \hat{y}_c \\
\text{(by def. of } \alpha\text{)} &= \textstyle\sum_{i=1}^{\bar{f}_{>D+1}} \hat{x}_i^a + \beta_a + \sum_{c \in \mathcal{C}_{\leq D} : a \in c} \hat{y}_c \\
\text{(by def. of } \beta_a\text{)} &= \textstyle\sum_{i=1}^{\bar{f}_{>D+1}} \hat{x}_i^a + \sum_{c \in \mathcal{C}_{\leq D+1} : a \in c} \hat{y}_c \;\text{(by feasibility of (4c) in } (\hat{x}, \hat{y})\text{)} = 1
\end{aligned}$$

Since $\mathcal{C}_{=D+1}$ contains only closed walks, we have $\sum_{u \in \mathrm{N}(v)} (\beta_{uv} - \beta_{vu}) = 0$ for all nodes v. Constraints (4d) hold, as we see by expanding their left-hand side:

$$\begin{aligned}
\textstyle\sum_{vu \in A} \tilde{x}_i^{vu} &= \textstyle\sum_{vu \in A} \hat{x}_i^{vu} + (\tilde{x}_i - \hat{x}_i)/\beta \cdot \sum_{vu \in A} \beta_{vu} \\
\text{(by (4d) in } \mathrm{ILP}_{\mathrm{Base}}^{D+1} \text{ and the above)} &= \textstyle\sum_{uv \in A} \hat{x}_i^{uv} + (\tilde{x}_i - \hat{x}_i)/\beta \cdot \sum_{uv \in A} \beta_{uv} = \sum_{uv \in A} \tilde{x}_i^{uv}
\end{aligned}$$

Constraints (4e) maintain their slack, as the first term increases by $\sum_{i=1}^{\bar{f}_{>D}} (\tilde{x}_i - \hat{x}_i) = \beta$ while the second decreases by β. Clearly, $\tilde{x}_i \geq \hat{x}_i$ and $\tilde{x}_i^a \geq \hat{x}_i^a$. By $\alpha \geq \beta$ we have $\tilde{x}_i \leq 1$. By (4c) we have $\hat{x}_i^a + \beta_a \leq 1$. Thus, $\tilde{x}_i^a > 1$ would imply $\tilde{x}_i - \hat{x}_i > \beta$. Clearly, we keep $0 \leq \tilde{y}_c \leq 1$. ◀

▶ **Theorem 6.** *Model* $\mathrm{ILP}_{\mathrm{Base}}^{D+1}$ *is stronger than* $\mathrm{ILP}_{\mathrm{Base}}^{D}$ *for any* $D \geq 2$.

Proof. Restricting ourselves to dense graphs of girth $> D + 1$, the claim immediately follows from Lemmata 3–5. An example of such graphs are the complete graphs on D nodes, where we subdivide each edge D times. They have girth $3(D + 1)$ and are dense enough such that the respective bounds differ: $f_{>D} > f_{>D+1}$. We note that there are also dense graphs with high girth that do not allow any general preprocessing techniques. ◀

5 Additional Tuning (“Add-Ons”)

In addition to the new models described above, there is a set of supplemental constraints that may be applied to several of these models. We discuss them in alphabetical order.

Arc-Face. We may require the below trivial constraints explicitly.

$$x_i \geq x_i^a \qquad \forall a \in A, i \in [\bar{f}] \tag{7}$$

Branching Rule. To facilitate the fast generation of strong primal bounds, we may initially restrict the solution space to explicitly modeled faces, e.g., by branching on

$$\textstyle\sum_{i \in \bar{f}_{>D}} x_i \overset{?}{=} 0. \tag{8}$$

deg 3-Model. There are only two possible rotations around any degree-3 node v. In ILP$_{\mathrm{Pre}}$, this can be modeled by a single binary variable for v and alternative constraints, partially replacing (2a–2d), as discussed in [2]. The same holds for ILP$^D_{\mathrm{Pre}}$, and we use this improvement in our benchmarks.

Long Faces. In several cases we can establish lower bounds on the length of faces modeled by the x-variables. Let $\mathrm{s}(v,w)$ denote the length of the shortest path between nodes v and w.

▶ **Lemma 7.** *For arcs uv, wx that traverse the same face f, we have* $\mathrm{s}(v,w)+\mathrm{s}(x,u)+2 \leq |f|$.

Proof. Tracing any such face f yields a path from v to w that neither contains arc uv (it may contain arc vu) nor arc vx. Similarly, an arc-disjoint path from x to u must exist in f. The total length of these paths is lower bounded by $\mathrm{s}(v,w) + \mathrm{s}(x,u)$. ◀

▶ **Lemma 8.** *Any face with a singular edge contains at least eight arcs and this is tight.*

Proof. Let uv denote the edge that is traversed in both directions when tracing face f. If tracing f would additionally yield a path from u to v that does not contain uv, the tracing would similarly yield a path from v to u that does not contain vu. This contradicts the assumption since f would contain two oppositely directed, closed walks that form two separate faces. Hence, there exist two arc-disjoint closed walks on the boundary of f, one for each node u, v. Since there are no deg-1 nodes in a biconnected graph, any subcycle in a face requires at least three arcs and the claim follows. Considering a genus-1 embedding of the K_4 we can see that it indeed contains such a face of length eight. ◀

▶ **Lemma 9.** *Any face with a singular node contains at least six arcs and this is tight.*

Proof. If the face f also traverses an edge twice, the bound follows from Lemma 8. Otherwise, the doubly traversed node has at least four arcs in f, belonging to pairwise different edges, and hence four incident nodes. A closed walk on this $K_{1,4}$ requires at least two additional arcs and the claimed bound follows. A face of length six can be observed in a genus-1 embedding of the following graph: Take two copies of the K_5, remove one edge each, join the graphs by identifying two deg-3 nodes, and add a new edge between the remaining two deg-3 nodes. ◀

Let $\ell_{uv,wx} := \max\{\mathrm{s}(v,w)+\mathrm{s}(x,u)+2, 6\cdot\mathbb{1}_{k=3}, 8\cdot\mathbb{1}_{k=2}\}$ with $k := |\{u,v,w,x\}|$. Lemmata 7–9 yield the following constraints. When using them in our benchmarks, we separate them.

$$\ell_{uv,wx}(x_i^{uv} + x_i^{wx} - 1) \leq x(i,A) \qquad \forall i \in [\bar{f}_{>D}],\ uv, wx \in A : uv \neq wx \tag{9}$$

Objective Parity. All above ILPs maximize the number f of faces and deduce γ via Euler's formula $n + f - m = 2 - 2\gamma$. Thus the parity of f is fixed. This gives room for improved bounding and cutting by the ILP solver. Using a new variable $z \in \mathbb{N}$ we may demand

$$(m - n \bmod 2) + 2z = \textstyle\sum_{i=1}^{\bar{f}_{>D}} f_i + \sum_{c \in \mathcal{C}_{\leq D}} y_c. \tag{10}$$

Symmetry Breaking. For ILP$_{\text{Pre}}$ it was observed in [2] that symmetry breaking does not seem to pay off. However, ILP$_{\text{Pre}}$ only solves genus-1 instances in practice (see below). Symmetry breaking may hinder heuristics from identifying trivially optimal solutions, but may be beneficial for harder instances. The approach in [2] enforces that face i has at most as many arcs as face $i + 1$. However, there are typically many faces with the same length in a low-genus embedding. We consider breaking symmetries by restricting the set of faces that may contain a given arc. Let $\prec$ denote an arbitrary but fixed order on the arc set A.

$$y(a) + x(\{1, \ldots, \ell\}, a) \geq 1 - x(\ell, \{a' \in A : a' \prec a\}) \qquad \forall \ell \in [\bar{f}], a \in A \tag{11}$$

These constraints ensure that any arc is contained either in an explicitly modeled closed walk or in the lowest-indexed face that it can be placed into. For ILP$_{\text{Pre}}$ and ILP$_{\text{Real}}$, i.e., when there are no explicitly modeled closed walks, we simply set $y(a) = 0$.

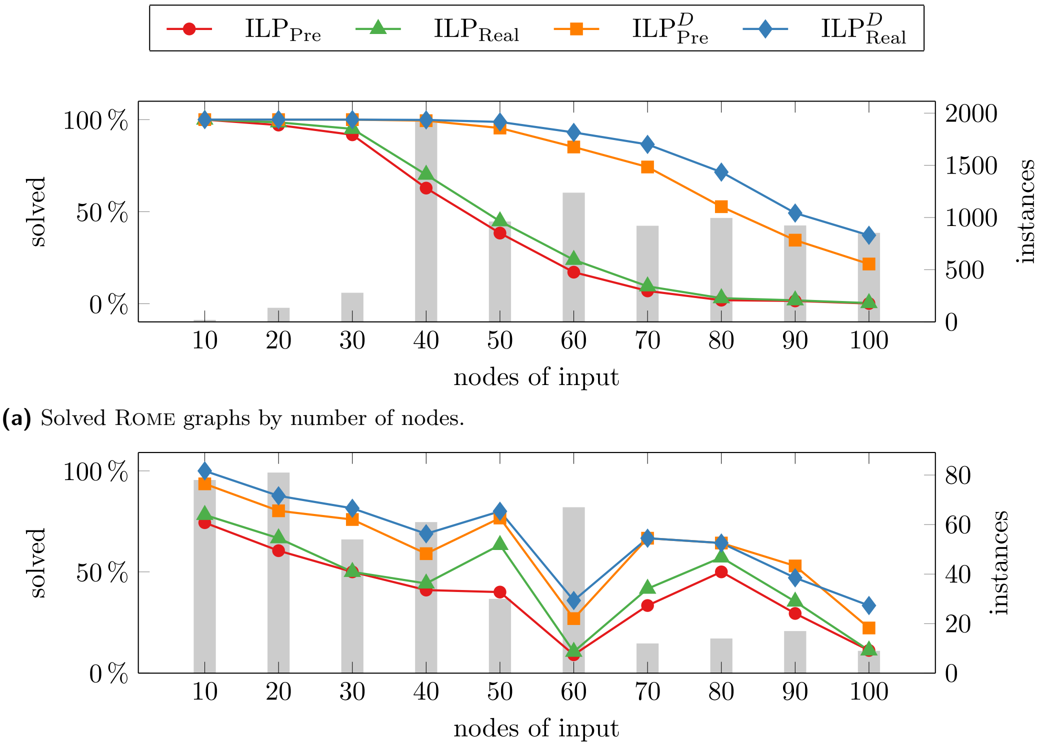
6 Experiments

All algorithms are implemented in `C++`, compiled with `gcc` 6.3.0, and use the OGDF (snapshot 2018-03-28) [9]. We use SCIP 6.0.0 [16] for solving ILPs, with CPLEX 12.8.0 as the underlying LP solver. Each computation uses a single physical core of a Xeon Gold 6134 CPU (3.2 GHz) with a memory speed of 2666 MHz. We employ a time limit of 10 minutes and a memory limit of 8 GB per computation. All instances and results, giving runtime and genus (if solved), are available for download at `http://tcs.uos.de/research/min-genus`. In our experiments, we increase parameter D – separately on each graph – until we obtain at least 1000 y-variables. As they are not required for integral solutions, we omit constraints (4e) by default.

Instance Sets. We consider the 423 and 8249 non-planar graphs of the two established real-world sets North [28] and Rome [14], respectively. In addition, we use the set of 600 Expander graphs, as established in [10, 11] for a related non-planarity measure (skewness): there are 20 graphs for each feasible parameterization $(|V(G)|, \Delta) \in \{10, 20, 30, 50, 100\} \times \{4, 6, 10, 20, 40\}$, where Δ denotes the node degree.

Discussion of SAT-based algorithms. In [2], the SAT-based approach was faster than the ILP-based one. However, we do not need to directly compare with it.

Both previous approaches solve only instances with genus ≤ 1 in practice. Since the respective dual bound is trivially given by planarity testing (and enforced in all previous models), the runtime difference can be attributed to the SAT-solver quickly finding a satisfying solution. In contrast, standard primal heuristics of ILP-sovlers are weaker, and the comparably time-consuming LP-relaxations are rarely profitable. However, w.r.t. success-rate, SAT is only marginally in the lead, if at all: on the Rome graphs, the ILP and SAT solve 2595 and 2667 instances, respectively. For North, "the success-rates of both approaches are [...] comparable" [2].



(a) Solved ROME graphs by number of nodes.

(b) Solved NORTH graphs by number of nodes.

Figure 1 Detailed success-rates of algorithms on established benchmark sets. We provide the relative number of solved instances over the number of nodes, clustered to the nearest multiple of 10. The gray bars denote the number of instances in each cluster.

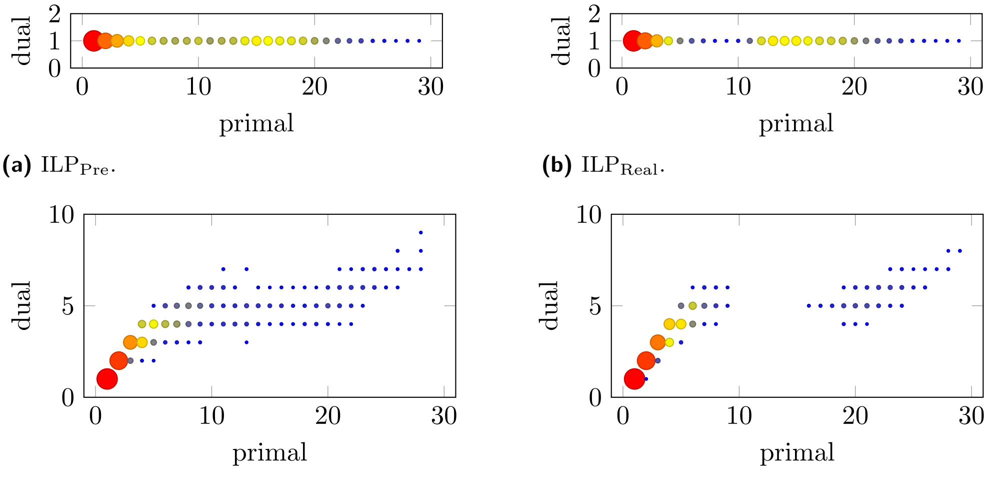
Since we can neither employ separation nor LP-relaxations in the setting of SAT-solvers, there also is no immediate way of using our strengthening results for SAT-based algorithms. We will see that the new ILP variants clearly dominate the SAT-based variants; e.g., we solve up to 6797 ROME instances.

Results. The experiments confirm that our new model is not only theoretically stronger but also better in practice: Using $\mathrm{ILP}^{D}_{\mathrm{Real}}$, we are now able to solve 82% instead of just 28% of the ROME graphs, cf. Table 1. Depending on the instance set, we achieve an average speed-up of factor 82 to 248. In [2], only graphs with genus 1 (and not all of them) could be solved. Surprisingly, and in contrast to the observations made in [2], the deg 3-model does not perform better than the respective base variant: SCIP's built-in preprocessing reduces the variable space to essentially the same dimension as obtained when manually applying the deg 3-model (while possibly retaining some additional information that helps in the solving process). Also somewhat to our surprise, none of the add-ons (8–11) pay off *reliably.*

Taking a closer look at the number of solved instances (Figure 1 and Table 2), we see that – on average – $\mathrm{ILP}^{D}_{\mathrm{Real}}$ is superior to all other variants for any graph size. We now solve real-world instances with non-trivial dual bounds, i.e., when the genus is > 1, e.g., we have solved a genus-7 instance on ROME and even a genus-21 instance on NORTH. We see very clearly, in particular on ROME, that we may order the models $\mathrm{ILP}_{\mathrm{Pre}}$, $\mathrm{ILP}_{\mathrm{Real}}$, $\mathrm{ILP}^{D}_{\mathrm{Pre}}$, $\mathrm{ILP}^{D}_{\mathrm{Real}}$ by increasing success rate. This means that, independent on whether we apply the small-faces model extension or not, the realizability model is more successful than

Table 1 Average success-rate s and runtime t on each instance set. Considering the runtime, we restrict the instances to those solved by all variants. Of the graphs solved by ILP_{Pre}, all variants solved at least 94% (ROME), 99% (NORTH), and 100% (EXPANDER). Particularly, algorithms based on $\text{ILP}^{D}_{\text{Base}}$ always solved all of the instances solved by ILP_{Pre}.

	ROME		NORTH		EXPANDER	
	s [%]	t [s]	s [%]	t [s]	s [%]	t [s]
ILP_{Pre}	27.79	190.23	45.86	139.09	5.26	58.29
ILP_{Pre} + deg 3	27.94	186.31	47.52	111.50	5.26	58.53
ILP_{Real}	**31.85**	**70.57**	**50.83**	**25.44**	**5.53**	**11.33**
$\text{ILP}^{D}_{\text{Pre}}$	73.08	2.92	67.14	12.21	17.37	2.24
$\text{ILP}^{D}_{\text{Pre}}$ + deg 3	68.09	3.86	65.01	12.47	17.37	2.25
$\text{ILP}^{D}_{\text{Real}}$	**81.65**	**0.91**	**73.52**	**7.26**	**23.95**	**0.95**
$\text{ILP}^{D}_{\text{Real}}$ + branch rule (8)	76.41	0.86	73.05	7.29	21.05	0.80
$\text{ILP}^{D}_{\text{Real}}$ + all symmetries (11)	81.56	0.91	73.52	7.27	23.95	0.95
$\text{ILP}^{D}_{\text{Real}}$ + sepa. symmetries (11)	81.59	0.91	73.76	7.26	23.95	0.94
$\text{ILP}^{D}_{\text{Real}}$ + sepa. long faces (9)	81.16	0.95	72.81	7.27	22.89	0.81
$\text{ILP}^{D}_{\text{Real}}$ + all #faces cons. (4e)	81.57	**0.75**	**74.00**	6.79	23.68	0.86
$\text{ILP}^{D}_{\text{Real}}$ + sepa. #faces cons. (4e)	81.60	1.13	**74.00**	7.02	23.95	1.05
$\text{ILP}^{D}_{\text{Real}}$ + parity model (10)	**82.33**	1.25	73.29	**1.11**	22.89	1.68
$\text{ILP}^{D}_{\text{Real}}$ + all arc-face cons. (7)	75.43	0.98	71.39	7.35	18.95	**0.71**
$\text{ILP}^{D}_{\text{Real}}$ + sepa. arc-face cons. (7)	81.71	1.48	73.76	7.28	23.42	0.76
$\text{ILP}^{D}_{\text{Real}}$ + (10) + sepa. (4e,7)	**82.40**	1.37	74.00	1.20	22.63	1.65
$\text{ILP}^{D}_{\text{Real}}$ + (8) + (10) + sepa. (4e,7)	78.07	1.19	**75.65**	1.18	21.58	1.62



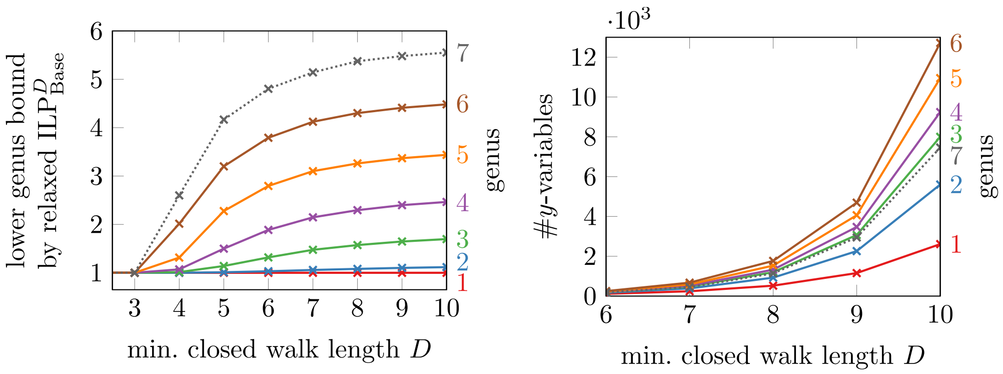
(a) ILP_{Pre}.

(b) ILP_{Real}.

(c) $\text{ILP}^{D}_{\text{Pre}}$.

(d) $\text{ILP}^{D}_{\text{Real}}$.

Figure 2 Final primal vs. dual bounds on the genus, generated by algorithmic variants on ROME (without any add-ons). Color and size indicate the number of instances with the respective bounds. We note that these bounds do not apply to the values of the formal objective value, i.e., the number of attained faces, but to the genus, which allows a more sensible comparison. Note that without the small faces extension, neither ILP_{Pre} nor ILP_{Real} obtains lower bounds > 1 (i.e., only trivial ones).



(a) genus bound of LP-relaxation.

(b) number of generated y-variables.

Figure 3 Average values of ILP$^D_{\text{Base}}$ on the solved ROME graphs, depending on the maximum length D of explicitly modeled cycles and the genus of the graph. Note that we only solved one instance with genus 7 on ROME.

Table 2 Number of solved instances in the EXPANDER set for selected variants without add-ons.

# nodes	10		20			30		≥ 50
node degree	4	6	4	6	10	4	6-20	4–40
ILP$_{\text{Pre}}$	20	0	0	0	0	0	0	0
ILP$_{\text{Real}}$	20	1	0	0	0	0	0	0
ILP$^D_{\text{Pre}}$	20	20	20	0	0	6	0	0
ILP$^D_{\text{Real}}$	20	20	20	18	0	13	0	0

the predecessor model. The most progress, however, is achieved by activating the small-faces model extension ILP$^D_{\text{Base}}$. As the shapes of the success-rate curves demonstrate, it benefits both underlying models roughly equally. In particular, we see (cf. also Figure 2 which shows the final bounds of our core variants on ROME) that even ILP$_{\text{Real}}$, like ILP$_{\text{Pre}}$, can only solve genus 1 instances. This is in accordance with Lemma 3, i.e., that the LP-relaxation of ILP$_{\text{Base}}$ always yields value $\bar{f}$. Nonetheless, the success-rates 46% and 51% on NORTH for ILP$_{\text{Pre}}$ and ILP$_{\text{Real}}$, respectively, demonstrate that ILP$_{\text{Pre}}$ is far from solving all toroidal instances. More complex instances require the small-faces extension ILP$^D_{\text{Base}}$. This is also reflected by the root relaxations of ILP$^D_{\text{Base}}$ for different values of D, cf. Figure 3. Consistent with theory, increasing the minimum length D leads to stronger LP-relaxations also in practice, but may drastically increase the number of variables. Interestingly, generating only triangles, i.e., $D = 3$, yields only a very slight increase on the average dual bound compared to ILP$_{\text{Base}}$ on ROME, possibly caused by the graphs' sparsity.

Genera in Graph Theory. Our new approach allows us to confirm results from literature, all with non-trivial dual bounds: In 2015, the circulants of genus ≤ 2 were characterized [12]. Thereby, the authors need to show that 12 specific graphs have genus ≥ 3. For these arguments alone, they require about nine pages, supplemented by several hours of computation. Using ILP$^D_{\text{Real}}$, we are able to confirm these results (and compute the respective genera) in a matter of seconds without employing any graph-specific theory. Before, using ILP$_{\text{Pre}}$, the arguably hardest case $C_{11}(1, 2, 4)$ required 180 hours [2]. In 2005, a full paper was dedicated to showing that the Gray graph has genus 7 [24]. Our tool confirms this result within 42 hours. Similarly, we confirm a result from 1989 [4] in 250 seconds: the group that is the semidirect product of $\mathbb{Z}_9$ with $\mathbb{Z}_3$ has genus 4 (the genus of group Γ is the smallest genus of a Caley graph of Γ).

7 Conclusion

We have presented novel ILP models for the graph genus problem, proved their theoretical strength, the existence of a hierarchy of ever stronger LP-relaxations, and positively evaluated them in practice, e.g., by solving 82% instead of the previous 28% of the Rome instances. We are now able to solve real-world instances with genera up to 21. This is in stark contrast to the previous models that – on the same set of instances – succeeded only on toroidal graphs.

It remains open whether even stronger models can be found by a more careful examination of the face structure. What additional properties of the embedding may be modeled? Is it possible to better exploit singular nodes or edges, particularly when they are adjacent? Further, we expect that our algorithms would benefit from strong primal heuristics but we are not aware of any general such algorithms. Currently, optimal dual bounds are often identified long before an optimal solution is found.

References

1 Amariah Becker, Philip N. Klein, and David Saulpic. A Quasi-Polynomial-Time Approximation Scheme for Vehicle Routing on Planar and Bounded-Genus Graphs. In *ESA 2017*, pages 12:1–12:15, 2017. doi:10.4230/LIPIcs.ESA.2017.12.

2 Stephan Beyer, Markus Chimani, Ivo Hedtke, and Michal Kotrbcík. A Practical Method for the Minimum Genus of a Graph: Models and Experiments. In *SEA 2016*, LNCS 9685, pages 75–88, 2016. doi:10.1007/978-3-319-38851-9_6.

3 C. Paul Bonnington and Tomaz Pisanski. On the orientable genus of the cartesian product of a complete regular tripartite graph with an even cycle. *Ars Comb.*, 70, 2004.

4 Matthew G. Brin, David E. Rauschenberg, and Craig C. Squier. On the genus of the semidirect product of $\mathbb{Z}_9$ by $\mathbb{Z}_3$. *Journal of Graph Theory*, 13(1):49–61, 1989. doi:10.1002/jgt.3190130108.

5 Matthew G. Brin and Craig C. Squier. On the Genus of $\mathbb{Z}_3 \times \mathbb{Z}_3 \times \mathbb{Z}_3$. *Eur. J. Comb.*, 9(5):431–443, 1988. doi:10.1016/S0195-6698(88)80002-7.

6 Chandra Chekuri and Anastasios Sidiropoulos. Approximation Algorithms for Euler Genus and Related Problems. In *FOCS 2013*, pages 167–176, 2013. doi:10.1109/FOCS.2013.26.

7 Jianer Chen. A Linear-Time Algorithm for Isomorphism of Graphs of Bounded Average Genus. *J. Disc. Math.*, 7(4):614–631, 1994. doi:10.1137/S0895480191196769.

8 Jianer Chen, Iyad A. Kanj, Ljubomir Perkovic, Eric Sedgwick, and Ge Xia. Genus characterizes the complexity of certain graph problems: Some tight results. *J. Comput. Syst. Sci.*, 73(6):892–907, 2007. doi:10.1016/j.jcss.2006.11.001.

9 Markus Chimani, Carsten Gutwenger, Mike Juenger, Gunnar W. Klau, Karsten Klein, and Petra Mutzel. The Open Graph Drawing Framework (OGDF). In Roberto Tamassia, editor, *Handbook on Graph Drawing and Visualization*, pages 543–569. Chapman and Hall/CRC, 2013. URL: https://crcpress.com/Handbook-of-Graph-Drawing-and-Visualization/Tamassia/9781584884125.

10 Markus Chimani, Ivo Hedtke, and Tilo Wiedera. Exact Algorithms for the Maximum Planar Subgraph Problem: New Models and Experiments. In *SEA 2018*, pages 22:1–22:15, 2018. doi:10.4230/LIPIcs.SEA.2018.22.

11 Markus Chimani and Tilo Wiedera. Cycles to the Rescue! Novel Constraints to Compute Maximum Planar Subgraphs Fast. In *ESA 2018*, LIPIcs 112, pages 19:1–19:14, 2018. doi:10.4230/LIPIcs.ESA.2018.19.

12 Marston Conder and Ricardo Grande. On Embeddings of Circulant Graphs. *Electr. J. Comb.*, 22(2):P2.28, 2015. URL: http://www.combinatorics.org/ojs/index.php/eljc/article/view/v22i2p28.

13 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Dimitrios M. Thilikos. The Bidimensional Theory of Bounded-Genus Graphs. *Disc. Math.*, 20(2):357–371, 2006. doi:10.1137/040616929.

14 Giuseppe Di Battista, Ashim Garg, Giuseppe Liotta, Roberto Tamassia, Emanuele Tassinari, and Francesco Vargiu. An Experimental Comparison of Four Graph Drawing Algorithms. *Comput. Geom.*, 7:303–325, 1997. doi:10.1016/S0925-7721(96)00005-3.

15 John A. Ellis, Hongbing Fan, and Michael R. Fellows. The dominating set problem is fixed parameter tractable for graphs of bounded genus. *J. Algorithms*, 52(2):152–168, 2004. doi:10.1016/j.jalgor.2004.02.001.

16 Ambros Gleixner, Michael Bastubbe, Leon Eifler, Tristan Gally, Gerald Gamrath, Robert Lion Gottwald, Gregor Hendel, Christopher Hojny, Thorsten Koch, Marco E. Lübbecke, Stephen J. Maher, Matthias Miltenberger, Benjamin Müller, Marc E. Pfetsch, Christian Puchert, Daniel Rehfeldt, Franziska Schlösser, Christoph Schubert, Felipe Serrano, Yuji Shinano, Jan Merlin Viernickel, Matthias Walter, Fabian Wegscheider, Jonas T. Witt, and Jakob Witzig. The SCIP Optimization Suite 6.0. Technical report, Optimization Online, July 2018. URL: http://www.optimization-online.org/DB_HTML/2018/07/6692.html.

17 David A. Hoelzeman and Saïd Bettayeb. On the genus of star graphs. *IEEE Transactions on Computers*, 43(6):755–759, 1994. doi:10.1109/12.286309.

18 Mark Jungerman and Gerhard Ringel. The genus of the n-octahedron: Regular cases. *J. Graph Theory*, 2(1):69–75, 1978. doi:10.1002/jgt.3190020109.

19 K. Kawarabayashi, B. Mohar, and B. Reed. A Simpler Linear Time Algorithm for Embedding Graphs into an Arbitrary Surface and the Genus of Graphs of Bounded Tree-Width. In *FOCS 2008*, pages 771–780, 2008. doi:10.1109/FOCS.2008.53.

20 Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Beyond the Euler Characteristic: Approximating the Genus of General Graphs. In *STOC 2015*, pages 675–682, 2015. doi:10.1145/2746539.2746583.

21 Michal Kotrbcík and Tomaz Pisanski. Genus of the Cartesian Product of Triangles. *Electr. J. Comb.*, 22(4):P4.2, 2015. URL: http://www.combinatorics.org/ojs/index.php/eljc/article/view/v22i4p2.

22 Valentas Kurauskas. On the genus of the complete tripartite graph $K_{n,n,1}$. *Disc. Math.*, 340(3):508–515, 2017. doi:10.1016/j.disc.2016.09.017.

23 Yury Makarychev, Amir Nayyeri, and Anastasios Sidiropoulos. A Pseudo-approximation for the Genus of Hamiltonian Graphs. In *APPROX 2013*, pages 244–259, 2013. doi:10.1007/978-3-642-40328-6_18.

24 Dragan Marusic, Tomaz Pisanski, and Steve Wilson. The genus of the GRAY graph is 7. *Eur. J. Comb.*, 26(3-4):377–385, 2005. doi:10.1016/j.ejc.2004.01.015.

25 Bojan Mohar. A Linear Time Algorithm for Embedding Graphs in an Arbitrary Surface. *J. Disc. Math.*, 12(1):6–26, 1999. doi:10.1137/S089548019529248X.

26 Bojan Mohar, Tomaz Pisanski, Martin Skoviera, and Arthur T. White. The cartesian product of three triangles can be embedded into a surface of genus 7. *Disc. Math.*, 56(1):87–89, 1985. doi:10.1016/0012-365X(85)90197-9.

27 Wendy Myrvold and William Kocay. Errors in graph embedding algorithms. *J. Comp. and Sys. Sciences*, 77(2):430–438, 2011. doi:10.1016/j.jcss.2010.06.002.

28 Stephen C. North. 5114 directed graphs, 1995. Manuscript.

29 Marcin Pilipczuk, Michal Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Network Sparsification for Steiner Problems on Planar and Bounded-Genus Graphs. *ACM Trans. Algorithms*, 14(4):53:1–53:73, 2018. doi:10.1145/3239560.

30 Gerhard Ringel. Das Geschlecht des vollständigen paaren Graphen. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 28(3):139–150, 1965. doi:10.1007/BF02993245.

31 Gerhard Ringel. On the genus of the graph $K_n \times K_2$ or the n-prism. *Disc. Math.*, 20:287–294, 1977. doi:10.1016/0012-365X(77)90067-X.

32 Gerhard Ringel and J. W. T. Youngs. SOLUTION OF THE HEAWOOD MAP-COLORING PROBLEM. *PNAS USA*, 60(2):438–445, 1968. doi:10.1073/pnas.60.2.438.

33 Gerhard Ringel and J. W. T. Youngs. Das Geschlecht des vollständigen dreifärbbaren Graphen. *Commentarii Mathematici Helvetici*, 45(1):152–158, 1970. doi:10.1007/BF02567322.

34 Neil Robertson and Paul D. Seymour. Graph minors. VIII. A Kuratowski theorem for general surfaces. *J. Comb. Theory, Ser. B*, 48(2):255–288, 1990. doi:10.1016/0095-8956(90)90121-F.

35 Carsten Thomassen. The graph genus problem is NP-complete. *J. Algorithms*, 10(4):568–576, 1989. doi:10.1016/0196-6774(89)90006-0.

36 Carsten Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. Comb. Theory, Series B*, 48(2):155–177, 1990. doi:10.1016/0095-8956(90)90115-G.

37 Arthur T. White. The genus of the complete tripartite graph $K_{mn,n,n}$. *J. Comb. Theory*, 7(3):283–285, 1969. doi:10.1016/S0021-9800(69)80027-X.