Gender-Aware Facility Location in Multi-Gender World

Valentin Polishchuk
Communications and Transport Systems, ITN, Linköping University, Sweden
valentin.polishchuk@liu.se

Leonid Sedov
Communications and Transport Systems, ITN, Linköping University, Sweden
leonid.sedov@liu.se

Abstract

This interdisciplinary (GS and CS) paper starts from considering the problem of locating restrooms or locker rooms in a privacy-preserving way, i.e., so that while following the path to one's room, one cannot peek into another room; the rooms are meant for a multitude of genders, one room per gender. We then proceed to showing that gender inequality (non-uniform treatment of genders by genders) makes the room placement hard. Finally, we delve into specifics of gender definition and consider locating facilities for the genders in a "perfect" way, i.e., so that navigating to the facilities involves only quick binary decisions; on the way, we indicate that there is room for interpretation the facilities under consideration (we outline several possibilities, depending on the application).

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases visibility, Strahler number, perfect tree, interval graphs, gender studies

Digital Object Identifier 10.4230/LIPIcs.FUN.2018.28

Acknowledgements We thank the anonymous reviewers for their helpful comments and Dr. Marietta Radomska from the Gender studies unit of Linköping University for bringing our attention to gender issues. The authors are partially supported by Swedish Transport Administration.

1 Introduction

The future progressive mankind, realizing and recognizing existence of more than two genders [5,14,22,39,43], faces social, political, economic, as well as algorithmic and scientistic[1] challenges of adjusting to practices of the multi-gender world and catering to the multiple genders needs. For instance, in the TEDx talk [32] on gendered innovations – a hot topic on both sides of the Atlantic [10,35] – a possibility is suggested to "require sophisticated sex and gender analysis when selecting papers for publication" (it would be fair to mention that the suggestion referred to doing it for journals and not for FUN). A lot of other inspiring material is available, produced not only by academics but also by recreational mathematicians and science popularizers [17].

One place where gender considerations come into play are locker rooms in a gym and public restrooms in a mall. As locking the locker room and restroom usage to biological sex

[1] We define a *scientistic*, or *scientistical* challenge as a non-scientific challenge faced by a scientist. Examples of scientistic challenges abound; one example relevant to this paper is learning to speak about genders without referring to sexes.

9th International Conference on Fun with Algorithms (FUN 2018).
Editors: Hiro Ito, Stefano Leonardi, Linda Pagli, and Giuseppe Prencipe; Article No. 28; pp. 28:1–28:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is being questioned (see e.g., [21]), solutions are explored both in terms of extreme gender segregation (cf. women-only gyms) and integration (cf. unisex public toilets).

For concreteness, let us speak about the gym application. We will often call the locker rooms just rooms. Privacy preservation stipulates that each gender has its own room, and that no gender g can see into another gender's room while going to g's room. We assume that the undesired peeking into other rooms may happen only from the path to the locker room, and not from anywhere in the gym (as no one wants to embarrass oneself by standing at a known spot of other-genders spotting, pretending to exercise; in addition, when exercising, people are busy with their own bodies).

Going deeper into the physics (optics) of peeking, observe that undesired exposures happen through rooms doors.[2] Locker room areas are a scarce resource, especially given the multitude of genders, so there may be people, in any locker room, arbitrarily close to the door (in fact, this is the case in many existing space-constrained gyms – those doubtful are welcome to visit a popular fitness facility at 6pm). Therefore, we ignore the locker room shapes, and require that no gender g sees another gender's door while going to g's door.

We model the gym by a simple polygon P. Let V be the vertices of P. One of the vertices $s \in V$ is the entrance to the gym. The potential locker room doors are a subset $D \subset V$ (we model doors as vertices, and not edges of P, because doors are small). For a point $p \in P$, let $\pi(s, p)$ denote the shortest s-p path. Our goal is to pick a maximum-cardinality subset $L \subseteq D$ of locker room doors so that no point on the shortest path $\pi(s, l)$ from s to a door $l \in L$ sees another door $l' \in L, l' \neq l$.

1.1 Related work

Genders (and perfectness – the other theme of this paper, developed below) are in focus of Hall's Marriage Theorem [42]: a bipartite graph, whose parts are two genders and whose edges indicate compatible pairs, has a perfect matching if and only if the vertices from any size-k subset of one part are collectively connected to at least k vertices (from the other part). Genders are central in research on Stable Marriage [16], where individuals rank members of the other gender and the goal is to match people so that there exists no pair of opposite-gender individuals each of which ranks the other higher than the current partner. Last but not least, three genders are the subject in 3D Matching – one of the six basic NP-complete problems [11, Section 3.1], in which the input is a set of compatible triples, each containing one member of each of the three genders, and the goal is to choose a subset of the triples so that each individual belongs to exactly one triple.

Hiding genders from the other genders is related to People Hiding, which is the problem of selecting a largest set of pairwise-invisible polygon vertices, or the Maximum Independent Set (MIS) in the visibility graph on V. Our problem is related to People Hiding because a feasible set L of locker room doors must be an independent set (IS) in the graph. People Hiding was proved NP-hard in [33], and [8] showed hardness of giving a PTAS.

More generally, both visibility and path planning are textbook subjects in geometric computing – see, e.g., the respective chapters in the handbook [13] and the books [12, 27]. Visibility meets path planning in a variety of computational-geometry tasks. Historically, the first approach to finding shortest paths was based on searching the visibility graph of the domain. Visibility is vital also in computing minimum-link paths, i.e., paths with fewest edges [23, 25, 37]. Last but not least, "visibility-driven" route planning is the subject in

[2] In addition, the door must be open, but the sensitive study of the birth–death process of people inside locker rooms is outside the scope of this paper; anyway, doors are regularly opened for drying.

Watchman Route problems [2, 6, 7, 24, 28] where the goal is to find the shortest path (or a closed loop) from which every point of the domain is seen (our door-picking algorithm uses the "essential cut" from watchman-route solutions). Apart from the above-mentioned theoretical considerations, visibility and motion planning are closely coupled in practice: computer vision and robot navigation go hand-in-hand in many courses and real-world applications.

1.2 Contributions and Roadmap

We open up the door (pun intended) to algorithmic gender studies. Specifically,

Section 2 gives an algorithm for choosing the maximum number of doors so that no door is seen from the shortest path between the entrance and another door. Note the contrast with the hardness of hiding people: while just picking a maximum subset of pairwise-invisible doors is equivalent to the NP-hard People Hiding (see Section 1.1), adding the restriction that no door is seen also from the *entire path* to another door makes our problem solvable in polynomial time (usually, adding restrictions makes problems harder). To make our solution efficient, we give a simple algorithm for MIS in "interval nest digraphs" (an extension of interval graphs), improving the currently-best cubic runtime [19] to quadratic; our algorithm also works for the weighted MIS, improving the currently-best runtime from quatric [18] to quadratic (the algorithms in [18, 19] work for more general class of graphs, however). We also give a simple reduction from People Hiding to our problem in *domains with holes*.

Section 3 proves (or, rather, gives another justification to the apparent fact) that non-equal treatment of genders by genders makes things hard(er): the door picking problem becomes NP-hard when every gender has a list of genders by whom it does not want to be seen (and the list does not simply contain all the other genders).

Section 4 considers picking facilities for gender segregation *without* taking into account the (in)visibility of genders to genders. We introduce the view on a gender as a (binary) string of attributes, formulate the Axiom of (Gender) Choice and derive from it the first result of the Gender Number Theory – the Fundamental Theorem On Multiple Genders. The theorem, along with navigation convenience, motivates us to consider choosing a subset of intervals on the polygon boundary so that the shortest paths from the entrance to the chosen intervals form a *perfect* binary tree. We give an algorithm for the problem, combining the greedy solution for MIS in an interval graph (which is perfect) with the bottom-up computation of *Strahler number* [41], or the *dimension* [9] of a tree, which is the height of the largest perfect tree minor of the given tree. We thus compute the *gender dimension* of a set of intervals on the boundary of a polygon – the number of genders that can be accommodated in a perfect way. The geometric nature of our MIS instances is used to bound the amount of information propagated up the tree by the algorithm; for general graphs, the requirement to have a perfect tree on the IS does not make the problem easier than vanilla MIS.

2 The gender number of a gym

We show how to find the maximum number of genders that can have the "pathview-independent" locker rooms in a gym. Recall the notation from Section 1: we are given a subset $D \subset V$ of vertices of a simple polygon P (the potential locker room doors) and a vertex $s \in V$ (the entrance), and want to pick a maximum-cardinality subset $L \subseteq D$ so that for any door $l \in L$, no point on the shortest s-l path $\pi(s, l)$ sees another door $l' \in L, l' \neq l$.

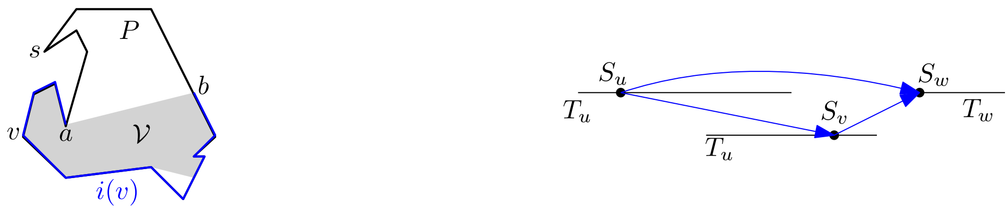

Figure 1 Left: $\mathcal{V}$ is shaded, $i(v)$ is blue; v is seen from $\pi(s,u)$ if and only if $u \in i(v)$. Right: Any IS is a path in the complement DAG, and vice versa: two consecutive vertices are connected in the DAG by definition; moreover, if uv, vw are edges, then S_v is right of T_u (since S_v is right of S_u, and there is no uv edge in the original graph) and hence S_w is also right of T_u (since S_w is right of S_v), implying that uw is also an edge, and inductively – that the whole path is a clique in the complement (the path coincides with its transitive closure).

Consider the visibility polygon $\mathcal{V}$ of a vertex $v \in D$, which is the set of points seen by v (Fig. 1, left). The polygon is bounded by edges and chords of P, with each chord connecting a vertex of P to a point on the boundary of P. One immediate observation is that if a vertex is visible from s, then the vertex cannot serve as the door to a room; thus it can be assumed w.l.o.g. that no vertex in D sees s.

For a vertex v that does not see s, there is a unique chord ab of $\mathcal{V}$ separating v from s. The chord is called the *essential cut* of v [2]. Let $i(v)$ denote the interval a-b along the boundary of P, to which v belongs. The vertex v is seen from the path $\pi(s,u)$ to a vertex u if and only if the path crosses ab, or, equivalently if and only if $u \in i(v)$. Thus, our problem reduces to the following: given the set of pairs $(v, i(v))_{v\in D}$ where v is a vertex and $i(v)$ is an interval on the boundary of P, find a maximum-cardinality subset $L \subseteq D$ so that there are no two vertices $l, l' \in L$ with $l \in i(l')$.

The above problem is an extension of MIS in interval graphs. Specifically, the *intersection graph* of a family of geometric objects has a vertex for every object and an edge for every pair of intersecting objects; the most prominent example, relevant for us, is the class of *interval graphs* in which the objects are intervals on a line. (A generalization of interval graphs are circular-arc graphs in which the objects are arcs on a circle; however, we have an interval graph, not circular-arc graph, because none of our intervals contains s, as s is separated by the essential cuts.) The following generalization of intersection graphs was proposed in [4]: the *intersection digraph* of a family $(S_v, T_v)_{v\in D}$ of ordered pairs of sets has D as the vertex set and a (directed) arc from vertex v to vertex u whenever $S_v \cap T_u \neq \emptyset$; in the *interval digraph* the sets S_v, T_v are intervals of the real line. Our problem reduces to finding MIS in an interval digraph (or more precisely, in the undirected *underlying* graph in which every arc is turned into undirected edge), simply by setting $S_v = v, T_v = i(v)$; in fact, we have an *interval catch digraph* [29], which are interval digraphs where S_v is a point and $S_v \subseteq T_v$. It was shown in [29] that interval catch digraphs are weakly triangulated (contain no chordless cycle of length at least 5 and no complement of such a cycle), and weakly triangulated graphs are perfect [20]. A well known property of perfect graphs (see, e.g., [15, Chapter 9]) is that MIS in them can be found in polynomial time.

▶ **Theorem 1.** *The maximum number of genders, for which a gym can set up locker rooms so that the shortest path from the entrance to any locker room does not see the door to another locker room, can be computed in polynomial time.*

▶ Remark. Instead of having $D \subset V$, we could consider a more general version where the doors are segments on the boundary of P. We would then redefine $\mathcal{V}$ to be the *weak visibility*

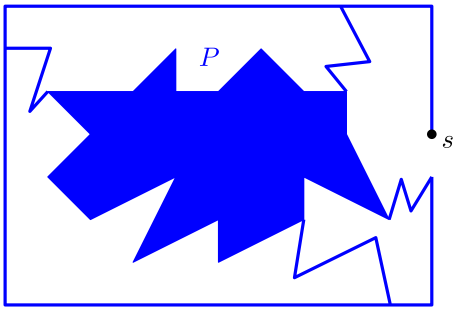

Figure 2 The free space (blue) is P, the corridor and the zigzags (zigzags are drawn not to all vertices); everything else (white) is obstacles.

polygon of v, i.e., the set of points seen from some point of v, or the union of the visibility polygons of points of v – as above, if s is inside $\mathcal{V}$, then v cannot be a locker room door. Otherwise, a point of the door is seen from a path iff the path crosses the essential cut of v, so the problem again reduces to MIS; this time – MIS in an *interval nest digraph*, in which S_v is a segment and $S_v \subseteq T_v$ (not interval catch digraph, as we had above, when the doors were vertices). When the doors are non-point objects, other models could be possible for the unwanted exposure: when a gender's door l is *fully* seen from a point on the path $\pi(s, l')$ to another gender's door, when all of l is seen (collectively) from a subset of $\pi(s, l')$ (i.e., when there exists $\pi' \subseteq \pi(s, l') : l \subset \cup_{p \in \pi(s,l')} \mathcal{V}(p)$ where $\mathcal{V}(p)$ is the visibility polygon of a point p), when π' has to be a contiguous subpath of $\pi(s, l')$, etc. – we leave these open.

▶ Remark. In non-simple gyms the problem becomes NP-hard. Indeed, given an instance of People Hiding (finding MIS in the visibility graph of vertices of a simple polygon P), we set the potential doors D to be the vertices of P, put a corridor around P and set the entrance s to the obtained domain at the start of the corridor; we connect the corridor to each vertex of P with a thin zigzag, so that no vertex sees any part of the corridor – this way, the path to any door may see another door if and only if the two doors are visible in P (Fig. 2). Thus, the maximum number of people that can hide from each other at vertices of P equals the maximum number of genders that can have the "pathview-independent" locker rooms in the domain.

2.1 A more efficient, combinatorial solution

General algorithms for MIS in perfect graphs [15, Chapter 9] are non-combinatorial and may have high complexity; therefore a lot of effort has been devoted to designing faster algorithms for special classes of perfect graphs. In particular, existing algorithms for interval catch and interval nest digraphs look for cliques in the complement [30]; the same is true also about general weakly triangulated graphs [1, 18, 19, 31, 34]. The fastest existing solution to our problem (MIS in an interval catch digraph) would run in worst-case cubic time [19]; here we give a simple quadratic algorithm. Our algorithm generalizes verbatim to interval nest graphs and to finding *weighted* MIS, improving from the currently-best quartic-time solution for the weighted case [18] (algorithms in [18, 19] work for more general classes of graphs though).[3]

[3] While going down from quartic to quadratic is a drastic theoretical improvement, we do not foresee an application of weighing the genders unequally. A (different) gender-unequal situation is considered in

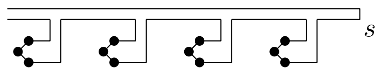

Figure 3 Solid circles are the doors; a gender will see the doors of the genders in the same "leg" of the caterpillar.

(Similarly to earlier algorithms,) we build the complement graph. (However, instead of looking for a clique in it,) we turn the complement into a DAG using S_v's for the vertices and directing the edges from left to right (if S_v is to the right of S_u, then there is a directed edge uv; Fig. 1, right), and find a longest path in it. The (quadratic) runtime is dominated by building the DAG and finding the path.

3 Gender inequality leads to hardness

In the previous section, the genders were gender-oblivious: it did not matter, for any gender, which other gender would see it naked – the exposure was undesirable regardless. In this section we consider the case when each gender has a list of genders to which it does not want to be exposed; being seen by the genders outside the list is not an issue. This generalization makes our problem NP-hard:

▶ **Theorem 2.** *If genders have lists of genders by whom they do not want to be seen, it is NP-hard to decide whether it is possible to set up locker rooms so that there are no unwanted exposures from the shortest paths between the entrance and the locker rooms.*

Proof. The reduction is from Partition Into Triangles [11, Problem GT11] (a close relative of 3D Matching, see Section 1.1): Can vertices of a graph be partitioned into triples so that the induced subgraph on each triple is a triangle? Given an instance of the problem, create a caterpillar-like polygon which consists of a long corridor to which short corridors are attached (Fig. 3), each ending with 3 doors; the total number of doors equals the number of vertices in the graph. We have as many genders as there are doors – thus, our problem is not to choose the doors (all doors need to be chosen), but to match perfectly the genders with the doors (in Section 2, the matching was not an issue since all genders were equal).

We associate each vertex of the graph with a gender. For every edge in the graph, the genders on the edge endpoints *do not* care about being exposed to each other, i.e., they are *not* in each other's lists (even though we are no longer in a gender-equal world, we at least have gender-symmetric lists: if one gender does not want to be seen by another, the latter does not want to be seen by the former). Thus, the vertices of the graph can be split into triangle-inducing triples if and only if the rooms can be assigned to the genders, avoiding undesirable exposures. ◀

4 Gender dimension and perfectness

In this section we stop taking (in)visibility between genders into account, consider premises other than gyms and go deeper into understanding what a gender is.

the next section.

4.1 Setting up the perfect scene

Women-only gyms are an extreme way to address (if at all) gender issues; an arguably milder approach to gender segregation, practiced e.g., in fast-food chains and airports in gender-segregating societies, is to have separate counters for the different genders. The ancient segregation tradition is unlikely to fade away, but maybe the modern ideas of acknowledging multiple genders existence could make their way into gender-segregating cultures? Being proactive, in anticipation of this, we consider the algorithmic question of picking disjoint counters for the many genders; here, hiding genders from genders is not an issue (as it was in Section 2) since people appear in the restaurant/airport well dressed.

Similar problems may arise e.g., in a ballet class where different genders use different barres lined up along the walls, and may even be supervised by different(-gender) coaches. A usual reason for separating genders in ballet classes is that they practice different, often complementary, parties. In ballet, gender questions have always been answered in the most progressive ways (and even were subjects of publications [26]); e.g., more-than-lightweight ballerinas and their spectacular performance are recurring topics in ballet circles [38, 40]. We may look forward ballet pieces in which presence of multiple genders is an essential part of the composition (as the two genders currently are) in the foreseeable future (directing such compositions may lead to interesting geometric questions). That is, solving the problem of optimally setting up the barres for multiple genders may be even more pressing than the above-discussed question of segregating genders at counters. Again, no issue of hiding (from) genders is present (on the contrary, genders might even want to show off to other genders).

Examples like above bear the common abstract gist, which may be formalized in the following problem statement:

Picking Intervals for Gender Segregation (PIGS) problem We are given a set D of intervals on the boundary of a simple polygon P – representing potential counters in a shop or restaurant, (sequences of) ballet barres, etc.; also given is a vertex s of P – the entrance. None of the intervals in D contains s, but otherwise, the intervals are not connected to P in any way (they may start and end in the middle of P's edges, may span multiple edges, etc.). It can be assumed w.l.o.g. that no interval is a subset of another interval. The goal is to pick a maximum-cardinality subset $L \subseteq D$ of non-overlapping intervals.

Giving a polynomial-time algorithm for PIGS is not interesting, since the intervals form an interval graph which is perfect (as in Section 2, we have an interval graph, and not circular-arc graph, since s belongs to no interval, implying that the boundary of P may be punctured at s). In the reminder of the section we take perfection one step further and require also that the shortest paths from s to the picked intervals form a *perfect* binary tree (which we will often call just "perfect" tree) – i.e., a binary tree in which every internal node has degree 2 (after vertices along paths are smoothed) and all leaves are at the same level (such a tree is sometimes called "complete" [44]). The motivation for the requirement comes from a deeper study of gender nature, presented next.

4.2 Gender definition and formal GS foundations

A moment of reflection suggests that a gender must be characterized by certain attributes – of social, biological, or any other nature. To expand and formalize this thought, we define gender as a string:

▶ **Definition 3.** Let $A_1, \ldots, A_K$ be a set of attributes; for $k = 1 \ldots K$, let $\mathcal{A}_k$ be the set of values that A_k may assume. A *gender* is a string $a_1 \ldots a_K$ where $a_k \in \mathcal{A}_k$.

Algorithmic gender studies, for which this paper makes the first steps, will become an essential part of Gender Science (GS). As any science, GS must be built on a solid axiomatic foundation (currently missing, which is forgivable for a science as young as GS). In particular, we believe that it would be unfair to deprive GS of the Axiom of Choice (while the good old sciences like mathematics enjoy the axiom in its full generally). The GS axiom reads:

▶ **Axiom 4** (Axiom of Choice). Given a set of attributes, there is a gender for each choice of values for the attributes.

For instance, assume that $a_K = baldness$ is a binary attribute (equal to 1 for a bald person and 0 otherwise), and let $S = a_1 \dots a_{K-1}$ be a length-$(K-1)$ string with $a_k \in \mathcal{A}_k, k = 1 \dots K-1$. Then both $S0$ and $S1$ should be genders, for it would be discriminative if bald people could identify themselves with a separate gender, while non-bald could not, or vice versa (boldness, or another attribute could be used in the example just as well). As computer scientists (who are adepts of the binary view of the world, reducing everything to 0s and 1s – cf. the classical joke about 10 types of people), we postulate that every attribute can assume only the values of 0 and 1 ($\forall k = 1 \dots K, \mathcal{A}_k = \{0, 1\}$); if not, just represent any kind of non-binary attribute by a set of binary attributes of appropriate size. With this interdisciplinary GS–CS postulate, we are ready to prove the first (and maybe the last) result in Gender Number Theory:

▶ **Theorem 5** (Fundamental Theorem On Multiple Genders (Fundamental OMG Theorem)). *The number of genders is a power of 2.*

Proof. By the Axiom of Choice, the number of genders is 2^K. ◀

We emphasize that the number of attributes K (and hence the number of genders) is not a constant; as the society gender awareness evolves, K may grow (and possibly decrease – but this is outside our scope).

4.3 Perfect counter/barre choice

We now return to choosing disjoint intervals for the genders (PIGS). By the Fundamental OMG Theorem, the number of picked intervals, $|L|$, must be a power of 2. As the number of genders will grow exponentially with new attributes being introduced/acknowledged/fashonable, it may become cumbersome to navigate from the entrance, s, to the intervals corresponding to people's genders. Returning for a moment to our gym application (Section 2), we ran extensive experiments on ourselves, in which the subject arrives to a new gym and finds the way to the locker room of the subject's gender; it was observed that, unsurprisingly, the task is much easier to perform in gyms with legible signs which first point in the general direction of the locker rooms, and then clearly mark the point where the paths to the different-gender rooms diverge.[4] Back to PIGS, we envision that in multi-gender society, it will also be convenient to have a "binary-split" shortest path tree (an extension of the classical "Boys to the left, girls to the right" directions from field trips to the wild), for navigation from s to the genders target intervals, with clearly marked split nodes. It might be confusing, however, to have a sign listing, say, 17 genders whose intervals are in one direction and 15 genders whose targets are in another – in fact, it could be embarrassing to stand near such a sign as

[4] Detailed report of the experimental results is deferred to a submission to the future sister conference on FUN with *Experimental* Algorithms.

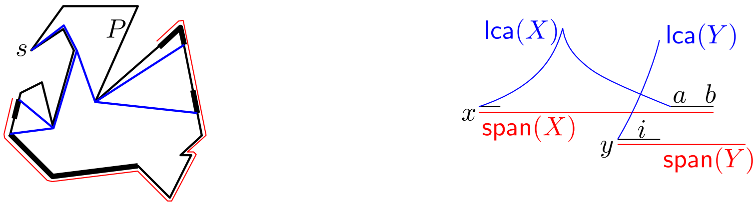

Figure 4 Left: Bold intervals are a perfect set Y of height 2. Red is the span of the set (Definition 6 below) and blue is $\mathsf{SPT}(Y)$; $\mathsf{lca}(Y) = s$. Right: Intervals are black; paths in $\mathsf{SPT}(D)$ are blue.

if confused about own gender (like the gender-confused Wolf from Shrek 1, mentioned as such in Shrek 2). On the contrary, it would be more natural (and gender-respectful) to split the genders paths based on the attributes, with each split involving only one attribute.

Thus, a gender-caring facility owner may face PIGS with the additional requirement that the shortest paths tree from s to the chosen intervals has maximum degree 3 (i.e., when viewed as rooted tree with s as the root, its every node has either 1 or 2 children). To state the full problem formally, let us introduce some notation.

We keep the assumptions from PIGS; in particular, the assumption that s does not belong to any interval from D. For an interval $i \in D$, we define the *left* endpoint $l(i)$ of i to be its (end)point encountered first when going from s counterclockwise around ∂P; more generally, we say that for two points $a, b \in \partial P$, a is to the left of b if a is between s and b when following the boundary counterclockwise from s. For a vertex v of P, the *shortest path from v to i* will mean the shortest path $\pi(v, l(i))$ to $l(i)$. (Alternatively, we could have used any other fixed points on the intervals to define the shortest paths or even use the "true" shortest paths to the intervals – i.e., shortest paths to their points that are closest to s– but dealing with paths to points makes the exposition cleaner; in terms of our applications, we may assume that the left endpoints are where the cashiers are at the counters or where the ballet dancers drop their stuff before using the barres.) For a subset $X \subseteq D$ of intervals, let $\mathsf{SPT}(X)$ be the shortest paths tree from s to the intervals in X, and let $\mathsf{lca}(X)$ be the intervals' least common ancestor in $\mathsf{SPT}(D)$. It can be assumed w.l.o.g. that all intervals in D are leaves of $\mathsf{SPT}(D)$ (otherwise, if an interval i corresponding to an internal node of the tree is picked, the interval can be replaced by a leaf interval i', the shortest path to which goes via i).

Let $Y \subseteq D$ be a set of pairwise-disjoint intervals; we say that intervals in Y are *independent* (since they form an IS in the interval graph). Let tree T be the union of the shortest paths from $\mathsf{lca}(Y)$ to intervals in Y (Fig. 4, left). Abusing the terminology, we say that T is *perfect* if it becomes a perfect binary tree after its degree-2 vertices are repeatedly smoothed (i.e., after every path of degree-2 vertices in T is replaced by a single edge). If T is perfect, we say that Y is a *perfect* IS, or simply a *perfect (sub)set*. Clearly, the cardinality of a perfect subset is a power of 2. We say that a size-2^k perfect set Y has *height* k (because k is the height of $\mathsf{lca}(Y)$ when T becomes perfect after the smoothing).

With the above notation, our problem may be stated as:

Perfect PIGS We are given a set D of intervals on the boundary of a simple polygon P; also given is a vertex s of P. The goal is to pick the largest perfect subset L of D (i.e., the largest subset $L \subseteq D$ such that $\mathsf{SPT}(L)$ is perfect).

That is, solution to Perfect PIGS gives the maximum number of genders for which the intervals can be picked so that people can navigate to their intervals in a "perfect" way – with only binary decisions at the branching points of $\mathsf{SPT}(L)$.

Perfect PIGS is an extension of not only PIGS but also of the problem of finding the *dimension* [9], or the *Strahler number* [41] of a tree – the height of the largest perfect minor of the tree (it is assumed that the tree is rooted and that the perfect minor has the same root). We therefore define the *gender dimension* of D as the height of the perfect tree $\mathsf{SPT}(L)$; with this definition, Perfect PIGS becomes the problem of computing the gender dimension of a set of intervals.

Our solution for Perfect PIGS, presented in Section 4.4, is based on two simple algorithms:

- the greedy Earliest-Endpoint algorithm for MIS in interval graphs: iteratively pick the interval with leftmost right endpoint and remove the intervals overlapping the picked one;
- the recursive procedure to compute the Strahler number of a tree (and in fact, the Strahler number of each subtree – the height of the largest perfect minor of the subtree): assign Strahler number $d(l) = 0$ to each leaf l, and then for an internal node v, let k be the maximum of the Strahler numbers of v's children – if the maximum is unique, assign $d(v) = k$, otherwise, $d(v) = k + 1$.

For Perfect PIGS, to make sure the leaves of $\mathsf{SPT}(L)$ are pairwise-disjoint, we propagate more information up the tree $\mathsf{SPT}(D)$: for every node v of $\mathsf{SPT}(D)$ we list, for every k, all height-k "tight" perfect sets of intervals from the subtree of v (where tight is an analog of earliest-endpoint). The details follow.

4.4 Algorithm for Perfect PIGS

We start from showing that when merging perfect sets from sibling nodes of $\mathsf{SPT}(D)$, it suffices to look at "spans" of the sets. Specifically, let $X \subseteq D$ be a perfect set.

▶ **Definition 6.** The *span* of X, denoted $\mathsf{span}(X)$, is the smallest interval containing all intervals in X (refer to Fig. 4, left).

Clearly, if spans of two independent sets do not overlap, their union is also an IS. The next lemma shows that if the sets live in different subtrees of $\mathsf{SPT}(D)$, the converse is also true:

▶ **Lemma 7.** *Let $X, Y \subset D$ be independent sets such that none of them is a subset of the other and $\mathsf{lca}(X) \neq \mathsf{lca}(Y)$. If $X \cup Y$ is independent, then $\mathsf{span}(X) \cap \mathsf{span}(Y) = \emptyset$.*

Proof. Let x, y be the left endpoints of $\mathsf{span}(X)$ and $\mathsf{span}(Y)$ resp. (Fig. 4, right); assume w.l.o.g. that y is to the right of x, and let $i \in Y$ be the interval whose left endpoint is y ($y = l(i)$). Let ab be the rightmost interval in X (so b is the right endpoint of $\mathsf{span}(X)$). The paths from $\mathsf{lca}(X)$ to x and a, together with the part of the boundary of P from x to a form a closed loop; since none of $\mathsf{lca}(X)$, $\mathsf{lca}(Y)$ is in the subtree of the other (for otherwise one of X, Y would be a subset of the other), $\mathsf{lca}(Y)$ is outside the loop. We claim that if the spans of X and Y intersect, then y is to the left of a, implying that the path $\pi(\mathsf{lca}(Y), y)$ from $\mathsf{lca}(Y)$ to i would intersect the loop, contradicting planarity of $\mathsf{SPT}(D)$ (which follows from the triangle inequality). Indeed, for the spans to intersect, y must be to the left of b, but if y is also to the right of a, then i intersects ab – contradicting that $X \cup Y$ is independent. ◀

Lemma 7 allows us to work with spans of perfect sets (instead of the sets themselves) when merging sets up $\mathsf{SPT}(D)$; moreover, where it creates no confusion, we will identify X with $\mathsf{span}(X)$, *and* with $\mathsf{lca}(X)$. E.g., we will speak about perfect sets X, Y (or the spans $\mathsf{span}(X)$, $\mathsf{span}(Y)$) being *siblings* – meaning that $\mathsf{lca}(X)$, $\mathsf{lca}(Y)$ are siblings in $\mathsf{SPT}(D)$. In addition, unless stated otherwise, whenever we speak about two perfect sets X, Y, we will assume that they are *siblings and have the same height* (i.e., the same number of intervals) – this is because we separately maintain perfect sets of each possible height and merge up the

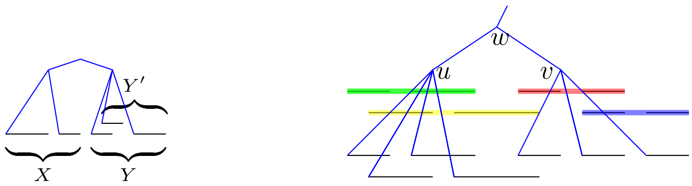

Figure 5 Left: $\mathsf{SPT}(D)$ is blue; Y is tight w.r.t. X, but Y' is not. Right: Tight height-1 sets at u are green and yellow, and the sets at v are red and blue; tight height-2 sets at w will be green+red and yellow+blue.

tree only pairs of sibling sets with the same height. Finally, whenever two spans have been merged in the course of the algorithm, we do a cleanup by removing "dominating" spans among siblings ($\mathsf{span}(Y)$ dominates a sibling span $\mathsf{span}(X)$ if $\mathsf{span}(X) \subseteq \mathsf{span}(Y)$) – this lets us speak about one set being *to the left* of another sibling set: X is to the left of a sibling Y if the left endpoint of $\mathsf{span}(X)$ is to the left of the left endpoint of $\mathsf{span}(Y)$ (due to removal of dominating spans, the right endpoint of X is also to the left of the right endpoint of Y).

Suppose now that X is to the left of Y and their spans do not overlap, or, to spell out all the assumptions, let $X, Y \subset D$ be perfect sets such that $|X| = |Y|$, (any point of) $\mathsf{span}(X)$ is to the left of $\mathsf{span}(Y)$, and $\mathsf{lca}(X)$ and $\mathsf{lca}(Y)$ are siblings in $\mathsf{SPT}(D)$.

▶ **Definition 8.** Y is *tight* w.r.t. X if there is no perfect set Y' such that (Fig 5, left): Y' is in the same subtree ($\mathsf{lca}(Y') = \mathsf{lca}(Y)$); $|Y'| = |Y|$; $X \cup Y'$ is an IS; Y' is to the left of Y.

Let $v = \mathsf{lca}(\mathsf{lca}(X), \mathsf{lca}(Y))$ be the common parent of $\mathsf{lca}(X)$ and $\mathsf{lca}(Y)$. Since (even) the spans of X and Y do not intersect, $Z = X \cup Y$ is an IS. Since X and Y have the same number of intervals and are siblings, Z is perfect: the height of the perfect tree rooted at v is by one larger than the height of the perfect trees rooted at its children; with our terminology, if each of X, Y had height k, the height of Z is $k + 1$.

Our algorithm merges, for each possible height k of perfect sets, only tight sets (Fig. 5, right). We initialize by making at each leaf of $\mathsf{SPT}(D)$ the list containing the interval at the leaf ($k = 0$). We then go up the tree. First of all, an internal node v of $\mathsf{SPT}(D)$ inherits the lists of its children for all k. In addition, we go through the lists of v's children for k, searching for tight merges of perfect sets. We do it brute force: for every child x of v, for every set X in the list of x, we go through each set Y in the list of every other child y checking whether $\mathsf{span}(X)$ and $\mathsf{span}(Y)$ are disjoint; out of the found sets, we keep only the tight one – for a fixed X, there is at most one tight union $X \cup Y$. The tight union becomes part of the list of v for $k + 1$. If no pair (X, Y) with disjoint spans is found, then v's list for $k + 1$ is empty.

▶ **Theorem 9.** *There is a polynomial-time algorithm for Perfect PIGS.*

Proof. Any tight set, after being created at a node v, appears also in the lists of all nodes on the path from v to the root s (unless the set is removed due to domination) – altogether at most $|V|$ times. To bound the number of sets created through the algorithm, note that since we remove dominating sets, at any node of $\mathsf{SPT}(D)$, for any k, we have at most one tight set "starting" at any interval i, i.e., the set whose span's left endpoint is $l(i)$ (in any case, the total number of possible spans is $O(|D|^2)$, as the spans endpoints come from endpoints of the original intervals). Since k is logarithmic in $|D|$, there are $O(\log |D|)$) lists at any node, so we can afford the propagation for all k – the total amount of propagated information is polynomial in $|D|$ and $|V|$, and the algorithm runs in polynomial time.

Figure 6 Left: Red and green spans are not tight, while red and blue are; using the tight union cannot lead to overlap with an interval i from another subtree, as such overlap would imply a crossing in $\mathsf{SPT}(D)$. Right: (Some of) the graph edges are drawn black; the inter-clause edges connect any variable to its negation.

Correctness of the solution can be shown by arguing that all "recursively tight sets" appear in our algorithm in the sets lists and that it suffices to look only at such sets. Specifically, let $\mathcal{T}$ be a perfect tree and let $L \subseteq D$ be its leaves. For a set $Z \subseteq L$ let $v = \mathsf{lca}_{\mathcal{T}}(Z)$ be the node of $\mathcal{T}$ whose subtree has Z as the leaves, and let X and Y be the sets at $\mathcal{T}$'s children of v (so $Z = X \cup Y$), with Y being tight w.r.t X. Say that Z is *recursively tight* if the sets at each internal node of the subtree of v is a tight union (i.e., if recursively, each of X and Y is a tight union, and – for the base of the recursion – a single interval is assumed to be a tight union). By induction on the node height, our algorithm lists all recursively tight sets, as it goes through all possible pairs of sets at each node of $\mathsf{SPT}(D)$. At the same time, there must exist a recursively tight optimal solution. Indeed, any feasible solution can be made recursively tight by (recursively) shuffling the spans to the left: start at the root of the perfect tree, and if the set $Z = X \cup Y$ at the root is not a tight union, then there exists a set Y' such that $Z' = X \cup Y'$ is a tight union – the solution with Y replaced by Y' is still an IS; recursively, if any of X, Y' is not a tight union, it can be fixed in the same way – the solution will remain feasible, which can be seen by a planarity argument analogous to the proof of Lemma 6 (Fig. 6, left). ◀

▶ Remark. Unsurprisingly, in general, putting a tree on top of an arbitrary graph and requiring to have a perfect tree minor on the IS from the graph, does not help finding MIS. To see this, use CLRS [3, p.1087] reduction from 3-SAT to MIS in the graph with a vertex for every literal, the literals in each clause connected into triangle (so at most one vertex from every clause may enter MIS) and edges between two literals whenever they are a variable and its negation (so only one can be in MIS); MIS size equals the number of clauses if and only if the 3-SAT instance is feasible (Fig. 6, right). It is easy to make the number of clauses a power of 2 and add a tree on the graph nodes, with vertices in each clause triangle being siblings of a height-1 parent, and the perfect binary tree up the leaves parents – any MIS will be perfect, so demanding the tree perfectness does not make the problem easier. It could be interesting to explore for which graphs, if any, the perfectness requirement changes the hardness of finding MIS; one question relevant to the gender study is whether it is possible to find perfect MIS in an interval catch digraph, to which reduces the problem of picking genders' locker room doors (like in Section 2) in the perfect way (like in Section 4) – we answer the question affirmatively in Appendix A.

5 Conclusion

We gave first algorithmic results for assessing how fit a gym is (i.e., how many genders it can accommodate) and related questions, presenting some polynomially-solvable and g-hard (for gender-hard, i.e., hard w.r.t. the number of genders) problems. Many extensions are possible:

- For a *given* number of genders, if it is not feasible to choose doors so that there is no exposure (Section 2), one may want to minimize various measures of the exposure – the number of undesirable peeks, the total length of the parts of the shortest paths from which the other(s') doors are seen, the total area seen behind the doors, the "depth" of the exposures (how far into locker rooms other genders see), etc. In the gender-oblivious scenario (when any gender does not want to be seen by all the others) the assignment of genders to the doors does not matter (as in Section 2); what matters is which vertices are chosen to be the doors.
- The gender-unequal setting in Section 3 may be generalized to the case when there is a whole matrix of numbers (possibly with positive entries, for exhibitionist genders) signifying (un)desirability of one gender being seen by every other. Here, one may want to minimize the *weighted* exposure.
- Last but not least, it would be interesting to know hardness of finding maximum perfect IS in a general perfect graph with a tree over its vertices. E.g., can the math programming algorithms for MIS in perfect graphs be made to work?

More generally, further mathematical studies on genders are to come, e.g., extending the differential equations for love [36] to many genders.

References

1. Srinivasa Arikati and C. Rangan. An efficient algorithm for finding a two-pair, and its applications. *Discrete Applied Mathematics*, 31(1):71–74, 1991.
2. Svante Carlsson, Håkan Jonsson, and Bengt J. Nilsson. Finding the shortest watchman route in a simple polygon. *Discrete & Computational Geometry*, 22(3):377–402, 1999. `doi:10.1007/PL00009467`.
3. Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.
4. Sandip Das, M Sen, AB Roy, and Douglas B West. Interval digraphs. *J. Graph Theory*, 13(2):189–202, 1989.
5. Donna Dean. Changing ones: 3rd & 4th genders in native America. *Women and Military*, 18(2):54–54, 2000.
6. Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph Mitchell. Touring a sequence of polygons. In Lawrence L. Larmore and Michel X. Goemans, editors, *SToC'03*, pages 473–482. ACM, 2003.
7. Adrian Dumitrescu and Csaba D. Tóth. Watchman tours for polygons with holes. *Comput. Geom.*, 45(7):326–333, 2012. `doi:10.1016/j.comgeo.2012.02.001`.
8. Stephan Eidenbenz. Inapproximability of finding maximum hidden sets on polygons and terrains. *CGTA*, 21(3):139–153, 2002.
9. Javier Esparza, Michael Luttenberger, and Maximilian Schlund. Fpsolve: A generic solver for fixpoint equations over semirings. *Intl J Foundations of Computer Science*, 26(07):805–825, 2015.
10. European Commission. Gendered Innovations.
11. Michael R. Garey and David S. Johnson. *Computers and Intractability*. Freeman & Co., New York, NY, USA, 1979.
12. Subir Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, New York, NY, USA, 2007.
13. J.E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. Discrete Mathematics and Its Applications. Taylor & Francis, 2nd edition, 2004.
14. Sharyn Graham. Sulawesi's fifth gender. *Inside Indonesia*, 66, 2001.

15 Martin Grötschel, László Lovász, and Alexander Schrijver. *Combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

16 Dan Gusfield and Robert Irving. *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, Cambridge, MA, USA, 1989.

17 Vi Hart. On Gender, 2015.

18 Ryan Hayward, Chính Hoàng, and Frédéric Maffray. Optimizing weakly triangulated graphs. *Graphs and Combinatorics*, 5(1):339–349, Dec 1989.

19 Ryan Hayward, Jeremy Spinrad, and R. Sritharan. Weakly chordal graph algorithms via handles. In *SODA'00*, pages 42–49, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

20 Ryan B Hayward. Weakly triangulated graphs. *J Comb Theory, Series B*, 39(3):200–208, 1985.

21 Aaron Homer. Man Uses Women's Locker Room At Seattle Pool, Says It's Legal Because Of Anti-Transgender Discrimination Laws, 2016.

22 M Kay Martin and Barbara Voorhies. *Female of the Species.* Columbia University Press, 1975.

23 J. Mitchell, G. Rote, and G. Woeginger. Minimum-link paths among obstacles. *Alg-ca'92*, 8(1):431–459, 1992.

24 Joseph Mitchell. Approximating watchman routes. In Sanjeev Khanna, editor, *Proc. 24th Annual ACM-SIAM Symp. on Discrete Algorithms, SODA'13, New Orleans, Louisiana, USA*, pages 844–855. SIAM, 2013.

25 Joseph S. B. Mitchell, Valentin Polishchuk, and Mikko Sysikaski. Minimum-link paths revisited. *Comput. Geom.*, 47(6):651–667, 2014. doi:10.1016/j.comgeo.2013.12.005.

26 Cynthia J Novack. Ballet, gender and cultural power. In *Dance, gender and culture*, pages 34–48. Springer, 1993.

27 Joseph O'Rourke. *Art Gallery Theorems and Algorithms.* The International Series of Monographs on Computer Science. Oxford University Press, New York, NY, 1987.

28 Eli Packer. Computing multiple watchman routes. In Catherine C. McGeoch, editor, *SEA'08*, volume 5038 of *Lecture Notes in Computer Science*, pages 114–128. Springer, 2008.

29 Erich Prisner. A characterization of interval catch digraphs. *Discrete Math*, 73(3):285–289, 1989.

30 Erich Prisner. Algorithms for interval catch digraphs. *Discrete Appl Math*, 51(1-2):147–157, 1994.

31 Arvind Raghunathan. Algorithms for weakly triangulated graphs, 1989. Tech Rep, UC Berkeley. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/1989/5196.html.

32 Londa Schiebinger. Gendered Innovations, 2013.

33 T Shermer. Hiding people in polygons. *Computing*, 42(2):109–131, 1989.

34 Jeremy P. Spinrad and R. Sritharan. Algorithms for weakly triangulated graphs. *Discrete Applied Mathematics*, 59(2):181–191, 1995. doi:10.1016/0166-218X(93)E0161-Q.

35 Stanford University. Gendered Innovations.

36 Steven H Strogatz. Love affairs and differential equations. *Mathematics Magazine*, 61(1):35, 1988.

37 Subhash Suri. A linear-time algorithm for minimum link paths inside a simple polygon. *Computer Vision, Graphics and Image Processing*, 35(1):99–110, 1986.

38 Erica Tempesta. Stereotypes are made to be broken!, 2017.

39 Randolph Trumbach. *From 3 sexes to 4 genders in the making of modern culture.* Routledge, 1991.

40 Peter Walker. Men need lots of energy to lift ballet dancers because women are getting taller, 2017.

41 Wikipedia contributors. Strahler number, 2017.

42 Wikipedia contributors. Hall's marriage theorem, 2018.
43 Wikipedia contributors. Third gender, 2018.
44 Yuming Zou and Paul Black. Perfect binary tree. In Vreda Pieterse and Paul E. Black, editors, *Dictionary of Algorithms and Data Structures*. National Institute of Standards and Technology, 2008.

A Maximum perfect subset of doors

This section extends the algorithm for perfect MIS in interval graphs (Section 4) to interval *catch digraphs*, thus solving the problem of picking genders' locker room doors (like in Section 2) in the perfect way (like in Section 4). The extension follows the same approach of merging "tight" sets up the tree, after appropriate modifications of definitions for span, being to the left, tightness, etc. The technical details follow.

We first recollect some definitions from Section 2. An *interval digraph* has two intervals (S_v, T_v) for each vertex v; two vertices u, v are connected if either $S_u \cap T_v \neq \emptyset$ or $S_v \cap T_u \neq \emptyset$ (or both). Strictly speaking, the graph is directed, but, similarly to almost all work on interval digraphs, we will look at the underlying undirected graph and often say simply "graph" for "digraph". An interval *nest* digraph has $S_v \subseteq T_v \, \forall v$; we call S_v and T_v the *inner* and the *outer* intervals, resp. An interval nest graph is an interval *catch* graph if S_v is a single point for all v. As usual, we will identify graph vertices with their intervals. The problem of picking maximum perfect subset of locker rooms reduces to the following: given an interval catch graph on the potential doors D, with $S_v = v, T_v = i(v) \, \forall v \in D$ (see Section 2 for notation), find a maximum independent set $L \subseteq D$ such that the shortest paths from s to L form a perfect tree (see Section 4 for terminology).

Our algorithm for picking a maximum perfect pathview-independent subset of locker room doors reuses two ideas from the algorithm of [30] for maximum clique in interval catch digraphs:

- instead of solving the problem for interval catch graphs, solve it for (more general) interval nest graphs
- if two vertices u, v of an interval nest graph are replaced by their "span", the span will be connected to those and only those vertices to which (at least) one of u, v was connected ([30, Theorem 3.6]; see also [18] for an application of the similar idea to finding maximum cliques in general weakly triangulated graphs).

We now define the span formally. Extending Definition 6 (Section 4) of the span of a set of segments, we obtain the span of two vertices in an interval nest digraph by taking the spans of both the inner and the outer intervals:

▶ **Definition 10.** The *span* of two vertices u, v consists of a pair of intervals (S, T) where S is the smallest interval containing S_u, S_v and T is the smallest interval containing T_u, T_v.

Clearly, if $S_v \subseteq T_v, S_u \subseteq T_u$, then $S \subseteq T$; thus, we can speak about the new interval nest graph in which u and v are replaced by their span (in [18, 30] such graph is denoted by $G(uv >> w)$ where G is the original graph and w is the vertex for the span of u, v). The span of more than two vertices is defined analogously, using associativity of taking the span. We retain all conventions from Section 4: a "set" will usually mean a perfect IS, two sets will generally have the same cardinality and be siblings in $\mathsf{SPT}(D)$, we will identify sets with their spans, and sometimes say "vertices" instead of "spans" – meaning that the spans are vertices in the interval nest graph (obtained by applying the $G(uv >> w)$ operation, possibly several times).

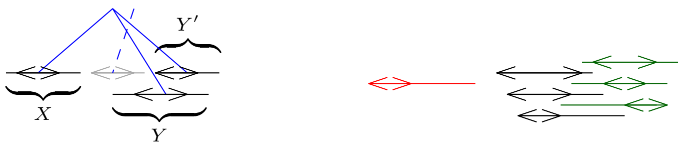

Figure 7 The inner intervals are represented by the arrows. Left: Y is tight w.r.t. X, but Y' is not; as in the proof of Lemma 7, if Y' were to be picked together with a set Z (grey) from another subtree, then $\mathsf{SPT}(D)$ (blue) would have had a crossing – to see this, ignore the outer intervals $T_X, T_{Y'}, T_Z$ and note that the paths to vertices in X, Y' and Z end inside $S_X, S_{Y'}, S_Z$ resp. (and for Z to be independent from both X and Y', S_Z must not overlap with any of $S_X, S_{Y'}$ – so S_Z would have to lie between S_X and $S_{Y'}$, implying the crossing). Right: Any black set is tight w.r.t. the red; no black sets "dominates" another black set because, depending on which green set is present, any black set may be the only one compatible with the green.

Analogously to merging intervals in Section 4, we merge (tight spans of) vertices up the tree $\mathsf{SPT}(D)$. The only difference is that for interval nest graphs the tightness is defined w.r.t. both inner and outer intervals. Specifically, let X and Y be two non-connected vertices, i.e., spelling out all our assumptions, two same-cardinality perfect independent sets such that $\mathsf{lca}(X)$, $\mathsf{lca}(Y)$ are siblings and $X \cup Y$ is a perfect IS (or two spans of such sets); assume w.l.o.g. that X is "to the left" of $Y-$ i.e., T_X is to the left of S_Y and (hence) S_X is to the left of T_Y.

▶ **Definition 11.** Y is *tight* w.r.t. X if there is no perfect set Y' such that (Fig 7, left):

- Y' is in the same subtree ($\mathsf{lca}(Y') = \mathsf{lca}(Y)$);
- $|Y'| = |Y|$;
- $X \cup Y'$ is an IS;
- the right endpoint of $S_{Y'}$ is to the left of the right endpoint of S_Y, or the right endpoint of $T_{Y'}$ is to the left of the right endpoint of T_Y.

The first three items are the same as in the tightness definition for interval graphs (Definition 8 in Section 4), while the last is slightly more involved since for interval nest graphs we have to look at both the inner and outer intervals (Fig. 7, right).

As in Section 4, considering only tight unions is enough, since any feasible solution can be made recursively tight by the same (recursive) procedure as in the proof of Theorem 9 – replace any non-tight set Y with a set Y' that certifies non-tightness if Y; also as in the theorem's proof, our algorithm lists all recursively tight sets, as it goes through all possible tight sets at every node of $\mathsf{SPT}(D)$. The only difference is that for interval nest graphs, more tight sets are created because there may be more than one tight set "starting" at any vertex $v \in D$ (i.e., the set whose span's inner interval has left endpoint at $l(S_v)$ or whose span's outer interval has left endpoint at $l(T_v)$); refer to Fig. 7, right. Still, for every k, at any node of $\mathsf{SPT}(D)$ there are $O(|D|^4)$ spans – this is because a span is defined by 4 points (endpoints of the inner and outer intervals), each of which is either $v \in D$ or an endpoint of $i(v)$; thus there are $O(|D|^4)$ different spans overall.