Improved Cardinality Bounds for Rectangle Packing Representations

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Vorgelegt von

Jannik Silvanus

aus

Gerolstein

Bonn, März 2019

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Erstgutachter:	Herr Professor Dr. Stefan Hougardy
Zweitgutachter:	Herr Professor Dr. Jens Vygen
Tag der Promotion:	27.05.2019
Erscheinungsjahr:	2019

Acknowledgments

First and foremost, I would like to deeply thank my advisors Professor Dr. Stefan Hougardy and Professor Dr. Jens Vygen for guiding and supporting me in the last years. Their invaluable advice, insights and expertise have been essential for my theoretical and practical work.

Special thanks go to Professor Dr. Dr. h.c. Bernhard Korte for providing excellent working conditions at the Research Institute for Discrete Mathematics.

I also need to thank my colleagues: First, I would like to particularly thank Philipp Ochsendorf for the productive joint work on BonnPlace; I will certainly miss those technical problem solving sessions. Furthermore, I would like to thank the other current and former members of the BonnPlace team, including Dr. Ulrich Brenner and Dr. Jan Schneider. I also thank the rest of the team, in particular Anna Hermann, Pascal Cremer, Markus Ahrens, Pietro Saccardi, and Siad Daboul. Lastly, I thank Professor Dr. Stephan Held for his help regarding BonnOpt and the IBM environment, and Dr. Nicolai Hähnle for many inspiring discussions.

I wish to thank my sister Anna Silvanus and my parents Gertrud and Manfred Silvanus. Sadly, my mother is no longer with us to see me complete this thesis. My parents' permanent support allowed me a care-free time as student, despite my father having to accept that I would study mathematics instead of studying something *proper* (i.e., becoming an engineer). Of course, I have to thank Anna Hermann again, not only for proofreading this thesis, but particularly for the enduring support during the last months.

Finally, I would also like to thank the *The On-Line Encyclopedia of Integer Sequences* which made me aware of plane permutations, an insight that proved to be crucial for this thesis.

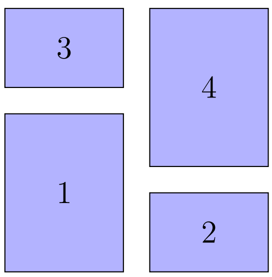
Contents

Chapter 1

Introduction

In this thesis, we consider axis-aligned rectangle packings. These can be characterized by the set of spatial relations that hold for pairs of rectangles (west, south, east, north). A *representation* of a packing consists of one satisfied spatial relation for each pair, see Figure 1.1. We call a set R of representations *complete* if R contains a representation of every packing of any n rectangles. Both in theory and practice, the fastest known algorithms for many rectangle packing problems enumerate a complete set R of representations. The running time of these algorithms is dominated by the (exponential) size of R.

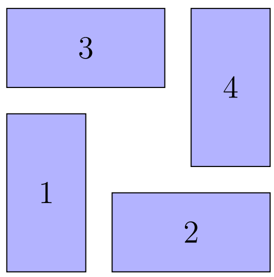
In this work, we improve the best known lower and upper bounds on the *minimum cardinality* of complete sets of representations. The new upper bound implies theoretically faster algorithms for many rectangle packing problems, while the new lower bound imposes a limit on the running time that can be achieved by any algorithm following this approach.



(a) A packing admitting two representations.

Pair	r	r'
$(1,2)$	west	west
$(1,3)$	south	south
$(1,4)$	west	west
$(2,3)$	*south*	*east*
$(2,4)$	south	south
$(3,4)$	west	west

(b) Two representations of the packing on the left.



(c) A packing represented by r, but not by r'.

Figure 1.1: Rectangle packing representations.

Rectangle Packing and Its Applications

In the simplest rectangle packing problem variant, we are given a set of small rectangles and one large rectangle, and the task is to find an axis-aligned packing of the small rectangles into the large rectangle. This problem is provably computationally difficult[1] ([GJ79]), and all known exact algorithms require exponential running time.

Often, additional constraints have to be satisfied, for example constraints on the positions of rectangles ([DLMT08]), or upper bounds on the distances of certain rectangles ([Och19]).

Moreover, one is usually interested in solutions that are not only feasible (i.e., disjoint packings that satisfy all constraints), but also optimize certain objectives. Examples of typical objectives include the perimeter or area of the smallest enclosing rectangle ([MFNK96]), or the total displacement of the small rectangles compared to an initial, overlapping solution ([BV04; Woc17]). Another popular variant fixes only the width of the enclosing rectangle, and asks for a packing that minimizes the total height ([BCR80]).

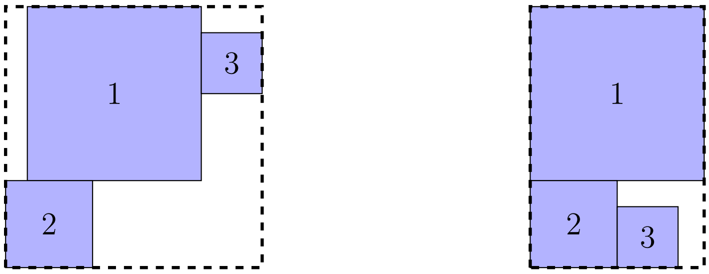
Rectangle packing problems naturally occur in pallet loading ([Hod82]), where a set of rectangular objects has to be packed onto a rectangular pallet using a single layer. The second obvious application of rectangle packing lies in two-dimensional cutting stock problems: Here, raw rectangular stock sheets need to be cut into small rectangular pieces ([GG65]), for example in the glass industry ([Mad79]).

More recently, (map) labeling problems ([FW91; Bar+14]) have been considered: Rectangular text labels, for example on a map, need to be arranged in such a way that no two labels intersect.

A less obvious application of rectangle packing is given by certain job scheduling problems with a shared resource: Each job is represented by a rectangle whose width corresponds to the contiguous amount of some resource that is blocked while processing the job, and whose height models the required time to process the job. Examples include the parallel execution of programs ([Cod60]) and assignments of container ships to berths ([LLQ04; DLMT08]).

However, the application that has driven most theoretical and practical advances in rectangle packing is chip design: Computer chips consist of hierarchical, rectangular modules which are connected by millions of electrical wires and need to be arranged in an axis-aligned packing ([HKRV11]). In a good packing, the total wire length should be small, as otherwise there might not be enough space to fit all wires, and moreover the power consumption of a chip is closely related to the total wire length. Moreover, performance requirements impose a limit on the time a signal may take to traverse a wire, and thus on the length of individual wires ([Och19]). See also Chapter 2.

[1] More precisely, the rectangle packing problem is strongly *NP*-complete, which can be shown by a simple reduction from 3-PARTITION ([GJ79]).



(a) A non-compacted rectangle packing.

(b) A compaction of the packing on the left.

Figure 1.2: Rectangle packing compactions.

Exact Algorithms

In some applications, the (absolute and relative) positions of the rectangles are not relevant, for example when only minimizing the area of the smallest rectangle enclosing the packing. In this case, one can restrict to so-called *compacted* packings, where no rectangle can be moved to the south or west without introducing overlap or leaving the enclosing rectangle, cf. Figure 1.2. If the dimensions of the rectangles are fixed, compacted packings can be efficiently encoded using O-trees ([GCY99; Tak00]) and B*-trees ([CCWW00]), both of which allow $\mathcal{O}\left(\frac{n!}{n^{1.5}}4^n\right)$ possible encodings for n rectangles and which are the basis of the theoretically fastest known algorithms for such problems.

However, often the actual positions of the rectangles are important: In chip design, wire connections need to be short. Similarly, in berth allocation, moving a rectangle horizontally changes the ship's position on the berth and hence the total distance that containers need to be moved, and vertical positions determine the ships' waiting times ([DLMT08]). In order to avoid confusion with the packing variants depicted above, we call these problems *placement problems*, consistent with the terminology used in chip design ([HKRV11]).

For placement problems, there is not necessarily an optimum solution that is compacted, and enumerating O-trees or B*-trees no longer suffices. Instead, the fastest known algorithms (both in theory and practice) enumerate a complete set of representations, and for each representation r compute an optimum placement that is represented by r ([KV08]). In the case of practical algorithms, the enumeration of representations usually follows a branch-and-bound scheme ([OTT91; FHS16]) or is implicit in an integer programming formulation ([Xu+17; Woc17; Och19]). Note that the set of placements represented by a fixed representation r forms a polyhedron with one inequality per rectangle pair, and hence can be efficiently optimized over using linear programming techniques. In many cases, the problem of finding an optimum placement that is represented by r even reduces to a more specific problem which can be solved more quickly, for example a minimum-cost flow problem ([CFS70; FHS16]).

Outline

In Chapter 2, we give more details on placement problems in chip design, and summarize our practical contributions in this area, which will not be dealt with in the remainder of this thesis.

In Chapter 3, we fix some general notation, formally introduce rectangle placements and their representations, and discuss previous work. Furthermore, we introduce pattern-avoiding permutations which are a key tool for our new results.

Then, in Chapter 4, the classical sequence pair representation of size $(n!)^2$ ([Jer85]) is revisited, which will be the basis for our work. We show a new construction from which the results of Jerrum [Jer85] can be recovered, and derive new properties of sequence pairs.

In Chapter 5, we prove a new upper bound of $\mathcal{O}\left(\frac{n!}{n^6} \cdot (\frac{11+5\sqrt{5}}{2})^n\right)$ on the minimum cardinality of complete sets of representations for n rectangles, where $\frac{11+5\sqrt{5}}{2} \leq 11.091$. This improves upon the previously best upper bound of $\mathcal{O}\left(\frac{n!}{n^{4.5}} \cdot 32^n\right)$ by Shen and Chu [SC03].

In Chapter 6, we improve the previously best lower bound of $n! \cdot 2^{n-1}$ ([Sil11]) to $\Omega\left(\frac{n!}{n^4} \cdot (4+2\sqrt{2})^n\right)$, where $4+2\sqrt{2} \geq 6.828$.

Finally, in Chapter 7, we empirically compute the minimum cardinality of complete sets of representations for small n. Our computations directly suggest two conjectures, connecting well-known Baxter permutations (cf. Section 3.4.3) with the set of permutations avoiding an apparently new pattern, which in turn seem to generate complete sets of representations of minimum cardinality.

Most results of Chapters 4, 5 and 6 are joint work with Jens Vygen ([SV17]).

Chapter 2
Placement in Chip Design

Now, we introduce practical placement problems in chip design in greater detail, giving more context for this work. Moreover, we briefly describe our contributions to these practical problems.

The Research Institute for Discrete Mathematics at the University of Bonn maintains a close cooperation with IBM on chip design (also called VLSI[1] design). As part of this cooperation, the software suite BonnTools ([KRV07; HKRV11]) is developed, which contains optimization algorithms for a wide range of problems occurring in VLSI design and which has been used for the design of hundreds of chips at IBM, including the latest POWER and mainframe processors. The placement engine of BonnTools is called BonnPlace ([BSV08; BHHO15]). The theoretical results in this work have been directly motivated by the work on BonnPlace algorithms. The details of this practical work will be briefly covered in this chapter.

The logical properties of a chip are modeled in a hardware description language (HDL), e.g., Verilog or VHDL. The HDL description is compiled to a *netlist* that consists of *cells* (also called *circuits*) that implement elementary logic functions, for example `NAND`, `NOR`, and `NOT`, as well as register cells that are used to store single bits. Each cell has a set of input and output connectors, called *pins*. Additionally to the set of cells, the netlist also contains the set of *nets*: Each net consists of a single input pin (which is the output pin of a cell, or an external input), and a set of output pins (which are input pins of cells, or external outputs), and models a required electrical connection.

In a design step called *physical design*, the cells of the netlist have to be

[1]Very Large Scale Integration

placed (called *placement*), and all nets have to be realized by physical wires (called *routing*). As part of physical design, the netlist may also be slightly changed: For long wire connections, repeaters may need to be inserted, and some cells may be replaced by different cells that implement the same logical function, but have different electrical properties, for example with respect to timing and power consumption.

A chip consists of many layers which are arranged on top of each other. The lowest layer is the placement layer, the only layer that contains transistors, followed by multiple wiring layers, which are connected by *via* layers. Manufacturing constraints require wires to have axis-aligned rectilinear shapes, and most layers are even uni-directional, i.e., only allow wires in a single direction.

Cells usually have a rectangular outline. Since there is only a single placement layer, cells must be placed disjointly, and technical constraints forbid cell rotations. Hence, a valid placement consists of an axis-aligned rectangle packing.

The quality of a placement is almost entirely determined by its routing properties: In the first place, the placement must be routable at all, i.e., there must not be areas where the available wiring space does not suffice to fit all wires. Moreover, the power consumption of a chip closely depends on the total wire length, and finally single wires may not be too long, since otherwise signals take too long to traverse them, limiting the frequency of the chip.

2.1 Global Placement

Each cell needs to be supplied with two different voltage levels which drive the CMOS transistors used in the implementation of the cell logic and which are used to encode binary information on a chip. The distribution of these voltages is implemented in the *power grid*. The power grid contains horizontal, equidistant power wires of alternating voltage levels which partition the chip area into *circuit rows*. The height of all cells must be a multiple of the circuit row height, and the height of almost all cells equals the circuit row height. Such cells are called *standard cells*. In a valid placement, standard cells need to align to the circuit rows.

Standard cell placement is also called *global placement*. Instances are often huge, containing millions of cells. On the other hand, since all rectangles have the same height and small width, finding a feasible placement is almost always trivial, and the difficulty only lies in finding a *good* placement.

Global placers commonly relax the disjointness constraint to a *density* constraint, which requires that everywhere on the chip, the local amount of cell area does not exceed (a certain fraction of) the available free area. After finding a placement that satisfies density constraints, the *legalization* step removes overlaps ([BV04]), which can usually be done using local changes only.

Thus, the characteristics of global placement are more similar to a continuous problem, and hence usually numerical methods are applied ([BV08]). BonnPlaceGlobal, the global placer of BonnPlace, also uses this idea, following a recursive approach that assigns cells into smaller and smaller rectangular regions, starting with a single region containing the whole chip area. In each recursion level, a quadratic programming (QP) relaxation (which ensures small net length) is solved, subject to the constraint that each cell remains in its region. Then, by solving a minimum-cost flow problem on a bipartite graph, cells are assigned to the smaller regions of the next recursion level, respecting the regions' capacities and minimizing the total induced cell movement.

Other global placers directly incorporate cell density into the relaxation: Solving the quadratic program is equivalent to minimizing the energy of a system of attracting forces, and cell spreading is achieved by adding forces to that system: Spindler, Schlichtmann, and Johannes [SSJ08] iteratively legalize the QP solution, and add forces pulling cells towards their legalized position, Lu et al. [Lu+14] add *repelling* forces, imitating an electrostatic system.

2.2 Macro Placement

Macros are large, non-standard cells. Macros come in two flavors:

Firstly, there are pre-designed macros that are repeatedly used on the same chip, for example *memory arrays* containing SRAM memory for processor-internal caches.

Secondly, there are macros that contain ordinary combinatorial logic and are often used only once. These occur in *hierarchical design*: The netlist of a large chip is partitioned into several clusters, and each cluster is assigned a rectangular shape that is large enough to contain its logic. Moreover, for nets crossing the boundary of a cluster the exact location of crossing points (called *ports*) is determined. Then, each cluster can be designed independently of the other clusters, often by different designers or even different teams. Afterwards, in a step called *integration*, finished designs of the clusters can simply be stitched together. Moreover, if the logic of a chip needs to be changed later on, it suffices to apply changes locally to the containing cluster, and the surrounding chip may remain untouched. Note that there may be multiple nested hierarchy levels on a chip. See also Section 2.3.

Since macro sizes may vary greatly, finding a feasible placement often is non-trivial, and finding a good placement is even more difficult. Due to the more discrete nature of the problem (compared to global placement), macro placement algorithms typically reduce the problem of finding a good placement to the problem of finding a good representation, since for a given representation, a good placement can usually be found efficiently.

2.2.1 BonnMacro Overview

BonnMacro, the macro placement algorithm of BonnPlace, works as follows (cf. Figure 2.1). First, in a step called *shredded placement*, we compute a macro placement with good net length properties which ignores macro overlaps, but ensures that the macros are well-distributed over the chip area. Then, in *macro legalization*, we eliminate all overlaps by solving local rectangle packing instances optimally. Finally, in *macro post-optimization*, we apply local changes to the placement that improve secondary objectives where possible.

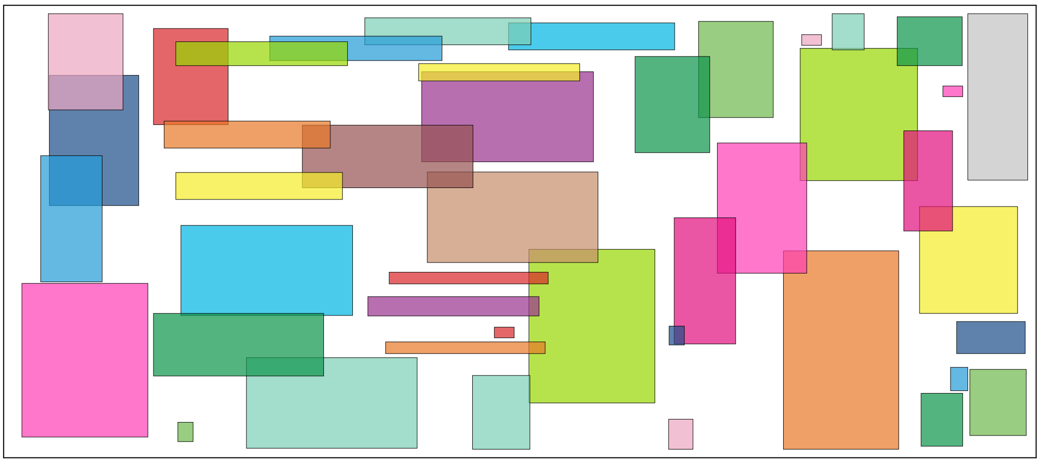
In shredded placement, every macro is cut into small pieces (called *fragments*), which are connected by artificial nets of high weights, and the resulting netlist is placed using BonnPlaceGlobal. An example of such a fragment placement is given in Figure 2.1(a). Then, the position of each macro is determined based on its fragments' positions, see Figure 2.1(b). Due to the large connectivity between the fragments of a macro, these are usually placed closely together. Moreover, the shredded global placement satisfies density constraints, and hence overlaps in the re-assembled macro placement can usually be resolved locally. Of course, there is no guarantee that this will always be the case.

In macro legalization, the objective is to find a feasible macro placement that is as close as possible to the initial, overlapping solution. More precisely, we want to minimize the (weighted) sum of L_1 distances of all macros' legalized center positions to their initial center positions. An alternative objective function is the minimization of the (weighted) sum of squared L_2 distances, where multiple small movements are preferred to single, large ones. In both cases, the objective function decomposes into two independent, dimension-specific components.

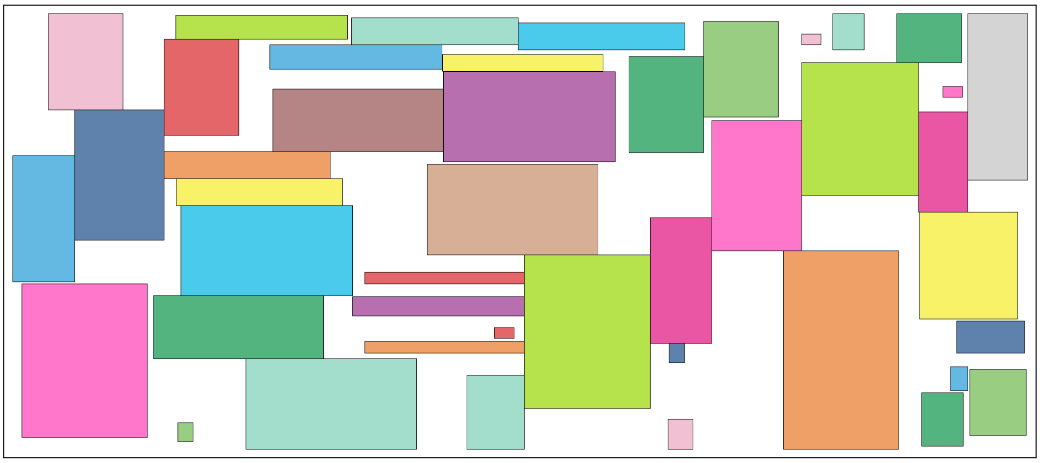
Overlaps are eliminated by solving local rectangle packing instances optimally using a branch-and-bound approach ([FHS16]). This algorithm solves the more general half-perimeter wirelength (HPWL) placement problem: As input, we are given a rectangular chip area, a set of movable rectangles, and possibly some rectangular blockages. Additionally, we are given a set of nets. Each net consists of a set of pins, and each pin has a specified position that can be on a movable rectangle (e.g., relative to its center), or on the chip area. The half-perimeter wirelength of a net is the half perimeter of the smallest axis-aligned rectangle containing all of its pins (also called *bounding box*), possibly weighted by a net-specific weight. Then, the HPWL placement problem asks for a disjoint packing of the rectangles into the chip area that respects all blockages, minimizing the sum of half-perimeter wirelengths of all nets. See Funke, Hougardy, and Schneider [FHS16] for a formal definition.

Note that the HPWL placement problem indeed generalizes the macro legalization problem: Instead of considering the real nets that are connected to the macros, for each macro we add an artificial net with two pins, one on the macro's center, and one on the chip area at the initial position of the macro's center. Then, the half-perimeter of the bounding box of the net's pins is exactly

(a) Placement of macro fragments.



(b) Re-assembled macro shapes.



(c) Result of macro legalization, which took less than a second.

Figure 2.1: BONNMACRO placement stages.

the L_1 distance between the macro's legalized and initial positions. Using multiple nets, we can also model piecewise linear convex cost functions, and hence also minimize an approximation of the sum of squared L_2 movements.

The branch-and-bound algorithm we use to solve HPWL placement problems ([FHS16]) branches on the spatial relations of the rectangles in order to find an optimum representation (and thereby an optimum placement). During the algorithm, we maintain a set of spatial relations that have already been assigned. In order to compute an optimum (possibly overlapping if not all spatial relations are assigned) placement represented by the partial representation, one can solve a linear program. However, this linear program has a special structure and turns out to be the dual of a minimum-cost flow problem, as already noted by Cabot, Francis, and Stary [CFS70]. Hence, we can use the network simplex algorithm ([Cun76]) which is very efficient in practice. Moreover, adding spatial relations during branching corresponds to adding edges in the network flow problem, which allows to incrementally run the network simplex algorithm instead of having to solve each minimum-cost flow problem from scratch.

If additional timing constraints need to be satisfied, the linear program no longer corresponds to a minimum-cost flow problem, and thus we instead solve an integer programming formulation ([Och19]).

2.2.2 Our Contributions

Compared to an earlier version of BonnMacro ([Fun11; Eng13]), we have significantly improved the algorithms used in macro legalization.

First, we have revised the branching scheme of the core branch-and-bound rectangle packing algorithm, using spatial relations that are satisfied in the input placement as a hint. As a consequence, good solutions are found earlier by the algorithm, and hence more partial solutions can be discarded by bounding. On instances occurring in macro legalization, the running time of the branch-and-bound algorithm is reduced by a factor of 5 on average.

In order to bound the running time of the algorithm, the old BonnMacro implementation limited local instances to 4 or 5 movable macros, and all other macros contained in local instances were fixed, i.e., replaced by blockages. In joint work with Michaelis [Mic15], instead of fixing the location of the surrounding macros, we only fix the spatial relations between these surrounding macros. The set of spatial relations that the algorithm has to branch on remains the same, and hence the running time is only slightly increased, in particular due to larger minimum-cost flow problems that need to be solved. On the other hand, the solution space is significantly expanded, allowing the algorithm to find much better solutions. In order to compensate for the increased running time, we slightly reduce the number of macros with unrestricted spatial relations.

In joint work with Wochnik [Woc17], we have improved the algorithm that determines the local instances which are solved by the branch-and-bound algorithm. Here, we want to compute a rectangle on the chip that contains a

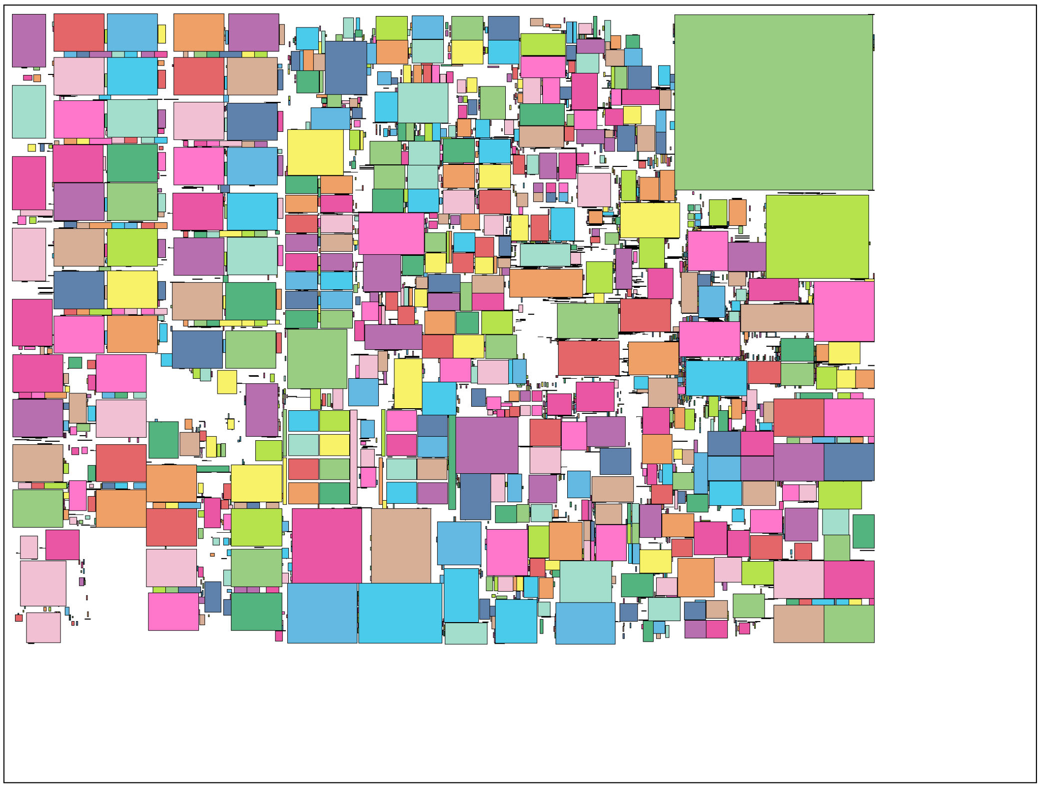


Figure 2.2: Floorplan prototype of a processor core designed with the help of BonnMacro, reducing area by 30 %. The area shown in this image corresponds to the outline of the original floorplan, the empty white border on the sides is the saved area.

given macro and is as large as possible, but does not intersect too many other macros. Using the inclusion-exclusion principle, we have designed an efficient algorithm for this problem, improving upon the previous heuristic approach.

Together with many further algorithmic and implementation improvements, these changes have significantly reduced the practical running time of BonnMacro legalization, while also improving the solution quality. A direct comparison is given in Table 2.1. On one chip, the total movement is slightly increased, on all other chips the total movement is (in some cases considerably) reduced. The running time is significantly improved on all chips, up to a factor of 90. Consequently, BonnMacro can now also be efficiently applied to large chips: For example, BonnMacro legalization was a crucial tool in the design of a floorplan prototype of an IBM processor core which reduced the total area by 30 % compared to the original layout, shown in Figure 2.2.

It remains an interesting open problem to apply our new theoretical results (cf. Chapter 5) to practical algorithms, e.g., to improve the worst-case running time of the core branch-and-bound rectangle packing algorithm.

Chip	Macros	Run	L_1 Movement [mm] Sum		Maximum		Time [s]	
1	37	old	0.6		0.07		9.1	
		new	0.5	**−6 %**	0.09	**+29 %**	0.2	**−98 %**
2	38	old	2.2		0.46		6.7	
		new	2.3	**+4 %**	0.50	**+8 %**	0.2	**−97 %**
3	72	old	3.2		0.73		17.1	
		new	2.9	**−12 %**	0.70	**−4 %**	0.5	**−97 %**
4	219	old	23.0		0.69		9.9	
		new	12.4	**−46 %**	0.41	**−41 %**	2.4	**−76 %**
5	292	old	4.8		0.11		297.8	
		new	4.2	**−14 %**	0.10	**−7 %**	3.3	**−99 %**
6	314	old	35.1		1.22		16.0	
		new	13.1	**−63 %**	0.28	**−77 %**	3.7	**−77 %**
7	2411	old	168.7		1.46		1221.5	
		new	111.3	**−34 %**	0.45	**−69 %**	19.8	**−98 %**

Table 2.1: Comparison of new BonnMacro legalization with an older version of BonnMacro from early 2015, essentially as in [Fun11; Eng13]. Column 1 identifies the chip, column 2 gives the number of macros. Columns 4 and 5 give the sum of L_1 movements of all macros in millimeters, and columns 6 and 7 give the maximum L_1 movement of any macro in millimeters. The last two columns give the running time of macro legalization in seconds. The overlapping macro placement to be legalized was a reassembled shredded placement which was placed with 85 % density. In all runs, the objective function was the minimization of the unweighted sum of L_1 movements. Note that the new BonnMacro version by default minimizes the area-weighted sum of squared L_2 movements which we modified for this experiment, allowing for a fair comparison. On chip 7, the old version failed to find a feasible placement, leaving 21 macros unlegalized. In all other cases, macro legalization was successful.

2.3 Floorplanning

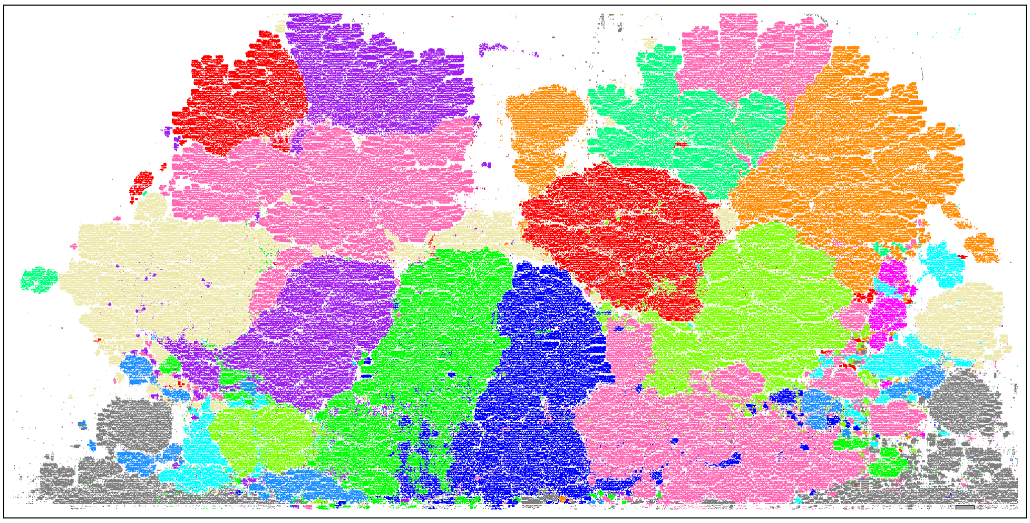
In the previous section, we have seen that in hierarchical design, the netlist of a chip is partitioned into clusters, and each cluster is assigned a rectangular shape. Then, each cluster can be designed independently of the other clusters, and the final cluster designs can then be stitched together in the integration step. *Floorplanning* is the problem of both determining the partition of the netlist into clusters, and computing rectangular shapes for these clusters. Here, we focus on the latter problem, and assume that a clustering is already given.

We have implemented a completely new tool for floorplanning, called BonnPlan. First, BonnPlan computes a global placement of the non-clustered netlist using BonnPlaceGlobal. See Figure 2.3(a) for such a global placement, colored by cluster. Then, for each cluster, we want to compute a rectangular shape that closely matches the shape of the cluster in the global placement such that the area of the shape is large enough to fit all cells of the cluster, and all cluster shapes are disjoint. Hence, the floorplanning problem is a rectangle packing problem with *flexible aspect ratios*. There may also be macros with fixed shapes, for example memory arrays.

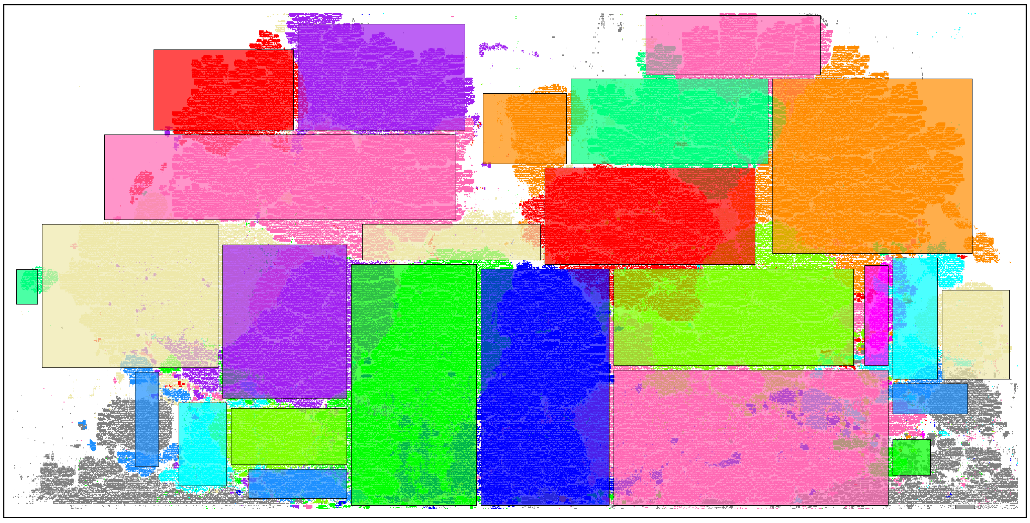
More precisely, our objective function is minimizing the sum of L_1 distances of cells to their clusters' shapes. One can show that this results in a piecewise linear cost function for each of the four boundary coordinates of every cluster shape, whose number of segments is roughly the number of cells in the cluster, which may be huge. We approximate these piecewise linear cost functions by reducing the number of segments. Moreover, for each cluster, the area of its shape is pre-determined by the sum of areas of its cell members and a target density. This induces a nonlinear dependency between the width w and the height h of the cluster shape. Again, this can be approximated by defining a piecewise linear function $f(w)$, and requiring $h \geq f(w)$.

Using these simplifications, BonnPlan computes an optimum solution for a given representation by solving a linear program (LP). More precisely, we compute the transitive reduction of the representation, that is, we eliminate redundant spatial relations which are already implied by other spatial relations. Then, we add a single disjointness constraint for each remaining spatial relation to the LP. For example, if 1 is west of 2, and 2 is west of 3, the constraint 1 west of 3 is redundant and can be omitted. A good representation is then found by local search, perturbing non-redundant spatial relations whose corresponding constraints are tight in the current LP solution. An example of a floorplan computed by BonnPlan is given in Figure 2.3(b).

BonnPlan has been successfully used at IBM to design floorplan prototypes which are then the basis for further manual adjustments.



(a) A global placement computed by BONNPLACEGLOBAL.



(b) Cluster shapes computed by BONNPLAN.

Figure 2.3: Global placement and resulting cluster shapes. The shown area is 2.2 mm wide and contains roughly 1.4 million cells. Cells are colored by cluster, gray cells do not belong to any cluster and will remain in the top hierarchy level.

Chapter 3
Preliminaries

After defining basic notation in Section 3.1, we can formally introduce rectangle placements and their representations in Section 3.2. Then, we discuss previous work in Section 3.3. In Section 3.4, we introduce the concept of pattern-avoiding permutations, which will be a key tool for our results.

3.1 Notation

First, we introduce the basic notation used in this thesis. We also refer to the glossary of notation on page 117 and the index on page 119.

3.1.1 Sets and Functions

We refer by $\mathbb{N}$ to the natural numbers excluding 0. Given a natural number $n \in \mathbb{N}$, we denote by $[\![n]\!]$ the set of integers $\{1, \ldots, n\}$.

Given a set S, the standard notation $S^2 = S \times S$ refers to the set of pairs on S. We denote by ${}^2S$ the set of ordered pairs consisting of distinct elements of S:

$$ {}^2S := S^2 \setminus \{ (i,i) : i \in S \} $$

Given a function $f\colon {}^2S \to X$, we sometimes abbreviate $f(i,j) := f\big((i,j)\big)$ for $(i,j) \in {}^2S$. We call such a function f **antisymmetric** if

$$ f(i,j) = -f(j,i) $$

for all $(i,j) \in {}^2S$.

Given a set S and an element i, we refer by $S + i$ to S together with i, and by $S - i$ to S without the element i:

$$S + i := S \cup \{ i \} \qquad\qquad S - i := S \setminus \{ i \}$$

3.1.2 Relations and Orders

Given a set S, we call a set $Q \subseteq S^2$ a **relation** on S. We will be interested in relations on $[\![n]\!]$. We say that a relation $Q \subseteq S^2$ is **transitive** if for all $(i, j), (j, k) \in Q$ we also have $(i, k) \in Q$. Moreover, we denote by $\mathrm{tr}(Q)$ the **transitive closure** of Q, that is,

$$\mathrm{tr}(Q) := \Big\{ (i, j) \in S^2 : \text{there are } i = a_1, \ldots, a_p = j \\ \text{with } (a_m, a_{m+1}) \in Q \text{ for all } 1 \leq m < p \Big\}.$$

Note that Q is transitive if and only if $\mathrm{tr}(Q) = Q$.
A relation $Q \subseteq S^2$ is called a **strict partial order** if $Q \subseteq {}^2S$ (**anti-reflexivity**) and Q is transitive. We say that i is **less** than j with respect to Q if $(i, j) \in Q$. The **reversed relation** $\overleftarrow{Q}$ of Q is given by

$$\overleftarrow{Q} := \{ (j, i) : (i, j) \in Q \}.$$

Moreover, the **symmetric closure** $\mathrm{sym}(Q)$ of Q is defined as

$$\mathrm{sym}(Q) := Q \cup \overleftarrow{Q}.$$

For example, if Q is a strict partial order, then $\mathrm{sym}(Q)$ consists of all pairs (i, j) such that i is less than j, or j is less than i with respect to Q. We call pairs $(i, j) \in \mathrm{sym}(Q)$ **comparable** with respect to Q.
A **strict total order** is a strict partial order in which every pair of distinct elements is comparable.

3.1.3 Graphs

For notation related to graphs, we closely follow Korte and Vygen [KV18].

An **undirected graph** $G = \big(V(G), E(G)\big)$ is a pair of a set of **vertices** $V(G)$ and a set of **edges** $E(G) \subseteq \big\{ \{ u, v \} : u, v \in V(G) \big\}$. We refer by $\Gamma_G(u)$ to the set of **neighbors** of a vertex u in G:

$$\Gamma_G(u) := \big\{ v \in V(G) : \{ u, v \} \in E(G) \big\}$$

A (loopless) **directed graph** (or **digraph**) $G = \big(V(G), E(G)\big)$ is a pair of a set of **vertices** $V(G)$ and a set of **edges** $E(G) \subseteq {}^2V(G)$. We refer by $\delta^+(u)$ to the set of edges leaving u, and refer by $\delta^-(u)$ to the set of edges entering u:

$$\delta^+(u) := \{ (u, v) \in E(G) \} \qquad\qquad \delta^-(u) := \{ (v, u) \in E(G) \}$$

Given a (directed or undirected) graph G and an edge e, we refer by $G+e$ to the graph G together with the new edge e, and by $G-e$ to G without the edge e. Similarly, given a set of edges F, we refer by $G+F$ to G together with the edges in F, and by $G-F$ to the graph obtained by removing all edges in F from G:

$$G+e := \big(V(G), E(G)+e\big) \qquad G+F := \big(V(G), E(G)\cup F\big)$$
$$G-e := \big(V(G), E(G)-e\big) \qquad G-F := \big(V(G), E(G)\setminus F\big)$$

Note that the edge set $E(G)$ of a digraph G is a relation on $V(G)$. Now, given a directed graph G, its **transitive closure** $\operatorname{tr}(G)$ is the directed graph on the same vertex with the transitive closure of $E(G)$ as edge set:

$$\operatorname{tr}(G) := \Big(V(G), \operatorname{tr}\big(E(G)\big)\Big)$$

A (directed) **path** is a digraph G of the form

$$V(G) = \{v_1, \ldots, v_n\}, \qquad E(G) = \big\{(v_i, v_{i+1}) : 1 \le i < n\big\}.$$

A (directed) **cycle** is a digraph G of the form

$$V(G) = \{v_1, \ldots, v_n\}, \qquad E(G) = \big\{(v_i, v_{i+1}) : 1 \le i < n\big\} \cup \big\{(v_n, v_1)\big\}.$$

A **subgraph** of a (directed or undirected) graph G is a graph H with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. We say that G **contains** H if H is a subgraph of G.

We call a digraph **acyclic** if it does not contain a cycle. Note that strict partial orders are exactly the edge sets of transitive closures of acyclic digraphs.

A **topological order** of an acyclic digraph G is a strict total order $Q \subseteq {}^2V(G)$ with $E(G) \subseteq Q$, in other words, for all edges $(u,v) \in E(G)$, the vertex u must precede the vertex v in Q. If $V(G) = [\![n]\!]$, we commonly encode a topological order of G using a permutation (cf. Section 3.4) $\pi\colon [\![n]\!] \to [\![n]\!]$, where we require $\pi(u) < \pi(v)$ for all $(u,v) \in E(G)$. Note that acyclic digraphs always have a topological order ([KV18]), but for digraphs containing a cycle a topological order trivially cannot exist.

3.2 Placements and Representations

Let $n \in \mathbb{N}$. A **rectangle placement** (also just called **placement**) is a tuple of coordinate functions $P = (\mathit{minc}_{\mathrm{x}}, \mathit{minc}_{\mathrm{y}}, \mathit{maxc}_{\mathrm{x}}, \mathit{maxc}_{\mathrm{y}})$ from $[\![n]\!]$ to $\mathbb{R}$ with, for $i \in [\![n]\!]$,

(i) $\mathit{minc}_{\mathrm{x}}(i) < \mathit{maxc}_{\mathrm{x}}(i)$, and

(ii) $\mathit{minc}_{\mathrm{y}}(i) < \mathit{maxc}_{\mathrm{y}}(i)$.

We often call the elements of $[\![n]\!]$ **rectangles**, and call P an n-placement. For each rectangle $i \in [\![n]\!]$, the **area** of i is the half-open rectangular area

$$\big[\mathit{minc}_{\mathrm{x}}(i), \mathit{maxc}_{\mathrm{x}}(i)\big) \times \big[\mathit{minc}_{\mathrm{y}}(i), \mathit{maxc}_{\mathrm{y}}(i)\big) \subseteq \mathbb{R}^2.$$

A placement P is called **feasible** if the areas of all rectangles are pairwise disjoint, that is, for all $(i, j) \in {}^2[\![n]\!]$ at least one of the following holds:

$$\begin{array}{rll} \mathit{maxc}_{\mathrm{x}}(i) \leq \mathit{minc}_{\mathrm{x}}(j) & & (i \text{ is } \textit{west} \text{ of } j \text{ in } P) \\ \mathit{maxc}_{\mathrm{y}}(i) \leq \mathit{minc}_{\mathrm{y}}(j) & & (i \text{ is } \textit{south} \text{ of } j \text{ in } P) \\ \mathit{maxc}_{\mathrm{x}}(j) \leq \mathit{minc}_{\mathrm{x}}(i) & & (i \text{ is } \textit{east} \text{ of } j \text{ in } P) \\ \mathit{maxc}_{\mathrm{y}}(j) \leq \mathit{minc}_{\mathrm{y}}(i) & & (i \text{ is } \textit{north} \text{ of } j \text{ in } P) \end{array}$$

An antisymmetric[1] function $r\colon {}^2[\![n]\!] \to \{\text{west}, \text{south}, \text{east}, \text{north}\}$ is called a **representation**, where

$$\begin{array}{ll} -\text{west} := \text{east}, & -\text{south} := \text{north}, \\ -\text{east} := \text{west}, & -\text{north} := \text{south}. \end{array}$$

We say that r **represents** a feasible placement P (or is a representation of P) if the following statements hold for all $(i, j) \in {}^2[\![n]\!]$:

$$\begin{array}{rcl} r(i, j) = \text{west} & \Longrightarrow & i \text{ is west of } j \text{ in } P \\ r(i, j) = \text{south} & \Longrightarrow & i \text{ is south of } j \text{ in } P \end{array}$$

We call a pair of functions $w, h\colon [\![n]\!] \to \mathbb{R}_{>0}$ **sizes** (or n-sizes if not clear from the context). A placement of given sizes (w, h) – also called (w, h)-placement – is a placement $(\mathit{minc}_{\mathrm{x}}, \mathit{minc}_{\mathrm{y}}, \mathit{maxc}_{\mathrm{x}}, \mathit{maxc}_{\mathrm{y}})$ with, for $i \in [\![n]\!]$,

(i) $w(i) = \mathit{maxc}_{\mathrm{x}}(i) - \mathit{minc}_{\mathrm{x}}(i)$, and

(ii) $h(i) = \mathit{maxc}_{\mathrm{y}}(i) - \mathit{minc}_{\mathrm{y}}(i)$.

In the following, we denote by R_n the set of representations on $[\![n]\!]$. Let $R \subseteq R_n$ be a set of representations. We say that R **covers** a placement P if R contains a representation of P. Moreover, R is called (w, h)-**complete** if R covers every (w, h)-placement. Finally, a set R of representations is **complete** for n if R is (w, h)-complete for all n-sizes (w, h). Note that R is complete if and only if it contains a representation of every placement of any n rectangles. We are interested in the following numbers:

$$\begin{array}{rcl} \mathit{CR}_n^{w,h} & := & \min\{\, |R| : R \subseteq R_n \text{ is } (w, h)\text{-complete} \} \\ \mathit{CR}_n^{\min} & := & \min\{\, \mathit{CR}_n^{w,h} : w, h : [\![n]\!] \to \mathbb{R}_{>0} \,\} \\ \mathit{CR}_n^{\max} & := & \max\{\, \mathit{CR}_n^{w,h} : w, h : [\![n]\!] \to \mathbb{R}_{>0} \,\} \\ \mathit{CR}_n & := & \min\{\, |R| : R \subseteq R_n \text{ is complete} \} \end{array}$$

Using this notation, the contributions of this work are new lower and upper bounds on CR_n.

[1] We require representations to be antisymmetric because the constraint of i being west of j is the *same* as requiring j to be east of i.

3.3 Previous Work

How small can a complete set of representations for n rectangles be? Obviously it needs to have cardinality at least $n!$ because for placements in which all rectangles have identical y-coordinates, we must represent all $n!$ horizontal orders. A trivial upper bound is $4^{\binom{n}{2}}$ because for each unordered pair there are four possibilities. Before this work, the best known bounds were:

$$n! \cdot 2^{n-1} \;\le\; CR_n^{\min} \;\le\; CR_n^{\max} \;\le\; CR_n \;=\; \mathcal{O}\left(\frac{n!}{n^{4.5}} \cdot 32^n\right)$$

The first inequality is due to Silvanus [Sil11], and the asymptotic upper bound is implied by a corresponding bound on the number of general floorplans by Shen and Chu [SC03], cf. Section 3.3.3. The other inequalities are trivial. For comparison, our new bounds are

$$CR_n = \Omega\left(\frac{n!}{n^4} \cdot \left(4 + 2\sqrt{2}\right)^n\right), \qquad 4 + 2\sqrt{2} \ge 6.828,$$

$$CR_n = \mathcal{O}\left(\frac{n!}{n^6} \cdot \left(\frac{11 + 5\sqrt{5}}{2}\right)^n\right), \qquad \frac{11 + 5\sqrt{5}}{2} \le 11.091.$$

3.3.1 Lower Bounds

The only known non-trivial lower bound on CR_n is the $n! \cdot 2^{n-1}$ lower bound on $CR_n^{\min}$ due to Silvanus [Sil11]. It implies that, given *any* n-sizes (w, h), every (w, h)-complete set of representations needs to contain at least $n! \cdot 2^{n-1}$ representations. The proof works as follows: Assume that rectangle 1 is placed arbitrarily. Then, given a string $s \in \{0, 1\}^{n-1}$, place rectangle $i + 1$ directly to the east of rectangle i if $s_i = 0$, and directly to the north of rectangle i if $s_i = 1$. Applying this procedure to every permutation of the rectangles we obtain $n! \cdot 2^{n-1}$ different feasible placements, and one can easily show that there are no two such placements that share a common representation.

On the contrary, the construction of our lower bound of $\Omega\left(\frac{n!}{n^4} \cdot (4 + 2\sqrt{2})^n\right)$ on CR_n relies on placements of different rectangle sizes, and hence does not apply to $CR_n^{\max}$ or even $CR_n^{\min}$.

Korte and Vygen [KV08, page 337] pose the challenge of solving the half-perimeter wirelength (HPWL) placement problem (cf. page 8 and [FHS16]) in $\mathcal{O}(n! \cdot 4^n)$ time (neglecting polynomial factors). Our new lower bound shows that no algorithm that follows the standard way of enumerating a complete set of representations can achieve this goal.

However, in the HPWL placement problem (and many other applications), the rectangle sizes are fixed, i.e., it suffices to enumerate a (w, h)-complete set of representations for some sizes (w, h) that are part of the problem input. Only lower bounds on $CR_n^{\max}$ apply to the worst-case running time of algorithms using this idea, but no such algorithm with a worst-case running time better than the best known upper bound on CR_n is known. In particular, no upper bound on $CR_n^{\max}$ stronger than CR_n is known.

On the other hand, enumerating a complete set of representations allows to optimize over *all* rectangle placements, that is, the determination of rectangle sizes can be part of the optimization problem ([IN06], also cf. Section 2.3).

3.3.2 Upper Bounds

The first non-trivial upper bound of $(n!)^2$ on CR_n was shown by Jerrum [Jer85] based on the *sequence pair* representation (rediscovered by Murata et al. [MFNK96]). Using Stirling's formula $n! = \Theta\left(\sqrt{n}\left(\frac{n}{e}\right)^n\right)$, see e.g. [KV18], we get the estimates

$$\begin{aligned} (n!)^2 &= \Theta\left(n \cdot \left(\frac{n}{e}\right)^{2n}\right) &&= \Theta\left(\frac{n}{e^{2n}} 2^{2n\log n}\right), \\ 4^{\binom{n}{2}} &= 4^{\frac{n(n-1)}{2}} &&= 2^{n^2-n}, \end{aligned} \tag{3.1}$$

which shows that $(n!)^2$ indeed is a dramatic improvement upon the trivial $4^{\binom{n}{2}}$ upper bound. The sequence pair representation maps pairs of permutations on $[\![n]\!]$ to representations. More precisely, for any pair of rectangles i, j there are four possibilities on the relative order of i and j in the two permutations. These four cases are then mapped to the four possible spatial relations. The sequence pair representation will be the basis of our new results and is discussed extensively in Chapter 4.

Nakatake et al. [NFMK96] proposed the *bounded sliceline grid* representation. Here, a two-dimensional $p \times q$ grid is considered. The rectangles are injectively assigned to vertices of the grid, and the relative positions of vertices in the grid induce the spatial relations of the rectangles. In order to ensure that all placements (in particular those consisting of n rectangles placed in a single row or column) are represented, we need to consider an $n \times n$ grid. Using

$$n^{2n} \cdot \left(1 - \frac{1}{n}\right)^n = \left(n^2 - n\right)^n \leq \frac{(n^2)!}{(n^2-n)!} \leq n^{2n},$$

it follows that the number of possible assignments of n rectangles into an $n \times n$ grid (and thus the resulting number of representations) is

$$n! \cdot \binom{n^2}{n} = \frac{(n^2)!}{(n^2-n)!} = \Theta\left(n^{2n}\right) = \Theta\left(2^{2n\log n}\right).$$

Comparing with (3.1), we see that this is worse than the sequence pair bound of $(n!)^2$ by a factor of $\Theta\left(\frac{e^{2n}}{n}\right)$.

Solution space	Size	Type	Reference
O-tree	$\Theta\left(\frac{n!}{n^{1.5}}4^n\right)$	compacted	[GCY99; Tak00]
B*-tree	$\Theta\left(\frac{n!}{n^{1.5}}4^n\right)$	compacted	[CCWW00]
corner sequence	$(n!)^2$	compacted	[LCL03]
v-h-tree	$\Theta\left(\frac{n!}{n^{1.5}}(3+\sqrt{8})^n\right)$	slicing	[SO80; YCCG03]
corner block list	$\mathcal{O}\left(\frac{n!}{n^{1.5}}8^n\right)$	mosaic	[Hon+04]
Q-sequence	$\mathcal{O}\left(\frac{n!}{n^{1.5}}8^n\right)$	mosaic	[SKM03]
twin binary sequence	$\mathcal{O}\left(\frac{n!}{n^{1.5}}8^n\right)$	mosaic	[YCS03]
twin binary tree	$\Theta\left(\frac{n!}{n^{4}}8^n\right)$	mosaic	[YCCG03]
sequence pair	$(n!)^2$	general	[Jer85; MFNK96]
bounded sliceline grid	$\Theta\left(n^{2n}\right)$	general	[NFMK96]
transitive closure graph	$(n!)^2$	general	[LC05]
TCG-S	$(n!)^2$	general	[LC04]

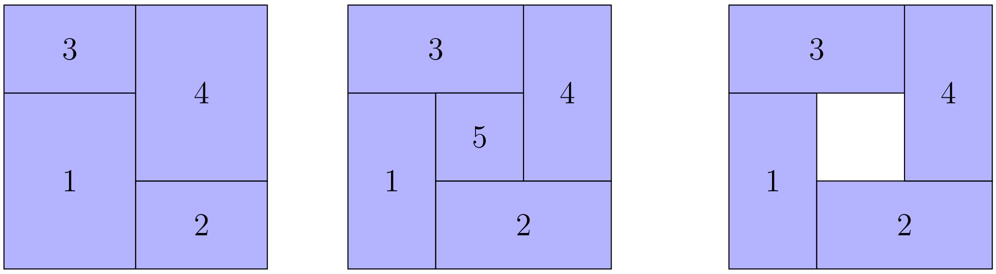
Table 3.1: Solution spaces for rectangle placements and floorplans. The second column gives the number of encodings in the solution space. The third column specifies the flexibility of the solution space: "compacted" means that only compacted placements can be represented (cf. page 3), "slicing" and "mosaic" mean that only floorplans of this type can be represented (cf. Section 3.3.3), and "general" means that the solution space induces a complete set of representations.

Many other solution spaces have been proposed, most of which can only represent placements with additional properties. An overview is given in Table 3.1, see also Young [You08] and Chen and Chang [CC08].

Note that these solution spaces are often used as the basis for a local search routine (e.g., simulated annealing), which not only depends on the size of the solution space. In this context, important properties of a solution space include the set of perturbations that can be applied to an encoding, and how fast the quality of an encoding can be evaluated.

3.3.3 Floorplans

A closely related concept uses a *floorplan* to represent the relative positions of rectangles. A floorplan is a dissection of a rectangle by horizontal and vertical



(a) A slicing floorplan.

(b) A non-slicing mosaic floorplan.

(c) A general, non-mosaic floorplan with a non-reducible empty room in the center.

Figure 3.1: Floorplan types.

line segments into m smaller rectangles, called *rooms*, some of which may be marked as *empty*. Then, $n \leq m$ rectangles can be assigned bijectively to the non-empty rooms. We refer to n (i.e., the number of non-empty rooms) as the *size* of the floorplan. A floorplan without empty rooms is called *mosaic* floorplan. A mosaic floorplan that can be obtained by recursively splitting a room vertically or horizontally into two rooms is called *slicing* floorplan. For example, the floorplan depicted in Figure 3.1(a) is slicing, and the floorplan in Figure 3.1(b) is a non-slicing mosaic floorplan. Moreover, the floorplan in Figure 3.1(b) is a general, non-mosaic floorplan.

The structure of a floorplan can be captured by *segment-room relations*: A segment s and a room r have the segment-room relation south if and only if s contains the bottom edge of r. The other cases west, north, and east are defined similarly. Then, we consider two floorplans as equivalent if there is a labeling of their rooms and segments which results in the same segment-room relations and which preserves empty rooms. Note that some authors consider an assignment of the rectangles to the non-empty rooms to be part of a floorplan. In [MFWK97, Property 5], it is shown that for each pair of rooms in a floorplan equivalence class, one can deduce a spatial relation that is satisfied by each floorplan in this equivalence class. This is proven by showing for each pair of rooms the existence of a sequence of segment-room relations that implies a spatial relation for the pair. In the remainder of this section, when we speak of floorplans, we mean equivalence classes of floorplans.

Using a bijection ([ABP06]) between Baxter permutations (cf. Section 3.4.3) and mosaic floorplans, the number of mosaic floorplans of size n is known to be $\Theta\left(\frac{8^n}{n^4}\right)$ (cf. Theorem 3.17), which was first shown by Yao et al. [YCCG03]. The same map, restricted to separable permutations, is a bijection to slicing floorplans, showing that the number of slicing floorplans of size n is $\Theta\left(\frac{(3+\sqrt{8})^n}{n^{1.5}}\right)$ (also first shown by Yao et al. [YCCG03]). Separable permutations are permutations avoiding the patterns 2413 and 3142, cf. Section 3.4.

General floorplans may contain an arbitrary number of empty rooms. Young, Chu, and Shen [YCS03] call an empty room *reducible* if it can be merged with adjacent rooms while keeping the spatial relations of the remaining non-empty rooms implied by the floorplan. For example, the empty room in the floorplan given in Figure 3.1(c) is not reducible. On the contrary, all rooms in the floorplan corresponding depicted in Figure 3.1(a) would be reducible if empty. We call a floorplan *redundant* if it contains a reducible empty room.

Zhuang et al. [ZSJK02] proved that a general floorplan of size n can contain at most $n - \left\lfloor \sqrt{4n-1} \right\rfloor \leq n$ non-reducible empty rooms. Hence, any non-redundant floorplan of size n can be obtained by starting with a mosaic floorplan of size $2n$, marking n rooms as empty, and removing any reducible empty rooms. Using Stirling's formula, this implies an upper bound of

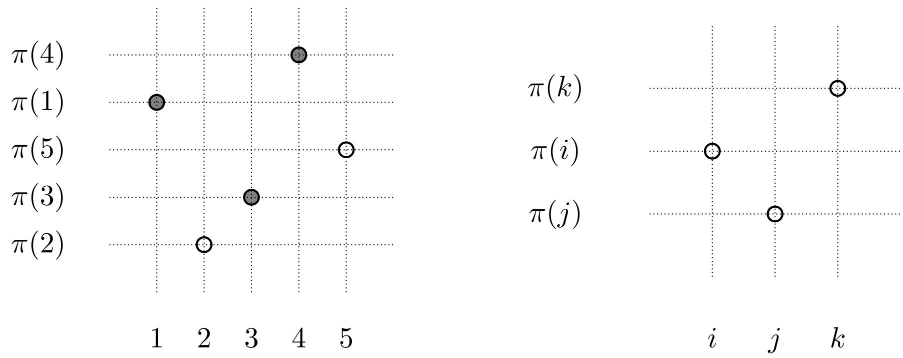
$$\mathcal{O}\left(\binom{2n}{n}\frac{8^{2n}}{n^4}\right) = \mathcal{O}\left(\frac{4^n}{\sqrt{n}} \cdot \frac{64^n}{n^4}\right) = \mathcal{O}\left(\frac{256^n}{n^{4.5}}\right)$$

on the number of general non-redundant floorplans of size n.

The best upper bound of $\mathcal{O}\left(\frac{32^n}{n^{4.5}}\right)$ was shown by Shen and Chu [SC03]. They prove that for each mosaic floorplan of size n (of which there are $\Theta\left(\frac{8^n}{n^4}\right)$ many) there are at most $\mathcal{O}\left(\frac{4^n}{\sqrt{n}}\right)$ possibilities to insert non-reducible empty rooms into the floorplan. No stronger lower bound than the number of mosaic floorplans is known.

Property 1 and Theorem 3 in [MFWK97] imply that for each placement of n rectangles, there exists a floorplan of size n and an assignment of the rectangles into the non-empty rooms such that each pair of rectangles satisfies the spatial relation implied by their rooms in the floorplan. Hence, an upper bound $U(n)$ on the number of general non-redundant floorplans of size n implies an upper bound of $U(n) \cdot n!$ on the minimum size of a complete set of representations for n rectangles.

On the contrary, lower bounds *cannot* be transferred in the same way: Our results in Chapter 5 imply $CR_4 \leq 23 \cdot 4!$ (in fact, we show $CR_4 = 22 \cdot 4!$ in Chapter 7), but there are 24 general non-redundant floorplans of size 4: There are $Baxter_4 = 22$ mosaic floorplans with 4 rooms, and there are 2 non-mosaic floorplans: the floorplan depicted in Figure 3.1(c), and the floorplan obtained by vertically flipping the one depicted in Figure 3.1(c).



(a) Dot diagram of the permutation $\pi = (\pi(1), \pi(2), \pi(3), \pi(4), \pi(5)) = (4, 1, 2, 5, 3)$. Highlighted elements form a match of the pattern on the right.

(b) The pattern 213.

Figure 3.2: Illustration of a permutation (left) and a pattern (right). The elements are ordered on the x- and y-axis according to their relative order in $<$ and $<_\pi$, respectively.

3.4 Pattern-Avoiding Permutations

Many results in this work use so-called *pattern-avoiding permutations*. We first define the basic concepts, and then, in Sections 3.4.1, 3.4.2 and 3.4.3, consider specific patterns that will be relevant later on.

A **permutation** is a bijection $\pi\colon \llbracket n \rrbracket \to \llbracket n \rrbracket$, and we denote the set of permutations on $\llbracket n \rrbracket$ by Π_n. Given any permutation π on $\llbracket n \rrbracket$, we associate π with a strict total order $<_\pi$ by defining $i <_\pi j \iff \pi(i) < \pi(j)$ for $i, j \in \llbracket n \rrbracket$. Similarly, the total order $\leq_\pi$ is given by $i \leq_\pi j \iff \pi(i) \leq \pi(j)$ for $i, j \in \llbracket n \rrbracket$.

We always denote permutations in the so-called *active* notation, that is, we write a permutation π as $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$. We illustrate permutations using dot diagrams: Given a permutation π, we draw each element i as a dot at position $(i, \pi(i))$, that is, elements are ordered on the x-axis according to $<$, and on the y-axis according to $<_\pi$. An example is given in Figure 3.2(a).

In the simplest case, a **pattern** p is just a permutation $p \in \Pi_m$. Given a permutation $\pi \in \Pi_n$, we say that π **avoids** p if there are no indices with the same pairwise comparison in p and π. More precisely, a **match** of p in π consists of indices $1 \leq a_1 < \ldots < a_m \leq n$ with $a_i <_\pi a_j \iff i <_p j$ for all $(i, j) \in {}^{2}\llbracket m \rrbracket$, and π avoids p if it does not contain a match of p. Using standard notation, we abbreviate $p = (p(1), p(2), \ldots, p(m))$ by $p = p(1)p(2)\ldots p(m)$. For example, given a permutation π, a match of the pattern 213 consists of elements $i < j < k$ with $j <_\pi i <_\pi k$. The pattern 213 is illustrated in Figure 3.2(b), and we see that the permutation given in Figure 3.2(a) contains a match of 213, given by the highlighted elements.

Sequence		Avoided patterns	Reference
powers of 2	2^{n-1}	$\{213, 312\}$	[SS85]
Catalan numbers	C_n	$\{p\}$ for any $p \in \Pi_3$	[Mac15; Knu68; Lov79]
Fibonacci numbers	F_{n+1}	$\{123, 132, 213\}$	[SS85]
Schröder[2] numbers	s_{n-1}	$\{3142, 2413\}$	[Wes95]
Bell numbers	B_n	$\{32\bar{4}1\}$	[Cal06]

Table 3.2: Well-known sequences counted by pattern-avoiding permutations in Π_n. For example, the number of permutations in Π_n avoiding both 213 and 312 is exactly 2^{n-1}. These permutations are precisely the permutations with a unique peak, and are completely determined by the set of elements that occur before that peak. Note that the last pattern $32\bar{4}1$ is a so-called *barred* battern.

In 1915, MacMahon [Mac15] proved that permutations that can be partitioned into two decreasing subsequences are counted by the Catalan numbers. Note that these permutations are exactly the 123-avoiding permutations. Pattern-avoiding permutations were first explicitly considered by Knuth [Knu68] who showed that a permutation can be sorted using a single stack if and only if it avoids the pattern 231, and that these permutations are also counted by the Catalan numbers. Note that, up to symmetry, *all* patterns of length 3 are equivalent to one of these patterns, depending on whether the 2 is in the middle or not. Hence, permutations avoiding *any* fixed pattern of length 3 are counted by the Catalan numbers. Similarly, many classical combinatorial sequences can be recovered as the number of permutations avoiding simple patterns, cf. Table 3.2.

The patterns we will be interested in are more complicated, adding additional constraints on valid matches:

A **barred** pattern contains a barred entry, and a match of a barred pattern is a match of the pattern without the barred entry that cannot be completed to a match of the pattern with the barred entry. For example, a match of $2\bar{1}3$ consists of a match of 23 that cannot be completed to a match of 213, that is, elements $i < k$ with $i <_\pi k$ such that there is no j (corresponding to the 1) with $i < j < k$ and $j <_\pi i <_\pi k$. More generally, one can define barred patterns with more than a single barred entry, see e.g. [Pud08].

Vincular patterns require that certain elements are adjacent in a match: For example, in a match of $2\underline{41}3$, the elements corresponding to 4 and 1 are required to be adjacent. This means that a match of $2\underline{41}3$ consists of elements $i < j < j+1 < l$ with $j+1 <_\pi i <_\pi l <_\pi j$. Note that in the literature, the notation 2-41-3 is more common to refer to the pattern $2\underline{41}3$.

[2]We refer to the *large* Schröder numbers.

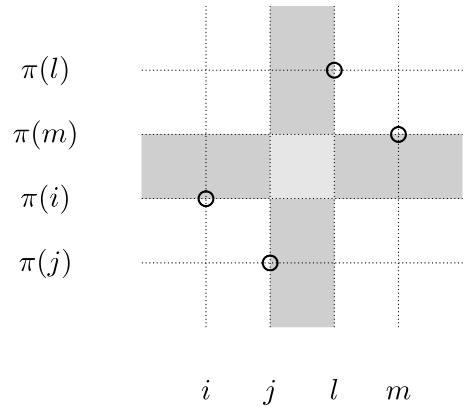


Figure 3.3: A bad quartet (i, j, l, m). The light gray square in the center is empty by the third condition in the definition of bad quartets. The bad quartet is extreme if the four darker areas are empty, too.

3.4.1 Plane Permutations

Our new upper bound will be based on *plane* permutations, which are defined using a barred pattern:

Definition 3.1. *Let π be a permutation on $[\![n]\!]$. We say that $(i, j, l, m) \in [\![n]\!]^4$ is a **bad quartet** of π if the following three conditions hold:*

(i) $i < j < l < m$,

(ii) $j <_\pi i <_\pi m <_\pi l$,

(iii) there is no $k \in [\![n]\!]$ *with* $j < k < l$ *and* $i <_\pi k <_\pi m$.

Definition 3.2. *We call a permutation π on $[\![n]\!]$ **plane** if it avoids the pattern* $21\bar{3}54$, *i.e., if π does not contain a bad quartet.*

Definition 3.3. *Let π be a permutation and let (i, j, l, m) be a bad quartet of π. We call (i, j, l, m) **extreme** if there is no $k \in [\![n]\!]$ with $j < k < l$ and there is no $k_\pi \in [\![n]\!]$ with $i <_\pi k_\pi <_\pi m$, that is, if j and l are consecutive, and i and m are consecutive in π.*

See Figure 3.3 for an illustration of (extreme) bad quartets.

Lemma 3.4. *Let π be a permutation that contains a bad quartet. Then, π contains an extreme bad quartet.*

Proof. For a permutation $\sigma \in \Pi_n$, let $d_\sigma(i, j) := \left|\{\, e \in [\![n]\!] : i <_\sigma e <_\sigma j \,\}\right|$ denote the number of elements in between i and j in permutation σ. Let (i, j, l, m)

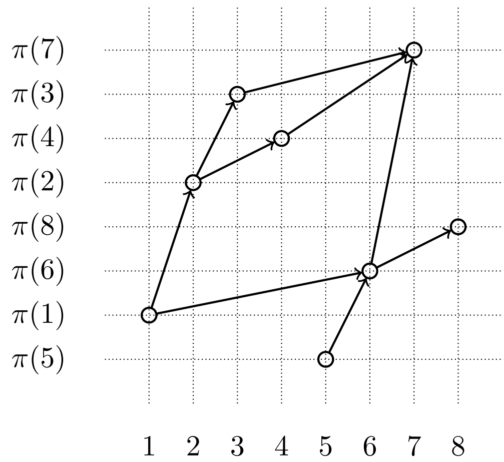


Figure 3.4: The natural embedding of G_π with $\pi = (2, 5, 7, 6, 1, 3, 8, 4)$, which is plane.

be a bad quartet such that $\Phi := d_{\mathrm{id}}(j, l) + d_\pi(i, m)$ is minimum. If Φ is zero, then (i, j, l, m) is extreme.

Otherwise, we consider two cases. Suppose first that there is $k \in [\![n]\!]$ with $j < k < l$. If $k <_\pi i$, then (i, k, l, m) is a bad quartet with smaller Φ. Otherwise, since (i, j, l, m) is a bad quartet, we have $m <_\pi k$, and (i, j, k, m) is a bad quartet with smaller Φ.

Secondly, suppose that there is $k_\pi \in [\![n]\!]$ with $i <_\pi k_\pi <_\pi m$. If $l < k_\pi$, then (i, j, l, k_π) is a bad quartet with smaller Φ. Otherwise, we have $k_\pi < j$, and (k_π, j, l, m) is a bad quartet with smaller Φ. $\square$

Corollary 3.5. *Let π a permutation. Then π is plane if and only if π avoids the vincular pattern $2\underline{14}3$, that is, if there are no indices $i < j < j+1 < m$ with $j <_\pi i <_\pi m <_\pi j+1$.*

Definition 3.6. *Given a permutation π on $[\![n]\!]$, we define an acyclic directed graph G_π with vertex set $[\![n]\!]$ whose edge set $E(G_\pi)$ consists of exactly the pairs (i, j) with*

(i) $i < j$ and $i <_\pi j$, and

(ii) there is no k with $i < k < j$ and $i <_\pi k <_\pi j$.

Observation 3.7. *Let π be a permutation on $[\![n]\!]$ and $1 \leq i, j \leq n$. Then j is reachable from i in G_π if and only if $i \leq j$ and $\pi(i) \leq \pi(j)$.*

To explain the name "plane", one can define a *natural embedding* of G_π into the plane by drawing i in $(i, \pi(i)) \in \mathbb{R}^2$ and drawing all edges as straight line segments (cf. Figure 3.4). It is known ([BB07]) that π is plane if and only if the natural embedding of G_π is plane, which we prove here for self-containedness:

Definition 3.8. *Let π be a permutation. The* ***natural embedding*** $\big(\psi, (J_e)_{e \in E(G_\pi)}\big)$ *of G_π is given by $\psi \colon V(G_\pi) \to \mathbb{R}^2$ with $\psi(i) := \big(i, \pi(i)\big)$ and $J_{(i,j)} = \big\{ (1-\lambda)\psi(i) + \lambda\psi(j) : 0 \leq \lambda \leq 1 \big\} \subseteq \mathbb{R}^2$.*

Proposition 3.9. *Let π be a permutation. Then, the natural embedding of G_π is plane if and only if π is plane.*

Proof. Let $\big(\psi, (J_e)_{e \in E(G_\pi)}\big)$ be the natural embedding of G_π. Since all edges are embedded as straight line segments with finite positive slope, for each edge (i, j) there is an affine function $f_{(i,j)} \colon [\,i, j\,] \to \mathbb{R}$ with $J_{(i,j)} = \Big\{ \big(x, f_{(i,j)}(x)\big) : x \in [\,i, j\,] \Big\}$. Note that by construction, the only vertices that can intersect the embedding of an edge are its endpoints, since for any vertex k with $\psi(k) \in J_{(i,j)}$ and $k \notin \{\, i, j \,\}$ we must have $i < k < j$ and $i <_\pi k <_\pi j$, contradicting $(i, j) \in G_\pi$.

For the first direction, assume that π is not plane. By Lemma 3.4, we know that π contains an extreme bad quartet (i, j, l, m). In particular, we then have $(i, m), (j, l) \in E(G_\pi)$, and clearly $J_{(i,m)}$ and $J_{(j,l)}$ must intersect: We have $f_{(j,l)}(j) = \pi(j) < \pi(i) < f_{(i,m)}(j)$ and $f_{(j,l)}(l) = \pi(l) > \pi(m) > f_{(i,m)}(l)$, so $f_{(j,l)}$ and $f_{(i,m)}$ intersect on the interval $[\,j, l\,]$.

For the other direction, assume that $\big(\psi, (J_e)_{e \in E(G_\pi)}\big)$ is not plane, and let $(i, m), (j, l) \in E(G_\pi)$ be two edges whose embeddings intersect, i.e., we have $J_{(i,m)} \cap J_{(j,l)} \neq \emptyset$ and i, m, j, l are pairwise different. By definition of G_π, we have $i < m$, $\pi(i) < \pi(m)$, $j < l$, and $\pi(j) < \pi(l)$. W.l.o.g. we can assume $i < j$, and since edges are embedded as straight line segments we must have $j < m$, otherwise $J_{(i,m)}$ and $J_{(j,l)}$ cannot intersect.

If $\pi(i) < \pi(j)$, then we must have $\pi(m) < \pi(j)$, otherwise (i, m) would not be an edge of G_π. But this implies $\pi(i) < \pi(m) < \pi(j) < \pi(l)$, contradicting $J_{(i,m)} \cap J_{(j,l)} \neq \emptyset$. Hence, we know that $\pi(j) < \pi(i)$ and summarize

$$i < j < l, m \qquad \text{and} \qquad \pi(j) < \pi(i) < \pi(l), \pi(m). \tag{3.2}$$

Now, we consider the two cases $m < l$ and $l < m$. The first case will lead to a contradiction, in the second case we will show that π is not plane.

In the first case, we have $i < j < m < l$, and then the edge (j, l) together with (3.2) implies $\pi(l) < \pi(m)$. Thus, we have $\pi(j) < \pi(i) < \pi(l) < \pi(m)$. But now $J_{(i,m)}$ and $J_{(j,l)}$ cannot intersect: Clearly, any point $(x, y) \in J_{(i,m)} \cap J_{(j,l)}$ needs to satisfy $j \leq x \leq m$. We have $f_{(j,l)}(j) = \pi(j) < \pi(i) < f_{(i,m)}(j)$ and $f_{(j,l)}(m) < \pi(l) < \pi(m) = f_{(i,m)}(m)$, so $f_{(j,l)}$ and $f_{(i,m)}$ do not intersect on the interval $[\,j, m\,]$.

In the second case, we have $i < j < l < m$. Now, the edge (i, m) together with (3.2) implies $\pi(m) < \pi(l)$. Hence, we have $\pi(j) < \pi(i) < \pi(m) < \pi(l)$. Finally, an element k with $j < k < l$ and $\pi(i) < k < \pi(m)$ would contradict that (i, m) and (j, l) are edges of G_π, so (i, j, l, m) is a bad quartet of π and thus π is not plane. $\square$

The number of plane permutations was recently analyzed by Bouvel et al. [BGRR18], solving an open problem due to Bousquet-Mélou and Butler [BB07]:

Theorem 3.10 ([BGRR18])**.** *For $n \in \mathbb{N}$, denote by $Plane_n$ the number of plane permutations on $[\![n]\!]$. Then, for all $n \geq 2$, we have*

$$\begin{aligned} Plane_n &= \frac{24}{(n-1)n^2(n+1)(n+2)} \sum_{k=0}^{n} \binom{n+1}{k+3}\binom{n+2}{k+1}\binom{n+k+3}{k} \\ &= \Theta\left(\frac{C^n}{n^6}\right), \end{aligned}$$

where $C = \frac{11+5\sqrt{5}}{2} < 11.091$.

For $n \leq 15$, $Plane_n$ is given in Table 3.3 (page 32). Further values can be obtained from *The On-Line Encyclopedia of Integer Sequences* (OEIS) [Slo18], sequence A117106.

Bouvel et al. [BGRR18] prove Theorem 3.10 as follows: First, they observe that given a plane permutation $\pi \in \Pi_n$, removing the last element n (i.e., restricting the total order of π to the first $n-1$ elements) again yields a plane permutation π'. Hence every plane permutation on $[\![n]\!]$ can be uniquely generated by starting with a plane permutation π' on $[\![n-1]\!]$, and inserting n into the strict total order of π'. This operation is called *local expansion*. Analyzing the structure of possible local expansions, they prove properties of the generating function of $Plane_n$. This allows to obtain a recurrence formula for $Plane_n$ which then yields the closed-form expression. In order to show the tight bound $Plane_n = \Theta(\frac{1}{n^6}(\frac{11+5\sqrt{5}}{2})^n)$, Bouvel et al. [BGRR18] apply a result of McIntosh [McI96] which allows to estimate sums of binomial coefficients.

Now, we briefly sketch how the asymptotic growth rate of $Plane_n$ can be derived directly from the closed-form expression, ignoring polynomial factors. Stirling's formula $n! = \Theta\left(\sqrt{n}\left(\frac{n}{e}\right)^n\right) \approx \left(\frac{n}{e}\right)^n$ allows to estimate

$$\binom{\beta n}{\alpha n} \approx \frac{\left(\frac{\beta n}{e}\right)^{\beta n}}{\left(\frac{\alpha n}{e}\right)^{\alpha n}\left(\frac{(\beta-\alpha)n}{e}\right)^{(\beta-\alpha)n}} = \left(\frac{\beta^\beta}{\alpha^\alpha(\beta-\alpha)^{\beta-\alpha}}\right)^n,$$

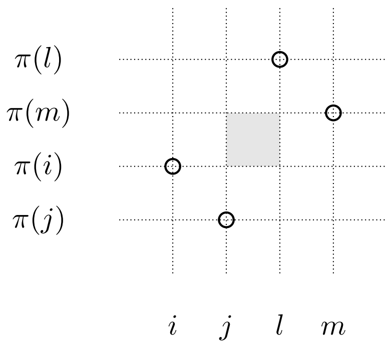
and hence, substituting $k = \alpha n$ with $0 < \alpha < 1$, we obtain the estimate

$$\binom{n}{k}\binom{n}{k}\binom{n+k}{k} \approx \left(\frac{(1+\alpha)^{1+\alpha}}{\alpha^{3\alpha}(1-\alpha)^{2-2\alpha}}\right)^n.$$

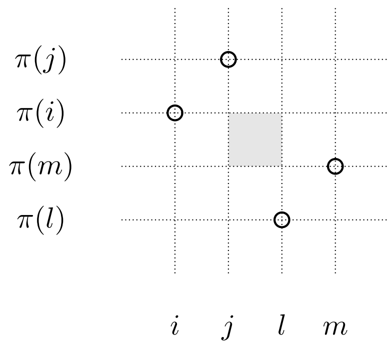
Then, the growth rate of $Plane_n$ is determined by

$$\max_{0<\alpha<1} \frac{(1+\alpha)^{1+\alpha}}{\alpha^{3\alpha}(1-\alpha)^{2-2\alpha}} = \frac{11+5\sqrt{5}}{2} = \phi^5,$$

which is attained at $\alpha = \frac{\sqrt{5}-1}{2} = \frac{1}{\phi}$, where $\phi = \frac{\sqrt{5}+1}{2}$ is the golden ratio.



(a) The pattern $21\bar{3}54$.



(b) The pattern $45\bar{3}12$.

Figure 3.5: Forbidden patterns of biplane permutations. Gray areas are assumed to be empty.

3.4.2 Biplane Permutations

Definition 3.11. *Let π be a permutation on $[\![n]\!]$. The **reversed** permutation $-\pi$ is defined by*

$$-\pi(i) := n + 1 - \pi(i)$$

for $i \in [\![n]\!]$.

Clearly, reversing a permutation π reverses the strict total order associated with π: We have $i <_\pi j$ if and only if $j <_{-\pi} i$.

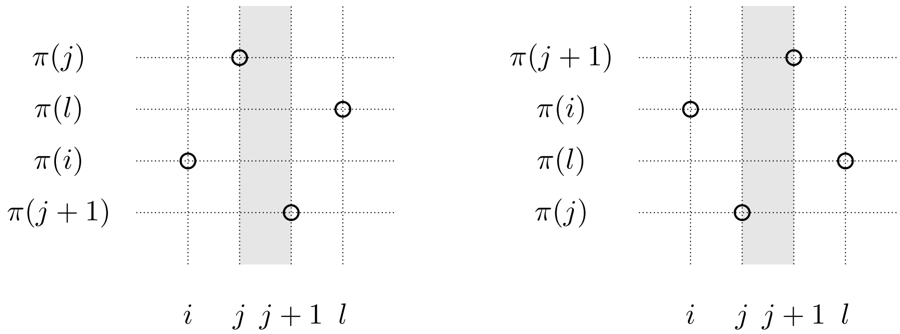
Observation 3.12. *Let π be a permutation on $[\![n]\!]$, and let $1 \leq i < j \leq n$. Then j is reachable from i in G_π if and only if j is not reachable from i in $G_{-\pi}$.*

Definition 3.13. *Let π be a permutation on $[\![n]\!]$. We call π **biplane** if π avoids the patterns $21\bar{3}54$ and $45\bar{3}12$, that is, if both π and $-\pi$ are plane.*

The patterns forbidden in biplane permutations are illustrated in Figure 3.5. For example, the permutation π depicted in Figure 3.4 is not biplane, as the elements $2, 4, 5, 8$ form a match of $45\bar{3}12$. Note that a permutation π is biplane if and only if $-\pi$ is biplane. Corollary 3.5 directly implies:

Corollary 3.14. *Let π be a permutation. Then π is biplane if and only if π avoids the vincular patterns $2\underline{14}3$ and $3\underline{41}2$.*

Asinowski et al. [Asi+13] have analyzed the number of permutations avoiding the patterns $2\underline{14}3$ and $3\underline{41}2$:



(a) The pattern $2\underline{41}3$. **(b)** The pattern $3\underline{14}2$.

Figure 3.6: The forbidden patterns of Baxter permutations. Gray areas are assumed to be empty.

Theorem 3.15 ([Asi+13])**.** *For $n \in \mathbb{N}$, denote by $Biplane_n$ the number of biplane permutations on $[\![n]\!]$. Then, we have*

$$Biplane_n = \Theta\left(\frac{C^n}{n^4}\right),$$

where $C = 4 + 2\sqrt{2} \geq 6.828$.

For $n \leq 15$, $Biplane_n$ is given in Table 3.3 (page 32). Further values can be obtained from the OEIS [Slo18], sequence A214358.

3.4.3 Baxter Permutations

Definition 3.16. *We call a permutation π on $[\![n]\!]$* ***Baxter*** *permutation if it avoids the patterns $2\underline{41}3$ and $3\underline{14}2$, that is, there are no indices $i < j < l$ with $j+1 <_\pi i <_\pi l <_\pi j$ or $j <_\pi l <_\pi i <_\pi j+1$.*

An illustration of the forbidden pattern of Baxter permutations is given in Figure 3.6. Baxter permutations were first considered by Baxter [Bax64] when studying the structure of fixpoints of certain functions. The number of Baxter permutations was first analyzed by Chung et al. [CGHK78]:

Theorem 3.17 ([CGHK78])**.** *For $n \in \mathbb{N}$, denote by $Baxter_n$ the number of Baxter permutations on $[\![n]\!]$. Then, we have*

$$Baxter_n = \sum_{k=1}^{n} \frac{\binom{n+1}{k-1}\binom{n+1}{k}\binom{n+1}{k+1}}{\binom{n+1}{1}\binom{n+1}{2}} = \Theta\left(\frac{8^n}{n^4}\right).$$

For $n \leq 15$, $Baxter_n$ is given in Table 3.3. Further values can be obtained from the OEIS [Slo18], sequence A001181.

n	$Biplane_n$	$Baxter_n$	$Plane_n$	$n!$
1	1	1	1	1
2	2	2	2	2
3	6	6	6	6
4	22	22	23	24
5	88	92	104	120
6	374	422	530	720
7	1 668	2 074	2 958	5 040
8	7 744	10 754	17 734	40 320
9	37 182	58 202	112 657	362 880
10	183 666	326 240	750 726	3 628 800
11	929 480	1 882 960	5 207 910	39 916 800
12	4 803 018	11 140 560	37 387 881	479 001 600
13	25 274 088	67 329 992	276 467 208	6 227 020 800
14	135 132 886	414 499 438	2 097 763 554	87 178 291 200
15	732 779 504	2 593 341 586	16 282 567 502	1 307 674 368 000

Table 3.3: The number of biplane (column 2), Baxter (column 3) and plane (column 4) permutations for $n \leq 15$, compared to $n!$ (last column).

Chapter 4

Sequence Pairs

In this chapter, we review the sequence pair representation of Jerrum [Jer85] (rediscovered by Murata et al. [MFNK96]), which will be the basis for our new results. Sequence pairs provide an elegant way to construct representations that inherently satisfy useful properties:

Definition 4.1. *Let (π, ρ) be a pair of permutations on $[\![n]\!]$ (called a* ***sequence pair****). Then, we define the representation $r_{\pi,\rho}$ by*

$$r_{\pi,\rho}(i,j) := \begin{cases} \textit{south} & \textit{if } i <_\pi j \textit{ and } i <_\rho j, \\ \textit{west} & \textit{if } i <_\pi j \textit{ and } j <_\rho i, \\ \textit{north} & \textit{if } j <_\pi i \textit{ and } j <_\rho i, \\ \textit{east} & \textit{if } j <_\pi i \textit{ and } i <_\rho j. \end{cases}$$

First, we observe that if $r_{\pi,\rho}(i,j) = r_{\pi,\rho}(j,k) = \alpha$, then we must have $r_{\pi,\rho}(i,k) = \alpha$. We will call this property transitivity.

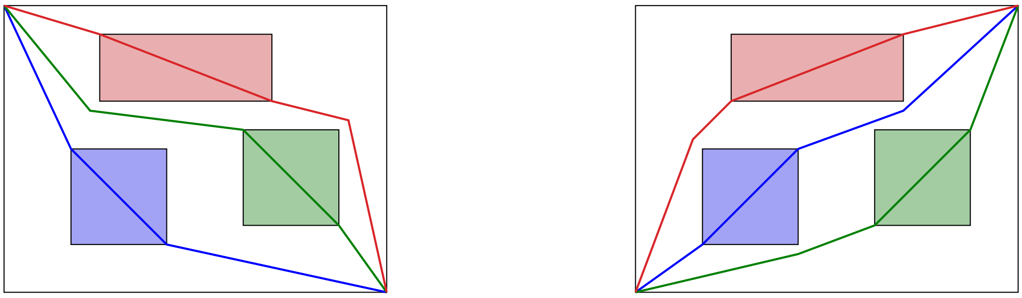
Moreover, we will see that for each sequence pair (π, ρ) there is a placement that is represented by $r_{\pi,\rho}$, and for each placement P there is a sequence pair that represents P. The latter result implies:

Theorem 4.2 ([Jer85])**.** *Let $n \in \mathbb{N}$. Then, the set*

$$\{\, r_{\pi,\rho} : (\pi, \rho) \textit{ is a sequence pair on } [\![n]\!] \,\}$$

is a complete set of representations.

In Section 4.1, we begin by briefly sketching the way Murata et al. [MFNK96] construct sequence pairs, and derive useful structural properties of sequence



(a) Negative steplines. **(b)** Positive steplines.

Figure 4.1: Negative and positive steplines of a feasible rectangle placement.

pairs. Then, in Section 4.2, we introduce the setting of our construction which is based on pairs of strict partial orders. In Section 4.3, we give a new construction (similar to [Jer85]) and recover the results of [Jer85]. The proof of our improved upper bound (Chapter 5) will be a direct improvement of this construction. Moreover, we will show that there is a complete set of representations of minimum cardinality that only consists of representations induced by sequence pairs, giving an even stronger motivation to study sequence pairs. This result is new and will be important in Chapters 6 and 7.

Finally, in Section 4.4, we describe an application of sequence pairs to a completely different problem: We give an efficient reachability oracle for arborescences based on sequence pairs. The result itself is not new, but shows that sequence pairs might be useful in different contexts.

4.1 Introduction

4.1.1 Geometric Construction: Steplines

Our new construction is very similar to the one given by Jerrum [Jer85], both of which are based on partial orders. Before we discuss these, we first briefly sketch the completely different, geometric construction of Murata et al. [MFNK96].

More precisely, the construction of Murata et al. [MFNK96] is based on so-called *steplines*, which are (usually piecewise linear) paths in the plane. Given a feasible placement P, we first draw a bounding box around P, that is, a rectangle containing all rectangles in P. Then, for each rectangle i, we construct a *negative stepline*, which is a strictly decreasing path that starts in the north west corner of the bounding box, visits the north west and south east corners of i, ends in the south east corner of the bounding box, and does not intersect any of the other rectangles. Similarly, one can define *positive steplines* that connect the south west corner of the bounding box with the north east corner of the bounding box. Furthermore, we require that any two distinct negative steplines may only intersect in their endpoints, and any two distinct positive steplines may only intersect in their endpoints. Murata et al. [MFNK96] show that it is always possible to find steplines that satisfy these requirements. This construction is illustrated in Figure 4.1.

It is easy to see that the negative steplines define an order of the rectangles from south west to north east, and the positive steplines define an order from south east to north west. These two orders are the permutations π and ρ of the sequence pair (π, ρ). Furthermore, it is not hard to show that if a rectangle i precedes a rectangle j in the south west to north east order π, then i must be south or west of j. Similarly, if i precedes j in ρ, then i must be south or east of j. Hence, if i precedes j in both orders, then i must be south of j, and if i precedes j in π but j precedes i in ρ, then i must be west of j. This way, Murata et al. [MFNK96] show that for every feasible placement P, there exists a sequence pair (π, ρ) such that $r_{\pi,\rho}$ represents P, implying Theorem 4.2.

Before we continue, we fix some notation related to sequence pairs: We refer by $\mathcal{SP}_n := \Pi_n^2$ to the set of sequence pairs on $[\![n]\!]$, and say that a sequence pair $(\pi, \rho) \in \mathcal{SP}_n$ **represents** a placement P if $r_{\pi,\rho}$ represents P. Moreover, we say that a set $\mathcal{SP} \subseteq \mathcal{SP}_n$ **covers** a placement P if $\{ r_{\pi,\rho} : (\pi, \rho) \in \mathcal{SP} \}$ covers P, that is, $\mathcal{SP}$ contains a sequence pair that represents P, and say that $\mathcal{SP}$ is **complete** if $\{ r_{\pi,\rho} : (\pi, \rho) \in \mathcal{SP} \}$ is complete.

4.1.2 Structural Permutations

Consider a feasible placement P and a sequence pair (π, ρ) that represents P. Moreover, assume that we re-label the rectangles in P according to some permutation τ, resulting in a placement P_τ. Now, also applying τ to (π, ρ) yields a new sequence pair $(\pi \circ \tau, \rho \circ \tau)$ that represents P_τ. Clearly, the sequence pairs (π, ρ) and $(\pi \circ \tau, \rho \circ \tau)$ have the same structure, which we call *structure-equivalence*:

Definition 4.3. *Let $n \in \mathbb{N}$, and let $(\pi, \rho), (\pi', \rho')$ be sequence pairs on $[\![n]\!]$. We say that (π, ρ) and (π', ρ') are **structure-equivalent** if there is a permutation $\tau \in \Pi_n$ with*

$$(\pi', \rho') = (\pi \circ \tau, \rho \circ \tau).$$

Definition 4.4. *Let (π, ρ) be a sequence pair. Then, we denote by*

$$\operatorname{struc}(\pi, \rho) \;:=\; \rho \circ \pi^{-1}$$

*the **structural permutation** of (π, ρ).*

Lemma 4.5. *Let $n \in \mathbb{N}$, and let $(\pi, \rho), (\pi', \rho')$ be sequence pairs on $[\![n]\!]$. Then, the sequence pairs (π, ρ) and (π', ρ') are structure-equivalent if and only if $\operatorname{struc}(\pi, \rho) = \operatorname{struc}(\pi', \rho')$.*

Proof. Clearly, if there is a permutation τ with $(\pi', \rho') = (\pi \circ \tau, \rho \circ \tau)$, then we have

$$\begin{aligned}
\operatorname{struc}\big(\pi', \rho'\big) &= \rho' \circ \pi'^{-1} \\
&= (\rho \circ \tau) \circ (\pi \circ \tau)^{-1} \\
&= (\rho \circ \tau) \circ (\tau^{-1} \circ \pi^{-1}) \\
&= (\rho \circ \pi^{-1}) \\
&= \operatorname{struc}(\pi, \rho).
\end{aligned}$$

On the other hand, if $\rho \circ \pi^{-1} = \rho' \circ \pi'^{-1}$, we can set $\tau := \pi^{-1} \circ \pi'$. Then, we have

$$\begin{aligned}
\pi \circ \tau &= \pi \circ (\pi^{-1} \circ \pi') \\
&= \pi'
\end{aligned}$$

and

$$\begin{aligned}
\rho \circ \tau &= \rho \circ (\pi^{-1} \circ \pi') \\
&= (\rho \circ \pi^{-1}) \circ \pi' \\
&= (\rho' \circ \pi'^{-1}) \circ \pi' \\
&= \rho'.
\end{aligned}$$

□

This observation is not new, in fact, the algorithm given by Bousquet-Mélou and Butler [BB07] that bijectively maps (unlabeled) mosaic floorplans to Baxter permutations first generates two labelings π and ρ, and then returns $\rho \circ \pi^{-1}$.

We extend the notion of pattern avoidance from permutations to sequence pairs using structural permutations: We say that a sequence pair (π, ρ) avoids a pattern if $\operatorname{struc}(\pi, \rho)$ avoids the pattern. For example, we call (π, ρ) plane if $\operatorname{struc}(\pi, \rho)$ is plane. Our new upper bound on CR_n presented in Chapter 5 considers plane sequence pairs, while the improved lower bound in Chapter 6 is based on biplane sequence pairs. Moreover, the empirical experiments in Chapter 7 suggest that complete sets of representations of minimum cardinality are also induced by sequence pairs avoiding a certain pattern.

Finally, we observe that $(\pi, \rho) = (\pi, \operatorname{struc}(\pi, \rho) \circ \pi)$, and hence the set of sequence pairs avoiding a certain pattern can be written as the set of sequence pairs of the form $(\pi, \sigma \circ \pi)$ where σ avoids the pattern. For example, the set of plane sequence pairs on $[\![n]\!]$ is the set

$$\big\{\, (\pi, \sigma \circ \pi) : \pi, \sigma \in \Pi_n,\ \sigma \text{ is plane} \,\big\}.$$

Using this notation, the permutation σ defines the structure of the sequence pair, and the permutation π defines a labeling of the elements. The classical

$(n!)^2$ upper bound on CR_n based on sequence pairs can thus be interpreted as enumerating $n!$ possible sequence pair structures, and all $n!$ possible labelings of the rectangles. Our new upper bound given in Chapter 5 still requires to enumerate all $n!$ possible labelings, but reduces the required number of structures from $n!$ to $\mathcal{O}\left(\frac{1}{n^6} \cdot (\frac{11+5\sqrt{5}}{2})^n\right)$.

4.2 Strict Partial Orders and Biorders

Similar to Jerrum [Jer85], we will show structural results on certain pairs of strict partial orders, and then apply these to rectangle placements. However, in contrast to Jerrum [Jer85], we will consider *strict* partial orders, which are equivalent to partial orders, but will simplify notation:

Definition 4.6. *Let $P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$ be a placement on $[\![n]\!]$. The strict partial orders $\mathcal{S}_P, \mathcal{W}_P \subset {}^2[\![n]\!]$ (corresponding to the spatial relations south and west) are defined as*

$$\begin{aligned} \mathcal{S}_P &:= \left\{ (i,j) \in {}^2[\![n]\!] : maxc_{\mathrm{y}}(i) \leq minc_{\mathrm{y}}(j) \right\}, \\ \mathcal{W}_P &:= \left\{ (i,j) \in {}^2[\![n]\!] : maxc_{\mathrm{x}}(i) \leq minc_{\mathrm{x}}(j) \right\}. \end{aligned}$$

Clearly, the set of representations of a feasible placement P only depends on $\mathcal{S}_P$ and $\mathcal{W}_P$. In a feasible placement, we know that each pair i, j of rectangles must be comparable in at least one of $\mathcal{S}_P$ and $\mathcal{W}_P$. We will see that this property is sufficient to obtain the $(n!)^2$ upper bound.

Definition 4.7. *Let $\mathcal{S}, \mathcal{W}$ be strict partial orders on $[\![n]\!]$. We say that $(\mathcal{S}, \mathcal{W})$ is a* ***biorder*** *(or* ***biordering*** *pair) if*

$${}^2[\![n]\!] = \mathrm{sym}(\mathcal{S}) \cup \mathrm{sym}(\mathcal{W}),$$

that is, if every pair $(i,j) \in {}^2[\![n]\!]$ is comparable in at least one of $\mathcal{S}$ and $\mathcal{W}$.

Most results will apply to general biorders, however, the used notation will be based on the application to rectangle placements, where the set $\mathcal{S}$ corresponds to the south-relations and the set $\mathcal{W}$ corresponds to the west-relations.

Observation 4.8. *Let P be a feasible placement. Then $(\mathcal{S}_P, \mathcal{W}_P)$ is a biorder.*

Every sequence pair naturally induces a pair of strict partial orders on $[\![n]\!]$:

Definition 4.9. *Let (π, ρ) be a sequence pair on $[\![n]\!]$. The strict partial orders $\mathcal{S}_{\pi,\rho}, \mathcal{W}_{\pi,\rho}$ on $[\![n]\!]$ are given by*

$$\begin{aligned} \mathcal{S}_{\pi,\rho} &:= \left\{ (i,j) \in {}^2[\![n]\!] : i <_\pi j \text{ and } i <_\rho j \right\} = r_{\pi,\rho}^{-1}(south), \\ \mathcal{W}_{\pi,\rho} &:= \left\{ (i,j) \in {}^2[\![n]\!] : i <_\pi j \text{ and } j <_\rho i \right\} = r_{\pi,\rho}^{-1}(west). \end{aligned}$$

We say that (π, ρ) ***represents*** *a biorder $(\mathcal{S}, \mathcal{W})$ if $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$.*

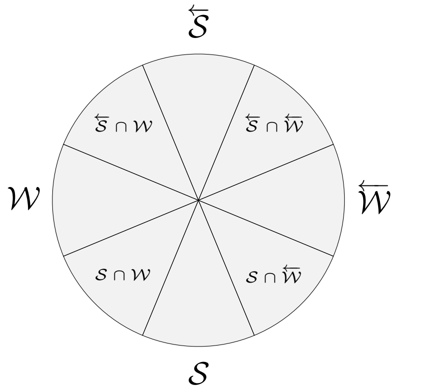


Figure 4.2: The sets $\mathcal{S}, \mathcal{W}, \overleftarrow{\mathcal{S}}, \overleftarrow{\mathcal{W}}$ and their possible intersections. The unlabeled segments contain the pairs that are in $\mathcal{S} \setminus \left(\mathcal{W} \cup \overleftarrow{\mathcal{W}}\right)$, $\mathcal{W} \setminus \left(\mathcal{S} \cup \overleftarrow{\mathcal{S}}\right)$, etc.

Clearly, (π, ρ) represents a placement P if and only if (π, ρ) represents $(\mathcal{S}_P, \mathcal{W}_P)$. It is easy to verify that $\mathcal{S}_{\pi,\rho}$ and $\mathcal{W}_{\pi,\rho}$ are not only strict partial orders, but also form a biorder. Moreover, every pair (i, j) is comparable in *exactly* one of $\mathcal{S}_{\pi,\rho}$ and $\mathcal{W}_{\pi,\rho}$. Pairs of partial orders with this property are called complementary in [Jer85]:

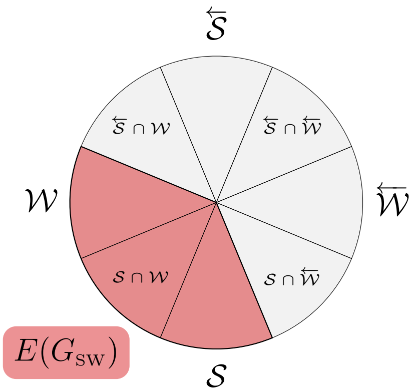
Definition 4.10. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$. We say that $(\mathcal{S}, \mathcal{W})$ is* ***complementary*** *if $\operatorname{sym}(\mathcal{S}) \cap \operatorname{sym}(\mathcal{W}) = \emptyset$, that is, if every pair $(i, j) \in {}^2[\![n]\!]$ is comparable in exactly one of $\mathcal{S}$ and $\mathcal{W}$.*

Jerrum [Jer85] then shows that the mapping from sequence pairs to complementary pairs given in Definition 4.9 is a bijection, which was already observed by Dushnik and Miller [DM41]. This means that every complementary pair of strict partial orders is of the form $(\mathcal{S}_{\pi,\rho}, \mathcal{W}_{\pi,\rho})$ for some sequence pair (π, ρ). Furthermore, Jerrum [Jer85] shows that for each biorder $(\mathcal{S}, \mathcal{W})$, there is a complementary pair $(\mathcal{S}', \mathcal{W}')$ with $\mathcal{S}' \subseteq \mathcal{S}$ and $\mathcal{W}' \subseteq \mathcal{W}$, and hence each biorder is represented by some sequence pair, which implies the $(n!)^2$ upper bound.

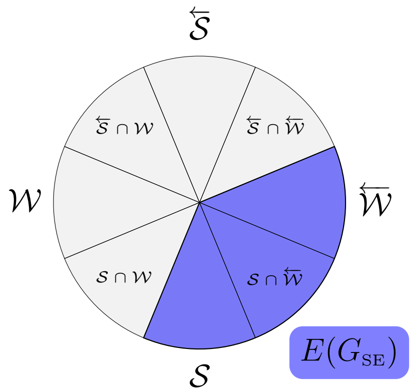
4.3 The New Construction: Biorder Digraphs

Now, we will show a construction slightly different to the one given by Jerrum [Jer85] from which these results can be recovered. Given a biorder $(\mathcal{S}, \mathcal{W})$, we want to find a sequence pair (π, ρ) representing $(\mathcal{S}, \mathcal{W})$, that is, $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$.

First, we observe that the strict partial orders $\mathcal{S}$ and $\mathcal{W}$ together with their reversed orders $\overleftarrow{\mathcal{S}}$ and $\overleftarrow{\mathcal{W}}$ exhibit a nice structure, depicted in Figure 4.2: Each pair $(i, j) \in {}^2[\![n]\!]$ is contained in exactly one of the eight segments, and (j, i) is in the segment opposite of the segment containing (i, j). The idea will be to



(a) The edge set of G_{SW}.



(b) The edge set of G_{SE}.

Figure 4.3: The edge sets of the south-west and south-east digraphs of a biorder $(\mathcal{S}, \mathcal{W})$.

encode π and ρ as topological orders of digraphs G_{SW} and G_{SE} on the vertex set $[\![n]\!]$ (formally defined later in Definition 4.11).

To motivate the construction, we derive some necessary properties of G_{SW} and G_{SE}: Assume that G_{SW} and G_{SE} are some digraphs with topological orders π and ρ and consider a pair (i, j) with $j <_\pi i$. Then, we have $(j, i) \in \mathcal{S}_{\pi,\rho} \cup \mathcal{W}_{\pi,\rho}$ (cf. Definition 4.9). Hence, if $(j, i) \notin \mathcal{S} \cup \mathcal{W}$, then $\mathcal{S}_{\pi,\rho} \nsubseteq \mathcal{S}$ or $\mathcal{W}_{\pi,\rho} \nsubseteq \mathcal{W}$, and thus (π, ρ) cannot represent $(\mathcal{S}, \mathcal{W})$. This implies that whenever

$$(j, i) \notin \mathcal{S} \cup \mathcal{W} \iff (i, j) \notin \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}},$$

we must guarantee that j is reachable from i in G_{SW} in order to avoid $j <_\pi i$. Since we are only interested in *topological orders* of G_{SW}, we can thus require that G_{SW} contains all edges (i, j) with $(i, j) \notin \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$. Similarly, if $(i, j) \notin \overleftarrow{\mathcal{S}} \cup \mathcal{W}$, we must ensure that j is reachable from i in G_{SE}, and hence require that G_{SE} contains all edges (i, j) with $(i, j) \notin \overleftarrow{\mathcal{S}} \cup \mathcal{W}$. We will show that these edges in fact do suffice:

Definition 4.11. *Let* $(\mathcal{S}, \mathcal{W})$ *be a biorder on* $[\![n]\!]$*. The* ***south-west digraph*** G_{SW} *and* ***south-east digraph*** G_{SE} *are digraphs with vertex set* $[\![n]\!]$*, and with edge sets:*

$$\begin{aligned} E(G_{\text{SW}}) &:= \Big(\mathcal{S} \cup \mathcal{W}\Big) \setminus \Big(\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}\Big) \\ E(G_{\text{SE}}) &:= \Big(\mathcal{S} \cup \overleftarrow{\mathcal{W}}\Big) \setminus \Big(\overleftarrow{\mathcal{S}} \cup \mathcal{W}\Big) \end{aligned}$$

Figure 4.3 gives an illustration of G_{SW} and G_{SE}. The edge set of G_{SW} can also be thought of as the disjoint union of the three segments "only south", "south and west" and "only west". If an element j is reachable from i in one of these three segments, then of course j is also reachable from i in G_{SW}. The following result implies Corollary 4.13 which states that the reverse implication does also hold, that is, whenever j is reachable from i in G_{SW}, we know that j is reachable from i in one of the three segments.

Lemma 4.12. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$, let G_{SW} and G_{SE} be the south-west and south-east digraphs of $(\mathcal{S}, \mathcal{W})$ and let $(i, j) \in {}^2[\![n]\!]$. Then, we have:*

(i) If H is a shortest i-j-path in G_{SW}, then either

- $E(H) \subseteq \mathcal{S} \setminus \operatorname{sym}(\mathcal{W})$,
- $E(H) = \{(i, j)\} \subseteq \mathcal{S} \cap \mathcal{W}$, *or*
- $E(H) \subseteq \mathcal{W} \setminus \operatorname{sym}(\mathcal{S})$.

(ii) If H is a shortest i-j-path in G_{SE}, then either

- $E(H) \subseteq \mathcal{S} \setminus \operatorname{sym}(\mathcal{W}) = \mathcal{S} \setminus \operatorname{sym}(\overleftarrow{\mathcal{W}})$,
- $E(H) = \{(i, j)\} \subseteq \mathcal{S} \cap \overleftarrow{\mathcal{W}}$, *or*
- $E(H) \subseteq \overleftarrow{\mathcal{W}} \setminus \operatorname{sym}(\mathcal{S})$.

Proof. We only show the first statement, the second statement then follows by exchanging $\mathcal{W}$ and $\overleftarrow{\mathcal{W}}$.

Let H be a shortest i-j-path in G_{SW}, so $E(H) \subseteq E(G_{\mathrm{SW}}) = (\mathcal{S} \setminus \overleftarrow{\mathcal{W}}) \cup (\mathcal{W} \setminus \overleftarrow{\mathcal{S}})$. If H consists of a single edge, there is nothing to show, so assume $|E(H)| \geq 2$.

Claim. There are no two different edges $e_1, e_2 \in E(H)$ with $e_1 \in \mathcal{S} \setminus \overleftarrow{\mathcal{W}}$ and $e_2 \in \mathcal{W} \setminus \overleftarrow{\mathcal{S}}$.

Let $(a, b), (b, c) \in E(H)$ be consecutive edges on H with $(a, b) \in \mathcal{S} \setminus \overleftarrow{\mathcal{W}}$ and $(b, c) \in \mathcal{W} \setminus \overleftarrow{\mathcal{S}}$. As $\overleftarrow{\mathcal{W}}$ is transitive, $(a, b) \notin \overleftarrow{\mathcal{W}}$ and $(c, b) \in \overleftarrow{\mathcal{W}}$ imply $(a, c) \notin \overleftarrow{\mathcal{W}}$. Similarly, $(b, a) \in \overleftarrow{\mathcal{S}}$ and $(b, c) \notin \overleftarrow{\mathcal{S}}$ imply $(a, c) \notin \overleftarrow{\mathcal{S}}$. We conclude that $(a, c) \notin \overleftarrow{\mathcal{W}} \cup \overleftarrow{\mathcal{S}}$, implying $(a, c) \in (\mathcal{S} \setminus \overleftarrow{\mathcal{W}}) \cup (\mathcal{W} \setminus \overleftarrow{\mathcal{S}}) = E(G_{\mathrm{SW}})$, contradicting that the edges (a, b) and (b, c) are consecutive on a shortest path. Analogously, if $(a, b) \in \mathcal{W} \setminus \overleftarrow{\mathcal{S}}$ and $(b, c) \in \mathcal{S} \setminus \overleftarrow{\mathcal{W}}$, then $(a, c) \in (\mathcal{S} \setminus \overleftarrow{\mathcal{W}}) \cup (\mathcal{W} \setminus \overleftarrow{\mathcal{S}}) = E(G_{\mathrm{SW}})$, which proves the claim.

Now, as $\mathcal{S} \cap \mathcal{W} = (\mathcal{S} \setminus \overleftarrow{\mathcal{W}}) \cap (\mathcal{W} \setminus \overleftarrow{\mathcal{S}})$, if H contains an edge in $\mathcal{S} \cap \mathcal{W}$, that edge must be the only edge of H, contradicting $|E(H)| \geq 2$. Hence, either all edges of H are in $(\mathcal{S} \setminus \overleftarrow{\mathcal{W}}) \setminus (\mathcal{S} \cap \mathcal{W}) = \mathcal{S} \setminus \operatorname{sym}(\mathcal{W})$, or all edges of H are in $(\mathcal{W} \setminus \overleftarrow{\mathcal{S}}) \setminus (\mathcal{S} \cap \mathcal{W}) = \mathcal{W} \setminus \operatorname{sym}(\mathcal{S})$. □

Recall that given a relation $Q \subseteq {}^2[\![n]\!]$, we denote by $\operatorname{tr}(Q)$ the transitive closure of Q.

Corollary 4.13. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$ and let G_{SW} and G_{SE} be the south-west and south-east digraphs of $(\mathcal{S}, \mathcal{W})$.*

Then, we have:

(i) $\operatorname{tr}\Big(E(G_{\text{SW}})\Big) = \operatorname{tr}\Big(\mathcal{S} \setminus \operatorname{sym}(\mathcal{W})\Big) \cup \Big(\mathcal{S} \cap \mathcal{W}\Big) \cup \operatorname{tr}\Big(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\Big)$

(ii) $\operatorname{tr}\Big(E(G_{\text{SE}})\Big) = \operatorname{tr}\Big(\mathcal{S} \setminus \operatorname{sym}(\overleftarrow{\mathcal{W}})\Big) \cup \Big(\mathcal{S} \cap \overleftarrow{\mathcal{W}}\Big) \cup \operatorname{tr}\Big(\overleftarrow{\mathcal{W}} \setminus \operatorname{sym}(\mathcal{S})\Big)$

Of course, when using topological orders of G_{SW} and G_{SE}, we must ensure that these graphs are acyclic:

Lemma 4.14. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$. Then, the south-west digraph G_{SW} and the south-east digraph G_{SE} are acyclic.*

Proof. By Lemma 4.12, we have $(i,j) \in \mathcal{S} \cup \mathcal{W}$ whenever j is reachable from i in G_{SW}. This implies $(j,i) \in \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$, and therefore (j,i) is not an edge of G_{SW}. So G_{SW} is indeed acyclic. Similarly, if j is reachable from i in G_{SE}, then by Lemma 4.12 we have $(i,j) \in \mathcal{S} \cup \overleftarrow{\mathcal{W}}$, and $(j,i) \in \overleftarrow{\mathcal{S}} \cup \mathcal{W}$ is not an edge of G_{SE}. $\square$

Now, we can show that topological orders of G_{SW} and G_{SE} indeed yield sequence pairs with the desired properties:

Lemma 4.15. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$, let G_{SW} and G_{SE} be the south-west and south-east digraphs of $(\mathcal{S}, \mathcal{W})$, and let (π, ρ) be a sequence pair. Then, the following statements are equivalent:*

(i) *π is a topological order of G_{SW}, and ρ is a topological order of G_{SE}.*

(ii) *$\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$.*

Proof. For the first direction, assume that π and ρ are topological orders of G_{SW} and G_{SE}.

If $(i,j) \in \mathcal{S}_{\pi,\rho}$, then $i <_\pi j$ and $i <_\rho j$, and (j,i) is neither an edge of G_{SW} nor of G_{SE}. Hence $(j,i) \notin E(G_{\text{SW}}) = (\mathcal{S} \setminus \overleftarrow{\mathcal{W}}) \cup (\mathcal{W} \setminus \overleftarrow{\mathcal{S}})$, so $(j,i) \in \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$. Similarly, as $(j,i) \notin E(G_{\text{SE}}) = (\mathcal{S} \setminus \mathcal{W}) \cup (\overleftarrow{\mathcal{W}} \setminus \overleftarrow{\mathcal{S}})$, we have $(j,i) \in \overleftarrow{\mathcal{S}} \cup \mathcal{W}$. This means $(j,i) \in (\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}) \cap (\overleftarrow{\mathcal{S}} \cup \mathcal{W}) = \overleftarrow{\mathcal{S}} \cup (\mathcal{W} \cap \overleftarrow{\mathcal{W}}) = \overleftarrow{\mathcal{S}}$, so $(i,j) \in \mathcal{S}$.

If $(i,j) \in \mathcal{W}_{\pi,\rho}$, then $i <_\pi j$ and $j <_\rho i$, and (j,i) is not an edge of G_{SW} and (i,j) is not an edge of G_{SE}. Again, by $(j,i) \notin E(G_{\text{SW}})$, it follows that $(j,i) \in \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$, and hence $(i,j) \in \mathcal{S} \cup \mathcal{W}$. Moreover, as $(i,j) \notin E(G_{\text{SE}})$, we have $(i,j) \in \overleftarrow{\mathcal{S}} \cup \mathcal{W}$. Hence $(i,j) \in (\mathcal{S} \cup \mathcal{W}) \cap (\overleftarrow{\mathcal{S}} \cup \mathcal{W}) = (\mathcal{S} \cap \overleftarrow{\mathcal{S}}) \cup \mathcal{W} = \mathcal{W}$.

For the other direction, if π is not a topological order of G_{SW}, then there is an edge $(j,i) \in E(G_{\text{SW}})$ with $i <_\pi j$, so $(i,j) \in \mathcal{S}_{\pi,\rho} \cup \mathcal{W}_{\pi,\rho}$. Now $(j,i) \in E(G_{\text{SW}}) = (\mathcal{S} \cup \mathcal{W}) \setminus (\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}})$, so $(i,j) \notin \mathcal{S} \cup \mathcal{W}$. It follows that $(i,j) \in \big(\mathcal{S}_{\pi,\rho} \setminus \mathcal{S}\big) \cup \big(\mathcal{W}_{\pi,\rho} \setminus \mathcal{W}\big)$.

Similarly, if ρ is not a topological order of G_{SE}, then there is an edge $(j,i) \in E(G_{\text{SE}})$ with $i <_\rho j$, so $(i,j) \in \mathcal{S}_{\pi,\rho} \cup \overleftarrow{\mathcal{W}_{\pi,\rho}}$. Again, $(i,j) \notin \mathcal{S} \cup \overleftarrow{\mathcal{W}}$, and it follows that $(i,j) \in \big(\mathcal{S}_{\pi,\rho} \setminus \mathcal{S}\big) \cup \big(\overleftarrow{\mathcal{W}_{\pi,\rho}} \setminus \overleftarrow{\mathcal{W}}\big)$. The result then follows by observing that $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$ if and only if $\overleftarrow{\mathcal{W}_{\pi,\rho}} \subseteq \overleftarrow{\mathcal{W}}$. $\square$

The two previous results imply:

Corollary 4.16. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$. Then, there is a sequence pair (π, ρ) with $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$.*

In particular, we can show Theorem 4.2 and thus have $CR_n \leq (n!)^2$:

Theorem 4.2 ([Jer85])**.** *Let $n \in \mathbb{N}$. Then, the set*

$$\left\{ r_{\pi,\rho} : (\pi, \rho) \text{ is a sequence pair on } [\![n]\!] \right\}$$

is a complete set of representations.

Proof. Let P be a feasible placement. Observation 4.8 and Corollary 4.16 imply that there is a sequence pair (π, ρ) with $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}_P$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}_P$, and hence $r_{\pi,\rho}$ is a representation of P. $\square$

Moreover, we can now recover Jerrum's result that Definition 4.9 establishes a bijection between sequence pairs and complementary pairs of strict partial orders:

Proposition 4.17 ([Jer85, Theorem 2])**.** *Let $(\mathcal{S}, \mathcal{W})$ be a complementary pair of strict partial orders on $[\![n]\!]$. Then, there is a unique sequence pair (π, ρ) with $\mathcal{S}_{\pi,\rho} = \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} = \mathcal{W}$.*

Proof. Let G_{SW} and G_{SE} be the south-west and south-east digraphs of $(\mathcal{S}, \mathcal{W})$. First, we observe that the topological orders of G_{SW} and G_{SE} are unique: As every pair (i, j) is comparable in exactly one of $\mathcal{S}$ and $\mathcal{W}$, we have

$$\begin{aligned} E(G_{\mathrm{SW}}) &= \left(\mathcal{S} \cup \mathcal{W}\right) \setminus \left(\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}\right) = \left(\mathcal{S} \setminus \overleftarrow{\mathcal{W}}\right) \cup \left(\mathcal{W} \setminus \overleftarrow{\mathcal{S}}\right) = \mathcal{S} \cup \mathcal{W}, \\ E(G_{\mathrm{SE}}) &= \left(\mathcal{S} \cup \overleftarrow{\mathcal{W}}\right) \setminus \left(\overleftarrow{\mathcal{S}} \cup \mathcal{W}\right) = \left(\mathcal{S} \setminus \mathcal{W}\right) \cup \left(\overleftarrow{\mathcal{W}} \setminus \overleftarrow{\mathcal{S}}\right) = \mathcal{S} \cup \overleftarrow{\mathcal{W}}, \end{aligned}$$

so both digraphs contain an edge between any pair of endpoints $(i, j) \in {}^2[\![n]\!]$. Hence, by Lemma 4.15, there is a unique sequence pair (π, ρ) with $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$. Finally, since both $(\mathcal{S}, \mathcal{W})$ and $(\mathcal{S}_{\pi,\rho}, \mathcal{W}_{\pi,\rho})$ are complementary, we must have $\mathcal{S}_{\pi,\rho} = \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} = \mathcal{W}$. $\square$

A natural strategy to improve upon the $(n!)^2$ upper bound would be to prove that only certain sequence pairs can appear as topological orders of G_{SW} and G_{SE}. However, there is no hope for this approach:

Proposition 4.18 ([MFNK96, Theorem 2])**.** *Let (π, ρ) be a sequence pair on $[\![n]\!]$. Then, there is a feasible placement P on $[\![n]\!]$ that is represented by (π, ρ).*

Proof. Let σ_{x} be a topological order of $(\llbracket n \rrbracket, \mathcal{W}_{\pi,\rho})$, and let σ_{y} be a topological order of $(\llbracket n \rrbracket, \mathcal{S}_{\pi,\rho})$. We define a placement $P = (\mathit{minc}_{\mathrm{x}}, \mathit{minc}_{\mathrm{y}}, \mathit{maxc}_{\mathrm{x}}, \mathit{maxc}_{\mathrm{y}})$ by

$$\begin{aligned} \mathit{minc}_{\mathrm{x}}(i) &:= \sigma_{\mathrm{x}}(i), & \mathit{minc}_{\mathrm{y}}(i) &:= \sigma_{\mathrm{y}}(i), \\ \mathit{maxc}_{\mathrm{x}}(i) &:= \sigma_{\mathrm{x}}(i) + 1, & \mathit{maxc}_{\mathrm{y}}(i) &:= \sigma_{\mathrm{y}}(i) + 1, \end{aligned}$$

for $i \in \llbracket n \rrbracket$. It is easy to verify that P is a feasible placement, and moreover we have $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}_P$ and $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}_P$. □

Now, we will characterize representations of the form $r_{\pi,\rho}$ and show that there is a complete set of representations of minimum cardinality that only consists of representations of this form.

Definition 4.19. *Let r a representation on $\llbracket n \rrbracket$. The sets $\mathcal{S}_r, \mathcal{W}_r$ are defined as*

$$\begin{aligned} \mathcal{S}_r &:= \Big\{ (i,j) \in {}^2\llbracket n \rrbracket : r(i,j) = \mathit{south} \Big\}, \\ \mathcal{W}_r &:= \Big\{ (i,j) \in {}^2\llbracket n \rrbracket : r(i,j) = \mathit{west} \Big\}. \end{aligned}$$

Note that $\mathcal{S}_r$ and $\mathcal{W}_r$ are not necessarily strict partial orders, as transitivity is not guaranteed. On the other hand, irreflexivity follows from the fact that we only consider pairs $(i,j) \in {}^2\llbracket n \rrbracket$, and antisymmetry is clear by the definition. Moreover, antisymmetry of r implies ${}^2\llbracket n \rrbracket = \mathrm{sym}(\mathcal{S}_r) \cup \mathrm{sym}(\mathcal{W}_r)$. Finally, a representation r represents a placement P if and only if $\mathcal{S}_r \subseteq \mathcal{S}_P$ and $\mathcal{W}_r \subseteq \mathcal{W}_P$.

Definition 4.20. *Let r be a representation. We call r **transitive** if $\mathcal{S}_r$ and $\mathcal{W}_r$ are transitive, that is, if $(\mathcal{S}_r, \mathcal{W}_r)$ is a biorder.*

Non-transitive representations can be interpreted as being overconstrained: There are rectangles i and j that must satisfy *two* fixed spatial relations in each placement represented by r. It turns out that transitive representations are exactly the representations induced by sequence pairs:

Theorem 4.21. *Let r be a representation. Then r is transitive if and only if there is a sequence pair (π, ρ) with $r = r_{\pi,\rho}$.*

Proof. For the first direction, let (π, ρ) be a sequence pair with $r = r_{\pi,\rho}$. Then $\mathcal{S}_r = \mathcal{S}_{\pi,\rho}$ and $\mathcal{W}_r = \mathcal{W}_{\pi,\rho}$ are strict partial orders, so r is transitive.

For the other direction, assume that r is transitive. Then $(\mathcal{S}_r, \mathcal{W}_r)$ is a complementary pair of strict partial orders on $\llbracket n \rrbracket$, and by Proposition 4.17 there is a unique sequence pair (π, ρ) with $\mathcal{S}_{\pi,\rho} = \mathcal{S}_r$ and $\mathcal{W}_{\pi,\rho} = \mathcal{W}_r$, implying $r = r_{\pi,\rho}$. □

In particular, we get:

Corollary 4.22. *Let $n \in \mathbb{N}$. Then, there are exactly $(n!)^2$ transitive representations on $[\![n]\!]$.*

This means that the worst-case running time of branch-and-bound based algorithms (e.g., [FHS16]) that enumerate representations can be bounded in terms of $(n!)^2$ if it is ensured that only partial representations that can be completed to transitive representations are considered.

Moreover, we can always replace non-transitive representations by transitive ones:

Lemma 4.23. *Let r be a representation on $[\![n]\!]$. Then, there is a sequence pair (π, ρ) such that every placement represented by r is also represented by $r_{\pi,\rho}$.*

Proof. Define

$$\mathcal{P}_r := \{ P : P \text{ is feasible placement on } [\![n]\!] \text{ represented by } r \}.$$

If $\mathcal{P}_r$ is empty, there is nothing to show, so assume $\mathcal{P}_r \neq \emptyset$. Now, set

$$\mathcal{S} := \bigcap_{P \in \mathcal{P}_r} \mathcal{S}_P, \qquad \mathcal{W} := \bigcap_{P \in \mathcal{P}_r} \mathcal{W}_P.$$

As strict partial orders are closed under intersection, $\mathcal{S}$ and $\mathcal{W}$ are strict partial orders on $[\![n]\!]$. For all placements $P \in \mathcal{P}_r$, we have $\mathcal{S}_r \subseteq \mathcal{S}_P$ and $\mathcal{W}_r \subseteq \mathcal{W}_P$, which implies $\mathcal{S}_r \subseteq \mathcal{S}$ and $\mathcal{W}_r \subseteq \mathcal{W}$. Hence, we have ${}^2[\![n]\!] = \mathrm{sym}(\mathcal{S}) \cup \mathrm{sym}(\mathcal{W})$, so $(\mathcal{S}, \mathcal{W})$ is a biorder on $[\![n]\!]$.

Thus, using Corollary 4.16, we know that there is a sequence pair (π, ρ) with $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$. In particular, we have $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}_P$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}_P$ for every placement $P \in \mathcal{P}_r$, so we know that $r_{\pi,\rho}$ represents every placement $P \in \mathcal{P}_r$. □

Lemma 4.23 directly implies:

Theorem 4.24. *Let $n \in \mathbb{N}$. There is a set $\mathcal{SP} \subseteq \mathcal{SP}_n$ of sequence pairs such that*

$$R := \{ r_{\pi,\rho} : (\pi, \rho) \in \mathcal{SP} \}$$

is a complete set of representations of minimum cardinality.
In particular, the minimum cardinality of a complete set of sequence pairs equals the minimum cardinality of a complete set of representations.

Hence, instead of considering sets of representations, one can restrict oneself to sets of sequence pairs.

4.4 Reachability in Arborescences

Before we proceed to prove a stronger upper bound on CR_n, we use the techniques developed so far for a completely different application, namely an efficient reachability oracle for (out-)arborescences. More precisely, we describe an algorithm that, given an arborescence T, answers reachability queries on T in $\mathcal{O}(1)$ time, requiring only $\mathcal{O}(|V(T)|)$ preprocessing time. This result is not new: In fact, Kameda [Kam75] showed that this algorithm can even be applied to general acyclic plane digraphs with the following property: All vertices with in-degree 0 or out-degree 0 are on the boundary of the same face, and the boundary of that face can be partitioned into two contiguous sections containing only vertices with in-degree 0 or out-degree 0, respectively. We restrict ourselves to arborescences, which clearly satisfy this property, simplifying the analysis. Holm, Rotenberg, and Thorup [HRT15] even achieve the same guarantees for general planar digraphs, using a much more complicated technique. Still, this result shows that sequence pairs might be useful in different contexts, in particular also for non-enumerative purposes.

The algorithm is based on two observations. First, if $(\mathcal{S}, \mathcal{W})$ is a complementary pair of strict partial orders on $[\![n]\!]$, then there is a sequence pair (π, ρ) with $(\mathcal{S}, \mathcal{W}) = (\mathcal{S}_{\pi,\rho}, \mathcal{W}_{\pi,\rho})$. Given a pair (i, j), we can decide in constant time whether $(i, j) \in \mathcal{S}_{\pi,\rho}$ without explicitly constructing the set $\mathcal{S}_{\pi,\rho}$ by just looking at the permutations π and ρ. Hence, if $\mathcal{S}$ is a strict partial order on $[\![n]\!]$ such that there exists a strict partial order $\mathcal{W}$ with $(\mathcal{S}, \mathcal{W})$ complementary, then we can encode $\mathcal{S}$ in a data structure of linear size that answers containment queries in constant time. We call such strict partial orders $\mathcal{S}$ **complementable** (called *reversible* by Dushnik and Miller [DM41]).

The second observation is that the reachability relation of an arborescence (excluding pairs of the form (u, u)) is indeed complementable, and we can compute the corresponding sequence pair in linear time:

Let T be an arborescence with root $r \in V(T) = [\![n]\!]$ and let $\mathcal{S}$ be the reachability relation of T, that is, $\mathcal{S} = \operatorname{tr}(E(T))$. Then, for a vertex $v \in V(T)$, we denote by $H_{[r,v]}$ the unique r-v path in T. Moreover, assume that for each vertex u, we are given an arbitrary strict total order $<_u$ on $\delta^+(u)$. We say that (u, v_1) is left of (u, v_2) if $(u, v_1) <_u (u, v_2)$. For example, we can imagine that T is embedded into the plane, and $<_u$ corresponds to a geometric order on the outgoing edges of u. Given two paths H_1, H_2 that both start in r and do not contain each other, let $(u, v_1) \in E(H_1) \setminus E(H_2)$ and $(u, v_2) \in E(H_2) \setminus E(H_1)$ be the unique edges on H_1 and H_2 that leave the last common vertex of H_1 and H_2. Then, we say that $H_1 < H_2$ if $(u, v_1) <_u (u, v_2)$. We can now define $\mathcal{W}$:

$$\mathcal{W} := \left\{(v_1, v_2) \in {}^2[\![n]\!] \setminus \operatorname{sym}(\mathcal{S}) : H_{[r,v_1]} < H_{[r,v_2]}\right\}$$

It is not hard to show that $\mathcal{W}$ indeed is a strict partial order. Then, by definition, $(\mathcal{S}, \mathcal{W})$ is a complementary pair, and in particular there is a unique sequence pair (π, ρ) with $\mathcal{S} = \mathcal{S}_{\pi,\rho}$. Moreover, one can verify that π is the left-first topological order of T, and ρ is the right-first topological order of T, which can be computed in linear time by left-first and right-first depth-first search, respectively.

We conclude:

Proposition 4.25 ([Kam75]). *Let T be an arborescence. Then, in $\mathcal{O}(|V(T)|)$ time, we can compute a data structure that allows to answer reachability queries on T in $\mathcal{O}(1)$ time.*

Chapter 5

Improved Upper Bound

In this chapter, we show a new upper bound of $\mathcal{O}\Big(\frac{n!}{n^6} \cdot (\frac{11+5\sqrt{5}}{2})^n\Big)$ on the minimum cardinality CR_n of complete sets of representations.

5.1 Augmented Digraphs

We improve the construction given in Chapter 4 by adding edges to the digraphs G_{SW} and G_{SE} (cf. Definition 4.11), which restrict their topological orders:

Definition 5.1. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$. The **augmented south-west digraph** $G_{\mathrm{SW+}}$ and **augmented south-east digraph** $G_{\mathrm{SE+}}$ of $(\mathcal{S}, \mathcal{W})$ are digraphs with vertex set $[\![n]\!]$, and with edge sets*

$$\begin{aligned} E(G_{\mathrm{SW+}}) &:= E(G_{\mathrm{SW}}) \cup \Big\{ (i,j) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W} : i \text{ is not reachable from } j \text{ in } G_{\mathrm{SW}} \Big\}, \\ E(G_{\mathrm{SE+}}) &:= E(G_{\mathrm{SE}}) \cup \Big\{ (i,j) \in \overleftarrow{\mathcal{S}} \cap \overleftarrow{\mathcal{W}} : i \text{ is not reachable from } j \text{ in } G_{\mathrm{SE}} \Big\}, \end{aligned}$$

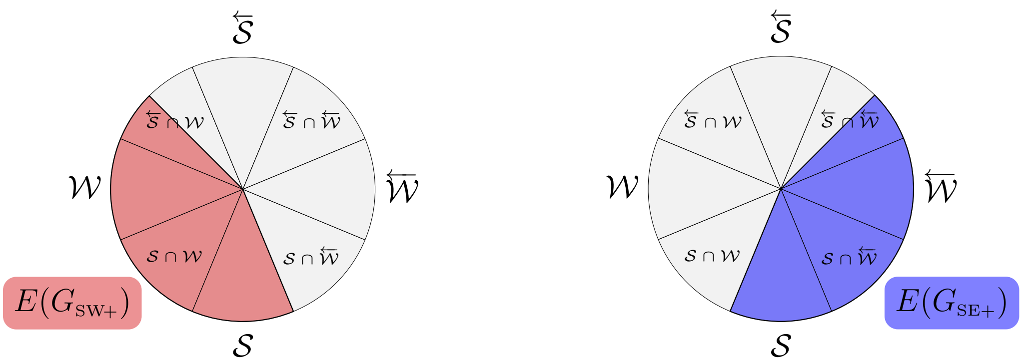
where G_{SW} and G_{SE} are the south-west and south-east digraphs of $(\mathcal{S}, \mathcal{W})$.

See Figure 5.1 for an illustration of $G_{\mathrm{SW+}}$ and $G_{\mathrm{SE+}}$. Again, we need to show that the constructed digraphs are acyclic:

Lemma 5.2. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$. Then the augmented digraphs $G_{\mathrm{SW+}}$ and $G_{\mathrm{SE+}}$ of $(\mathcal{S}, \mathcal{W})$ are acyclic.*

Proof. It suffices to consider $G_{\mathrm{SW+}}$ (for $G_{\mathrm{SE+}}$, exchange $\mathcal{W}$ and $\overleftarrow{\mathcal{W}}$).

Suppose $G_{\mathrm{SW+}}$ contains a cycle. Consider a cycle C with smallest number of edges. Of course, C must contain at least two edges from $E(G_{\mathrm{SW+}}) \setminus E(G_{\mathrm{SW}})$



(a) The edge set of $G_{\text{SW+}}$. **(b)** The edge set of $G_{\text{SE+}}$.

Figure 5.1: The edge sets of augmented digraphs. Partially colored segments indicate that a subset of the segment is used in the edge set.

because G_{SW} is acyclic (Lemma 4.14) and any single added edge does not create a cycle by construction.

We can partition C into paths that only consist of edges in $E(G_{\text{SW}})$, except for their last edge, see Figure 5.2. Let $v_k, v_{k-1}, \ldots, v_1, v_0, b$ be the vertices of such a path in C, i.e., only the last edge (v_0, b) does not belong to G_{SW}.

Claim. $(v_i, b) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$ for all $i = 0, \ldots, k$.

We show the claim by induction on i. It is true for $i = 0$ because $(v_0, b) \in E(G_{\text{SW+}}) \setminus E(G_{\text{SW}})$. Let now $i \geq 1$. As (v_i, v_{i-1}) is an edge of G_{SW}, $(v_i, v_{i-1}) \notin \overleftarrow{\mathcal{W}}$. Moreover, $(b, v_{i-1}) \in \overleftarrow{\mathcal{W}}$ by the induction hypothesis. As $\overleftarrow{\mathcal{W}}$ is transitive, $(v_i, b) \notin \overleftarrow{\mathcal{W}}$.

Now (v_i, b) is not an edge of G_{SW} because C is a shortest cycle. As $(v_i, b) \notin \overleftarrow{\mathcal{W}}$, this implies $(v_i, b) \in \overleftarrow{\mathcal{S}}$.

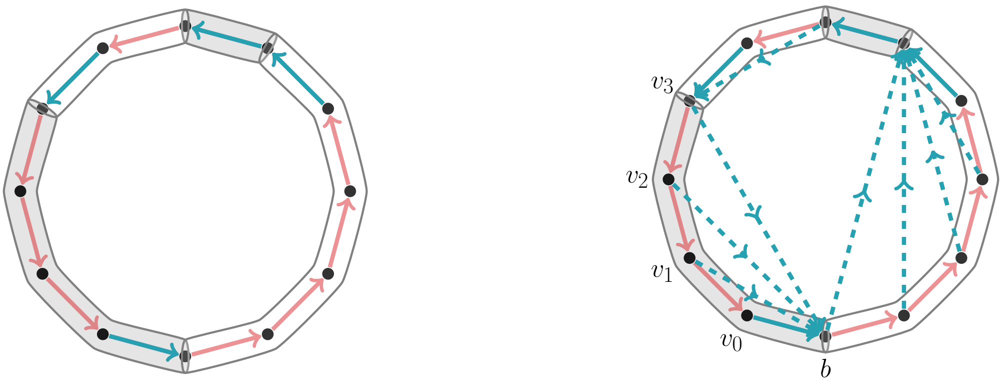
Finally suppose that $(v_i, b) \notin \mathcal{W}$. Then $(b, v_i) \in \mathcal{S} \setminus \overleftarrow{\mathcal{W}}$, and hence $(b, v_i) \in E(G_{\text{SW}})$. Then $b, v_i, v_{i-1}, \ldots, v_0$ is a path from b to v_0 in G_{SW}. This is a contradiction to the fact that $(v_0, b) \in E(G_{\text{SW+}}) \setminus E(G_{\text{SW}})$. The claim is proven.

Now let (a_i, b_i), $i = 1, \ldots, l$, be the edges of C that do not belong to G_{SW}. We have $l \geq 2$, and by the claim $(b_{i-1}, b_i) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$ for all $i = 1, \ldots, l$ (where $b_0 := b_l$). This is impossible because $\overleftarrow{\mathcal{S}}$ and $\mathcal{W}$ are strict partial orders. $\square$

Although the following statement is not required for the improved upper bound, we note:

Proposition 5.3. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$. Then, the augmented digraphs $G_{\text{SW+}}$ and $G_{\text{SE+}}$ of $(\mathcal{S}, \mathcal{W})$ have unique topological orders.*

Proof. We show that $G_{\text{SW+}}$ has a unique topological order π. The statement for $G_{\text{SE+}}$ then follows by exchanging $\mathcal{W}$ and $\overleftarrow{\mathcal{W}}$. So let $(i, j) \in {}^2[\![n]\!]$ and w.l.o.g.



(a) The decomposition of C into paths that only consist of red edges, except for the last one, indicated by alternating background colors.

(b) We show that all dashed edges, joining startpoints of red edges with the endpoint of their respective path, must exist in $\overleftarrow{\mathcal{S}} \cap \mathcal{W}$. This leads to a cycle in $\overleftarrow{\mathcal{S}} \cap \mathcal{W}$ and hence a contradiction.

Figure 5.2: Proof of Lemma 5.2: A shortest cycle in $G_{\mathrm{SW}+}$ with edges in $E(G_{\mathrm{SW}})$ in red.

we can assume that $(i, j) \in E(G_{\mathrm{SW}}) \cup \left(\overleftarrow{\mathcal{S}} \cap \mathcal{W}\right)$, otherwise consider (j, i). Let π be an arbitrary topological order of $G_{\mathrm{SW}+}$.
If $(i, j) \in E(G_{\mathrm{SW}+})$, then we must have $i <_\pi j$.
If $(i, j) \notin E(G_{\mathrm{SW}+})$, then $(i, j) \in \left(\overleftarrow{\mathcal{S}} \cap \mathcal{W}\right) \setminus E(G_{\mathrm{SW}+})$ and hence i is reachable from j in G_{SW} (otherwise we would have added the edge (i, j) to $G_{\mathrm{SW}+}$). But then i is also reachable from j in $G_{\mathrm{SW}+}$ and hence $j <_\pi i$.

We conclude that the relative order of all pairs in $<_\pi$ is fixed, and hence π is unique. $\square$

We now consider topological orders π and ρ of $G_{\mathrm{SW}+}$ and $G_{\mathrm{SE}+}$, respectively. Since we only added edges, π and ρ are topological orders of G_{SW} and G_{SE}, and hence Lemma 4.15 still implies $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$.

5.2 New Upper Bound

Now, we show that only certain sequence pairs can occur as topological orders of $G_{\mathrm{SW}+}$ and $G_{\mathrm{SE}+}$ if we restrict to biorders of the form $(\mathcal{S}_P, \mathcal{W}_P)$ for feasible placements P.

Lemma 5.4. *Let $P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$ be a feasible placement, and let $G_{\mathrm{SW}+}$ and $G_{\mathrm{SE}+}$ be the augmented digraphs of $(\mathcal{S}_P, \mathcal{W}_P)$. Furthermore, let π and ρ be topological orders of $G_{\mathrm{SW}+}$ and $G_{\mathrm{SE}+}$, respectively. Then (π, ρ) is plane.*

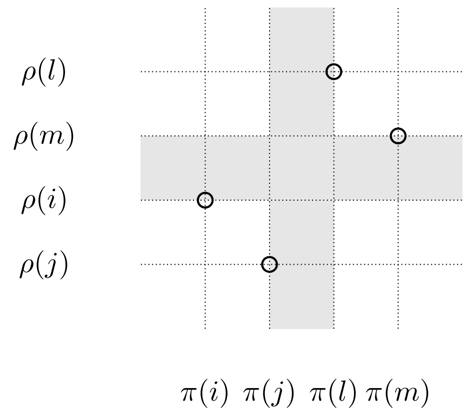


Figure 5.3: Illustration of (i, j, l, m) and (π, ρ) in the proof of Lemma 5.4. Gray areas are empty.

Proof. First note that (π, ρ) represents P by Lemma 4.15. For the sake of contradiction, assume that (π, ρ) is not plane. Then, $\sigma := \rho \circ \pi^{-1}$ is not plane, and by Lemma 3.4 σ contains an extreme bad quartet (i', j', l', m'). We have $i' < j' < l' < m'$ and there is no element between j' and l', i.e., $l' = j' + 1$. Moreover, we have $j' <_\sigma i' <_\sigma m' <_\sigma l'$ and there is no element between i' and m' in σ. Define $i := \pi^{-1}(i')$, $l := \pi^{-1}(l')$, $j := \pi^{-1}(j')$, $m := \pi^{-1}(m')$. Then, we have $i <_\pi j <_\pi l <_\pi m$ and there is no element between j and l in π. Furthermore, we have $j <_\rho i <_\rho m <_\rho l$ and there is no element between i and m in ρ. See Figure 5.3.

Since (π, ρ) represents P, we have $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}_P$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}_P$, and hence $(i, l), (i, m), (j, l), (j, m) \in \mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}_P$ and $(i, j), (l, m) \in \mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}_P$.

Claim 1. We have $(l, j) \notin \mathcal{W}_P$.

Suppose that $(l, j) \in \mathcal{W}_P$. Then $(j, l) \in \overleftarrow{\mathcal{W}_P}$, and thus $(j, l) \notin E(G_{\text{SW}})$. Therefore l is not reachable from j in G_{SW} as any vertex on a j-l-path would have to be in between j and l in the topological order π. But then $(l, j) \in \overleftarrow{\mathcal{S}_P} \cap \mathcal{W}_P$ would be an edge of $G_{\text{SW+}}$, contradicting $j <_\pi l$.

Claim 2. We have $(i, m) \notin \mathcal{W}_P$.

Suppose that $(i, m) \in \mathcal{W}_P$. Then $(i, m) \notin E(G_{\text{SE}})$. Therefore m is not reachable from i in G_{SE} as any vertex on an i-m-path would have to be in between i and m in the topological order ρ. But then $(m, i) \in \overleftarrow{\mathcal{S}_P} \cap \overleftarrow{\mathcal{W}_P}$ would be an edge of $G_{\text{SE+}}$, contradicting $i <_\rho m$.

The two claims are proved. However, they contradict each other: together with $(i, j), (l, m) \in \mathcal{W}_P$ they imply

$$\begin{aligned} maxc_{\text{x}}(i) &\overset{(i,j)\in\mathcal{W}_P}{\leq} minc_{\text{x}}(j) \overset{\text{Claim 1}}{<} maxc_{\text{x}}(l) \\ &\overset{(l,m)\in\mathcal{W}_P}{\leq} minc_{\text{x}}(m) \overset{\text{Claim 2}}{<} maxc_{\text{x}}(i). \end{aligned}$$

□

In Section 7.3.3, our empirical experiments will show that for $n \leq 8$, all plane sequence pairs occur as topological orders of $G_{\text{SW+}}$ and $G_{\text{SE+}}$, and hence the analysis of Lemma 5.4 is best possible for $n \leq 8$.

Observation 5.5. *Let $n \in \mathbb{N}$. Then, we have*

$$\begin{aligned} &\Big\{(\pi,\rho) \quad\;\; : (\pi,\rho) \text{ is plane sequence pair on } [\![n]\!]\Big\} \\ =&\Big\{(\pi,\rho) \quad\;\; : \pi,\rho \text{ are permutations on } [\![n]\!],\ \rho\circ\pi^{-1} \text{ is plane}\Big\} \\ =&\Big\{(\pi,\sigma\circ\pi) : \pi,\sigma \text{ are permutations on } [\![n]\!],\ \sigma \text{ is plane}\Big\}. \end{aligned}$$

In particular, the number of plane sequence pairs is $n!$ times the number of plane permutations.

We conclude:

Theorem 5.6. *Let $n \in \mathbb{N}$. Then, the set of plane sequence pairs on $[\![n]\!]$ is complete for n.*

Proof. The result is a direct consequence of Lemmata 4.15, 5.2 and 5.4. □

Theorem 5.7. *Let $n \in \mathbb{N}$. Then*

$$CR_n \leq n! \cdot Plane_n = \mathcal{O}\left(\frac{n!}{n^6} \cdot C^n\right),$$

where $C = \frac{11+5\sqrt{5}}{2} \leq 11.091$.

Proof. By Observation 5.5, the number of plane sequence pairs is $n!$ times the number of plane permutations. By Theorem 3.10, the number of plane permutations is $\Theta\left(\frac{C^n}{n^6}\right)$. The result then follows from Theorem 5.6. □

Note that this result does not only imply an improved asymptotic behavior compared to classical sequence pairs, but also yields a strict improvement for all $n \geq 4$, cf. Table 3.3 (page 32).

Chapter 6

Improved Lower Bound

In this chapter, we prove a lower bound of $\Omega\Big(\frac{n!}{n^4} \cdot (4+2\sqrt{2})^n\Big)$ on CR_n. By Theorem 4.24, instead of considering complete sets of representations, we can restrict ourselves to complete sets of sequence pairs. Now, the idea of the new lower bound is to generate a large set of feasible placements $\mathcal{P}$, each of which is only represented by a single unique sequence pair, where no sequence pair occurs twice. Then, every complete set of sequence pairs must contain all these sequence pairs, and hence we have $CR_n \geq |\mathcal{P}|$.

We construct these placements using biplane permutations, which have been examined by Asinowski et al. [Asi+13]. They study orders on *segments* of floorplans, which have a very similar structure to the *rectangles* in the placements considered here.

At the end of this chapter, we show that $\mathcal{P}$ has maximum possible cardinality among all sets of placements with the properties described above, and thus our lower bound is best possible when using this technique.

6.1 Forcing Placements

Consider a feasible placement P and a pair of rectangles i, j in P. If i is only west of j in P, then for any sequence pair (π, ρ) representing P we must have $(i, j) \in \mathcal{W}_{\pi,\rho}$. If we further assume that j is only west of some rectangle k, then also $(j, k) \in \mathcal{W}_{\pi,\rho}$, and transitivity implies $(i, k) \in \mathcal{W}_{\pi,\rho}$, even if i is also south or north of k. We will exploit this observation (which will be formalized in Lemma 6.3) to construct placements that are represented by a unique sequence pair only.

Given a biorder $(\mathcal{S}, \mathcal{W})$ on $[\![n]\!]$ and a pair $(i, j) \in {}^2[\![n]\!]$, we say that i is

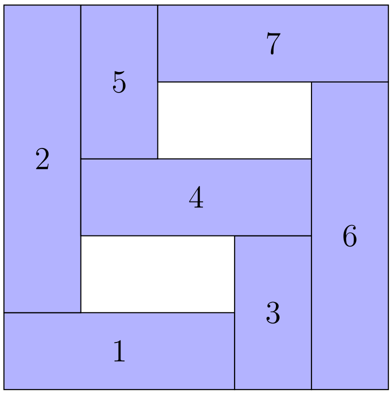


Figure 6.1: A forcing placement.

south of j in $(\mathcal{S}, \mathcal{W})$ if $(i, j) \in \mathcal{S}$, i is west of j if $(i, j) \in \mathcal{W}$, and so on, even if $(\mathcal{S}, \mathcal{W})$ is not of the form $(\mathcal{S}_P, \mathcal{W}_P)$ for some placement P.

Definition 6.1. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder and $(i, j) \in {}^2[\![n]\!]$. We say that a spatial relation $\alpha \in \{$ west, south, east, north$\}$ is **forced** for (i, j) in $(\mathcal{S}, \mathcal{W})$ if there is a sequence of elements $i = a_1, \ldots, a_k = j$ such that for all $1 \leq m < k$, the only spatial relation of (a_m, a_{m+1}) in $(\mathcal{S}, \mathcal{W})$ is α.*

Observation 6.2. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$, let $i, j, k \in [\![n]\!]$ and let $\alpha \in \{$ west, south, east, north$\}$ be a spatial relation. If α is forced for (i, j) and (j, k) in $(\mathcal{S}, \mathcal{W})$, then α is also forced for (i, k) in $(\mathcal{S}, \mathcal{W})$.*

Lemma 6.3. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $[\![n]\!]$, let (π, ρ) be a sequence pair representing $(\mathcal{S}, \mathcal{W})$ and $(i, j) \in {}^2[\![n]\!]$. If a relation $\alpha \in \{$ west, south, east, north$\}$ is forced for (i, j) in $(\mathcal{S}, \mathcal{W})$, then $r_{\pi,\rho}(i, j) = \alpha$.*

Proof. If α is forced for (i, j), then there is a sequence $i = a_1, \ldots, a_k = j$ such that for all $1 \leq m < k$, the only spatial relation of (a_m, a_{m+1}) in $(\mathcal{S}, \mathcal{W})$ is α. Since $r_{\pi,\rho}$ represents $(\mathcal{S}, \mathcal{W})$, for all $1 \leq m < k$ we have $r_{\pi,\rho}(a_m, a_{m+1}) = \alpha$ and, by transitivity of $r_{\pi,\rho}$ (cf. Theorem 4.21), we conclude that $r_{\pi,\rho}(i, j) = \alpha$. $\square$

Definition 6.4. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder. We call $(\mathcal{S}, \mathcal{W})$ **forcing** if for all pairs $(i, j) \in {}^2[\![n]\!]$, there is a forced spatial relation for (i, j) in $(\mathcal{S}, \mathcal{W})$. We call a placement P **forcing** if $(\mathcal{S}_P, \mathcal{W}_P)$ is forcing.*

Note that in particular a forcing placement is feasible. Examples of forcing placements are given in Figure 1.1(c) (page 1) and Figure 6.1. Using Lemma 6.3, we note:

Observation 6.5. *Let $(\mathcal{S}, \mathcal{W})$ be a forcing biorder. Then, there is a unique sequence pair (π, ρ) representing $(\mathcal{S}, \mathcal{W})$.*

Hence, we can assign each forcing placement its corresponding sequence pair:

Definition 6.6. *Let P be a forcing placement. The* ***forced sequence pair*** *(π, ρ) of P is the unique sequence pair (π, ρ) that represents P. We denote by $r_P := r_{\pi,\rho}$ the* ***forced representation*** *of P. Moreover, we call a sequence pair (π, ρ)* ***forced*** *if it is the forced sequence pair of a forcing placement.*

Theorem 4.24 directly implies:

Observation 6.7. *Let $n \in \mathbb{N}$ and let $\mathcal{SP}_{forced}$ be a set of forced sequence pairs on $[\![n]\!]$. Furthermore, let $\mathcal{SP}$ be a complete set of sequence pairs on $[\![n]\!]$. Then, we have $\mathcal{SP}_{forced} \subseteq \mathcal{SP}$ and hence $CR_n \geq \left|\mathcal{SP}_{forced}\right|$.*

6.2 Many Forced Sequence Pairs

Now we get to the main part of the proof: We show the existence of a large set of forced sequence pairs. The plan is to prove that each biplane sequence pair is forced, that is, to prove that for each sequence pair of the form $(\pi, \sigma \circ \pi)$ with σ biplane there is a forcing placement P such that $(\pi, \sigma \circ \pi)$ is the forced sequence pair of P.

This will be done in two steps: In this section, we will prove that every sequence pair of the form (id, σ) with σ biplane is forced. Then, in Section 6.3, we will see that one can apply all permutations π to such sequence pairs, resulting in all sequence pairs of the form $(\pi, \sigma \circ \pi)$ being forced as desired.

Set $r_\sigma := r_{\mathrm{id},\sigma}$. Recall that for a permutation σ on $[\![n]\!]$, there is a digraph G_σ on the vertex set $[\![n]\!]$ whose edge set contains all pairs (i, k) such that $i <_\sigma k$ and there is no j with $i <_\sigma j <_\sigma k$, cf. Definition 3.6.

Lemma 6.8. *Let σ be a permutation on $[\![n]\!]$ and let P be a feasible n-placement. Then P is a forcing placement with $r_P = r_\sigma$ if and only if*

(i) for all $(i, j) \in E(G_\sigma)$, i is only south of j in P, and

(ii) for all $(i, j) \in E(G_{-\sigma})$, i is only west of j in P.

Proof. First, we prove that if (i) and (ii) hold, then P is forcing with $r_P = r_\sigma$. Let $i, j \in [\![n]\!]$ with $i < j$. By Observation 3.12, j is reachable from i in either G_σ or $G_{-\sigma}$, but not both. Assume j is reachable from i in G_σ. Then there is a sequence of vertices $i = a_1, \ldots, a_k = j$ with $(a_m, a_{m+1}) \in E(G_\sigma)$ for $1 \leq m < k$, so by (i), i south of j is forced. Furthermore, since j is reachable from i in G_σ, we have $i <_\sigma j$, so $r_\sigma(i, j) = \text{south}$. The case that j is reachable from i in $G_{-\sigma}$ is proven analogously.

For the other direction, let P be forcing with $r_P = r_\sigma$ and $(i, j) \in E(G_\sigma)$. Since $i < j$ and $i <_\sigma j$, we have $r_P(i, j) = r_\sigma(i, j) = \text{south}$, so i is south of j. It remains to be shown that south is the only spatial relation of (i, j). Since $r_P(i, j) = \text{south}$, i south of j is forced, and there are indices $i = a_1, \ldots, a_k = j$ such that a_m is only south of a_{m+1} in P for $1 \leq m < k$. Since $r_\sigma = r_P$ represents

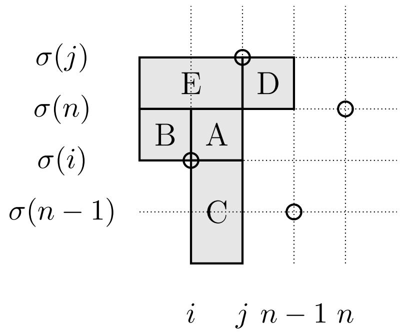


Figure 6.2: Configuration with $i < j < n-1 < n$. Gray areas are claimed to be empty.

P, we have $r_\sigma(a_m, a_{m+1}) =$ south for $1 \le m < k$, so $i = a_1 < \cdots < a_k = j$ and $i = a_1 <_\sigma \cdots <_\sigma a_k = j$. Hence, due to $(i,j) \in E(G_\sigma)$, we have $k = 2$, and thus i is only south of j. Again, the case $(i,j) \in E(G_{-\sigma})$ is proven analogously. $\square$

Recall that a permutation σ is called biplane if it avoids the patterns $21\bar{3}54$ and $45\bar{3}12$, cf. Definition 3.13. Before we prove the main lemma, we need a technical auxiliary result:

Lemma 6.9. *Let σ be a biplane permutation on $[\![n]\!]$ with $\sigma(n-1) < \sigma(n) < n$. Let $(j,n) \in E(G_{-\sigma})$ such that j has no outgoing edges in G_σ and let $i < j$ with $(i,n) \in E(G_\sigma)$. Furthermore, let $P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$ be a forcing placement with $r_P = r_\sigma$. Then, i is the only index with this property, and there is a forcing placement $P' = (minc'_{\mathrm{x}}, minc'_{\mathrm{y}}, maxc'_{\mathrm{x}}, maxc'_{\mathrm{y}})$ with $r_{P'} = r_\sigma$ and $maxc'_{\mathrm{x}}(j) < maxc'_{\mathrm{x}}(i)$.*

Proof. First, note that $j \neq n-1$, so $j < n-1$, and due to $i < j < n-1 < n$ and $(n-1,n), (i,n) \in E(G_\sigma)$, we must have

$$n - 1 <_\sigma i <_\sigma n <_\sigma j.$$

Claim. There is no $l \in [\![n]\!]$ with either

(A) $i < l < j$ and $i <_\sigma l <_\sigma n$, or

(B) $l < i$ and $i <_\sigma l <_\sigma n$, or

(C) $i < l < j$ and $l <_\sigma i$, or

(D) $j < l < n-1$ and $n <_\sigma l <_\sigma j$, or

(E) $l < j$ and $n <_\sigma l <_\sigma j$.

Figure 6.2 illustrates the setting and the five statements.

To prove the claim, first observe that an l with (A) would contradict $(i,n) \in E(G_\sigma)$. Next, this implies that an l with (B) or with (C) would yield (with i, j and n) a match of the pattern $21\bar{3}54$, contradicting that σ is plane. Third, an l with (D) would contradict $(j,n) \in E(G_{-\sigma})$. Finally, this implies that an l with (E) would yield (together with j, $n-1$ and n) a match of the pattern $45\bar{3}12$, contradicting that $-\sigma$ is plane. The claim is proven.

Now, by (A), (B), and (C) of the claim, there is no $l < j$ with $l \neq i$ and $(l,n) \in E(G_\sigma)$.

Part (A) and (E) of the claim imply that $(i,j) \in E(G_\sigma)$. Hence, by Lemma 6.8, i is only south of j in P – in particular i is not west of j – so $maxc_{\mathrm{x}}(i) > minc_{\mathrm{x}}(j)$. If $maxc_{\mathrm{x}}(j) < maxc_{\mathrm{x}}(i)$, there is nothing to show (i.e., set $P' = P$), so assume $maxc_{\mathrm{x}}(j) \geq maxc_{\mathrm{x}}(i)$.
Set $(minc'_{\mathrm{x}}, minc'_{\mathrm{y}}, maxc'_{\mathrm{x}}, maxc'_{\mathrm{y}}) = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$, except for

$$maxc'_{\mathrm{x}}(j) := \frac{\max\{\, minc_{\mathrm{x}}(i), minc_{\mathrm{x}}(j) \,\} + maxc_{\mathrm{x}}(i)}{2}.$$

Then, we have

$$maxc'_{\mathrm{x}}(j) < maxc_{\mathrm{x}}(i) \leq maxc_{\mathrm{x}}(j)$$

and

$$maxc'_{\mathrm{x}}(j) \geq \frac{minc'_{\mathrm{x}}(j) + maxc'_{\mathrm{x}}(i)}{2} > minc'_{\mathrm{x}}(j).$$

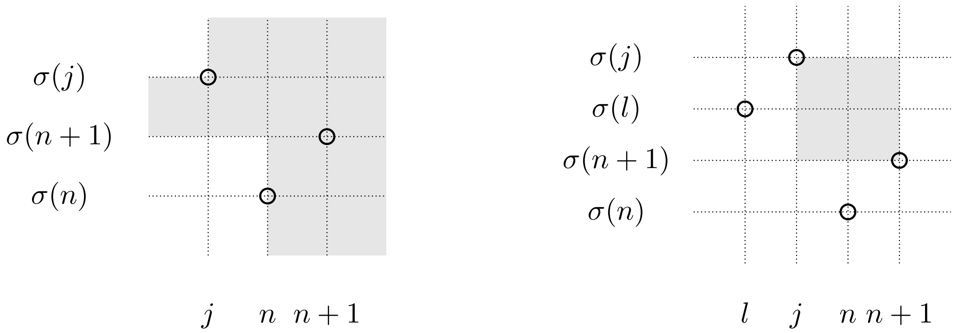
Hence, P' is still a placement. Since we only decreased the width of j, P' is still feasible, and all only-west and only-east relations of j are still intact. Moreover, as j has no outgoing edges in G_σ, in order to see that P' is still forcing with $r_{P'} = r_\sigma$, we only need to verify that for all edges $(k,j) \in E(G_\sigma)$, k is still only south of j. But, by (A), (B), (C) and (E) of the claim, i is the only predecessor of j in G_σ, and since we only reduced $maxc'_{\mathrm{x}}(j)$, j is still not east of i. Moreover, we have

$$\begin{aligned} maxc'_{\mathrm{x}}(j) &= \frac{\max\left\{\, minc'_{\mathrm{x}}(i), minc'_{\mathrm{x}}(j) \,\right\} + maxc'_{\mathrm{x}}(i)}{2} \\ &\geq \frac{minc'_{\mathrm{x}}(i) + maxc'_{\mathrm{x}}(i)}{2} \\ &> minc'_{\mathrm{x}}(i), \end{aligned}$$

so j is not west of i in P'. □

Lemma 6.10. *Let σ be a biplane permutation on $[\![n]\!]$. Then there is a forcing placement P of n rectangles with $r_P = r_\sigma$.*

Proof. We prove the lemma by induction. The case $n = 1$ is trivial, so assume the claim holds for $n \in \mathbb{N}$ and let σ be a biplane permutation on $[\![n+1]\!]$.



(a) Situation with $j < n < n+1$ and $n <_\sigma n+1 <_\sigma j$.

(b) If there exists a predecessor $l < j$ of $n+1$ in $G_{-\sigma}$, then $-\sigma$ is not plane.

Figure 6.3: Illustrations of order of elements in σ. Gray areas do not contain any other elements.

First, we consider the case $n <_\sigma n+1$. The other case will later be reduced to this case. Let σ' be the permutation on $[\![n]\!]$ given by $\sigma'(i) := \sigma(i)$ if $i <_\sigma n+1$, and $\sigma'(i) := \sigma(i) - 1$ otherwise. Clearly, for $i, j \in [\![n]\!]$, we have $i <_\sigma j \iff i <_{\sigma'} j$. In particular, σ' is a biplane permutation, so by the induction hypothesis, there is a forcing placement $P' = (\mathit{minc}'_{\mathrm{x}}, \mathit{minc}'_{\mathrm{y}}, \mathit{maxc}'_{\mathrm{x}}, \mathit{maxc}'_{\mathrm{y}})$ with $r_{P'} = r_{\sigma'}$. Note that $G_{\sigma'}$ is an induced subgraph of G_σ, and $G_{-\sigma'}$ is an induced subgraph of $G_{-\sigma}$. This means that if we extend P' to some placement P of $n+1$ rectangles, we only need to check edges incident to $n+1$ when applying Lemma 6.8.

If $\sigma(n+1) = n+1$, then we can just place $n+1$ north of all other rectangles: extend

$$P' = (\mathit{minc}'_{\mathrm{x}}, \mathit{minc}'_{\mathrm{y}}, \mathit{maxc}'_{\mathrm{x}}, \mathit{maxc}'_{\mathrm{y}})$$

to

$$P = (\mathit{minc}_{\mathrm{x}}, \mathit{minc}_{\mathrm{y}}, \mathit{maxc}_{\mathrm{x}}, \mathit{maxc}_{\mathrm{y}})$$

by

$$\begin{aligned} \mathit{minc}_{\mathrm{x}}(n+1) &:= \min_{i \in [\![n]\!]} \mathit{minc}'_{\mathrm{x}}(i), & \mathit{minc}_{\mathrm{y}}(n+1) &:= \max_{i \in [\![n]\!]} \mathit{maxc}'_{\mathrm{y}}(i), \\ \mathit{maxc}_{\mathrm{x}}(n+1) &:= \max_{i \in [\![n]\!]} \mathit{maxc}'_{\mathrm{x}}(i), & \mathit{maxc}_{\mathrm{y}}(n+1) &:= \max_{i \in [\![n]\!]} \mathit{maxc}'_{\mathrm{y}}(i) + 1. \end{aligned}$$

By *extending*, we mean that P and P' agree for $i = 1, \ldots, n$. Then, $n+1$ does not overlap with any rectangle, so P is a feasible placement. For $(i, n+1) \in E(G_\sigma)$, by the construction of P, we have that i is only south of $n+1$ in P. Since there are no edges $(i, n+1) \in E(G_{-\sigma})$, we can apply Lemma 6.8 to conclude that P is forcing with $r_P = r_\sigma$.

So assume $\sigma(n+1) < n+1$. Let j be maximum with $(j, n+1) \in E(G_{-\sigma})$. Note that j exists since $n+1$ is reachable from $\sigma^{-1}(n+1)$ in $G_{-\sigma}$. This configuration is illustrated in Figure 6.3(a). Then j is the only predecessor of $n+1$ in $G_{-\sigma}$: If $l < j$ with $n+1 <_\sigma l <_\sigma j$, then $(l, j, n, n+1)$ shows that $-\sigma$ is not plane, cf. Figure 6.3(b).

Moreover, j has no outgoing edges in $G_{\sigma'}$, since if $(j, l) \in E(G_{\sigma'})$, then $l < n+1$ and $n+1 <_\sigma l$, so $n+1$ is reachable from l in $G_{-\sigma}$, contradicting that j is the only predecessor of $n+1$ in $G_{-\sigma}$. Hence, $r_{\sigma'}(j, l) \neq$ north for all other rectangles l, and as $r_{\sigma'}$ represents P', there is no rectangle only north of j in P'. We can thus w.l.o.g. assume that

$$maxc'_{\mathrm{y}}(j) \geq \max\Big\{1 + maxc'_{\mathrm{y}}(i) : i \in [\![n]\!] \setminus \{\, j\,\}\Big\}, \tag{6.1}$$

since we can increase the height of j as required. Increasing the size of rectangles while maintaining a feasible placement does not destroy forced relations, so P' is still forcing with $r_{P'} = r_{\sigma'}$.

Now, we consider the predecessors of $n+1$ in G_σ, which represent the rectangles that $n+1$ has to be north of. Let i be minimum with $(i, n+1) \in E(G_\sigma)$. Again, i exists since $(n, n+1) \in E(G_\sigma)$. If $i < j$, by Lemma 6.9 (applied to the case $n+1$) there is no $i < l < j$ with $(l, n+1) \in E(G_\sigma)$ and w.l.o.g. we can assume that $maxc'_{\mathrm{x}}(j) < maxc'_{\mathrm{x}}(i)$. Note that (6.1) can still be assumed.
We extend

$$P' = (minc'_{\mathrm{x}}, minc'_{\mathrm{y}}, maxc'_{\mathrm{x}}, maxc'_{\mathrm{y}})$$

to

$$P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$$

by

$$\begin{aligned} minc_{\mathrm{x}}(n+1) &:= maxc_{\mathrm{x}}(j), & minc_{\mathrm{y}}(n+1) &:= maxc_{\mathrm{y}}(j) - 1, \\ maxc_{\mathrm{x}}(n+1) &:= \max_{l \in [\![n]\!]} maxc_{\mathrm{x}}(l), & maxc_{\mathrm{y}}(n+1) &:= maxc_{\mathrm{y}}(j). \end{aligned}$$

First, since j is west of n, we have $maxc_{\mathrm{x}}(n+1) \geq maxc_{\mathrm{x}}(n) > minc_{\mathrm{x}}(n) \geq maxc_{\mathrm{x}}(j) = minc_{\mathrm{x}}(n+1)$. Furthermore, $n+1$ is east of j and (using (6.1)) north of all other rectangles, so in particular $n+1$ does not intersect any rectangle, showing that P is a feasible placement.

Now, we verify for all $(k, n+1) \in E(G_\sigma)$ that k is only south of $n+1$, and for all $(k, n+1) \in E(G_{-\sigma})$ that k is only west of $n+1$.

Clearly, by construction of P, j is only west of $n+1$ in P, and j is the only predecessor of $n+1$ in $G_{-\sigma}$. As $n+1$ is north of all rectangles other than j, it remains to be shown that for $(k, n+1) \in E(G_\sigma)$, we have that k is not west of $n+1$ and not east of $n+1$. The latter already directly follows from the choice of $maxc_{\mathrm{x}}(n+1)$.

So let $(k, n+1) \in E(G_\sigma)$. If $k < j$ we have that $k = i$, and by $maxc_{\mathrm{x}}(i) > maxc_{\mathrm{x}}(j) = minc_{\mathrm{x}}(n+1)$ we have that i is not west of $n+1$.

Otherwise, i.e., $j < k$, we have $k <_\sigma n+1 <_\sigma j$, so j is west of k. Then $minc_{\mathrm{x}}(n+1) = maxc_{\mathrm{x}}(j) \leq minc_{\mathrm{x}}(k) < maxc_{\mathrm{x}}(k)$, so k is not west of $n+1$. We conclude, using Lemma 6.8, that P is a forcing placement with $r_P = r_\sigma$.

Finally, consider the case $n+1 <_\sigma n$. Since σ is biplane, $-\sigma$ is biplane as well, and $n <_{-\sigma} n+1$, so there exists a forcing placement $P' = (minc'_{\mathrm{x}}, minc'_{\mathrm{y}}, maxc'_{\mathrm{x}}, maxc'_{\mathrm{y}})$ with $r_{P'} = r_{-\sigma}$. Now let $P = (minc'_{\mathrm{y}}, minc'_{\mathrm{x}}, maxc'_{\mathrm{y}}, maxc'_{\mathrm{x}})$, i.e., exchange the role of x-coordinates and y-coordinates in P'. As the definition of forcingness is symmetric, clearly P is still a forcing placement. Moreover, for $(i,j) \in E(G_\sigma)$, we have $(i,j) \in E(G_{-(-\sigma)})$, so i is only west of j in P', resulting in i only south of j in P. Similarly, if $(i,j) \in E(G_{-\sigma})$, then i is only south of j in P', so i is only west of j in P. Hence, by Lemma 6.8, P is a forcing placement with $r_P = r_\sigma$. □

6.3 Completing the Lower Bound

Now, we show that one can apply all permutations on $[\![n]\!]$ to the forcing placements obtained from Lemma 6.10:

Lemma 6.11. *Let (π, ρ) be a biplane sequence pair. Then (π, ρ) is forced.*

Proof. As (π, ρ) is biplane, we can write $(\pi, \rho) = (\pi, \sigma \circ \pi)$ with σ biplane. We prove that there is a forcing placement P' with $r_{P'} = r_{\pi, \sigma\circ\pi}$. Since σ is biplane, by Lemma 6.10, there is a forcing placement P such that $r_P = r_\sigma$. We now show that permuting the rectangles in P according to π yields a forcing placement P' with $r_{P'} = r_{\pi,\sigma\circ\pi}$. So let $P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$ and define $P' = (minc'_{\mathrm{x}}, minc'_{\mathrm{y}}, maxc'_{\mathrm{x}}, maxc'_{\mathrm{y}})$ by, for $i \in [\![n]\!]$,

$$\begin{aligned} minc'_{\mathrm{x}}(i) &:= minc_{\mathrm{x}}(\pi(i)), & minc'_{\mathrm{y}}(i) &:= minc_{\mathrm{y}}(\pi(i)), \\ maxc'_{\mathrm{x}}(i) &:= maxc_{\mathrm{x}}(\pi(i)), & maxc'_{\mathrm{y}}(i) &:= maxc_{\mathrm{y}}(\pi(i)). \end{aligned}$$

Obviously, P' is still a forcing placement. Furthermore, for $i, j \in [\![n]\!]$ with $i \neq j$, we have

$$\begin{aligned} r_{P'}(i,j) &= r_P\big(\pi(i), \pi(j)\big) \\ &= r_\sigma\big(\pi(i), \pi(j)\big) \\ &= r_{\mathrm{id},\sigma}\big(\pi(i), \pi(j)\big) \\ &= r_{\pi,\sigma\circ\pi}(i,j). \end{aligned}$$

□

Hence, there is a large set of forced sequence pairs, and we finally obtain the new lower bound:

Theorem 6.12. *Let $n \in \mathbb{N}$. Then, we have*

$$CR_n \geq n! \cdot Biplane_n = \Omega\Big(n! \cdot \frac{c^n}{n^4}\Big),$$

where $c = 4 + 2\sqrt{2} \geq 6.828$.

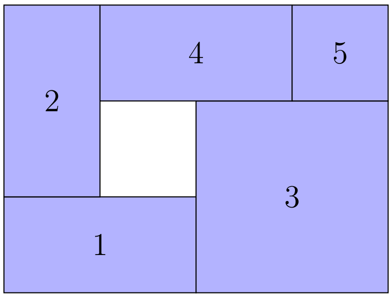


Figure 6.4: A feasible placement that is not representable by a forced representation.

Proof. By Lemma 6.11, there is a set of forced sequence pairs for n that contains a separate element for each pair π, σ of permutations where σ is biplane. By Theorem 3.15, the number of biplane permutations is $\Theta\left(\frac{(4+2\sqrt{2})^n}{n^4}\right)$. The result now follows from Observation 6.7. $\square$

In Chapter 7, we will see that this lower bound is tight for $n \leq 4$. However, for $n \geq 5$, we observe:

Proposition 6.13. *Let $n \in \mathbb{N}$ with $n \geq 5$. Then $CR_n > n! \cdot Biplane_n$.*

Proof. First, we prove the case $n = 5$. Consider the feasible placement P as depicted in Figure 6.4. We show that P is not representable by a forced representation.

So suppose that $P' = (minc'_{\mathrm{x}}, minc'_{\mathrm{y}}, maxc'_{\mathrm{x}}, maxc'_{\mathrm{y}})$ is a forcing placement such that P is represented by $r_{P'}$. The pair $(1, 5)$ is the only pair without a forced relation in P. Moreover, there is no $1 < i < 5$ such that $(1, i)$ and $(i, 5)$ have the same relation in P. Hence, the only way to force a relation for $(1, 5)$ in P' is to either let 1 be only west of 5 or let 1 be only south of 5 in P'.

For all pairs $1 \leq i < j \leq 4$, there is no k such that (i, k) and (k, j) have the same relation in P. Hence, all such (i, j) may only have one relation in P' as well. Since 3 is south of 4, but not east of 4, we have $minc'_{\mathrm{x}}(3) < maxc'_{\mathrm{x}}(4)$. This implies

$$maxc'_{\mathrm{x}}(1) \leq minc'_{\mathrm{x}}(3) < maxc'_{\mathrm{x}}(4) \leq minc'_{\mathrm{x}}(5),$$

so 1 is west of 5 in P'. Similarly, 2 is west of 3, but not north of 3, so we have $minc'_{\mathrm{y}}(2) < maxc'_{\mathrm{y}}(3)$. Then

$$maxc'_{\mathrm{y}}(1) \leq minc'_{\mathrm{y}}(2) < maxc'_{\mathrm{y}}(3) \leq minc'_{\mathrm{y}}(5),$$

so 1 is south of 5 in P'. This contradicts the requirement that 1 and 5 have only one relation in P'.

For the case $n > 5$, the same argument works after adding $n - 5$ rectangles to P that are east of $\{1, \ldots, 5\}$. $\square$

6.4 Characterization of Forced Sequence Pairs

We have seen that all biplane sequence pairs are forced. Now, we will show that *all* forced sequence pairs are of this type, that is, a sequence pair is forced if and only if it is biplane. This means that the lower bound of Theorem 6.12 is the best possible result that can be obtained from forcing placements and their forced sequence pairs.

Lemma 6.14. *Let $n \in \mathbb{N}$ and*

$$\mathcal{SP}_1 = \Big\{(\pi, \sigma \circ \pi) : \pi, \sigma \in \Pi_n, \quad \sigma \text{ is plane}\Big\},$$
$$\mathcal{SP}_2 = \Big\{(\pi, \sigma \circ \pi) : \pi, \sigma \in \Pi_n, \ -\sigma \text{ is plane}\Big\}.$$

Then, both $\mathcal{SP}_1$ and $\mathcal{SP}_2$ are complete sets of sequence pairs.

Proof. The fact that $\mathcal{SP}_1$ is complete is just a reformulation of Theorem 5.6.

To show that $\mathcal{SP}_2$ is complete, let $P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$ be a feasible placement. We obtain a new feasible placement $P' = (minc_{\mathrm{y}}, minc_{\mathrm{x}}, maxc_{\mathrm{y}}, maxc_{\mathrm{x}})$ by exchanging the role of x- and y-coordinates in P. Now, we choose a sequence pair $(\pi, \sigma' \circ \pi) \in \mathcal{SP}_1$ that represents P' and set $\sigma := -\sigma'$. As $-\sigma = -(-\sigma') = \sigma'$ is plane, we know that $(\pi, \sigma \circ \pi) \in \mathcal{SP}_2$. Moreover, $\sigma \circ \pi = (-\sigma) \circ \pi = -(\sigma' \circ \pi)$, which implies that $r_{\pi,\sigma\circ\pi}$ can obtained from $r_{\pi,\sigma'\circ\pi}$ by exchanging south and west and exchanging north and east (cf. Definition 4.1), so $(\pi, \sigma \circ \pi)$ represents P. □

Lemma 6.15. *Let P be a feasible placement that is only represented by a unique sequence pair (π, ρ). Then (π, ρ) is biplane.*

Proof. Set $\sigma := \mathrm{struc}(\pi, \rho)$. As (π, ρ) is the only sequence pair representing P, Lemma 6.14 directly implies that both σ and $-\sigma$ must be plane, so σ is biplane. □

We conclude:

Theorem 6.16. *Let (π, ρ) be a sequence pair. Then, the following statements are equivalent:*

(i) (π, ρ) *is forced.*

(ii) There is a feasible placement P such that (π, ρ) is the unique sequence pair representing P.

(iii) (π, ρ) *is biplane.*

Proof.
$(i) \implies (ii)$ is implied by Observation 6.5,
$(ii) \implies (iii)$ is implied by Lemma 6.15, and
$(iii) \implies (i)$ is implied by Lemma 6.11. □

Chapter 7
Computational Bounds

Recall that by CR_n, we refer to the minimum cardinality of a complete set of representations, that is, a set of representations that contains a representation for each feasible placement of n rectangles. By Theorem 4.24, CR_n also is the minimum cardinality of a complete set of sequence pairs. In this chapter, we will compute CR_n for small n.

Clearly, the set of sequence pairs representing a placement P only depends on the set of spatial relations satisfied in P, that is, the pair of strict partial orders $(\mathcal{S}_P, \mathcal{W}_P)$. We will call this pair the configuration of P (to be defined formally later). In Section 7.1, we will identify configurations that are not relevant for the computation of CR_n (as these are in a sense dominated by other configurations) and demonstrate how to detect them. In Section 7.2, we then show how to efficiently enumerate the set of all relevant configurations for fixed n. Finally, in Section 7.3, we reduce the computation of CR_n to a set cover problem and solve it for all $n \leq 8$. As the main result, we observe that $CR_n = n! \cdot \mathit{Baxter}_n$ for $n \leq 8$. Moreover, we introduce the notion of *symmetric* sets of sequence pairs, and demonstrate that the minimum cardinality CR_n^{sym} of symmetric complete sets of sequence pairs even satisfies $CR_n^{\text{sym}} = n! \cdot \mathit{Baxter}_n$ for all $n \leq 12$, again using optimum set cover solutions. Lastly, we observe that complete sets of sequence pairs of minimum cardinality seem to be induced by the set of permutations avoiding a certain pattern. These permutations (called *pseudo-biplane*) seem to be equinumerous with Baxter permutations.

7.1 Theoretical Foundation: Configurations

7.1.1 Interval Orders

Before we consider placements and their strict partial orders, we first deal with the one dimensional case of intervals.

Definition 7.1. *Let $n \in \mathbb{N}$. An* ***interval placement*** *is a pair of coordinate functions $I = (minc, maxc)$ with $minc, maxc\colon [\![n]\!] \to \mathbb{R}$ and $minc(i) < maxc(i)$ for all $i \in [\![n]\!]$.*

Every interval placement I induces a strict partial order:

Definition 7.2. *Let I be an interval placement. Then, the strict partial order Q_I is given by*

$$Q_I := \Big\{ (i,j) \in {}^2[\![n]\!] : maxc(i) \leq minc(j) \Big\}.$$

Definition 7.3. *Let $n \in \mathbb{N}$ and let Q be a strict partial order on $[\![n]\!]$. We say that Q is an* ***interval order*** *if there is an interval placement I with $Q_I = Q$.*

Observation 7.4. *Let P be a placement. Then $\mathcal{S}_P$ and $\mathcal{W}_P$ are interval orders.*

This means that when enumerating pairs $(\mathcal{S}, \mathcal{W})$ that are candidates for $(\mathcal{S}_P, \mathcal{W}_P)$ of some placement P, we need to ensure that $\mathcal{S}$ and $\mathcal{W}$ are interval orders. To this end, as the next step, we will characterize interval orders. First, we observe that there are strict partial orders that are not interval orders:

Proposition 7.5. *Let $Q \subset {}^2[\![4]\!]$ be given by*

$$Q := \big\{ (1,2), (3,4) \big\}.$$

Then Q is a strict partial order, but not an interval order.

Proof. One easily verifies that indeed Q is a strict partial order.

Now assume there is an interval placement $I = (minc, maxc)$ with $Q = Q_I$. Then

$$maxc(1) \overset{(1,2)\in Q}{\leq} minc(2) \overset{(3,2)\notin Q}{<} maxc(3) \overset{(3,4)\in Q}{\leq} minc(4),$$

contradicting $(1,4) \notin Q$. □

In fact, Fishburn [Fis70] showed that a strict partial order is an interval order if and only if it does not contain four elements that compare as in Q.

In order to efficiently detect interval orders, given a relation $Q \subseteq {}^2[\![n]\!]$, we model constraints on the coordinate functions $minc, maxc$ of an interval placement I with $Q_I = Q$ using a system of linear inequalities. Then, we will efficiently detect whether that system of inequalities is feasible in $\mathcal{O}(n^2)$ time, exploiting the special structure of that system.

Before we proceed to describe that approach in detail, we need some definitions:

Definition 7.6. *A* ***weighted digraph*** (G, w) *is a digraph* G *together with a weight function* $w\colon E(G) \to \mathbb{R}$*. We call* (G, w) ***conservative*** *if* (G, w) *does not contain any cycle* C *with*

$$w(C) := \sum_{e \in E(C)} w(e) < 0.$$

Definition 7.7. *Let* (G, w) *be a weighted digraph and let* $\lambda\colon V(G) \to \mathbb{R}$*. We define the reduced cost function* $w_\lambda\colon E(G) \to \mathbb{R}$ *by*

$$w_\lambda((u, v)) \;:=\; w((u, v)) + \lambda(u) - \lambda(v).$$

We call λ *a* ***feasible potential*** *if* $w_\lambda(e) \geq 0$ *for all* $e \in E(G)$*.*

Clearly, if λ is a feasible potential of (G, w), then (G, w) must be conservative, since $w(C) = w_\lambda(C) \geq 0$ for any cycle C. One can show that the reverse statement is also true, that is, there exists a feasible potential λ of (G, w) if and only if (G, w) is conservative ([KV18]).

Feasible potentials are most prominently used in flow and shortest path problems, where one exploits that computing a shortest path with respect to w_λ yields a shortest path with respect to w. In particular, feasible potentials allow to use Dijkstra's algorithm ([Dij59]), which requires nonnegative edge weights, to compute shortest paths in general, conservative graphs (G, w) ([BL74]). Moreover, feasible potentials (also called future cost estimates in this context) allow to speed up Dijkstra's algorithm in practice ([HNR68]). Note that feasible potentials are exactly the feasible solutions of the dual of the shortest path LP.

The problem of finding a feasible potential is equivalent to determining values $\lambda(v)$ for all $v \in V(G)$ such that a certain set of linear inequalities is satisfied. This means that we can also use algorithms to compute feasible potentials to solve systems of linear inequalities of this form.

In general graphs (G, w), one can compute a feasible potential (or detect that none exists) in $\mathcal{O}\Big(\big|V(G)\big| \cdot \big|E(G)\big|\Big)$ time ([KV18]) using the Moore-Bellman-Ford algorithm ([Moo59; Bel58; For56]). However, in our case, we only need to compute feasible potentials for acyclic digraphs:

Lemma 7.8. *Let* (G, w) *be a weighted, acyclic digraph. Then, we can compute a feasible potential* λ *of* (G, w) *in* $\mathcal{O}\Big(\big|V(G)\big| + \big|E(G)\big|\Big)$ *time.*

Proof. First, we remark that if (G, w) is acyclic, then (G, w) must also be conservative, so a feasible potential λ exists.

To compute λ, simply set

$$\lambda(v) \;=\; 0$$

for all vertices $v \in V(G)$ without ingoing edges. Then, process the remaining vertices in topological order and set

$$\lambda(v) \;=\; \min\big\{\, \lambda(u) + w((u, v)) : (u, v) \in E(G) \,\big\}.$$

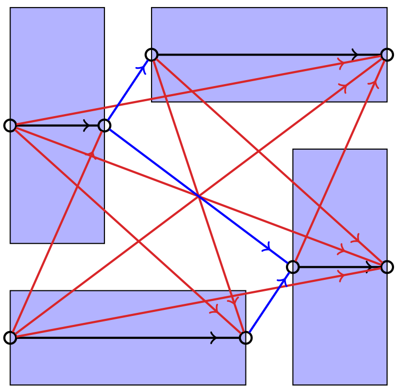


Figure 7.1: Constraint graph of $\mathcal{W}_P$ for a placement P. For a rectangle i, we draw the vertex v_i^{min} at the center of i's left border, and v_i^{max} at the center of i's right border. Edges in E_1 are black, edges in E_2 red, and edges in E_3 blue.

The procedure above is well-defined, since $\lambda(u)$ is computed before $\lambda(v)$ if $(u, v) \in E(G)$. Moreover, the resulting function λ clearly is a feasible potential, and the running time guarantee is satisfied, where we exploit that a topological order of G can be computed in $\mathcal{O}\Big(\big|V(G)\big| + \big|E(G)\big|\Big)$ time ([KV18]). $\square$

Now, we introduce a weighted digraph that contains two vertices v_i^{min}, v_i^{max} for each interval index $i \in [\![n]\!]$. The feasible potential will yield coordinates of intervals, and weighted edges are used to model constraints on distances of coordinates. A similar construction was considered by Fekete and Schepers [FS97].

Definition 7.9. *Let $Q \subseteq {}^2[\![n]\!]$ and let*

$$E_1 := \Big\{ (v_i^{min}, v_i^{max}) : i \in [\![n]\!] \Big\},$$
$$E_2 := \Big\{ (v_j^{min}, v_i^{max}) : (i, j) \notin Q \Big\},$$
$$E_3 := \Big\{ (v_i^{max}, v_j^{min}) : (i, j) \in Q \Big\}.$$

*The **constraint graph** of Q is the weighted digraph (G_Q, w_Q) with vertex set*

$$V(G_Q) := \bigcup_{i \in [\![n]\!]} \Big\{ v_i^{min}, v_i^{max} \Big\}$$

and edge set

$$E(G_Q) := E_1 \cup E_2 \cup E_3$$

with

$$w_Q((u, v)) := \begin{cases} -1 & \text{if } (u, v) \in E_1 \cup E_2, \\ 0 & \text{if } (u, v) \in E_3. \end{cases}$$

An illustration of G_Q with $Q = \mathcal{W}_P$ is given in Figure 7.1.

Lemma 7.10. *Let $Q \subseteq {}^2[\![n]\!]$. Then Q is an interval order if and only if (G_Q, w_Q) is conservative. Moreover, assume that λ is a feasible potential of (G_Q, w_Q), and let $minc, maxc\colon [\![n]\!] \to \mathbb{R}$ be given by*

$$\begin{aligned} minc(i) &:= -\lambda(v_i^{min}), \\ maxc(i) &:= -\lambda(v_i^{max}). \end{aligned}$$

Then $I_\lambda = \big(minc, maxc\big)$ is an interval placement with $Q_{I_\lambda} = Q$.

Proof. Let $\lambda'\colon V(G_Q) \to \mathbb{R}$ be arbitrary and let $I_{\lambda'}$ be defined as above. Then $I_{\lambda'}$ is an interval placement if and only if

$$-\lambda'\Big(v_i^{min}\Big) < -\lambda'\big(v_i^{max}\big) \qquad\qquad \forall i \in [\![n]\!].$$

Moreover, we have $Q = Q_I$ if and only if

$$\begin{aligned} -\lambda'\big(v_i^{max}\big) &> -\lambda'\Big(v_j^{min}\Big) && \forall (i,j) \notin Q, \\ -\lambda'\big(v_i^{max}\big) &\leq -\lambda'\Big(v_j^{min}\Big) && \forall (i,j) \in Q. \end{aligned}$$

If λ' satisfies these constraints, we can w.l.o.g. assume that all strict inequalities are satisfied with a slack of at least 1 by scaling. Hence, after multiplying with -1, we see that λ' exists if and only if there is $\lambda\colon V(G_Q) \to \mathbb{R}$ with

$$\begin{aligned} \lambda\Big(v_i^{min}\Big) - 1 &\geq \lambda\big(v_i^{max}\big) && \forall i \in [\![n]\!], \\ \lambda\Big(v_j^{min}\Big) - 1 &\geq \lambda\big(v_i^{max}\big) && \forall (i,j) \notin Q, \\ \lambda\big(v_i^{max}\big) &\geq \lambda\Big(v_j^{min}\Big) && \forall (i,j) \in Q, \end{aligned}$$

which are exactly the requirements for λ to be a feasible potential of (G_Q, w_Q). Moreover, the constraints for λ imply the constraints for λ', showing that indeed I_λ is an interval placement with $Q_{I_\lambda} = Q$. □

Lemma 7.10 implies that we can use the Moore-Bellman-Ford algorithm ([Moo59; Bel58; For56]) to detect whether a given set $Q \subseteq {}^2[\![n]\!]$ is an interval order in $\mathcal{O}(n^3)$ time. However, Corollary 7.12 shows that we can improve the running time to $\mathcal{O}(n^2)$:

Lemma 7.11. *Let $Q \subseteq {}^2[\![n]\!]$. Then (G_Q, w_Q) is conservative if and only if G_Q is acyclic.*

Proof. We show that G_Q does not contain any cycles of nonnegative weight. As all edges in G_Q have nonpositive weight, it suffices to show that G_Q does not contain cycles consisting only of zero weight edges. But all zero weight edges are of the form (v_i^{max}, v_j^{min}) and hence clearly do not form a cycle. □

Corollary 7.12. *Let $Q \subseteq {}^2[\![n]\!]$. Then, in $\mathcal{O}(n^2)$ time, we can detect that Q is not an interval order or compute an interval placement I with $Q_I = Q$.*

Proof. First, in $\mathcal{O}(n^2)$ time, we compute a topological order of G_Q or detect that G_Q contains a cycle [KV18], in which case Lemma 7.10 and Lemma 7.11 show that Q is not an interval order. Otherwise, we compute a feasible potential of G_Q in $\mathcal{O}(n^2)$ time using Lemma 7.8, which directly induces an interval placement I with $Q_I = Q$ using Lemma 7.10. □

The following condition will be useful later on:

Lemma 7.13. *Let Q be an interval order and $(i, j) \in Q$. Then $Q - (i, j)$ is not an interval order if and only if v_j^{min} is reachable from v_i^{max} in $G_Q - (v_i^{max}, v_j^{min})$, that is,*

$$\operatorname{tr}\bigl(G_Q\bigr) = \operatorname{tr}\Bigl(G_Q - (v_i^{max}, v_j^{min})\Bigr).$$

Proof. By Lemmata 7.10 and 7.11, G_Q is acyclic, and $Q - (i, j)$ is not an interval order if and only if its constraint graph $G_{Q-(i,j)}$ contains a cycle. But $G_{Q-(i,j)}$ is obtained from G_Q by reversing the direction of the edge (v_i^{max}, v_j^{min}), and hence $G_{Q-(i,j)}$ contains a cycle if and only if v_j^{min} is reachable from v_i^{max} in $G_Q - (v_i^{max}, v_j^{min})$. □

7.1.2 Configurations

Recall that a pair $(\mathcal{S}, \mathcal{W})$ of strict partial orders on $[\![n]\!]$ is *biordering* if each pair of elements $(i, j) \in {}^2[\![n]\!]$ is comparable in at least one of $\mathcal{S}$ and $\mathcal{W}$.

Observation 7.14. *Let $(\mathcal{S}, \mathcal{W})$ be a pair of strict partial orders on $[\![n]\!]$. Then, we have:*

(i) There is a placement P with $(\mathcal{S}_P, \mathcal{W}_P) = (\mathcal{S}, \mathcal{W})$ if and only if $(\mathcal{S}, \mathcal{W})$ is a pair of interval orders.

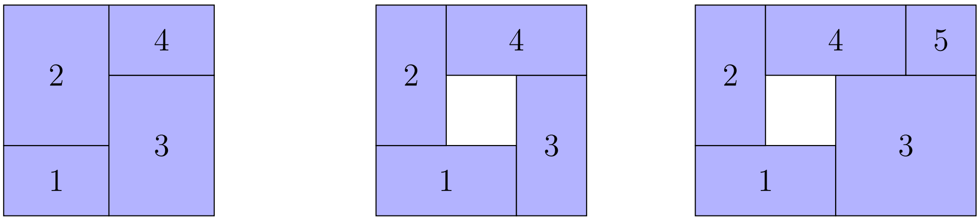
(ii) There is a feasible placement P with $(\mathcal{S}_P, \mathcal{W}_P) = (\mathcal{S}, \mathcal{W})$ if and only if $(\mathcal{S}, \mathcal{W})$ is a biordering pair of interval orders.

This fact motivates the notion of configurations:

Definition 7.15. *A **configuration** $(\mathcal{S}, \mathcal{W})$ on $[\![n]\!]$ is a biordering pair of interval orders on $[\![n]\!]$. For a feasible placement P, we refer to $(\mathcal{S}_P, \mathcal{W}_P)$ as the configuration of P.*

Now, let $\mathcal{SP} \subseteq \mathcal{SP}_n$ be a set of sequence pairs. We say that $\mathcal{SP}$ **covers** a configuration $(\mathcal{S}, \mathcal{W})$ if it contains a sequence pair (π, ρ) representing $(\mathcal{S}, \mathcal{W})$, i.e., $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$. Clearly, we have:

$$\begin{aligned} & \mathcal{SP} \text{ is complete for } n \\ \iff\ & \mathcal{SP} \text{ covers } (\mathcal{S}_P, \mathcal{W}_P) \text{ for all feasible } n\text{-placements } P \\ \iff\ & \mathcal{SP} \text{ covers all configurations } (\mathcal{S}, \mathcal{W}) \text{ on } [\![n]\!] \end{aligned} \tag{7.1}$$



(a) A non-tight placement.

(b) A tight placement.

(c) A tight placement containing the pair $(1, 5)$ satisfying two spatial relations.

Figure 7.2: Examples of (non-)tight placements.

This means that we can compute CR_n by explicitly enumerating all configurations and solving a set cover problem (formally defined in Section 7.3.1): Every sequence pair $(\pi, \rho) \in \mathcal{SP}_n$ corresponds to a set of represented configurations, and we want to find a minimum cardinality set of sequence pairs $\mathcal{SP}$ that covers all configurations.

7.1.3 Tight Configurations

Next, we show a sufficient condition for configurations to be irrelevant for the computation of CR_n. This will not only result in smaller set cover instances, but will also allow us to avoid the enumeration of many configurations.

As a motivating example, consider the placement P_a given in Figure 7.2(a) and its configuration: Rectangle 1 is both west and south of rectangle 4, but there is a feasible placement P_b (given in Figure 7.2(b)) where 1 is only south of 4, and all remaining pairs of rectangles satisfy the same unique spatial relation in both placements. This means that any representation of P_b also represents P_a and hence we can ignore the configuration of P_a for the computation of CR_4. This fact motivates the concept of *tightness*:

Definition 7.16. *Let $(\mathcal{S}, \mathcal{W})$ be a configuration.*

- *We say that a configuration $(\mathcal{S}', \mathcal{W}') \neq (\mathcal{S}, \mathcal{W})$ **dominates** $(\mathcal{S}, \mathcal{W})$ if $\mathcal{S}' \subseteq \mathcal{S}$ and $\mathcal{W}' \subseteq \mathcal{W}$.*

- *We call $(\mathcal{S}, \mathcal{W})$ **tight** if there is no configuration that dominates $(\mathcal{S}, \mathcal{W})$.*

- *We call a feasible placement P **tight** if $(\mathcal{S}_P, \mathcal{W}_P)$ is tight.*

The placement P_c depicted in Figure 7.2(c) is tight although the rectangles 1 and 5 satisfy two spatial relations. Note that the proof of Proposition 6.13 implies that P_c is indeed tight. Now, if some configuration $(\mathcal{S}, \mathcal{W})$ is not tight, there is a configuration $(\mathcal{S}', \mathcal{W}') \neq (\mathcal{S}, \mathcal{W})$ that dominates $(\mathcal{S}, \mathcal{W})$. If (π, ρ) is a sequence pair representing $(\mathcal{S}', \mathcal{W}')$, then $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}' \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}' \subseteq \mathcal{W}$, so (π, ρ) also represents $(\mathcal{S}, \mathcal{W})$.

Hence, using (7.1) and Theorem 4.24, we conclude:

Observation 7.17. *Let $n \in \mathbb{N}$ and $\mathcal{C}_n^T$ be the set of tight configurations on $[\![n]\!]$. Then, we have*

$$CR_n = \min\Big\{ |\mathcal{SP}| : \mathcal{SP} \subseteq \mathcal{SP}_n \text{ covers all } (\mathcal{S}, \mathcal{W}) \in \mathcal{C}_n^T \Big\}.$$

Before we proceed to characterize tightness, we first need a useful result on general acyclic digraphs:

Lemma 7.18. *Let G be an acyclic digraph and $F \subseteq E(G)$ be a set of edges. Then,* $\mathrm{tr}(G - F) = \mathrm{tr}(G)$ *if and only if* $\mathrm{tr}(G - f) = \mathrm{tr}(G)$ *for all $f \in F$.*

Proof. For the first direction, assume that $\mathrm{tr}(G - F) = \mathrm{tr}(G)$ and let $f \in F$. Then

$$E\big(\mathrm{tr}(G)\big) = E\big(\mathrm{tr}(G - F)\big) \subseteq E\big(\mathrm{tr}(G - f)\big) \subseteq E\big(\mathrm{tr}(G)\big),$$

hence $\mathrm{tr}(G - f) = \mathrm{tr}(G)$.

For the other direction, assume that $\mathrm{tr}(G - f) = \mathrm{tr}(G)$ for all $f \in F$. We show $F \subseteq \mathrm{tr}(G - F)$, implying $\mathrm{tr}(G - F) = \mathrm{tr}(G)$.

For every $(u, v) = f \in F$, there is a u-v path H_f with $\big|E(H_f)\big| \geq 2$ in $G - f$. Consider some $(u, v) = f \in F$. If H_f is a path in $G - F$, we are done. Otherwise, there is $f' \in E(H_f) \cap F$, and we can replace f' in H_f by the path $H_{f'}$. As G is acyclic, the result is a strictly longer u-v path. Since the length of any path in G is bounded, this procedure must terminate after finitely many steps with a u-v path in $G - F$. □

Lemmata 7.13 and 7.18 imply:

Observation 7.19. *Let Q be an interval order and $Q' \subseteq Q$. Furthermore, let $F := \Big\{ (v_i^{max}, v_j^{min}) : (i, j) \in Q' \Big\}$. Then, the following conditions are equivalent:*

(i) $\mathrm{tr}\big(G_Q - F\big) = \mathrm{tr}\big(G_Q\big)$.

(ii) $Q - (i, j)$ is not an interval order for all $(i, j) \in Q'$.

Now, we can characterize tightness which in particular allows to efficiently detect tightness:

Lemma 7.20. *Let $(\mathcal{S}, \mathcal{W})$ be a configuration.*
Then $(\mathcal{S}, \mathcal{W})$ is tight if and only if

(i) $\mathcal{S} - (i, j)$ is not an interval order for all $(i, j) \in \mathcal{S} \cap \mathrm{sym}(\mathcal{W})$, and

(ii) $\mathcal{W} - (i, j)$ is not an interval order for all $(i, j) \in \mathcal{W} \cap \mathrm{sym}(\mathcal{S})$.

Proof. For the first direction, assume in case (ii) that $\mathcal{W} - (i,j)$ is an interval order for a pair $(i,j) \in \mathcal{W} \cap \operatorname{sym}(\mathcal{S})$. Then $(\mathcal{S}, \mathcal{W} - (i,j))$ is a biordering pair of interval orders (i.e., a configuration) and hence dominates $(\mathcal{S}, \mathcal{W})$. The other case follows by symmetry.

For the other direction, let $(\mathcal{S}', \mathcal{W}')$ be a configuration that dominates $(\mathcal{S}, \mathcal{W})$ and w.l.o.g. assume that $\mathcal{W} \setminus \mathcal{W}'$ is not empty. We will show that there is a pair $(i,j) \in \mathcal{W} \setminus \mathcal{W}'$ such that $\mathcal{W} \setminus \{(i,j)\}$ is an interval order. This will imply the result: As $(\mathcal{S}', \mathcal{W}')$ is a biorder and $(i,j) \notin \mathcal{W}'$, we must have $(i,j) \in \mathcal{W} \cap \operatorname{sym}(\mathcal{S})$.

Set $\Delta := \mathcal{W} \setminus \mathcal{W}'$ and assume, for the sake of contradiction, that $\mathcal{W} - (i,j)$ is not an interval order for all $(i,j) \in \Delta$. Then Observation 7.19 implies that for all $(i,j) \in \Delta$, the vertex v_j^{min} is reachable from v_i^{max} in $G' := G_{\mathcal{W}} - \left\{ (v_{i'}^{max}, v_{j'}^{min}) : (i',j') \in \Delta \right\}$. But $E(G') \cup \left\{ (v_{j'}^{min}, v_{i'}^{max}) : (i',j') \in \Delta \right\} \subseteq E(G_{\mathcal{W}'})$, contradicting that $G_{\mathcal{W}'}$ is acyclic. □

Proposition 7.21. *Let* $(\mathcal{S}, \mathcal{W})$ *be a configuration on* $[\![n]\!]$*, and* $2 < \omega < 2.373$ *the current best matrix multiplication constant. Then, in* $\mathcal{O}(n^\omega)$ *time, we can detect whether* $(\mathcal{S}, \mathcal{W})$ *is tight.*

Proof. Observation 7.19 implies that in order to test the conditions of Lemma 7.20, it suffices to compute the transitive closure of two digraphs on $2n$ vertices. Munro [Mun71] and Furman [Fur70] have shown that an algorithm that computes the product of two Boolean $k \times k$ matrices in $\mathcal{O}(k^\omega)$ time implies an algorithm to compute the transitive closure of a k-vertex digraph in $\mathcal{O}(k^\omega)$ time. The currently fastest matrix multiplication algorithm due to Le Gall [LeG14] achieves a running time of $\mathcal{O}(k^\omega)$ with $\omega < 2.373$. □

In practice, we use Observation 7.19 to detect tightness in $\mathcal{O}(n^3)$ time: It suffices to compute the transitive closures of two acyclic digraphs, which we do by processing the vertices in reverse topological order. Note that this is still faster than a naïve implementation of Lemma 7.20, which would take $\mathcal{O}(n^4)$ time by applying Corollary 7.12 $\mathcal{O}(n^2)$ times.

Now, consider a configuration $(\mathcal{S}, \mathcal{W})$. By Lemma 7.20, we know that in order to check whether $(\mathcal{S}, \mathcal{W})$ is tight, we only need to verify whether there is any pair (i,j) that is comparable in both $\mathcal{S}$ and $\mathcal{W}$ and can be removed from $\mathcal{S}$ or $\mathcal{W}$ while maintaining an interval order.

For example, assume there is a pair $(i,l) \in \mathcal{S} \cap \mathcal{W}$ (the other case $(i,l) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$ is symmetric): If $\mathcal{W} - (i,l)$ is not an interval order, then there is a path from v_i^{max} to v_l^{min} in the constraint graph $G_{\mathcal{W}}$ that does not use the edge (v_i^{max}, v_l^{min}). Lemma 7.23 shows that there always is such a path that visits at most two other rectangles j and k, and gives additional properties of j and k that will be useful later on. In the case of Figure 7.2(c) with $i = 1$ and $l = 5$, we get $j = 3$ and $k = 4$. Before we proceed, we need a technical result used in the proof of Lemma 7.23:

Lemma 7.22. *Let $(\mathcal{S}, \mathcal{W})$ be a configuration on $[\![n]\!]$. Then, there is an integral placement $P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$ with $(\mathcal{S}_P, \mathcal{W}_P) = (\mathcal{S}, \mathcal{W})$ s.t. all* x*-coordinates are pairwise different and all* y*-coordinates are pairwise different.*

Proof. First, choose a placement $P' = (minc'_{\mathrm{x}}, minc'_{\mathrm{y}}, maxc'_{\mathrm{x}}, maxc'_{\mathrm{y}})$ with $(\mathcal{S}_{P'}, \mathcal{W}_{P'}) = (\mathcal{S}, \mathcal{W})$ and integral coordinates, which can be obtained by scaling any placement with rational coordinates by a sufficiently large integer.
Define $P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$ using, for $i \in [\![n]\!]$,

$$
\begin{aligned}
minc_{\mathrm{x}}(i) &:= minc'_{\mathrm{x}}(i) + \frac{1}{3i}, & minc_{\mathrm{y}}(i) &:= minc'_{\mathrm{y}}(i) + \frac{1}{3i}, \\
maxc_{\mathrm{x}}(i) &:= maxc'_{\mathrm{x}}(i) - \frac{1}{3i}, & maxc_{\mathrm{y}}(i) &:= maxc'_{\mathrm{y}}(i) - \frac{1}{3i}.
\end{aligned}
$$

As the difference between any pair of coordinates is reduced by at most $\frac{2}{3} < 1$, any $<$-relations on pairs of coordinates in P' are preserved in P.
Hence, in particular $minc_{\mathrm{x}}(i) < maxc_{\mathrm{x}}(i)$ and $minc_{\mathrm{y}}(i) < maxc_{\mathrm{y}}(i)$ for all $i \in [\![n]\!]$, so P is a placement. Moreover, since min-coordinates were only increased and max-coordinates only decreased, all spatial relations that hold in P' still hold in P. On the other hand, all spatial relations that hold in P also hold in P': If i is not west of j in P', then $maxc'_{\mathrm{x}}(i) > minc'_{\mathrm{x}}(j)$ and thus $maxc_{\mathrm{x}}(i) > minc_{\mathrm{x}}(j)$. The other cases are shown in the same way. Finally, of course all x-coordinates and all y-coordinates in P are pairwise different by construction. Finally, to obtain an integral placement with the same properties, we can again scale all coordinates. □

To keep notation simple, Lemma 7.23 is only formulated and shown for one of the possible cases, the other cases then follow by symmetry. Its setting is illustrated in Figure 7.3.

Lemma 7.23. *Let $(\mathcal{S}, \mathcal{W})$ be a tight configuration on $[\![n]\!]$ and $(i, l) \in \mathcal{S} \cap \mathcal{W}$. Then, there are rectangles $j, k \in [\![n]\!]$ with*

(i) $(i, j) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$,

(ii) $(k, l) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$.

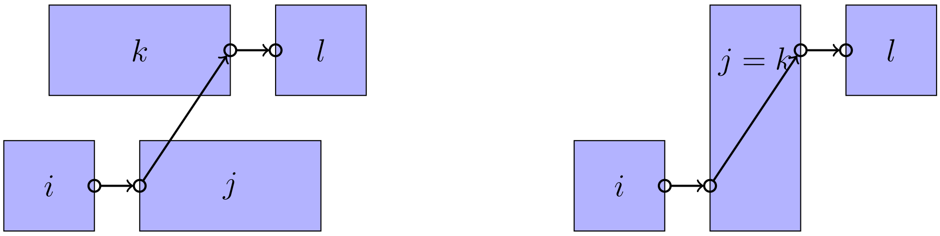
Moreover, if $j \neq k$, then

(iii) $(j, k) \in (\mathcal{S} \cup \mathcal{W}) \setminus \left(\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}\right)$.

In particular, $v_i^{max}, v_j^{min}, v_k^{max}, v_l^{min}$ are the vertices of a path of cost -1 in $G_{\mathcal{W}}$.

Proof. By Lemma 7.22, there is a feasible placement

$$P = (minc_{\mathrm{x}}, minc_{\mathrm{y}}, maxc_{\mathrm{x}}, maxc_{\mathrm{y}})$$



(a) The case $j \neq k$ with $(j,k) \in \mathcal{S} \setminus \mathrm{sym}(\mathcal{W})$. Note that $(j,k) \in \mathcal{S} \cap \mathcal{W}$ or $(j,k) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$ are also possible.

(b) The case $j = k$.

Figure 7.3: The setting of Lemma 7.23: If $(i,j) \in \mathcal{S} \cap \mathcal{W}$, then Lemma 7.23 guarantees the existence of j, k such that j restricts i in east direction, and k restricts l in west direction. Moreover, either $j = k$ (left), or $(j,k) \in \mathcal{S} \cup \mathcal{W}$ (right). In particular, $v_i^{max}, v_j^{min}, v_k^{max}, v_l^{min}$ form a path in $G_{\mathcal{W}}$. By symmetry, there also must be rectangles that restrict i in north direction, and l in south direction, yielding a path in $G_{\mathcal{S}}$.

with all different coordinates and $(\mathcal{S}_P, \mathcal{W}_P) = (\mathcal{S}, \mathcal{W})$. Now, choose j minimizing $minc_{\mathrm{x}}(j)$ with

$$(i,j) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S}).$$

If there is no such j, then $(\mathcal{S}, \mathcal{W})$ cannot be tight: There is no rectangle only east of i, so setting $maxc_{\mathrm{x}}(i) := minc_{\mathrm{x}}(l) + 1$ yields a feasible placement P' where i is not west of l, so $(\mathcal{S}_{P'}, \mathcal{W}_{P'})$ dominates $(\mathcal{S}, \mathcal{W})$. Hence, j exists. Furthermore, we must have $minc_{\mathrm{x}}(j) < minc_{\mathrm{x}}(l)$: Otherwise, as all coordinates in P are pairwise different, we have $minc_{\mathrm{x}}(j) > minc_{\mathrm{x}}(l)$, and setting $maxc_{\mathrm{x}}(i) := minc_{\mathrm{x}}(j)$ leads to a feasible placement P' such that $(\mathcal{S}_{P'}, \mathcal{W}_{P'})$ dominates $(\mathcal{S}, \mathcal{W})$, contradicting tightness of $(\mathcal{S}, \mathcal{W})$.

Using the same argument, we can choose k maximizing $maxc_{\mathrm{x}}(k)$ with

$$(k,l) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S}),$$

and we know that $maxc_{\mathrm{x}}(i) < maxc_{\mathrm{x}}(k)$. If $j = k$, we are done, so assume $j \neq k$. It remains to be shown that

$$(j,k) \in (\mathcal{S} \cup \mathcal{W}) \setminus \left(\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}\right).$$

As P is feasible, it suffices to show $(j,k) \notin \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$. We have

$$minc_{\mathrm{y}}(j) \overset{(i,j)\notin\mathcal{S}}{<} maxc_{\mathrm{y}}(i) \overset{(i,l)\in\mathcal{S}}{\leq} minc_{\mathrm{y}}(l) \overset{(k,l)\notin\mathcal{S}}{<} maxc_{\mathrm{y}}(k),$$

so $(j,k) \notin \overleftarrow{\mathcal{S}}$.

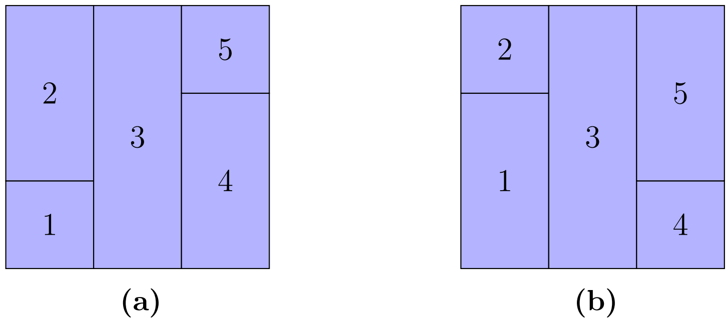


Figure 7.4: Two tight placements with different configurations that are represented by the same unique sequence pair.

Finally, assume that $(j,k) \in \overleftarrow{\mathcal{W}}$. Then $maxc_{\mathrm{x}}(k) < minc_{\mathrm{x}}(j)$, and setting $maxc_{\mathrm{x}}(i) := minc_{\mathrm{x}}(j)$ and $minc_{\mathrm{x}}(l) := maxc_{\mathrm{x}}(k)$ yields a placement P' where i is not west of l. The choice of j and k implies that P' is feasible. Again, now $(\mathcal{S}_{P'}, \mathcal{W}_{P'})$ dominates $(\mathcal{S}, \mathcal{W})$, contradicting tightness of $(\mathcal{S}, \mathcal{W})$. □

7.1.4 SP-Equivalence

In Section 7.1.3, we have seen how to identify configurations that are dominated by other configurations and hence do not need to be considered for the computation of CR_n.

In this section, we first observe that there are *different* (possibly tight) configurations that are represented by the *same* set of sequence pairs, in which case we only need to consider one of the two. We will call configurations **SP-equivalent** if they are represented by the same set of sequence pairs, and give a characterization of SP-equivalence. In contrast to tightness, which is only defined for configurations, all results in this section apply to general biorders.

For example, consider the two placements depicted in Figure 7.4 that have different configurations. As these are forcing, both are represented by a unique sequence pair, and clearly these sequence pairs must be the same, as for all pairs (i,j) the same relation is forced (cf. Definition 6.1) in both placements for i and j.

Definition 7.24. *Let $(\mathcal{S}, \mathcal{W})$ and $\big(\mathcal{S}', \mathcal{W}'\big)$ be biorders on $[\![n]\!]$. We say that $(\mathcal{S}, \mathcal{W})$ and $\big(\mathcal{S}', \mathcal{W}'\big)$ are **SP-equivalent** if $(\mathcal{S}, \mathcal{W})$ and $\big(\mathcal{S}', \mathcal{W}'\big)$ are represented by the same set of sequence pairs, that is,*

$$\big\{ (\pi,\rho) : \mathcal{S}_{\pi,\rho} \subseteq \mathcal{S} \text{ and } \mathcal{W}_{\pi,\rho} \subseteq \mathcal{W} \big\} = \big\{ (\pi,\rho) : \mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}' \text{ and } \mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}' \big\}.$$

Using Observation 7.17, we get:

Observation 7.25. *Let $n \in \mathbb{N}$ and $\mathcal{C}_n^T$ be the set of tight configurations on $[\![n]\!]$. Moreover, let $\mathcal{C}_n^{T,SP} \subseteq \mathcal{C}_n^T$ be a set that contains a representative of each SP-equivalence class of $\mathcal{C}_n^T$.*

Then, we have

$$CR_n = \min\Big\{ |\mathcal{SP}| : \mathcal{SP} \subseteq \mathcal{SP}_n \text{ covers all } (\mathcal{S},\mathcal{W}) \in \mathcal{C}_n^{T,SP} \Big\}.$$

In Figure 7.4, we see that for all pairs (i,j) that have different satisfied spatial relations in the two placements (i.e., $(1,5)$ and $(2,4)$), there is a common forced relation for (i,j) in both placements. We will show that this is always the case.

Lemma 7.26. *Let G and G' be two acyclic digraphs on the same vertex set. Then the set of topological orders of G equals the set of topological orders of G' if and only if* $\operatorname{tr}(G) = \operatorname{tr}(G')$.

Proof. A permutation π is a topological order of G if and only if π is a topological order of $\operatorname{tr}(G)$, which shows the first direction.

Now, assume that $\operatorname{tr}(G) \neq \operatorname{tr}(G')$, and w.l.o.g. there is an edge $(i,j) \in E(G')$ such that j is not reachable from i in G. Then $G + (j,i)$ is acyclic, so let π be a topological order of $G + (j,i)$. Then π is also a topological order of G, but not a topological order of G'. □

Lemmata 4.15 and 7.26 imply that we can express SP-equivalence in terms of the transitive closures of G_{SW} and G_{SE}:

Corollary 7.27. *Let $(\mathcal{S},\mathcal{W})$ and $\big(\mathcal{S}',\mathcal{W}'\big)$ be biorders on $[\![n]\!]$, and let G_{SW}, G_{SE}, G'_{SW} and G'_{SE} be the south-west and south-east digraphs of $(\mathcal{S},\mathcal{W})$ and $\big(\mathcal{S}',\mathcal{W}'\big)$, respectively.*
Then $(\mathcal{S},\mathcal{W})$ and $\big(\mathcal{S}',\mathcal{W}'\big)$ are SP-equivalent if and only if $\operatorname{tr}(G_{\mathrm{SW}}) = \operatorname{tr}(G'_{\mathrm{SW}})$ *and* $\operatorname{tr}(G_{\mathrm{SE}}) = \operatorname{tr}(G'_{\mathrm{SE}})$.

Consider a biorder $(\mathcal{S},\mathcal{W})$. Then, the set $\operatorname{tr}\big(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\big)$ consists of exactly the pairs (i,j) for which west is forced in $(\mathcal{S},\mathcal{W})$. With this in mind, we can define the reduction of a biorder, which will play a crucial role to determine SP-equivalence classes:

Definition 7.28. *Let $(\mathcal{S},\mathcal{W})$ be a biorder on $[\![n]\!]$. The* ***reduction***

$$\operatorname{red}\big((\mathcal{S},\mathcal{W})\big) := (\mathcal{S}_{\mathrm{red}}, \mathcal{W}_{\mathrm{red}})$$

of $(\mathcal{S},\mathcal{W})$ is given by

$$\mathcal{S}_{\mathrm{red}} := \mathcal{S} \setminus \left(\operatorname{tr}\Big(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\Big) \cup \operatorname{tr}\Big(\overleftarrow{\mathcal{W}} \setminus \operatorname{sym}(\mathcal{S})\Big) \right),$$

$$\mathcal{W}_{\mathrm{red}} := \mathcal{W} \setminus \left(\operatorname{tr}\Big(\mathcal{S} \setminus \operatorname{sym}(\mathcal{W})\Big) \cup \operatorname{tr}\Big(\overleftarrow{\mathcal{S}} \setminus \operatorname{sym}(\mathcal{W})\Big) \right).$$

In other words, in the reduction of $(\mathcal{S}, \mathcal{W})$, we remove all pairs (i, j) from $\mathcal{S}$ that have a forced relation different from south, and remove all pairs (i, j) from $\mathcal{W}$ that have a forced relation different from west.

Lemma 7.29. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder. Then, its reduction $\operatorname{red}\big((\mathcal{S}, \mathcal{W})\big)$ is a biorder.*

Proof. Let $\operatorname{red}\big((\mathcal{S}, \mathcal{W})\big) = (\mathcal{S}_{\text{red}}, \mathcal{W}_{\text{red}})$ be the reduction of $(\mathcal{S}, \mathcal{W})$. Since at most one relation can be forced for a given pair (i, j) in $(\mathcal{S}, \mathcal{W})$, clearly every pair (i, j) is comparable in at least one of $\mathcal{S}_{\text{red}}$ and $\mathcal{W}_{\text{red}}$. As $\mathcal{S}_{\text{red}}$ and $\mathcal{W}_{\text{red}}$ are subsets of strict partial orders, we only need to show that $\mathcal{S}_{\text{red}}$ and $\mathcal{W}_{\text{red}}$ are transitive, and by symmetry it suffices to consider $\mathcal{S}_{\text{red}}$.

So let $(i, j), (j, k) \in \mathcal{S}_{\text{red}}$ and assume, for the sake of contradiction, that $(i, k) \notin \mathcal{S}_{\text{red}}$. As $\mathcal{S}$ is transitive and $\mathcal{S}_{\text{red}} \subseteq \mathcal{S}$, we know that $(i, k) \in \mathcal{S}$, and hence i west of k is forced, or i east of k is forced. W.l.o.g. assume that i west of k is forced, that is, k is reachable from i in $\mathcal{W} \setminus \operatorname{sym}(\mathcal{S})$. Let H be a shortest i-k path in $\big(\llbracket n \rrbracket, \mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\big)$, and assume that we have chosen a counterexample i, j, k minimizing $|E(H)|$.

As $(i, k) \in \mathcal{S}$, we have $|E(H)| \geq 2$, and let v be the predecessor of k in H. Then $(v, k) \in \mathcal{W} \setminus \operatorname{sym}(\mathcal{S})$, and in particular $v \neq j$. Moreover, as v is on an i-k-path in $\big(\llbracket n \rrbracket, \mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\big)$, i west of v is forced, and v west of k is forced. Now, we consider the pair (j, v).
If j west of v is forced, then j west of k is forced, contradicting $(j, k) \in \mathcal{S}_{\text{red}}$.
If v west of j is forced, then i west of j is forced, contradicting $(i, j) \in \mathcal{S}_{\text{red}}$. In particular, we get $(j, v) \in \operatorname{sym}(\mathcal{S})$. Moreover, $(v, k) \notin \mathcal{S}$ and $(j, k) \in \mathcal{S}$ imply $(j, v) \notin \overleftarrow{\mathcal{S}}$, and hence $(j, v) \in \mathcal{S}$. As neither west nor east is forced for (j, v), this implies $(j, v) \in \mathcal{S}_{\text{red}}$. But then $(i, j), (j, v) \in \mathcal{S}_{\text{red}}$ and $(i, v) \notin \mathcal{S}_{\text{red}}$ (as i west of v is forced), so i, j, v is a counterexample with smaller $|E(H)|$. □

Note that the reduction of a configuration is not necessarily a configuration again: For example, the configurations depicted in Figure 7.4 have the same reduction $(\mathcal{S}_{\text{red}}, \mathcal{W}_{\text{red}})$. We have $(1, 2), (4, 5) \in \mathcal{S}_{\text{red}}$ and $(4, 2) \notin \mathcal{S}_{\text{red}}$, but $(1, 5) \notin \mathcal{S}_{\text{red}}$, hence $\mathcal{S}_{\text{red}}$ is not an interval order.

We observe that the set of forced relations does not change when reducing:

Lemma 7.30. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder and $(\mathcal{S}_{\text{red}}, \mathcal{W}_{\text{red}})$ its reduction. Then, we have*

$$\begin{aligned}
\operatorname{tr}\Big(\mathcal{S} \setminus \operatorname{sym}(\mathcal{W})\Big) &= \operatorname{tr}\Big(\mathcal{S}_{\text{red}} \setminus \operatorname{sym}(\mathcal{W}_{\text{red}})\Big), \\
\operatorname{tr}\Big(\overleftarrow{\mathcal{S}} \setminus \operatorname{sym}(\mathcal{W})\Big) &= \operatorname{tr}\Big(\overleftarrow{\mathcal{S}_{\text{red}}} \setminus \operatorname{sym}(\mathcal{W}_{\text{red}})\Big), \\
\operatorname{tr}\Big(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\Big) &= \operatorname{tr}\Big(\mathcal{W}_{\text{red}} \setminus \operatorname{sym}(\mathcal{S}_{\text{red}})\Big), \textit{ and} \\
\operatorname{tr}\Big(\overleftarrow{\mathcal{W}} \setminus \operatorname{sym}(\mathcal{S})\Big) &= \operatorname{tr}\Big(\overleftarrow{\mathcal{W}_{\text{red}}} \setminus \operatorname{sym}(\mathcal{S}_{\text{red}})\Big).
\end{aligned}$$

Proof. By symmetry, it suffices to show

$$\operatorname{tr}\big(\mathcal{S} \setminus \operatorname{sym}(\mathcal{W})\big) = \operatorname{tr}\big(\mathcal{S}_{\mathrm{red}} \setminus \operatorname{sym}(\mathcal{W}_{\mathrm{red}})\big).$$

As for each pair at most one relation can be forced in $(\mathcal{S}, \mathcal{W})$, we have $\mathcal{S} \setminus \operatorname{sym}(\mathcal{W}) \subseteq \mathcal{S}_{\mathrm{red}}$. Furthermore, $\mathcal{W}_{\mathrm{red}} \subseteq \mathcal{W}$ implies $\operatorname{sym}(\mathcal{W}_{\mathrm{red}}) \subseteq \operatorname{sym}(\mathcal{W})$, and hence $\mathcal{S} \setminus \operatorname{sym}(\mathcal{W}) \subseteq \mathcal{S}_{\mathrm{red}} \setminus \operatorname{sym}(\mathcal{W}_{\mathrm{red}})$. This implies $\operatorname{tr}(\mathcal{S} \setminus \operatorname{sym}(\mathcal{W})) \subseteq \operatorname{tr}(\mathcal{S}_{\mathrm{red}} \setminus \operatorname{sym}(\mathcal{W}_{\mathrm{red}}))$.

For the other direction, let $(i, j) \in \mathcal{S}_{\mathrm{red}} \setminus \operatorname{sym}(\mathcal{W}_{\mathrm{red}})$. We need to show that j is reachable from i in $(\llbracket n \rrbracket, \mathcal{S} \setminus \operatorname{sym}(\mathcal{W}))$. Clearly $(i, j) \in \mathcal{S}$, and if $(i, j) \notin \operatorname{sym}(\mathcal{W})$, we are done, so assume $(i, j) \in \operatorname{sym}(\mathcal{W})$. Then, $(i, j) \in \operatorname{sym}(\mathcal{W}) \setminus \operatorname{sym}(\mathcal{W}_{\mathrm{red}})$, so south is forced for (i, j) in $(\mathcal{S}, \mathcal{W})$, that is, j is reachable from i in $(\llbracket n \rrbracket, \mathcal{S} \setminus \operatorname{sym}(\mathcal{W}))$. □

Hence, taking the reduction is an idempotent operation:

Corollary 7.31. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder. Then, we have*

$$\operatorname{red}\Big(\operatorname{red}\big((\mathcal{S}, \mathcal{W})\big)\Big) = \operatorname{red}\big((\mathcal{S}, \mathcal{W})\big).$$

Moreover, reducing does not change reachability in G_{SW} and G_{SE}, and hence also preserves the set of representing sequence pairs:

Lemma 7.32. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder on $\llbracket n \rrbracket$ and $(\mathcal{S}_{\mathrm{red}}, \mathcal{W}_{\mathrm{red}})$ its reduction. Furthermore, let G_{SW}, G_{SE}, $G^{\mathrm{red}}_{\mathrm{SW}}$ and $G^{\mathrm{red}}_{\mathrm{SE}}$ be the south-west and south-east digraphs of $(\mathcal{S}, \mathcal{W})$ and $(\mathcal{S}_{\mathrm{red}}, \mathcal{W}_{\mathrm{red}})$, respectively.*
Then, we have $\operatorname{tr}(G_{\mathrm{SW}}) = \operatorname{tr}(G^{\mathrm{red}}_{\mathrm{SW}})$ and $\operatorname{tr}(G_{\mathrm{SE}}) = \operatorname{tr}(G^{\mathrm{red}}_{\mathrm{SE}})$.

Proof. By symmetry, it suffices to show $\operatorname{tr}(G_{\mathrm{SW}}) = \operatorname{tr}(G^{\mathrm{red}}_{\mathrm{SW}})$.
Using Corollary 4.13 and Lemma 7.30, we get:

$$\begin{aligned}
E\Big(\operatorname{tr}(G^{\mathrm{red}}_{\mathrm{SW}})\Big) &= \operatorname{tr}\big(\mathcal{S}_{\mathrm{red}} \setminus \operatorname{sym}(\mathcal{W}_{\mathrm{red}})\big) \cup \operatorname{tr}\big(\mathcal{W}_{\mathrm{red}} \setminus \operatorname{sym}(\mathcal{S}_{\mathrm{red}})\big) \cup (\mathcal{S}_{\mathrm{red}} \cap \mathcal{W}_{\mathrm{red}}) \\
&= \operatorname{tr}\big(\mathcal{S} \setminus \operatorname{sym}(\mathcal{W})\big) \cup \operatorname{tr}\big(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\big) \cup (\mathcal{S}_{\mathrm{red}} \cap \mathcal{W}_{\mathrm{red}}) \\
E\Big(\operatorname{tr}(G_{\mathrm{SW}})\Big) &= \operatorname{tr}\big(\mathcal{S} \setminus \operatorname{sym}(\mathcal{W})\big) \cup \operatorname{tr}\big(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\big) \cup (\mathcal{S} \cap \mathcal{W})
\end{aligned}$$

Clearly, $\mathcal{S}_{\mathrm{red}} \cap \mathcal{W}_{\mathrm{red}} \subseteq \mathcal{S} \cap \mathcal{W}$, and hence it suffices to show that $\mathcal{S} \cap \mathcal{W} \subseteq E\big(\operatorname{tr}(G^{\mathrm{red}}_{\mathrm{SW}})\big)$. But if $(i, j) \in \mathcal{S} \cap \mathcal{W}$, then $(i, j) \notin \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$, so $(i, j) \notin \overleftarrow{\mathcal{S}_{\mathrm{red}}} \cup \overleftarrow{\mathcal{W}_{\mathrm{red}}}$ and hence $(i, j) \in E(G^{\mathrm{red}}_{\mathrm{SW}})$. □

Now, we can characterize SP-equivalence:

Theorem 7.33. *Let $(\mathcal{S}, \mathcal{W})$ and $\big(\mathcal{S}', \mathcal{W}'\big)$ be biorders on $\llbracket n \rrbracket$. Then $(\mathcal{S}, \mathcal{W})$ and $\big(\mathcal{S}', \mathcal{W}'\big)$ are SP-equivalent if and only if $\operatorname{red}\Big((\mathcal{S}, \mathcal{W})\Big) = \operatorname{red}\Big(\big(\mathcal{S}', \mathcal{W}'\big)\Big)$.*

Proof. If the reductions of $(\mathcal{S},\mathcal{W})$ and $(\mathcal{S}',\mathcal{W}')$ are equal, then by Corollary 7.27 and Lemma 7.32 we know that $(\mathcal{S},\mathcal{W})$ and $(\mathcal{S}',\mathcal{W}')$ are SP-equivalent.

So assume that $(\mathcal{S},\mathcal{W})$ and $(\mathcal{S}',\mathcal{W}')$ are SP-equivalent. Then, by Corollary 7.27 we have $\operatorname{tr}(G_{\text{SW}}) = \operatorname{tr}(G'_{\text{SW}})$ and $\operatorname{tr}(G_{\text{SE}}) = \operatorname{tr}(G'_{\text{SE}})$, where G_{SW}, G_{SE}, G'_{SW} and G'_{SE} are the south-west and south-east digraphs of $(\mathcal{S},\mathcal{W})$ and $(\mathcal{S}',\mathcal{W}')$, respectively. Furthermore, let $\operatorname{red}\big((\mathcal{S},\mathcal{W})\big) = (\mathcal{S}_{\text{red}},\mathcal{W}_{\text{red}})$ and $\operatorname{red}\Big(\big(\mathcal{S}',\mathcal{W}'\big)\Big) = (\mathcal{S}'_{\text{red}},\mathcal{W}'_{\text{red}})$. We need to show that $(\mathcal{S}_{\text{red}},\mathcal{W}_{\text{red}}) = (\mathcal{S}'_{\text{red}},\mathcal{W}'_{\text{red}})$. First, by symmetry, it suffices to show $\mathcal{S}_{\text{red}} = \mathcal{S}'_{\text{red}}$. Furthermore, we only need to show $\mathcal{S}_{\text{red}} \subseteq \mathcal{S}'_{\text{red}}$, for the other direction one can exchange $(\mathcal{S},\mathcal{W})$ and $(\mathcal{S}',\mathcal{W}')$. Finally, as $\mathcal{S}_{\text{red}} = (\mathcal{S}_{\text{red}} \setminus \mathcal{W}) \cup (\mathcal{S}_{\text{red}} \setminus \overleftarrow{\mathcal{W}})$, it suffices to prove

$$\mathcal{S}_{\text{red}} \setminus \overleftarrow{\mathcal{W}} \subseteq \mathcal{S}'_{\text{red}},$$

for the other case consider $(\mathcal{S},\overleftarrow{\mathcal{W}})$ and $(\mathcal{S}',\overleftarrow{\mathcal{W}'})$.

Let $(i,j) \in \mathcal{S}_{\text{red}} \setminus \overleftarrow{\mathcal{W}} \subseteq \mathcal{S} \setminus \overleftarrow{\mathcal{W}} \subseteq E(G_{\text{SW}})$ and assume that $(i,j) \notin \mathcal{S}'_{\text{red}}$. As $(i,j) \in E(G_{\text{SW}})$ and $\operatorname{tr}(G_{\text{SW}}) = \operatorname{tr}(G'_{\text{SW}})$, we know that j must be reachable from i in G'_{SW}.

Claim 1. We have $(i,j) \in \operatorname{tr}(\mathcal{W}' \setminus \operatorname{sym}(\mathcal{S}'))$, that is, i west of j is forced in $(\mathcal{S}',\mathcal{W}')$.

If $(i,j) \in \mathcal{S}'$, then $(i,j) \notin \mathcal{S}'_{\text{red}}$ implies that we have $(i,j) \in \operatorname{tr}(\mathcal{W}' \setminus \operatorname{sym}(\mathcal{S}')) \cup \operatorname{tr}(\overleftarrow{\mathcal{W}'} \setminus \operatorname{sym}(\mathcal{S}'))$, and as j is reachable from i in G'_{SW}, we must have $(i,j) \in \operatorname{tr}(\mathcal{W}' \setminus \operatorname{sym}(\mathcal{S}'))$.

If $(i,j) \notin \mathcal{S}'$, Corollary 4.13 applied to G'_{SW} implies $(i,j) \in \operatorname{tr}(\mathcal{W}' \setminus \operatorname{sym}(\mathcal{S}'))$, proving Claim 1.

Now, let H' be an i-j-path in $(\llbracket n \rrbracket, \mathcal{W}' \setminus \operatorname{sym}(\mathcal{S}'))$. The following claim implies that i west of j is forced in $(\mathcal{S},\mathcal{W})$, contradicting $(i,j) \in \mathcal{S}_{\text{red}}$:

Claim 2. Let $(a,b) \in E(H')$. Then $(a,b) \in \operatorname{tr}(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S}))$.

As $(a,b) \in \mathcal{W}' \setminus \operatorname{sym}(\mathcal{S}')$, b is reachable from a in G'_{SW} (and hence also in G_{SW}), and a is reachable from b in both G'_{SE} and G_{SE}. Then, by Corollary 4.13 applied to G_{SW}, we know that one of the following conditions holds:

(i) $(a,b) \in \operatorname{tr}(\mathcal{S} \setminus \operatorname{sym}(\mathcal{W}))$

(ii) $(a,b) \in \mathcal{S} \cap \mathcal{W}$

(iii) $(a,b) \in \operatorname{tr}(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S}))$

Case (i) would imply that b is reachable from a in G_{SE}, a contradiction. In case (ii), we have $(b,a) \in \overleftarrow{\mathcal{S}}$, and as a is reachable from b in G_{SE}, Corollary 4.13 applied to G_{SE} implies $(b,a) \in \operatorname{tr}\Big(\overleftarrow{\mathcal{W}} \setminus \operatorname{sym}(\mathcal{S})\Big)$, and hence $(a,b) \in \operatorname{tr}\Big(\mathcal{W} \setminus \operatorname{sym}(\mathcal{S})\Big)$. In the last case, there is nothing to show, which proves the claim. $\square$

Theorem 7.33 allows us to compute the set of SP-equivalence classes of a set of biorders by simply computing the set of their reductions, and removing duplicates. As we already observed in the case of Figure 7.4, the reduction of a configuration is not necessarily a configuration. This means that directly enumerating *reductions* of tight configurations would require to work with biorders that are not configurations, severely complicating the enumeration algorithm. Hence, we will not consider SP-equivalence within the configuration enumeration algorithm, but instead use Theorem 7.33 to filter the enumerated configurations and only keep a single representative of each SP-equivalence-class.

7.1.5 Normalized Configurations

In this section, we consider a different equivalence relation on biorders, namely the equivalence relation induced by relabeling the elements of the ground set (e.g., the rectangles):

Consider a biorder $(\mathcal{S}, \mathcal{W})$ on $[\![n]\!]$. For each permutation $\pi \in \Pi_n$, we obtain a different biorder $(\pi(\mathcal{S}), \pi(\mathcal{W}))$ (formally defined in Definition 7.35) that has the same structure as $(\mathcal{S}, \mathcal{W})$ by permuting the elements of $[\![n]\!]$ according to π. It is easy to see that

$$(\mathcal{S}, \mathcal{W}) \sim (\mathcal{S}', \mathcal{W}') \;:\Longleftrightarrow\; (\mathcal{S}', \mathcal{W}') = (\pi(\mathcal{S}), \pi(\mathcal{W})) \text{ for some } \pi \in \Pi_n$$

defines an equivalence relation on biorders which preserves tightness (in the case of configurations) and SP-equivalence. This means that in order to enumerate *all* tight configurations, it suffices to enumerate a *representative* of each equivalence class of tight configurations, and then applying all permutations to the found representatives.

Recall that the augmented south-west digraph $G_{\text{sw+}}$ (cf. Definition 5.1) of a biorder $(\mathcal{S}, \mathcal{W})$ has a unique topological order (Proposition 5.3). This fact motivates the following definition:

Definition 7.34. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder and let $G_{\text{sw+}}$ be the augmented south-west digraph of $(\mathcal{S}, \mathcal{W})$.*
We call $(\mathcal{S}, \mathcal{W})$ ***normalized*** *if the unique topological order π of $G_{\text{sw+}}$ is $\mathrm{id}_{[\![n]\!]}$, i.e., we have $1 <_\pi \ldots <_\pi n$.*

Now, we give a formal definition of $(\pi(\mathcal{S}), \pi(\mathcal{W}))$:

Definition 7.35. *Let $\pi \in \Pi_n$ be a permutation and $Q \subseteq {}^2[\![n]\!]$. The set $\pi(Q) \subseteq {}^2[\![n]\!]$ is defined as*

$$\pi(Q) := \left\{ (\pi(i), \pi(j)) : (i, j) \in Q \right\}.$$

We observe that the edges of $G_{\text{sw+}}$ are permuted according to π:

Observation 7.36. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder and $\pi \in \Pi_n$. Furthermore, let $G_{\text{sw+}}$ and $G^\pi_{\text{sw+}}$ be the augmented south-west digraphs of $(\mathcal{S}, \mathcal{W})$ and $(\pi(\mathcal{S}), \pi(\mathcal{W}))$, respectively. Then, we have $E(G^\pi_{\text{sw+}}) = \pi\big(E(G_{\text{sw+}})\big)$.*

Hence, π also permutes the elements in the topological order of $G_{\mathrm{SW}+}$ and we get:

Observation 7.37. *Let $(\mathcal{S}, \mathcal{W})$ be a biorder. Then, there is a unique normalized biorder $(\mathcal{S}', \mathcal{W}')$ with $(\mathcal{S}, \mathcal{W}) \sim (\mathcal{S}', \mathcal{W}')$.*

Thus, from now on we only consider the efficient enumeration of SP-equivalence classes of *normalized* tight configurations.

7.2 Configuration Enumeration

In this section, we describe an algorithm that, given $n \in \mathbb{N}$, computes the set of SP-equivalence classes of normalized tight configurations on $[\![n]\!]$. The largest part of this section will cover the core enumeration algorithm that ignores SP-equivalence and enumerates the set of normalized tight configurations. Then, in Section 7.2.6, we will apply Theorem 7.33 to only keep a single representative of each SP-equivalence class.

The basic idea of the core enumeration algorithm will be to recursively enumerate partial configurations and use pruning rules to cut off enumeration subtrees that cannot lead to normalized tight configurations. In Section 7.2.1, we will describe the basic algorithm and then, in Sections 7.2.2, 7.2.3 and 7.2.4, introduce pruning rules to eliminate non-interval orders, non-normalized configurations and non-tight configurations.

For each new group of pruning rules, we demonstrate its impact by comparing the algorithm with and without these pruning rules, in both cases using all previously introduced pruning rules. Hence, the best obtained results are given at the end of Section 7.2.4. Moreover, Section 7.2.5 covers implementation details of the core enumeration algorithm.

7.2.1 Partial Configurations and Enumeration Algorithm

The central concept of the algorithm is a *partial* configuration, which in addition to $\mathcal{S}$ and $\mathcal{W}$ is equipped with a set $\mathcal{A}$. The set $\mathcal{A}$ consists of all pairs (i, j) for which we already have decided on the relation in $\mathcal{S}$ and $\mathcal{W}$:

Definition 7.38. *Let $n \in \mathbb{N}$. A* ***partial configuration*** *is a triple $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ with $\mathcal{S}, \mathcal{W}, \mathcal{A} \subseteq {}^2[\![n]\!]$, $\mathcal{A} = \overleftarrow{\mathcal{A}} = \mathrm{sym}(\mathcal{A})$ and $\mathcal{S} \cup \mathcal{W} \subseteq \mathcal{A}$.*

We say that a configuration $(\mathcal{S}^*, \mathcal{W}^*)$ is a ***completion*** *of $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ if $\mathcal{S} = \mathcal{S}^* \cap \mathcal{A}$ and $\mathcal{W} = \mathcal{W}^* \cap \mathcal{A}$. Moreover, we refer by $\mathrm{compl}(\mathcal{S}, \mathcal{W}, \mathcal{A})$ to the set of normalized tight completions of $(\mathcal{S}, \mathcal{W}, \mathcal{A})$.*

Finally, we say that a partial configuration $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ is ***invalid*** *if $\mathrm{compl}(\mathcal{S}, \mathcal{W}, \mathcal{A})$ is empty.*

Clearly, the task of computing all normalized tight configurations is equivalent to computing all normalized tight completions of the partial configuration

$(\emptyset, \emptyset, \emptyset)$. Moreover, the only possible completion of a partial configuration $(\mathcal{S}, \mathcal{W}, {}^2[\![n]\!])$ is $(\mathcal{S}, \mathcal{W})$ itself.

These facts suggest a simple recursive algorithm to enumerate all normalized configurations: Start with the trivial partial configuration $(\emptyset, \emptyset, \emptyset)$, and recursively enumerate the set of possible completions of $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ by adding single pairs (i, j) to $\mathcal{A}$ and enumerating all possible relations of i and j in $(\mathcal{S}, \mathcal{W})$.

This algorithm is formally described in Algorithm 7.1 (page 83). The following result immediately implies that it works correctly:

Lemma 7.39. *Let $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ be a partial configuration and $(i, j) \in {}^2[\![n]\!] \setminus \mathcal{A}$ with $i < j$. Furthermore, let*

$$\begin{aligned}\mathcal{C} := \Big\{ &\big(\mathcal{S} + (j,i), \mathcal{W} + (i,j)\ \big),\\ &\big(\mathcal{S}, \qquad\qquad \mathcal{W} + (i,j)\ \big),\\ &\big(\mathcal{S} + (i,j), \mathcal{W} + (i,j)\ \big),\\ &\big(\mathcal{S} + (i,j), \mathcal{W} \qquad\qquad \big),\\ &\big(\mathcal{S} + (i,j), \mathcal{W} + (j,i)\big)\Big\}\end{aligned}$$

and $\mathcal{A}' := \mathcal{A} \cup \{ (i,j), (j,i) \}$.
Then, we have

$$\operatorname{compl}(\mathcal{S}, \mathcal{W}, \mathcal{A}) = \bigcup_{(\mathcal{S}', \mathcal{W}') \in \mathcal{C}} \operatorname{compl}(\mathcal{S}', \mathcal{W}', \mathcal{A}').$$

Proof. For the first direction, let $(\mathcal{S}^*, \mathcal{W}^*) \in \operatorname{compl}(\mathcal{S}, \mathcal{W}, \mathcal{A})$ be a normalized tight completion of $(\mathcal{S}, \mathcal{W}, \mathcal{A})$. We need to show that $(\mathcal{S}^* \cap \mathcal{A}', \mathcal{W}^* \cap \mathcal{A}') \in \mathcal{C}$. As $(\mathcal{S}^*, \mathcal{W}^*)$ is a biorder, we know that $(i,j) \in \operatorname{sym}(\mathcal{S}^*) \cup \operatorname{sym}(\mathcal{W}^*)$. There are eight possible cases, five of which are covered by $\mathcal{C}$. The three remaining cases are

- $(i,j) \in \overleftarrow{\mathcal{S}^*} \setminus \operatorname{sym}(\mathcal{W}^*)$,
- $(i,j) \in \overleftarrow{\mathcal{S}^*} \cap \overleftarrow{\mathcal{W}^*}$,
- $(i,j) \in \overleftarrow{\mathcal{W}^*} \setminus \operatorname{sym}(\mathcal{S}^*)$.

However, in all of these three cases we must have an edge $(j,i) \in E(G_{\mathrm{sw+}})$ in the augmented south-west digraph $G_{\mathrm{sw+}}$ of $(\mathcal{S}^*, \mathcal{W}^*)$ (cf. Definitions 4.11 and 5.1), which together with $i < j$ contradicts that $(\mathcal{S}^*, \mathcal{W}^*)$ is normalized.

For the other direction, let $(\mathcal{S}', \mathcal{W}') \in \mathcal{C}$ and $(\mathcal{S}^*, \mathcal{W}^*)$ be a normalized tight completion of $(\mathcal{S}', \mathcal{W}', \mathcal{A})$. Then

$$\mathcal{S}^* \cap \mathcal{A} = \big(\mathcal{S}^* \cap \mathcal{A}'\big) \setminus \{ (i,j), (j,i) \} = \mathcal{S}' \setminus \{ (i,j), (j,i) \} = \mathcal{S}$$

and

$$\mathcal{W}^* \cap \mathcal{A} = \big(\mathcal{W}^* \cap \mathcal{A}'\big) \setminus \{\,(i,j),(j,i)\,\} = \mathcal{W}' \setminus \{\,(i,j),(j,i)\,\} = \mathcal{W},$$

so $(\mathcal{S}^*, \mathcal{W}^*)$ is a normalized tight completion of $(\mathcal{S}, \mathcal{W})$. □

Corollary 7.40. *Algorithm 7.1 works correctly.*

Note that the order in which we process pairs (i, j) is irrelevant for the correctness of the algorithm, but will be important later on for some pruning rules. Moreover, as the proof of Lemma 7.39 does not exploit tightness, we can easily modify Algorithm 7.1 to enumerate all normalized (including non-tight) configurations by simply not testing for tightness and not using any pruning rules that exploit tightness.

The program was implemented in the C++17 programming language and compiled using `clang-7.0.0` with the `-O3` compiler flag. All results were obtained on a machine with two `AMD EPYC 7601` 32-core processors and 512 GB of main memory running `CentOS Linux 7.6`, using 64 threads.

7.2.2 Consistency Pruning

The simplest pruning rule exploits that in a configuration $(\mathcal{S}, \mathcal{W})$, the sets $\mathcal{S}$ and $\mathcal{W}$ need to be interval orders. Analogously to Definition 7.38, we define:

Definition 7.41. *Let $n \in \mathbb{N}$. A* ***partial interval order*** *is a pair $(Q, \mathcal{A})$ with $Q, \mathcal{A} \subseteq {}^2[\![n]\!]$, $\mathcal{A} = \overleftarrow{\mathcal{A}}$ and $Q \subseteq \mathcal{A}$.*

We say that an interval order Q^* is a ***completion*** *of $(Q, \mathcal{A})$ if $Q = Q^* \cap \mathcal{A}$. Moreover, we refer by $\operatorname{compl}(Q, \mathcal{A})$ to the set of completions of $(Q, \mathcal{A})$.*

Finally, we call a partial interval order $(Q, \mathcal{A})$ ***invalid*** *if $\operatorname{compl}(Q, \mathcal{A})$ is empty, and* ***valid*** *otherwise.*

Not surprisingly, we will detect whether a partial interval order is valid based on whether an appropriately chosen constraint graph is acyclic:

Definition 7.42. *Let $(Q, \mathcal{A})$ be a partial interval order and let (G_Q, w_Q) be the constraint graph of Q (cf. Definition 7.9). Furthermore, let*

$$\begin{aligned}
E_1' &:= \Big\{\, (v_i^{min}, v_i^{max}) : i \in [\![n]\!] \,\Big\}, \\
E_2' &:= \Big\{\, (v_j^{min}, v_i^{max}) : (i,j) \in (\mathcal{A} \setminus Q) \,\Big\}, \\
E_3' &:= \Big\{\, (v_i^{max}, v_j^{min}) : (i,j) \in (\mathcal{A} \cap Q) \,\Big\}.
\end{aligned}$$

The ***partial constraint graph*** *$(G_{Q,\mathcal{A}}, w_{Q,\mathcal{A}})$ of $(Q, \mathcal{A})$ is the subgraph of G_Q on the same vertex set, with edge set*

$$E(G_{Q,\mathcal{A}}) := E_1' \cup E_2' \cup E_3' \subseteq E(G_Q),$$

and the same edge weights on these edges.

Algorithm 7.1: Normalized tight configuration enumeration

Input: Integer $n \in \mathbb{N}$.

Output: Set of all normalized tight configurations on $[\![n]\!]$.

return enumerate_recursively$(\emptyset, \emptyset, \emptyset, 1, 2, n)$

`//` Returns all normalized tight completions of $(\mathcal{S}, \mathcal{W}, \mathcal{A})$.
`//` (i, j) is the next pair to be assigned.

procedure enumerate_recursively$(\mathcal{S}, \mathcal{W}, \mathcal{A}, i, j, n)$

  `//` Prune invalid partial configurations.
  if *we can prove that* $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ *is invalid* **then**
    return $\emptyset$
  `//` Check if all pairs are assigned.
  if $\mathcal{A} = {}^2[\![n]\!]$ **then**
    if $(\mathcal{S}, \mathcal{W})$ *is normalized tight configuration* **then**
      return $\{(\mathcal{S}, \mathcal{W})\}$
    else
      return $\emptyset$
  `//` Enumerate relations of (i, j) in $\mathcal{S}$ and $\mathcal{W}$ and recurse.
  $\mathcal{C} \leftarrow \{$
    $(\mathcal{S} + (j, i),\ \mathcal{W} + (i, j))$, `//` i north and west of j
    $(\mathcal{S},\ \mathcal{W} + (i, j))$, `//` i only west of j
    $(\mathcal{S} + (i, j),\ \mathcal{W} + (i, j))$, `//` i south and west of j
    $(\mathcal{S} + (i, j),\ \mathcal{W})$, `//` i only south of j
    $(\mathcal{S} + (i, j),\ \mathcal{W} + (j, i))$ `//` i south and east of j
  $\}$
  $\mathcal{A}' \leftarrow \mathcal{A} \cup \{(i, j), (j, i)\}$
  $(i', j') \leftarrow$ next_pair(i, j)
  return $\bigcup_{(\mathcal{S}', \mathcal{W}') \in \mathcal{C}}$ enumerate_recursively$(\mathcal{S}', \mathcal{W}', \mathcal{A}', i', j', n)$

`//` Returns pair to be processed after (i, j).
`//` Order is increasing in j and decreasing in i:
`//` $(1, 2); (2, 3), (1, 3); (3, 4), (2, 4), (1, 4); \ldots$

procedure next_pair(i, j)
  if $i = 1$ **then**
    return $(j, j + 1)$
  else
    return $(i - 1, j)$

Lemma 7.43. *Let $(Q, \mathcal{A})$ be a partial interval order on $[\![n]\!]$. Then $(Q, \mathcal{A})$ is valid if and only if $G_{Q,\mathcal{A}}$ is acyclic.*

Proof. For the first direction, assume that $(Q, \mathcal{A})$ is valid and let Q^* be a completion of $(Q, \mathcal{A})$. Then $G_{Q,\mathcal{A}}$ is a subgraph of G_{Q^*}. By Lemmata 7.10 and 7.11, we know that G_{Q^*} is acyclic, so $G_{Q,\mathcal{A}}$ must be acyclic.

For the other direction, assume that $G_{Q,\mathcal{A}}$ is acyclic. As (G_Q, w_Q) does not contain cycles of nonnegative cost (Lemma 7.11), its subgraph $(G_{Q,\mathcal{A}}, w_{Q,\mathcal{A}})$ also does not contain cycles of nonnegative cost and hence is conservative. Thus, there is a feasible potential λ of $(G_{Q,\mathcal{A}}, w_{Q,\mathcal{A}})$. As $G_{Q,\mathcal{A}}$ still contains all edges of the form (v_i^{min}, v_i^{max}), we have $\lambda(v_i^{min}) - \lambda(v_i^{max}) - 1 \geq 0$ for all $i \in [\![n]\!]$, so $I_\lambda = \big(minc, maxc\big)$ given by

$$minc(i) := -\lambda(v_i^{min}), \qquad maxc(i) := -\lambda(v_i^{max}),$$

is an interval placement. It is easy to verify that its interval order Q_{I_λ} is a completion of $(Q, \mathcal{A})$. □

In the algorithm, we will maintain the partial constraint graphs of $(\mathcal{S}, \mathcal{A})$ and $(\mathcal{W}, \mathcal{A})$. Moreover, for both graphs we will maintain an all-pairs reachability table that tells us for all pairs of vertices u, v whether there is a path from u to v in $G_{\mathcal{S},\mathcal{A}}$ or $G_{\mathcal{W},\mathcal{A}}$, respectively. Hence, we can determine whether a new edge induces a cycle in constant time, and discard all partial assignments that would lead to such a cycle. In Section 7.2.5, we explain how to update these reachability tables very efficiently when adding new edges.

We will refer to pruning based on Lemma 7.43 by **consistency pruning**, its impact is given in Table 7.1: As expected, the number of enumeration nodes is significantly reduced, and the set of normalized tight configurations can be enumerated up to $n = 8$. We observe that for $n \leq 4$, the number of normalized tight configurations equals the number of biplane permutations (cf. Table 3.3 on page 32), and hence the lower bound introduced in Chapter 6 is tight for $n \leq 4$, complementing Proposition 6.13.

7.2.3 Normalization Pruning

Algorithm 7.1 only allows five of the eight possibilities for the spatial relation between rectangles $i < j$, as the other three assignments "only north", "north and east", "only east" cannot lead to a normalized configuration. However, this restriction is not sufficient to guarantee normalization: For example, the algorithm would consider the configuration on two rectangles where 1 is both south and east of 2, but as the augmented south-west digraph contains an edge $(2, 1)$ in this case, this configuration is not normalized.

The following result gives an alternative characterization of normalized configurations in the special case that $(i, j) \in \mathcal{S} \cup \mathcal{W}$ for all $i < j$, which is guaranteed by Algorithm 7.1.

n	Pruning	Nodes	Time [s]	Configurations
1	no pruning	1	0.00	1
	consistency	1	0.00	
2	no pruning	6	0.00	2
	consistency	6	0.00	
3	no pruning	156	0.00	6
	consistency	100	0.00	
4	no pruning	19 531	0.00	22
	consistency	3 389	0.00	
5	no pruning	12 207 031	0.08	98
	consistency	202 033	0.01	
6	no pruning	$3.8 \cdot 10^{10}$	116.08	516
	consistency	$1.9 \cdot 10^{7}$	0.31	
7	no pruning	–	–	3 140
	consistency	$2.7 \cdot 10^{9}$	26.02	
8	no pruning	–	–	21 684
	consistency	$5.4 \cdot 10^{11}$	5465.41	

Table 7.1: Impact of consistency pruning. Column 3 gives the number of nodes in the enumeration tree. Column 4 gives the running time of the algorithm in seconds. The last column lists the number of normalized tight configurations for the given n.

Lemma 7.44. *Let $(\mathcal{S}, \mathcal{W})$ be a configuration on $[\![n]\!]$ with $(i, j) \in \mathcal{S} \cup \mathcal{W}$ for all $1 \leq i < j \leq n$.*

Then $(\mathcal{S}, \mathcal{W})$ is normalized if and only if there is no $i \in [\![n-1]\!]$ with $(i, i+1) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$.

Proof. Let G_{sw} and $G_{\mathrm{sw+}}$ be the (augmented) south-west digraphs of $(\mathcal{S}, \mathcal{W})$. First, we observe that for all $i < j$ the condition $(i, j) \in \mathcal{S} \cup \mathcal{W}$ implies $(j, i) \in \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$, so (j, i) is not an edge of G_{sw}. In particular, every vertex k can only reach vertices l with $k < l$.

For the first direction, assume there is $i \in [\![n-1]\!]$ with $(i, i+1) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$. Then $(i, i+1) \notin E(G_{\mathrm{sw}})$, implying that $i+1$ is not reachable from i in G_{sw}: Otherwise, every path H from i to $i+1$ needs to contain inner vertices, and every such inner vertex $k \in [\![n]\!]$ needs to satisfy $i < k < i+1$, a contradiction. Hence, we have $(i+1, i) \in E(G_{\mathrm{sw+}})$ and $(\mathcal{S}, \mathcal{W})$ is not normalized.

For the other direction, assume that $(i, i+1) \notin \mathcal{S} \cap \overleftarrow{\mathcal{W}}$ for all $i \in [\![n-1]\!]$. We show that $(i, i+1) \in E(G_{\mathrm{sw+}})$ for all $i \in [\![n-1]\!]$, which implies that $(\mathcal{S}, \mathcal{W})$ is normalized. So let $i \in [\![n-1]\!]$.

n	Pruning	Nodes	Leaves	Time [s]	Configurations
1	consistency	1	1	0.00	1
	normalization	1	1	0.01	
2	consistency	6	5	0.00	2
	normalization	5	4	0.01	
3	consistency	100	69	0.00	6
	normalization	61	40	0.01	
4	consistency	3 389	1 997	0.00	22
	normalization	1 393	772	0.01	
5	consistency	202 033	103 507	0.01	98
	normalization	52 009	24 840	0.01	
6	consistency	19 200 156	8 660 521	0.31	516
	normalization	2 901 007	1 211 968	0.07	
7	consistency	$2.7 \cdot 10^9$	$1.1 \cdot 10^9$	26.02	3 140
	normalization	$2.3 \cdot 10^8$	$8.4 \cdot 10^7$	2.59	
8	consistency	$5.4 \cdot 10^{11}$	$1.9 \cdot 10^{11}$	5465.41	21 684
	normalization	$2.4 \cdot 10^{10}$	$7.9 \cdot 10^9$	274.63	

Table 7.2: Impact of normalization pruning. Columns 3 and 4 give the number of nodes and leaves in the enumeration tree, respectively. Column 5 gives the running time of the algorithm in seconds. The last column lists the number of normalized tight configurations for the given n.

If $(i, i+1) \notin \overleftarrow{\mathcal{S}}$, then $(i, i+1) \in (\mathcal{S} \cup \mathcal{W}) \setminus (\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}) = E(G_{\mathrm{sw}}) \subseteq E(G_{\mathrm{sw+}})$. If $(i, i+1) \in \overleftarrow{\mathcal{S}}$, then $(i, i+1) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$. By the observation, we know that i is not reachable from $i+1$ in G_{sw}, so $(i, i+1) \in E(G_{\mathrm{sw+}})$. □

This means that we can guarantee to only enumerate normalized configurations by simply excluding $(i, i+1) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$. For complete configurations we can then also skip the test for normalization.

We call this pruning rule (always used together with consistency pruning) **normalization** pruning. Table 7.2 shows the impact of normalization pruning. While the number of nodes in the enumeration tree is reduced compared to consistency pruning, numbers are still in a similar order of magnitude.

Before we proceed to apply pruning rules exploiting tightness, we can now also enumerate all normalized (not necessarily tight) configurations by simply not testing for tightness in line 6 of Algorithm 7.1. As expected, the number of normalized configurations on $[\![n]\!]$ equals the number of enumeration leaves for the normalization pruning rule as given in Table 7.2, as the algorithm does not enumerate partial configurations that have no normalized completion.

We see that the number of normalized *tight* configurations on $[\![n]\!]$ is *much* smaller than the number of normalized configurations on $[\![n]\!]$, stressing the importance of tightness for the computation of CR_n. For example, for $n = 7$, we would need to consider $7! \cdot 8.4 \cdot 10^7 > 4 \cdot 10^{11}$ configurations, which would not even fit in memory, while there are only $7! \cdot 3140 < 2 \cdot 10^7$ tight configurations.

Moreover, it is evident that in order to enumerate all normalized *tight* configurations for larger n, we need to exploit tightness already during the enumeration of partial configurations, that is, we need pruning rules based on tightness.

7.2.4 Tightness Pruning

All tightness pruning rules we will describe are based on Lemmata 7.20 and 7.23:

Lemma 7.20. *Let $(\mathcal{S}, \mathcal{W})$ be a configuration.*
Then $(\mathcal{S}, \mathcal{W})$ is tight if and only if

(i) $\mathcal{S} - (i, j)$ is not an interval order for all $(i, j) \in \mathcal{S} \cap \operatorname{sym}(\mathcal{W})$, and

(ii) $\mathcal{W} - (i, j)$ is not an interval order for all $(i, j) \in \mathcal{W} \cap \operatorname{sym}(\mathcal{S})$.

Lemma 7.23. *Let $(\mathcal{S}, \mathcal{W})$ be a tight configuration on $[\![n]\!]$ and $(i, l) \in \mathcal{S} \cap \mathcal{W}$. Then, there are rectangles $j, k \in [\![n]\!]$ with*

(i) $(i, j) \in \mathcal{W} \setminus \operatorname{sym}(\mathcal{S})$,

(ii) $(k, l) \in \mathcal{W} \setminus \operatorname{sym}(\mathcal{S})$.

Moreover, if $j \neq k$, then

(iii) $(j, k) \in (\mathcal{S} \cup \mathcal{W}) \setminus \left(\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}\right)$.

In particular, $v_i^{max}, v_j^{min}, v_k^{max}, v_l^{min}$ are the vertices of a path of cost -1 in $G_{\mathcal{W}}$.

In Lemma 7.23, one can of course exchange the role of $\mathcal{S}$ and $\mathcal{W}$ to obtain a path in $G_{\mathcal{S}}$. Moreover, by replacing $\mathcal{W}$ with $\overleftarrow{\mathcal{W}}$, we get analogous statements for $(i, l) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$.

The pruning rules to be described exploit the order in which we assign pairs (i, j) in Algorithm 7.1. First, we formalize this order:

Definition 7.45. *Let $1 \leq i < l \leq n$. We say that a partial configuration $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ is **(i, l)-ready** if*

$$\mathcal{A} = {}^2[\![l-1]\!] \cup \left\{ (j, k) \in {}^2[\![l]\!] : i < j \text{ and } i < k \right\}.$$

In particular, if $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ is (i, l)-ready, then $(i, l) \notin \mathcal{A}$.

Clearly, whenever Algorithm 7.1 considers a pair (i, j), the current partial configuration is (i, j)-ready.

Weak Tightness Pruning

First, we consider the case $(i,l) \in \mathcal{S} \cap \mathcal{W}$, which can be dealt with much easier than the case $(i,l) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$:

Lemma 7.46. *Let $1 \leq i < l \leq n$ and $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ be an (i,l)-ready partial configuration. Furthermore, let $(\mathcal{S}^*, \mathcal{W}^*)$ be a normalized tight completion of $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ with $(i,l) \in \mathcal{S}^* \cap \mathcal{W}^*$.*
Then v_l^{min} is reachable from v_i^{max} in both $G_{\mathcal{W},\mathcal{A}}$ and $G_{\mathcal{S},\mathcal{A}}$.

Proof. Let G_{sw} be the south-west digraph of $(\mathcal{S}^*, \mathcal{W}^*)$.
By Lemma 7.23, there are $j,k \in [\![n]\!]$ with

- $(i,j) \in \mathcal{W}^* \setminus \mathrm{sym}(\mathcal{S}^*)$,
- $(k,l) \in \mathcal{W}^* \setminus \mathrm{sym}(\mathcal{S}^*)$, and additionally
- $(j,k) \in (\mathcal{S}^* \cup \mathcal{W}^*) \setminus \left(\overleftarrow{\mathcal{S}}^* \cup \overleftarrow{\mathcal{W}}^*\right)$ if $j \neq k$.

In particular, $v_i^{max}, v_j^{min}, v_k^{max}, v_l^{min}$ are the vertices of a path H in $G_{\mathcal{W}^*}$. We show that H is also a path in $G_{\mathcal{W},\mathcal{A}}$:

The conditions on j and k above imply $(i,j),(k,l) \in E(G_{\mathrm{sw}})$, and $(j,k) \in E(G_{\mathrm{sw}})$ if $j \neq k$. As $(\mathcal{S}^*, \mathcal{W}^*)$ is normalized, we get $i < j \leq k < l$. Moreover, $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ is (i,l)-ready, so $(i,j),(k,l) \in \mathcal{A}$ and $(j,k) \in \mathcal{A}$ if $j \neq k$. Thus, $\mathcal{S} = \mathcal{S}^* \cap \mathcal{A}$ and $\mathcal{W} = \mathcal{W}^* \cap \mathcal{A}$ imply

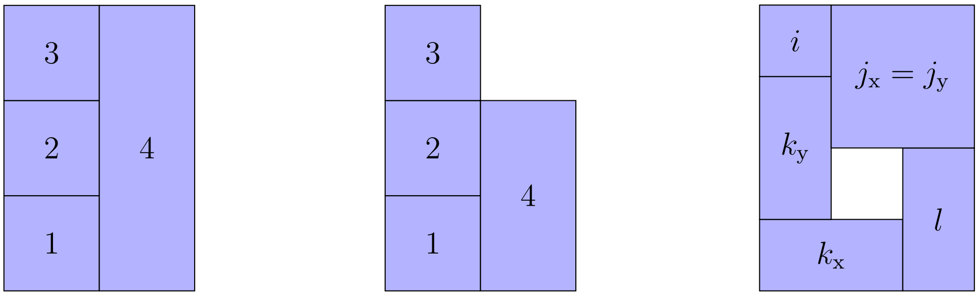
- $(i,j) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$,
- $(k,l) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$, and additionally
- $(j,k) \in (\mathcal{S} \cup \mathcal{W}) \setminus \left(\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}\right)$ if $j \neq k$.

It follows that H is a path in $G_{\mathcal{W},\mathcal{A}}$.

The statement that v_l^{min} is reachable from v_i^{max} in $G_{\mathcal{S},\mathcal{A}}$ follows by symmetry: Although the definition of normalizedness is not symmetric in $\mathcal{S}$ and $\mathcal{W}$ (as $G_{\mathrm{sw}+}$ is not symmetric in $\mathcal{S}$ and $\mathcal{W}$), we only used that $\mathrm{id}_{[\![n]\!]}$ is a topological order of G_{sw} (not $G_{\mathrm{sw}+}$), which is symmetric in $\mathcal{S}$ and $\mathcal{W}$. □

Lemma 7.46 implies that we only need to consider the case $(i,l) \in \mathcal{S} \cap \mathcal{W}$ if it is already implied. That is: Either v_l^{min} is reachable from v_i^{max} both in $G_{\mathcal{W},\mathcal{A}}$ and $G_{\mathcal{S},\mathcal{A}}$, in which case by consistency pruning $(i,l) \in \mathcal{S} \cap \mathcal{W}$ is the only assignment of (i,l) that we need to consider, or $(i,l) \in \mathcal{S} \cap \mathcal{W}$ cannot lead to a normalized tight configuration.

The cases $(i,l) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$ and $(i,l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$ are more difficult: We can have an (i,l)-ready partial configuration $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ that has a normalized tight completion $(\mathcal{S}^*, \mathcal{W}^*)$ with $(i,l) \in \overleftarrow{\mathcal{S}}^* \cap \mathcal{W}^*$, but v_l^{min} is not reachable from



(a) A placement dominating the placement depicted in the center.

(b) A non-tight placement P'.

(c) A tight placement P with labels according to Lemma 7.47.

Figure 7.5: If a partial configuration $(\mathcal{S}, \mathcal{W}, {}^2[\![k]\!])$ has a tight completion, then $(\mathcal{S}, \mathcal{W})$ is not necessarily tight: Here, $(\mathcal{S}_{P'}, \mathcal{W}_{P'})$ is not tight (as can be seen on the left), but $(\mathcal{S}_P, \mathcal{W}_P)$ is a tight completion of $(\mathcal{S}_{P'}, \mathcal{W}_{P'}, {}^2[\![4]\!])$, where P' is the placement in the center, P is the placement on the right, and $k_\mathrm{x} = 1$, $k_\mathrm{y} = 2$, $i = 3$, $l = 4$, and $j_\mathrm{x} = j_\mathrm{y} = 5$.

v_i^{max} in $G_{\mathcal{W},\mathcal{A}}$, and v_i^{min} is not reachable from v_l^{max} in $G_{\mathcal{S},\mathcal{A}}$. The reason is that Lemma 7.23 applied to $(\mathcal{S}^*, \mathcal{W}^*)$ still guarantees the existence of j and k, but these no longer necessarily precede i or l in the topological order of $G_{\mathrm{sw}+}$. For example, consider the placement P' as depicted in Figure 7.5(b): Set $\mathcal{A} := {}^2[\![4]\!] \setminus \{\,(3,4),(4,3)\,\}$. Then $(\mathcal{S}_{P'} \cap \mathcal{A}, \mathcal{W}_{P'} \cap \mathcal{A}, \mathcal{A})$ satisfies the conditions above.

In the following, instead of dealing with the two different cases $(i,l) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$ and $(i,l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$ where $i < l$, we instead consider only $(i,l) \in \mathcal{S} \cap \overleftarrow{\mathcal{W}}$, no longer necessarily requiring $i < l$.

Lemma 7.47. *Let $(\mathcal{S}, \mathcal{W})$ be a normalized tight configuration and $(i,l) \in {}^2[\![n]\!]$ with $(i,l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$.*
Then, there are $k_\mathrm{x} < l$ and $i < j_\mathrm{x}$ with

(i) $(i, j_\mathrm{x}) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$.

(ii) $(k_\mathrm{x}, l) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$.

(iii) $(j_\mathrm{x}, k_\mathrm{x}) \in (\overleftarrow{\mathcal{S}} \cup \mathcal{W}) \setminus (\mathcal{S} \cup \overleftarrow{\mathcal{W}})$ *if* $j_\mathrm{x} \neq k_\mathrm{x}$.

Moreover, there are $k_\mathrm{y} < i$ and $l < j_\mathrm{y}$ with

(iv) $(l, j_\mathrm{y}) \in \mathcal{S} \setminus \mathrm{sym}(\mathcal{W})$

(v) $(k_\mathrm{y}, i) \in \mathcal{S} \setminus \mathrm{sym}(\mathcal{W})$

(vi) $(j_{\mathrm{y}}, k_{\mathrm{y}}) \in (\mathcal{S} \cup \overleftarrow{\mathcal{W}}) \setminus (\overleftarrow{\mathcal{S}} \cup \mathcal{W})$ *if* $j_{\mathrm{y}} \neq k_{\mathrm{y}}$.

Finally, we have

(vii) $k_{\mathrm{x}} \neq k_{\mathrm{y}}$ *or* $j_{\mathrm{x}} \neq j_{\mathrm{y}}$.

In particular,
$v_i^{max}, v_{j_{\mathrm{x}}}^{min}, v_{k_{\mathrm{x}}}^{max}, v_l^{min}$ *are the vertices of a path in* $G_{\mathcal{W}}$*, and*
$v_l^{max}, v_{j_{\mathrm{y}}}^{min}, v_{k_{\mathrm{y}}}^{max}, v_i^{min}$ *are the vertices of a path in* $G_{\mathcal{S}}$.

Proof. The existence of $j_{\mathrm{x}}, k_{\mathrm{x}}, j_{\mathrm{y}}, k_{\mathrm{y}}$ satisfying (i) to (vi) is clear by Lemma 7.23. Moreover, as $(\mathcal{S}, \mathcal{W})$ is normalized, (i) to (vi) imply $k_{\mathrm{x}} < l$, $i < j_{\mathrm{x}}$, $k_{\mathrm{y}} < i$ and $l < j_{\mathrm{y}}$ as required.

It remains to show that $k_{\mathrm{x}} \neq k_{\mathrm{y}}$ or $j_{\mathrm{x}} \neq j_{\mathrm{y}}$, so assume for the sake of contradiction that $k_{\mathrm{x}} = k_{\mathrm{y}} = k$ and $j_{\mathrm{x}} = j_{\mathrm{y}} = j$. If $j \neq k$, then (iii) and (vi) clearly contradict each other. But if $j_{\mathrm{x}} = j_{\mathrm{y}} = j = k = k_{\mathrm{x}} = k_{\mathrm{y}}$, then $j_{\mathrm{x}} > i > k_{\mathrm{y}}$ leads to a contradiction. □

Note that $k_{\mathrm{x}} = k_{\mathrm{y}}$ or $j_{\mathrm{x}} = j_{\mathrm{y}}$ can indeed occur as seen in Figure 7.5(c).

Definition 7.48. *Let* $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ *be a partial configuration on* $[\![n]\!]$.
We denote by

$$K_{\mathrm{x}}^{(i,l)} := \Big\{ k_{\mathrm{x}} \in [\![n]\!] : (k_{\mathrm{x}}, l) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S}) \text{ and } v_i^{max} \text{ is not reachable from } v_{k_{\mathrm{x}}}^{max} \text{ in } G_{\mathcal{W},\mathcal{A}} \Big\}$$

the set of ***k*$_{\mathrm{x}}$*-candidates*** *of* (i, l) *in* $(\mathcal{S}, \mathcal{W}, \mathcal{A})$.
Moreover, we denote by

$$K_{\mathrm{y}}^{(i,l)} := \Big\{ k_{\mathrm{y}} \in [\![n]\!] : (k_{\mathrm{y}}, i) \in \mathcal{S} \setminus \mathrm{sym}(\mathcal{W}) \text{ and } v_l^{max} \text{ is not reachable from } v_{k_{\mathrm{y}}}^{max} \text{ in } G_{\mathcal{S},\mathcal{A}} \Big\}$$

the set of ***k*$_{\mathrm{y}}$*-candidates*** *of* (i, l) *in* $(\mathcal{S}, \mathcal{W}, \mathcal{A})$.

Lemma 7.49. *Let* $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ *be a partial configuration and let* $(i, l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$. *Then, we have:*

(i) If ${}^2[\![l]\!] \subseteq \mathcal{A}$ *and the set of* k_{x}*-candidates* $K_{\mathrm{x}}^{(i,l)}$ *is empty, then* $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ *is invalid.*

(ii) If ${}^2[\![i]\!] \subseteq \mathcal{A}$ *and the set of* k_{y}*-candidates* $K_{\mathrm{y}}^{(i,l)}$ *is empty, then* $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ *is invalid.*

Proof. We only show (i), the other case (ii) is shown similarly.

Assume ${}^2[\![l]\!] \subseteq \mathcal{A}$. It suffices to prove that if $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ has a normalized tight completion $(\mathcal{S}^*, \mathcal{W}^*)$, then $K_{\mathrm{x}}^{(i,l)}$ is not empty. So let $(\mathcal{S}^*, \mathcal{W}^*)$ be a normalized tight completion of $(\mathcal{S}, \mathcal{W}, \mathcal{A})$. By Lemma 7.47, there are $k_{\mathrm{x}} < l$ and $i < j_{\mathrm{x}}$ with $(k_{\mathrm{x}}, l) \in \mathcal{W}^* \setminus \mathrm{sym}(\mathcal{S}^*)$ such that $v_i^{max}, v_{j_{\mathrm{x}}}^{min}, v_{k_{\mathrm{x}}}^{max}, v_l^{min}$ are the vertices of a path in $G_{\mathcal{W}^*}$. Note that $k_{\mathrm{x}} < l$ implies $(k_{\mathrm{x}}, l) \in {}^2[\![l]\!] \subseteq \mathcal{A}$. Now, using $\mathcal{W} = \mathcal{W}^* \cap \mathcal{A}$ and $\mathcal{S} = \mathcal{S}^* \cap \mathcal{A}$, we have $(k_{\mathrm{x}}, l) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$. Finally, since $G_{\mathcal{W}^*}$ is acyclic and $v_{k_{\mathrm{x}}}^{max}$ is reachable from v_i^{max} in $G_{\mathcal{W}^*}$, we know that v_i^{max} is not reachable from $v_{k_{\mathrm{x}}}^{max}$ in $G_{\mathcal{W}^*}$. But $G_{\mathcal{W},\mathcal{A}}$ is a subgraph of $G_{\mathcal{W}^*}$, and hence v_i^{max} is not reachable from $v_{k_{\mathrm{x}}}^{max}$ in $G_{\mathcal{W},\mathcal{A}}$, which implies $k_{\mathrm{x}} \in K_{\mathrm{x}}^{(i,l)}$. □

Note that due to the order in which we process pairs in Algorithm 7.1, we always have ${}^2[\![\min\{i, l\}]\!] \subseteq \mathcal{A} \subseteq {}^2[\![\max\{i, l\}]\!]$ when assigning the relation between i and l. Hence, we can directly apply the test above to $\min\{i, l\}$ and postpone the test for $\max\{i, l\}$ until ${}^2[\![\max\{i, l\}]\!] \subseteq \mathcal{A}$.

Moreover, Lemma 7.47 implies that in a normalized tight placement $(\mathcal{S}, \mathcal{W})$ with $(i, l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$, we must have $i < n$ and $l < n$, as $i < j_{\mathrm{x}}$ and $l < j_{\mathrm{y}}$. Hence, we can forbid $(i, l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$ whenever $n \in \{i, l\}$.

We refer to the tightness pruning rules described so far (when used together with normalization pruning) as **weak tightness** pruning, results are given in Table 7.3. Clearly, the number of enumerated partial configurations is drastically reduced, and now is in a similar order of magnitude as the number of normalized tight configurations.

Strong Tightness Pruning

Again, consider a normalized tight configuration $(\mathcal{S}, \mathcal{W})$ with $(i, l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$. We will now introduce pruning rules that exploit the existence of $i < j_{\mathrm{x}}$ and $l < j_{\mathrm{y}}$, apart from the trivial implication $i < n$ and $l < n$.

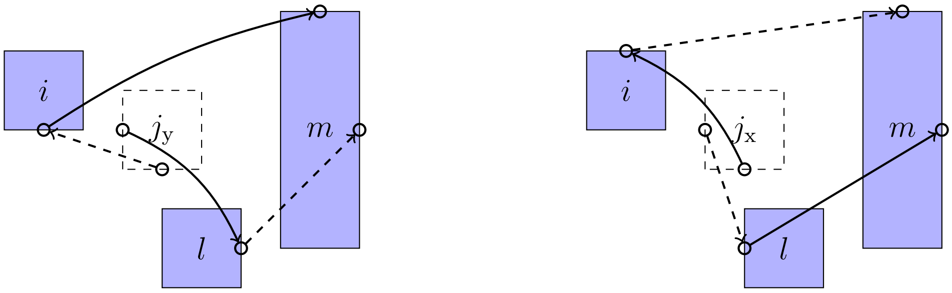
The following simple observation yields a sufficient condition (Lemma 7.52) to prove that a partial configuration is invalid:

Lemma 7.50. *Let $(\mathcal{S}, \mathcal{W})$ be a normalized tight configuration on $[\![n]\!]$. Assume that there are $i, l, j_{\mathrm{x}}, j_{\mathrm{y}} \in [\![n]\!]$ with*

- $(i, l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$,
- $(i, j_{\mathrm{x}}) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S})$,
- *v_l^{min} is reachable from $v_{j_{\mathrm{x}}}^{min}$ in $G_{\mathcal{W}}$,*
- *$(l, j_{\mathrm{y}}) \in \mathcal{S} \setminus \mathrm{sym}(\mathcal{W})$, and*
- *v_i^{min} is reachable from $v_{j_{\mathrm{y}}}^{min}$ in $G_{\mathcal{S}}$.*

n	Pruning	Nodes	Time [s]	Configurations
1	normalization	1	0.01	1
	weak tightness	1	0.00	
2	normalization	5	0.01	2
	weak tightness	3	0.00	
3	normalization	61	0.01	6
	weak tightness	13	0.01	
4	normalization	1 393	0.01	22
	weak tightness	77	0.00	
5	normalization	52 009	0.01	98
	weak tightness	577	0.01	
6	normalization	2 901 007	0.07	516
	weak tightness	5 321	0.00	
7	normalization	$2.3 \cdot 10^8$	2.59	3 140
	weak tightness	$5.9 \cdot 10^4$	0.01	
8	normalization	$2.4 \cdot 10^{10}$	274.63	21 684
	weak tightness	$7.6 \cdot 10^5$	0.03	
9	normalization	–	–	167 450
	weak tightness	$1.1 \cdot 10^7$	0.23	
10	normalization	–	–	1 429 100
	weak tightness	$1.9 \cdot 10^8$	2.09	
11	normalization	–	–	13 350 964
	weak tightness	$3.6 \cdot 10^9$	34.79	
12	normalization	–	–	135 452 972
	weak tightness	$7.5 \cdot 10^{10}$	732.93	
13	normalization	–	–	1 482 478 624
	weak tightness	$1.7 \cdot 10^{12}$	17 085.22	

Table 7.3: Impact of weak tightness pruning. Column 3 gives the number of nodes in the enumeration tree. Column 4 gives the running time of the algorithm in seconds. The last column lists the number of normalized tight configurations for the given n.



(a) v_m^{max} is reachable from v_l^{max} in $G_{\mathcal{W}}$. **(b)** v_m^{max} is reachable from v_i^{max} in $G_{\mathcal{S}}$.

Figure 7.6: Illustration of Lemma 7.50 with $(i, m), (l, m) \in \mathcal{W} \setminus \mathcal{S}$, including relevant edges (solid) and paths (dashed) in $G_{\mathcal{S}}$ and $G_{\mathcal{W}}$. All drawn edges and paths do not depend on the placement chosen in this example, but only on the preconditions of Lemma 7.50. Rectangles implying the indicated paths (e.g., k_{y} and k_{x} in the sense of Lemma 7.47) are omitted. Note that both (a) and (b) illustrate parts of both graphs $G_{\mathcal{S}}$ and $G_{\mathcal{W}}$: Vertices in $G_{\mathcal{S}}$ are drawn at the lower and upper border of rectangles, and vertices in $G_{\mathcal{W}}$ are drawn at the left and right border of rectangles.

Moreover, assume that there is $m \in [\![n]\!] \setminus \{i, l, j_{\mathrm{x}}, j_{\mathrm{y}}\}$ with $(i, m), (l, m) \in (\mathcal{S} \cup \mathcal{W}) \setminus (\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}})$. Then, we have:

(i) If v_m^{max} is reachable from v_l^{max} in $G_{\mathcal{W}}$, then $(j_{\mathrm{y}}, m) \notin \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$, so $j_{\mathrm{y}} < m$.

(ii) If v_m^{max} is reachable from v_i^{max} in $G_{\mathcal{S}}$, then $(j_{\mathrm{x}}, m) \notin \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$, so $j_{\mathrm{x}} < m$.

Proof. In the first case, the given conditions directly imply (cf. Figure 7.6(a)) that v_m^{max} is reachable from $v_{j_{\mathrm{y}}}^{min}$ in both $G_{\mathcal{S}}$ and $G_{\mathcal{W}}$, so $(j_{\mathrm{y}}, m) \notin \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$.

In the second case, v_m^{max} is reachable from $v_{j_{\mathrm{x}}}^{min}$ in both $G_{\mathcal{S}}$ and $G_{\mathcal{W}}$ (cf. Figure 7.6(b)), so $(j_{\mathrm{x}}, m) \notin \overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}}$. □

Definition 7.51. *Let $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ be a partial configuration and let $(i, l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$. We say that (i, l) is **x-uncovered** in $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ if there is no path from v_i^{max} to v_l^{min} in $G_{\mathcal{W},\mathcal{A}}$ that does not consist of a single edge.*
*We say that (i, l) is **y-uncovered** in $(\mathcal{S}, \mathcal{W}, \mathcal{A})$ if there is no path from v_l^{max} to v_i^{min} in $G_{\mathcal{S},\mathcal{A}}$ that does not consist of a single edge.*

Now Lemmata 7.47 and 7.50 imply:

Lemma 7.52. *Let $(\mathcal{S}, \mathcal{W}, {}^2[\![m]\!])$ be a partial configuration. Let $(i, l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$ such that both $(i, m), (l, m) \in (\mathcal{S} \cup \mathcal{W}) \setminus (\overleftarrow{\mathcal{S}} \cup \overleftarrow{\mathcal{W}})$.*
Then, we have:

(i) If v_m^{max} is reachable from v_l^{max} in $G_{\mathcal{W},^2[\![m]\!]}$ and (i,l) is y-uncovered in $(\mathcal{S},\mathcal{W},{}^2[\![m]\!])$, then $(\mathcal{S},\mathcal{W},{}^2[\![m]\!])$ is invalid.

(ii) If v_m^{max} is reachable from v_i^{max} in $G_{\mathcal{S},^2[\![m]\!]}$ and (i,l) is x-uncovered in $(\mathcal{S},\mathcal{W},{}^2[\![m]\!])$, then $(\mathcal{S},\mathcal{W},{}^2[\![m]\!])$ is invalid.

Proof. Again, we only show the first statement (ii), the second statement is proven analogously.

So assume that v_m^{max} is reachable form v_l^{max} in $G_{\mathcal{W},^2[\![m]\!]}$. We prove that if $(\mathcal{S}^*,\mathcal{W}^*)$ is a normalized tight completion of $(\mathcal{S},\mathcal{W},\mathcal{A})$, then (i,l) is not y-uncovered.

Let $(\mathcal{S}^*,\mathcal{W}^*)$ be a normalized tight completion of $(\mathcal{S},\mathcal{W},{}^2[\![m]\!])$, and let $j_\mathrm{x}, j_\mathrm{y}, k_\mathrm{x}, k_\mathrm{y} \in [\![n]\!]$ as in Lemma 7.47 applied to $(\mathcal{S}^*,\mathcal{W}^*)$. Note that in particular we have $k_\mathrm{y} < i < m$. Moreover, since $G_{\mathcal{W},^2[\![m]\!]}$ is a subgraph of $G_\mathcal{W}$, we know that v_m^{max} is reachable from v_l^{max} in $G_\mathcal{W}$. Now, $(\mathcal{S}^*,\mathcal{W}^*)$ together with $i, l, j_\mathrm{x}, j_\mathrm{y}, m$ satisfy the conditions of Lemma 7.50, and thus we get $j_\mathrm{y} < m$. Finally, $v_l^{max}, v_{j_\mathrm{y}}^{min}, v_{k_\mathrm{y}}^{max}, v_i^{min}$ form a path in $G_{\mathcal{S}^*}$, and now $\{l, j_\mathrm{y}, k_\mathrm{y}, i\} \subseteq [\![m]\!]$ implies that the same vertices form a path in $G_{\mathcal{S},^2[\![m]\!]}$. Hence, (i,l) is not y-uncovered in $(\mathcal{S},\mathcal{W},{}^2[\![m]\!])$. □

To motivate the next (and last) pruning rule, consider the placement depicted in Figure 7.7(a): With $i = 3$ and $l = 5$, we are still lacking both $j_\mathrm{x}^{(3,5)}$ and $j_\mathrm{y}^{(3,5)}$ to satisfy Lemma 7.47. Moreover, with $i = 5$ and $l = 6$, we are also lacking both $j_\mathrm{x}^{(5,6)}$ and $j_\mathrm{y}^{(5,6)}$, and as $k_\mathrm{x}^{(5,6)} = k_\mathrm{y}^{(5,6)} = 4$ is the only possible assignment for $k_\mathrm{x}^{(5,6)}$ and $k_\mathrm{y}^{(5,6)}$ in this case, we know that $j_\mathrm{x}^{(5,6)} \neq j_\mathrm{y}^{(5,6)}$. Finally $j_\mathrm{y}^{(3,5)}$ must be only north of 5, which is not possible for both $j_\mathrm{x}^{(5,6)}$ and $j_\mathrm{y}^{(5,6)}$, so we need to add at least three rectangles to obtain a normalized tight placement. In particular, if $n < 9$, we can prune the partial configuration. In Figure 7.7(b), we see that indeed three rectangles do suffice: $j_\mathrm{x}^{(3,5)} = j_\mathrm{y}^{(3,5)} = 9$, $j_\mathrm{x}^{(5,6)} = 8$ and $j_\mathrm{y}^{(5,6)} = 7$.

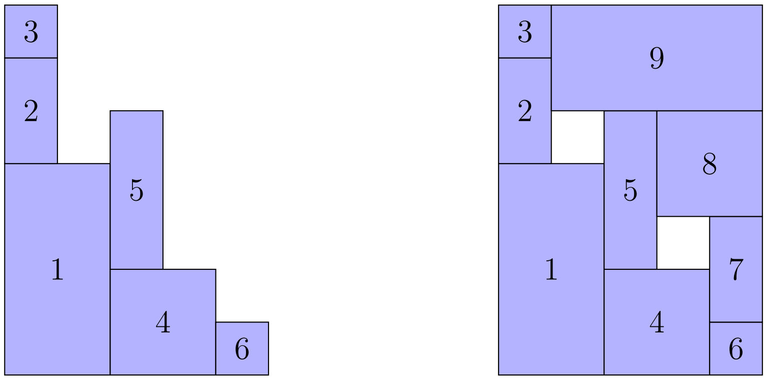
Lemma 7.53. *Let $(i,l) \in {}^2[\![n]\!]$ and $(\mathcal{S},\mathcal{W})$ be a normalized tight configuration on $[\![n]\!]$ with $(i,l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$. Furthermore, let $j_\mathrm{x}, j_\mathrm{y}, k_\mathrm{x}, k_\mathrm{y} \in [\![n]\!]$ as in Lemma 7.47.*
Then, for all $v \in \{j_\mathrm{x}, j_\mathrm{y}, k_\mathrm{x}, k_\mathrm{y}\}$, we have $(i,v),(v,l) \in (\overleftarrow{\mathcal{S}} \cup \mathcal{W}) \setminus (\mathcal{S} \cup \overleftarrow{\mathcal{W}})$.

Proof. We only show $(i,v) \in (\overleftarrow{\mathcal{S}} \cup \mathcal{W}) \setminus (\mathcal{S} \cup \overleftarrow{\mathcal{W}})$ for all $v \in \{j_\mathrm{x}, j_\mathrm{y}, k_\mathrm{x}, k_\mathrm{y}\}$, one can prove $(v,l) \in (\overleftarrow{\mathcal{S}} \cup \mathcal{W}) \setminus (\mathcal{S} \cup \overleftarrow{\mathcal{W}})$ analogously. We have

$$(i, j_\mathrm{x}) \in \mathcal{W} \setminus \mathrm{sym}(\mathcal{S}) \subseteq (\overleftarrow{\mathcal{S}} \cup \mathcal{W}) \setminus (\mathcal{S} \cup \overleftarrow{\mathcal{W}})$$

and

$$(i, k_\mathrm{y}) \in \overleftarrow{\mathcal{S}} \setminus \mathrm{sym}(\mathcal{W}) \subseteq (\overleftarrow{\mathcal{S}} \cup \mathcal{W}) \setminus (\mathcal{S} \cup \overleftarrow{\mathcal{W}}).$$



(a) A partial placement that has a normalized tight completion.

(b) A normalized tight placement that is a completion of the placement on the left.

Figure 7.7: At least three rectangles have to be added to the placement on the left to make it tight.

Furthermore v_i^{max} is reachable from $v_{j_\mathrm{y}}^{min}$ in $G_\mathcal{S}$, so $(i, j_\mathrm{y}) \notin \mathcal{S}$. As $(i,l) \in \mathcal{W}$ and l is only south of j_y, we get a path $v_i^{min}, v_i^{max}, v_l^{min}, v_{j_\mathrm{y}}^{max}$ in $G_\mathcal{W}$, so $(i, j_\mathrm{y}) \notin \overleftarrow{\mathcal{W}}$.

Similarly, $v_{k_\mathrm{x}}^{max}$ is reachable from v_i^{min} in $G_\mathcal{W}$, so $(i, k_\mathrm{x}) \notin \overleftarrow{\mathcal{W}}$, and $v_{k_\mathrm{x}}^{min}, v_l^{max}, v_i^{min}, v_i^{max}$ is a path in $G_\mathcal{S}$, so $(i, k_\mathrm{x}) \notin \mathcal{S}$. □

Lemma 7.53 implies that if $(a,b), (b,c) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W}$, then any sets of rectangles

$$X = \left\{ j_\mathrm{x}^{(a,b)}, j_\mathrm{y}^{(a,b)}, k_\mathrm{x}^{(a,b)}, k_\mathrm{y}^{(a,b)} \right\}$$

and

$$Y = \left\{ j_\mathrm{x}^{(b,c)}, j_\mathrm{y}^{(b,c)}, k_\mathrm{x}^{(b,c)}, k_\mathrm{y}^{(b,c)} \right\}$$

satisfying the conditions of Lemma 7.47 must be disjoint, as all elements of X can only be north or west of b, while elements of Y can only be south or east of b. Hence, we can obtain a lower bound on the number of rectangles that need to be added to a partial configuration by considering paths that only use edges in $\overleftarrow{\mathcal{S}} \cap \mathcal{W}$ that are uncovered.

More precisely, we construct a weighted graph G whose edges correspond to pairs (i,l) that are uncovered, and whose edge weights $w((i,l))$ are a lower bound on the number of rectangles that need to be added in order to cover (i,l). The edge weight will be 1, except if both $j_\mathrm{x}^{(i,l)}$ and $j_\mathrm{y}^{(i,l)}$ are missing, and we already know that $k_\mathrm{x}^{(i,l)} = k_\mathrm{y}^{(i,l)}$, implying $j_\mathrm{x}^{(i,l)} \neq j_\mathrm{y}^{(i,l)}$:

Corollary 7.54. *Let $(\mathcal{S}, \mathcal{W}, {}^2[\![m]\!])$ be a partial configuration on $[\![n]\!]$ and let*

$$E_{\mathrm{x}} := \Big\{ (i,l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W} : (i,l) \text{ is x-uncovered in } (\mathcal{S}, \mathcal{W}, {}^2[\![m]\!]) \Big\},$$
$$E_{\mathrm{y}} := \Big\{ (i,l) \in \overleftarrow{\mathcal{S}} \cap \mathcal{W} : (i,l) \text{ is y-uncovered in } (\mathcal{S}, \mathcal{W}, {}^2[\![m]\!]) \Big\}.$$

Furthermore, for $(i,l) \in E_{\mathrm{x}} \cup E_{\mathrm{y}}$, we denote by $K_{\mathrm{x}}^{(i,l)}$ the set of k_{x}-candidates of (i,l), and by $K_{\mathrm{y}}^{(i,l)}$ the set of k_{y}-candidates of (i,l) in $(\mathcal{S}, \mathcal{W}, {}^2[\![m]\!])$ (cf. Definition 7.48).
Let (G, w) be the weighted directed graph with vertex set $[\![m]\!]$, edge set

$$E(G) := E_{\mathrm{x}} \cup E_{\mathrm{y}},$$

and weights

$$w((i,l)) := \begin{cases} 2 & \text{if } (i,l) \in E_{\mathrm{x}} \cap E_{\mathrm{y}} \text{ and } \left| K_{\mathrm{x}}^{(i,l)} \cup K_{\mathrm{y}}^{(i,l)} \right| = 1, \\ 1 & \text{otherwise.} \end{cases}$$

Let W be the weight of a longest path in (G, w). If $n < m + W$, then $(\mathcal{S}, \mathcal{W}, {}^2[\![m]\!])$ is invalid.

Note that since G is acyclic, we can compute W in $\mathcal{O}(n^2)$ time by processing its vertices in topological order.

We refer to the pruning rules according to Lemma 7.52 and Corollary 7.54 when used together with weak tightness pruning by **strong tightness pruning**, results are given in Table 7.4: For small n, the benefit of using strong tightness pruning compared to weak tightness pruning is only marginal, but for large n there is a substantial reduction in both number of enumeration nodes and running time, allowing the enumeration of all normalized tight configurations up to $n = 14$. For $n = 14$, the number of leaves of the enumeration tree is $1.6 \cdot 10^{11}$ (not given in Table 7.4), less than ten times the number of normalized tight configurations, demonstrating the effectiveness of pruning.

Note that we only *enumerate* all normalized tight configurations, but do not *store* them, as for $n = 14$ all normalized tight configurations would not fit into 512 GB of main memory. Hence, the enumeration of all normalized tight configurations is only limited by the available memory to store the result, and hence further speedups of Algorithm 7.1 would be of limited use.

7.2.5 Implementation Details

Finally, we describe a few details of the implementation of Algorithm 7.1. We represent a strict partial order Q (e.g., $\mathcal{S}$ and $\mathcal{W}$) on $[\![n]\!]$ as a function that assigns each pair $1 \leq i < j \leq n$ a value indicating whether $(i,j) \in Q$, $(j,i) \in Q$, or none of the two, allowing to query whether $(i,j) \in Q$ holds in constant time. Moreover, we store all digraphs using adjacency lists.

n	Pruning	Nodes	Time [s]	Configurations
1	weak tightness	1	0.00	1
	strong tightness	1	0.01	
2	weak tightness	3	0.00	2
	strong tightness	3	0.01	
3	weak tightness	13	0.01	6
	strong tightness	13	0.01	
4	weak tightness	77	0.00	22
	strong tightness	68	0.01	
5	weak tightness	577	0.01	98
	strong tightness	472	0.01	
6	weak tightness	5 321	0.00	516
	strong tightness	3 959	0.01	
7	weak tightness	58 827	0.01	3 140
	strong tightness	37 757	0.01	
8	weak tightness	761 900	0.03	21 684
	strong tightness	394 300	0.02	
9	weak tightness	11 342 792	0.23	167 450
	strong tightness	4 471 047	0.13	
10	weak tightness	$1.9 \cdot 10^8$	2.09	1 429 100
	strong tightness	$5.5 \cdot 10^7$	0.91	
11	weak tightness	$3.6 \cdot 10^9$	34.79	13 350 964
	strong tightness	$7.1 \cdot 10^8$	10.06	
12	weak tightness	$7.5 \cdot 10^{10}$	732.93	135 452 972
	strong tightness	$10.0 \cdot 10^9$	138.07	
13	weak tightness	$1.7 \cdot 10^{12}$	17 085.22	1 482 478 624
	strong tightness	$1.5 \cdot 10^{11}$	2078.47	
14	weak tightness	–	–	17 403 502 928
	strong tightness	$2.3 \cdot 10^{12}$	33 302.86	

Table 7.4: Impact of strong tightness pruning. Column 3 gives the number of nodes in the enumeration tree. Column 4 gives the running time of the algorithm in seconds. The last column lists the number of normalized tight configurations for the given n.

Bitsets

Given a directed graph G and a vertex $u \in V(G)$, we denote by $V_G^+(u)$ the set of vertices reachable from u in G. As all occurring graphs contain at most $2n$ vertices (cf. the constraint graphs of interval orders) and $n \leq 14$, we have at most 28 vertices. We can thus encode $V_G^+(u)$ in a bitset containing 28 bits. The union of two such sets can be computed in constant time using a single bitwise `OR`-operation. Then, the observation

$$V_G^+(u) = \{u\} \cup \bigcup_{(u,v)\in\delta^+(u)} V_G^+(v)$$

allows to compute an all-pairs reachability table of an acyclic directed graph G by processing its vertices in reverse topological order. Note that a topological order of G can be computed in $\mathcal{O}\big(|V(G)| + |E(G)|\big)$ time [KV18]. The total number of instructions required by this procedure is *linear* in $|V(G)| + |E(G)|$. Moreover, when adding an edge (a, b) to G, we can use the observation

$$V_{G+(a,b)}^+(u) = \begin{cases} V_G^+(u) \cup V_G^+(b) & \text{if } a \in V_G^+(u) \\ V_G^+(u) & \text{otherwise} \end{cases}$$

to update the reachability table of G. The number of instructions required by this procedure is *linear* in $\big|V(G)\big|$.

Fixed-Capacity Dynamic-Length Arrays

Again, as $n \leq 14$, we can bound the length of most dynamic-length arrays used in the algorithm by a small constant. In particular, this applies to the storage of strict partial orders, adjacency lists of graphs and their reachability tables. For these arrays, instead of using ordinary dynamic-length arrays that rely on dynamic allocations (e.g., `std::vector<T>`), we use `boost::container::static_vector<T, N>`, which is provided by the `boost` C++ library [Boo18]. It can only be used to store up to `N` elements of type `T` and uses a fixed-size buffer to store these elements. We can thus completely avoid the cost of dynamic allocations for these arrays. Moreover, since the contained elements are stored *within* the memory of the container itself (and not in a different, dynamically allocated memory region that needs to be accessed by following a pointer), cache locality is improved.

Parallelization

Clearly, the enumeration of all normalized tight configurations can easily be parallelized by simply processing multiple enumeration subtrees in parallel. To this end, we first run the algorithm with bounded recursion depth, and collect the set of enumeration nodes that were cut off. Each of these nodes defines a subtree of the enumeration tree that can be processed independently of the others, which we then do in parallel, using 64 threads.

Parallel	Fixed cap. arrays	Bitsets	Relative running time
Yes	Yes	Yes	1.0
Yes	Yes	–	2.5
Yes	–	Yes	4.2
Yes	–	–	16.9
–	Yes	Yes	52.3
–	Yes	–	130.7
–	–	Yes	210.7
–	–	–	841.5

Table 7.5: Impact of speedup techniques to enumeration running time in the case $n = 11$ with strong tightness pruning. Column 1 specifies whether parallelization was used (with 64 threads). Column 2 specifies whether we replace generic dynamic arrays by fixed-capacity arrays that avoid dynamic allocations. Column 3 specifies whether we use bitsets to speed up reachability computations. The last column gives the running time relative to the case that all three techniques are used.

Results

Experimental results are given in Table 7.5. Running time is reduced significantly both by using bitsets to quickly compute reachability data and using fixed-capacity arrays to avoid dynamic allocations. Moreover, parallelization gives a reasonable speedup compared to the number of threads.

7.2.6 SP-Equivalence Filtering

As already mentioned, we ignore SP-equivalence within Algorithm 7.1. It would be possible to exploit SP-equivalence and directly work on SP-equivalence classes, which however would significantly complicate both the algorithm and its analysis. We will see that for small n, the number of SP-equivalence classes of normalized tight configurations is not drastically smaller than the number of normalized tight configurations, and hence the running time cost of not exploiting SP-equivalence within configuration enumeration is limited as well.

We apply Theorem 7.33 to determine the set of SP-equivalence classes by simply computing the set of reductions, removing duplicate reductions and then keeping a single arbitrary configuration for each found reduction.

Results are given in Table 7.6, which are restricted to $n \leq 12$ due to memory limitations. First, we remark that by selecting a separate sequence pair for each SP-equivalence class of tight configurations, one obtains a complete set of representations, and hence the number of SP-equivalence classes of normalized tight configurations is an upper bound on $\frac{CR_n}{n!}$. For $n \leq 4$, the number of

n	Configurations	SP-eq. classes
1	1	1
2	2	2
3	6	6
4	22	22
5	98	96
6	516	478
7	3 140	2 624
8	21 684	15 550
9	167 450	98 036
10	1 429 100	650 464
11	13 350 964	4 504 774
12	135 452 972	32 356 774

Table 7.6: The number of normalized tight configurations (column 2) and the number of SP-equivalence classes of normalized tight configurations (column 3) for $n \leq 12$.

SP-equivalence classes equals the number of normalized tight configurations. This is expected: For $n \leq 4$, the number of normalized tight configurations equals the number of biplane permutations, which by Theorem 6.12 is a lower bound on $\frac{CR_n}{n!}$ and hence the number of normalized tight SP-equivalence classes.

For $n = 5$, there are two non-trivial SP-equivalence classes: The configurations depicted in Figure 7.4 (page 74) are SP-equivalent, and rotating these by 90° again yields two SP-equivalent configurations.

Comparing with Table 3.3 (page 32), we also observe that, at least for $4 \leq n \leq 12$, the number of SP-equivalence classes of normalized tight configurations is *strictly less* than the number of plane permutations. This implies that the upper bound of Theorem 5.7 is not tight for $4 \leq n \leq 12$.

7.3 Set Cover Results

In this section, we finally reduce the computation of CR_n to a set cover problem, where sets correspond to sequence pairs and elements correspond to configurations. In Section 7.3.1, we formally define the MINIMUM SET COVER PROBLEM and describe the reduction-based algorithm that we will use to solve all occurring set cover instances. Then, in Section 7.3.2, we compute CR_n for $n \leq 8$. Moreover, in Section 7.3.3, we show that the analysis of our improved upper bound construction which is based on topological orders of augmented digraphs is essentially optimal, and that it is not possible to obtain complete sets of sequence pairs of minimum cardinality using plane sequence pairs only. Finally, in Section 7.3.4, we discuss symmetric sets of sequence pairs with computational results up to $n = 12$.

7.3.1 Set Cover Algorithm

First, we formally introduce the MINIMUM SET COVER PROBLEM. A **set system** $(\mathcal{U}, \mathcal{M})$ consists of an arbitrary set $\mathcal{U}$, called **universe**, and a set $\mathcal{M}$ of subsets of the universe.

MINIMUM SET COVER PROBLEM	
Instance:	A set system $(\mathcal{U}, \mathcal{M})$.
Task:	Find a minimum cardinality cover $\mathcal{N}$ of $(\mathcal{U}, \mathcal{M})$, i.e., a subset $\mathcal{N} \subseteq \mathcal{M}$ with $\mathcal{U} = \bigcup_{N \in \mathcal{N}} N$ minimizing $\lvert \mathcal{N} \rvert$.

The MINIMUM SET COVER PROBLEM is well-known to be *NP*-complete, even in the unweighted variant considered here ([KV18]).

In order to solve the arising set cover instances, we will use well-known reduction techniques that eliminate elements of $\mathcal{M}$ or $\mathcal{U}$, ensuring that optimum solutions of the reduced instance can be trivially extended to optimum solutions of the original instance. For example, if we have two sets $M_1, M_2 \in \mathcal{M}$ with $M_1 \subseteq M_2$, then we can safely remove M_1 from $\mathcal{M}$, as it can be replaced by M_2 in any feasible solution. These reductions are more commonly used as a preprocessing step before solving the reduced instance using an exact algorithm with exponential worst-case running time. However, in our case, we will see that all considered instances will be solved entirely by reductions.

Now, let $(\mathcal{U}, \mathcal{M})$ be an instance of the MINIMUM SET COVER PROBLEM. Given an element $u \in \mathcal{U}$, we denote by $\mathcal{M}_u \subseteq \mathcal{M}$ the set of sets in $\mathcal{M}$ that cover u:

$$\mathcal{M}_u := \{ M \in \mathcal{M} : u \in M \}$$

We use the following reduction rules which were first observed by Garfinkel and Nemhauser [GN72]:

(i) If $\mathcal{M}_u = \emptyset$ for some $u \in \mathcal{U}$, then $(\mathcal{U}, \mathcal{M})$ is infeasible.

(ii) If $\mathcal{M}_u = \{ N \}$ for some $u \in \mathcal{U}$, then every feasible solution contains N. Reduce the instance to $(\mathcal{U} \setminus N, \{ M \setminus N : M \in \mathcal{M} - N \})$.

(iii) If $\mathcal{M}_{u_1} \subseteq \mathcal{M}_{u_2}$ for some $u_1, u_2 \in \mathcal{U}$ with $u_1 \neq u_2$, then every cover of $\mathcal{U} - u_2$ also covers $\mathcal{U}$ and we say that u_1 **dominates** u_2.
Reduce the instance to $(\mathcal{U} - u_2, \{ M - u_2 : M \in \mathcal{M} \})$.

(iv) If $M_1 \subseteq M_2$ for some $M_1, M_2 \in \mathcal{M}$ with $M_2 \neq M_1$, then M_1 can be replaced by M_2 in any solution and we say that M_2 **dominates** M_1.
Reduce the instance to $(\mathcal{U}, \mathcal{M} - M_1)$.

The instances to be solved will be huge, more precisely, the universe $\mathcal{U}$ may contain many millions of elements. On the other hand, instances will be *sparse*, that is, the sets $M \in \mathcal{M}$ will be small, and $|\mathcal{M}|$ will have the same order of magnitude as $|\mathcal{U}|$.

With that in mind, we represent the set system $(\mathcal{U}, \mathcal{M})$ by an undirected sparse bipartite graph G with $V(G) := \mathcal{U} \cup \mathcal{M}$ and

$$E(G) := \big\{ \{u, M\} : u \in \mathcal{U}, M \in \mathcal{M}, \text{ and } u \in M \big\}.$$

The neighbors of an element $u \in \mathcal{U}$ are exactly the sets in $\mathcal{M}_u$, and the neighbors of a set $M \in \mathcal{M}$ are exactly the elements of M itself. We store G using adjacency lists.

Of course, when applying a reduction rule, we do not create a new reduced instance to be solved, but instead change the current instance with respect to the reduction rule, and store that the set N is part of the solution in case of reduction rule (ii).

Given an element $u \in \mathcal{U}$, implementing the reduction rules (i) and (ii) is trivial. The reduction rules (iii) and (iv) both can be implemented in terms of the following problem: Given a vertex $v_1 \in V(G)$, compute the set of vertices

$$\operatorname{dom}(v_1) := \big\{ v_2 \in V(G) - v_1 : \Gamma_G(v_1) \subseteq \Gamma_G(v_2) \big\}.$$

For an element $u \in \mathcal{U}$, the set $\operatorname{dom}(u)$ is the set of elements *dominated* by u, and for a set $M \in \mathcal{M}$, the set $\operatorname{dom}(M)$ consists of the sets *dominating* M. In order to compute $\operatorname{dom}(v)$ for a vertex v with $\Gamma_G(v) \neq \emptyset$, we observe

$$\operatorname{dom}(v) + v = \bigcap_{w \in \Gamma_G(v)} \Gamma_G(w).$$

Hence, $\operatorname{dom}(v_1)$ can be computed by visiting all neighbors $v_2 \in \Gamma_G(w)$ of all neighbors $w \in \Gamma_G(v_1)$ of v_1. Then, $\operatorname{dom}(v_1) + v_1$ consists of exactly the vertices v_2 which were visited $\big|\Gamma_G(v_1)\big|$ times. After once initializing a counter for every vertex $v \in V(G)$ in $\Theta\Big(\big|V(G\big|\Big)$ time, we can thus compute $\operatorname{dom}(v)$ in $\mathcal{O}\Big(\sum_{w \in \Gamma_G(v)} \big|\Gamma_G(w)\big|\Big)$ time, which is sufficiently fast in our application where G can assumed to be sparse.

Finally, we need to efficiently detect candidate vertices to apply reduction rules to. Simply repeatedly scanning all vertices and testing all applicable reduction rules easily leads to a quadratic running time, which is infeasible if G contains millions of vertices. Instead, we maintain a reduction candidate queue that initially contains all vertices. In each iteration, we remove a vertex v from the candidate queue and apply reduction rules to v. If a reduction rule is successful, we add all vertices whose neighborhood changed back to the candidate queue.

Note that removing a set from the neighborhood of an element u does not potentially lead to be u being dominated, but instead may lead to now

u dominating other elements. Hence, when applying the element dominance reduction rule (iii) to an element u, we need to check whether u dominates other elements, and not whether u is dominated by other elements. This is consistent with the implementation of this reduction rule in terms of $\mathrm{dom}(u)$, which gives the set of elements dominated by u. Hence, whenever a reduction rule is applicable to a vertex v, we know that v is contained in the candidate queue.

7.3.2 Main Result: CR_n for $n \leq 8$

Recall that we can use a set $\mathcal{C}_n^{\mathrm{T,SP}}$ of SP-equivalence representatives of the set of tight configurations to compute CR_n:

Observation 7.25. *Let $n \in \mathbb{N}$ and $\mathcal{C}_n^T$ be the set of tight configurations on $[\![n]\!]$. Moreover, let $\mathcal{C}_n^{T,SP} \subseteq \mathcal{C}_n^T$ be a set that contains a representative of each SP-equivalence class of $\mathcal{C}_n^T$.*
Then, we have

$$CR_n = \min\Big\{ |\mathcal{SP}| : \mathcal{SP} \subseteq \mathcal{SP}_n \textit{ covers all } (\mathcal{S}, \mathcal{W}) \in \mathcal{C}_n^{T,SP} \Big\}.$$

The algorithm described in Section 7.2 allows to enumerate a set $\mathcal{C}_n^{\mathrm{T,SP,N}}$ of SP-equivalence representatives of *normalized* tight configurations. Moreover, recall that given a set $Q \subseteq {}^2[\![n]\!]$ and a permutation $\pi \in \Pi_n$, the set $\pi(Q) \subseteq {}^2[\![n]\!]$ is obtained by relabeling the elements of $[\![n]\!]$ according to π, cf. Definition 7.35. Identifying configurations that can be transformed into each other by this operation yields an equivalence relation, and normalized configurations are unique representatives of the equivalence classes of this equivalence relation.

Hence, we can compute a set $\mathcal{C}_n^{\mathrm{T,SP}}$ satisfying the conditions of Observation 7.25 by applying all $n!$ permutations $\pi \in \Pi_n$ to each configuration $(\mathcal{S}, \mathcal{W}) \in \mathcal{C}_n^{\mathrm{T,SP,N}}$:

$$\mathcal{C}_n^{\mathrm{T,SP}} := \Big\{ \big(\pi(\mathcal{S}), \pi(\mathcal{W})\big) : (\mathcal{S}, \mathcal{W}) \in \mathcal{C}_n^{\mathrm{T,SP,N}}, \pi \in \Pi_n \Big\}$$

We can finally define the set cover instance $(\mathcal{U}, \mathcal{M})$ to be solved: Set $\mathcal{U} := \mathcal{C}_n^{\mathrm{T,SP}}$ and

$$\mathcal{M} := \big\{ M_{\pi,\rho} : (\pi, \rho) \in \mathcal{SP}_n \big\},$$

where

$$M_{\pi,\rho} := \big\{ (\mathcal{S}, \mathcal{W}) \in \mathcal{U} : (\mathcal{S}, \mathcal{W}) \text{ is represented by } (\pi, \rho) \big\}$$

for $(\pi, \rho) \in \mathcal{SP}_n$. Then, the cardinality of an optimum solution of $(\mathcal{U}, \mathcal{M})$ clearly equals CR_n.

n	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{M}\rvert$	CR_n	$T_{\text{constr.}}$ [s]	T_{solve} [s]
1	1	1	1	0.00	0.00
2	4	4	4	0.00	0.00
3	36	36	36	0.00	0.00
4	528	576	528	0.00	0.00
5	11 520	14 400	11 040	0.04	0.00
6	344 160	518 400	303 840	0.61	0.14
7	13 224 960	25 401 600	10 452 960	29.38	9.37
8	626 976 000	1 625 702 400	433 601 280	2121.54	1159.56

Table 7.7: Computational set cover results. Column 1 gives the number of rectangles n, columns 2 and 3 give the size of the computed set cover instance $(\mathcal{U}, \mathcal{M})$. Note that $\lvert\mathcal{U}\rvert$ equals $n!$ times the number of SP-equivalence classes of normalized tight configurations (cf. Table 7.6), and $\lvert\mathcal{M}\rvert = (n!)^2$. Column 4 gives the minimum cardinality CR_n of a complete set of representations for n, and the last two columns give the running time of constructing and solving the set cover instance, respectively. Note that $T_{\text{constr.}}$ covers *both* the construction of $\mathcal{U}$ by applying all $n!$ permutations to the normalized configurations computed earlier, and the computation of $\mathcal{M}$ by enumerating the set of sequence pairs representing given configurations.

The computation of $\mathcal{M}$ is not trivial, as naïvely enumerating all pairs of sequence pairs and configurations would be computationally infeasible even for small n. Instead, it suffices to solve the following problem: Given a configuration $(\mathcal{S}, \mathcal{W})$, enumerate the set of sequence pairs (π, ρ) representing $(\mathcal{S}, \mathcal{W})$. However, by Lemma 4.15, computing the set of sequence pairs representing a configuration can be reduced to the problem of enumerating the set of topological orders of an acyclic digraph G.

So let G be an acyclic digraph. Recall that one can compute a topological order of G in $\mathcal{O}(\lvert V(G)\rvert^2)$ time by repeatedly removing a vertex v with in-degree zero from G ([Kah62]). By maintaining a queue that contains the set of vertices with in-degree zero, the running time of this algorithm can be improved to $\mathcal{O}(\lvert V(G)\rvert + \lvert E(G)\rvert)$, but we do not need this as our graphs are dense. Clearly, we can instead *enumerate* the *set of all topological orders* of G by recursively enumerating all possible choices for v in each iteration.

Computational results are given in Table 7.7. All set cover results were obtained using the same machine and compiler as in Section 7.2 (cf. page 82), using a single thread.

Using this method, we can solve all occurring set cover instances, and thereby determine CR_n for all $n \leq 8$. The fact that the set cover instances

n	$Biplane_n$	$Baxter_n$	$Plane_n$	$\lvert\mathcal{U}\rvert/n!$	$CR_n/n!$
1	1	1	1	1	1
2	2	2	2	2	2
3	6	6	6	6	6
4	22	22	23	22	22
5	88	92	104	96	92
6	374	422	530	478	422
7	1 668	2 074	2 958	2 624	2 074
8	7 744	10 754	17 734	15 550	10 754

Table 7.8: The number of biplane, Baxter and plane permutations on $[\![n]\!]$ together with columns 2 and 4 of Table 7.7 normalized by $(n!)$.

can be solved entirely by simple reduction routines only is quite surprising, and suggests that the set cover instances exhibit a rich structure that could possibly also be exploited for new proofs, in particular for stronger lower bound constructions.

In the largest case $n = 8$, the average and maximum number of sets $M \in \mathcal{M}$ covering elements $u \in \mathcal{U}$ are 2.67 and 25, respectively, and the average and maximum sizes of sets $M \in \mathcal{M}$ are 1.03 and 8, respectively, so $(\mathcal{U}, \mathcal{M})$ is indeed very sparse. Hence, applying the reduction rules given in Section 7.3.1 is very fast, and takes even less time than the construction of $(\mathcal{U}, \mathcal{M})$. The computation of CR_8 required 235 GB of memory, and thus determining CR_9 using this approach is clearly infeasible memory-wise.

Both the lower bound (Theorem 6.12) and the upper bound (Theorem 5.7) on CR_n are multiples of $n!$, and we observe that also CR_n is a multiple of $n!$ for $n \leq 8$. Results normalized by $n!$ are given in Table 7.8, and we see that $CR_n = n! \cdot Baxter_n$ for $n \leq 8$. This motivates the main conjecture of this chapter:

Conjecture 7.55. *Let $n \in \mathbb{N}$. Then, we have*

$$CR_n = n! \cdot Baxter_n.$$

Note that Conjecture 7.55 would imply $CR_n = \Theta\left(\frac{n!}{n^4} \cdot 8^n\right)$. In the remainder of this chapter, we will collect further results supporting Conjecture 7.55. Of course, Conjecture 7.55 does *not* imply that the set of Baxter sequence pairs is a complete set of sequence pairs of minimum cardinality. Clearly, not all biplane permutations are Baxter permutations, and hence the set of Baxter sequence pairs is not complete. Instead, it seems that there is a different set of permutations which are equinumerous to Baxter permutations and yield a complete set of sequence pairs of minimum cardinality. Note that Baxter permutations also count mosaic floorplans (cf. Section 3.3.3).

7.3.3 Analysis of Upper Bound Construction

Although we already know that the upper bound of Theorem 5.7 on CR_n is not tight, this does not yet imply that the *construction* (i.e., pairs of topological orders of $G_{\text{SW+}}$ and $G_{\text{SE+}}$) is suboptimal, as the *analysis* is not necessarily best-possible. In this section, we will see that for small n, the analysis indeed is best-possible unless restricted to a *subset* of configurations (e.g., tight configurations), since *all* plane sequence pairs occur as topological orders of $G_{\text{SW+}}$ and $G_{\text{SE+}}$. Furthermore, we will show that no complete set of sequence pairs of minimum cardinality can be constructed by considering only *plane* sequence pairs, that is, sequence pairs that are obtained as topological orders of $G_{\text{SW+}}$ and $G_{\text{SE+}}$.

In the first experiment, we determine the set of sequence pairs that arise as topological orders of the augmented digraphs $G_{\text{SW+}}$ and $G_{\text{SE+}}$ of arbitrary, not necessarily tight configurations. Recall that we can easily modify Algorithm 7.1 to enumerate the set of all normalized configurations on $[\![n]\!]$ for $n \leq 8$ (cf. page 86). Hence, we can explicitly compute the set of sequence pairs (π, ρ) that appear as topological orders of $G_{\text{SW+}}$ and $G_{\text{SE+}}$ of normalized configurations $(\mathcal{S}, \mathcal{W})$. For all such sequence pairs, we must have $\pi = \mathrm{id}_{[\![n]\!]}$ by definition of normalization, and furthermore ρ must be plane by Lemma 5.4. The computational experiment shows that for $n \leq 8$, *all* plane permutations ρ do occur. Clearly, applying all permutations $\pi \in \Pi_n$ to the set of normalized configurations by re-labeling their elements yields the set of all configurations on $[\![n]\!]$, resulting in the set of all sequence pairs of the form $(\pi, \sigma \circ \pi)$ with σ plane. This means that the analysis of Lemma 5.4 is best possible for $n \leq 8$.

However, the fact that all plane sequence pairs occur as topological orders of $G_{\text{SW+}}$ and $G_{\text{SE+}}$ does not imply that all of these are required. In fact, for $n \leq 4$, we know that the set of biplane sequence pairs is a complete set of minimum cardinality, which is a subset of the set of plane sequence pairs. This means that it could be possible to obtain a complete set of sequence pairs of minimum cardinality by only using a subset of plane sequence pairs, e.g., by only considering tight configurations and their augmented digraphs. To answer this question, we repeat the experiment of Section 7.3.2, this time restricted to plane sequence pairs. This allows us to compute the minimum cardinality CR_n^{plane} of a complete set of plane sequence pairs on $[\![n]\!]$.

Results are given in Table 7.9. As expected, for $n \leq 4$, we have $CR_n^{\text{plane}} = CR_n$. However, for $n = 5$, we need strictly more sequence pairs to cover all configurations when restricted to plane sequence pairs. This proves that for $5 \leq n \leq 8$, there is no set of configurations on $[\![n]\!]$ such that topological orders of the augmented south-west and south-east digraphs $G_{\text{SW+}}$ and $G_{\text{SE+}}$ of these configurations leads to a complete set of sequence pairs of minimum cardinality.

We now discuss the case $n = 5$ in more detail. In the following, we are only interested in the *structure* of configurations and ignore the labeling of

n	$Biplane_n$	$Plane_n$	$\lvert\mathcal{U}\rvert/n!$	$CR_n/n!$	$CR_n^{\text{plane}}/n!$
1	1	1	1	1	1
2	2	2	2	2	2
3	6	6	6	6	6
4	22	23	22	22	22
5	88	104	96	92	94
6	374	530	478	422	450
7	1 668	2 958	2 624	2 074	2 349
8	7 744	17 734	15 550	10 754	13 128

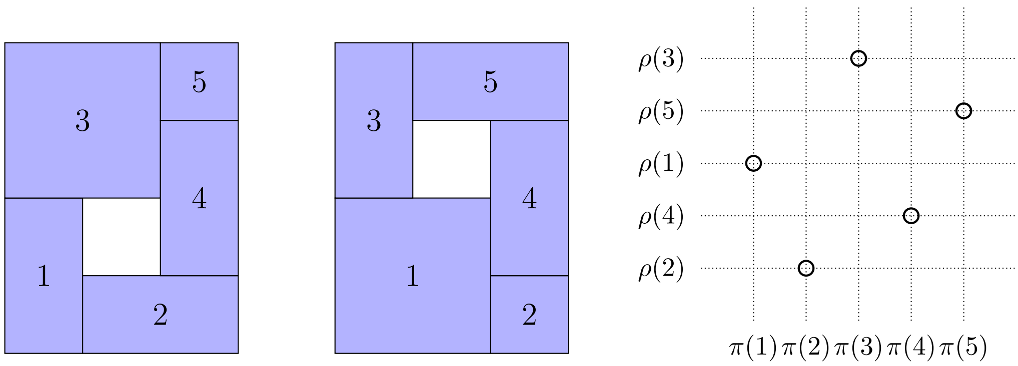
Table 7.9: The minimum cardinality CR_n^{plane} of a complete set of plane sequence pairs on $[\![n]\!]$, normalized by $n!$, is given in the last column. The remaining columns are copied from Table 7.8 in order to facilitate comparisons.

the rectangles. In other words, we only consider normalization equivalence classes of configurations. Note that this is different from considering *normalized* configurations, which are representatives of these equivalence classes with a fixed labeling. Consequently, we say that two equivalence classes *share a sequence pair* if there is a labeling of their rectangles that allows the resulting configurations to share a sequence pair. This way, we can only make statements about so-called *symmetric* sets of sequence pairs, which will be formally defined and discussed in Section 7.3.4. In particular, we stress that the analysis below does not prove properties of CR_n or CR_n^{plane}, but rather gives empirical details of the solved set cover instances. Still, the resulting observations may be useful in the construction of stronger lower or upper bounds.

There are 88 equivalence classes whose elements are represented by a unique sequence pair, counted by biplane permutations. The elements of the remaining $96 - 88 = 8$ equivalence classes (cf. columns $Biplane_n$ and $|\mathcal{U}|/n!$ in Table 7.9) are represented by exactly two sequence pairs each. Moreover, these 8 equivalence classes can be partitioned into 4 pairs that each share a sequence pair. This leads to $CR_5/5! = 88 + 4 = 92$. However, for two of these pairs, the resulting shared sequence pairs are not plane, and hence the corresponding four configurations are covered by a separate sequence pair each when restricted to plane sequence pairs, leading to $CR_5^{\text{plane}}/5! = 88 + 2 + 4 = 94$.

One of these pairs of equivalence classes is illustrated in Figure 7.8, using a suitable exemplary rectangle labeling. The common sequence pair (π, ρ) is not plane, as the elements $1, 2, 3, 5$ form a bad quartet. Note that (π, ρ) assigns the pairs $(1, 5)$ and $(2, 3)$ to south, while topological orders of $G_{\text{SW}+}$ and $G_{\text{SE}+}$ assign $(1, 5)$ to west in the case of Figure 7.8(a), and assign $(2, 3)$ to east in the case of Figure 7.8(b).

Finally, we remark that all 8 equivalence classes that allow to share a



(a) **(b)** **(c)**

Figure 7.8: Two placements with different configurations (left) that share a non-plane sequence pair (π, ρ) illustrated on the right.

sequence pair are, up to reflection and rotation, of the type depicted in Figure 7.8. This includes the placement given in Figure 6.4 (page 61), which we used to show that the lower bound of Theorem 6.12 is not tight for $n \geq 5$.

7.3.4 Symmetric Sets of Sequence Pairs

In this section, we introduce the concept of *symmetric* sets of sequence pairs, and compute the minimum cardinality CR_n^{sym} of complete symmetric sets of sequence pairs for $n \leq 12$. In particular, we observe that $CR_n = CR_n^{\text{sym}}$ for $n \leq 8$, and furthermore $CR_n^{\text{sym}} = n! \cdot \mathit{Baxter}_n$ for $n \leq 12$, supporting Conjecture 7.55. Finally, we will observe that the resulting set of sequence pairs (i.e., a complete symmetric set of sequence pairs of minimum cardinality) is induced by certain pattern-avoiding permutations which we will call *pseudo-biplane*. The number of pseudo-biplane permutations seems to equal the number of Baxter permutations (which we verify for $n \leq 15$), and moreover we will verify that the set of pseudo-biplane sequence pairs is complete for $n \leq 14$.

First, we define the symmetry property of a set $\mathcal{SP}$ of sequence pairs which means that if a sequence pair (π, ρ) is contained in $\mathcal{SP}$, then all sequence pairs (π', ρ') that are structure-equivalent to (π, ρ) are also contained in $\mathcal{SP}$, cf. Definition 4.3.

Definition 7.56. *Let $n \in \mathbb{N}$ and $\mathcal{SP} \subseteq \mathcal{SP}_n$. We say that $\mathcal{SP}$ is* ***symmetric*** *if*

$$\mathcal{SP} = \left\{ (\pi \circ \tau, \rho \circ \tau) : (\pi, \rho) \in \mathcal{SP}, \tau \in \Pi_n \right\}.$$

Note that if $\mathcal{SP}$ is a complete symmetric set of sequence pairs and $(\mathcal{S}, \mathcal{W})$ is a configuration represented by $(\pi, \rho) \in \mathcal{SP}$, then a sequence pair representing the configuration obtained by relabeling elements in $(\mathcal{S}, \mathcal{W})$ can be obtained by simply relabeling the elements in (π, ρ). Moreover, we remark that the sets of sequence pairs used in the proofs of our new lower and upper bounds on CR_n (Theorems 5.7 and 6.12) are indeed symmetric.

Recall that given a sequence pair (π, ρ), we refer by $\operatorname{struc}(\pi, \rho) = \rho \circ \pi^{-1}$ to the structural permutation of (π, ρ), cf. Definition 4.4. We now extend this notion to *sets* of sequence pairs: Given a set $\mathcal{SP}$ of sequence pairs, we refer by

$$\operatorname{struc}(\mathcal{SP}) := \big\{ \operatorname{struc}(\pi, \rho) : (\pi, \rho) \in \mathcal{SP} \big\}$$

to the set of structural permutations of $\mathcal{SP}$. Symmetric sets of sequence pairs are uniquely determined by the set of their structural permutations:

Lemma 7.57. *Let $n \in \mathbb{N}$ and $\mathcal{SP} \subseteq \mathcal{SP}_n$ be a symmetric set of sequence pairs. Then, we have*

$$\begin{aligned} \mathcal{SP} &= \Big\{ (\pi, \rho) : \pi, \rho \in \Pi_n,\ \operatorname{struc}(\pi, \rho) \in \operatorname{struc}(\mathcal{SP}) \Big\} \\ &= \Big\{ (\pi, \sigma \circ \pi) : \pi \in \Pi_n,\ \sigma \in \operatorname{struc}(\mathcal{SP}) \Big\}. \end{aligned}$$

In particular, we have $|\mathcal{SP}| = n! \cdot \big|\operatorname{struc}(\mathcal{SP})\big|$.

Proof. We prove

$$\begin{aligned} \mathcal{SP} &\subseteq \Big\{ (\pi, \rho) : \pi, \rho \in \Pi_n,\ \operatorname{struc}(\pi, \rho) \in \operatorname{struc}(\mathcal{SP}) \Big\} \\ &= \Big\{ (\pi, \sigma \circ \pi) : \pi \in \Pi_n,\ \sigma \in \operatorname{struc}(\mathcal{SP}) \Big\} \\ &\subseteq \mathcal{SP}. \end{aligned}$$

The first inclusion directly follows from the definition of $\operatorname{struc}(\mathcal{SP})$, and the subsequent equality is obtained by replacing $\operatorname{struc}(\pi, \rho) = \rho \circ \pi^{-1}$ by σ. To show the second inclusion, let $\sigma \in \operatorname{struc}(\mathcal{SP})$ and $\pi \in \Pi_n$ be arbitrary. Then, there are $(\pi', \rho') \in \mathcal{SP}$ with $\operatorname{struc}(\pi, \sigma \circ \pi) = \sigma = \operatorname{struc}\big(\pi', \rho'\big)$. Now, by Lemma 4.5 we know that $(\pi, \sigma \circ \pi)$ and (π', ρ') are structure-equivalent. Hence, since $\mathcal{SP}$ is symmetric and $(\pi', \rho') \in \mathcal{SP}$, we have $(\pi, \sigma \circ \pi) \in \mathcal{SP}$. □

Recall that we say that a set of sequence pairs $\mathcal{SP}$ *covers* a set of configurations $\mathcal{C}$ if for every configuration $(\mathcal{S}, \mathcal{W}) \in \mathcal{C}$, there is a sequence pair $(\pi, \rho) \in \mathcal{SP}$ representing $(\mathcal{S}, \mathcal{W})$.

Definition 7.58. *We say that a permutation $\sigma \in \Pi_n$* ***structure-represents*** *a configuration $(\mathcal{S}, \mathcal{W})$ if there is a sequence pair (π, ρ) representing $(\mathcal{S}, \mathcal{W})$ with $\sigma = \rho \circ \pi^{-1}$.*
Moreover, given a set of configurations $\mathcal{C}$ and a set of permutations Π, we say that Π ***structure-covers*** *$\mathcal{C}$ if for every configuration $(\mathcal{S}, \mathcal{W}) \in \mathcal{C}$, there is a permutation $\sigma \in \Pi$ that structure-represents $(\mathcal{S}, \mathcal{W})$.*

The following result implies that the computation of CR_n^{sym} can be reduced to the computation of a set Π of permutations of minimum cardinality that structure-covers all configurations. This leads to a set cover problem whose sets correspond to *permutations* instead of *sequence pairs*, dramatically reducing the number of candidate sets.

Lemma 7.59. *Let $n \in \mathbb{N}$, let $\mathcal{C}$ be a set of configurations on $[\![n]\!]$, and let $\mathcal{SP} \subseteq \mathcal{SP}_n$ be a symmetric set of sequence pairs.*
Then $\mathcal{SP}$ covers $\mathcal{C}$ if and only if $\operatorname{struc}(\mathcal{SP})$ *structure-covers $\mathcal{C}$.*

Proof. If $\mathcal{SP}$ covers $\mathcal{C}$, then by definition clearly $\operatorname{struc}(\mathcal{SP})$ structure-covers $\mathcal{C}$.

For the other direction, assume that $\operatorname{struc}(\mathcal{SP})$ structure-covers $\mathcal{C}$, and let $(\mathcal{S}, \mathcal{W}) \in \mathcal{C}$. Then, there is a sequence pair (π, ρ) representing $(\mathcal{S}, \mathcal{W})$ with $\operatorname{struc}(\pi, \rho) \in \operatorname{struc}(\mathcal{SP})$, and by Lemma 7.57 we have $(\pi, \rho) \in \mathcal{SP}$. $\square$

Recall that given a set $Q \subseteq {}^2[\![n]\!]$ and a permutation $\pi \in \Pi_n$, we denote by $\pi(Q)$ the relation obtained from Q by re-labeling the elements of $[\![n]\!]$ according to π, cf. Definition 7.35.

Lemma 7.60. *Let $n \in \mathbb{N}$ and let $\mathcal{C}$ be a set of configurations on $[\![n]\!]$. Define*

$$\mathcal{C}' := \Big\{ \big(\tau(\mathcal{S}), \tau(\mathcal{W})\big) : (\mathcal{S}, \mathcal{W}) \in \mathcal{C},\ \tau \in \Pi_n \Big\}.$$

Furthermore, let $\Pi \subseteq \Pi_n$ be a set of permutations.
Then Π structure-covers $\mathcal{C}$ if and only if Π structure-covers $\mathcal{C}'$.

Proof. As $\mathcal{C} \subseteq \mathcal{C}'$, clearly Π structure-covers $\mathcal{C}$ if Π structure-covers $\mathcal{C}'$.

For the other direction, assume that Π structure-covers $\mathcal{C}$, and let $\big(\tau(\mathcal{S}), \tau(\mathcal{W})\big) \in \mathcal{C}'$ with $(\mathcal{S}, \mathcal{W}) \in \mathcal{C}$ and $\tau \in \Pi$. Then, there is a sequence pair (π, ρ) with $\operatorname{struc}(\pi, \rho) \in \Pi$ such that (π, ρ) represents $(\mathcal{S}, \mathcal{W})$, i.e., $\mathcal{S}_{\pi,\rho} \subseteq \mathcal{S}$ and $\mathcal{W}_{\pi,\rho} \subseteq \mathcal{W}$. Set $\big(\pi', \rho'\big) := \big(\pi \circ \tau^{-1}, \rho \circ \tau^{-1}\big)$. Then, we see

$$\begin{aligned}
\mathcal{S}_{\pi',\rho'} &= \Big\{ (i,j) \in {}^2[\![n]\!] : \pi'(i) < \pi'(j) \text{ and } \rho'(i) < \rho'(j) \Big\} \\
&= \Big\{ (i,j) \in {}^2[\![n]\!] : \pi\big(\tau^{-1}(i)\big) < \pi\big(\tau^{-1}(j)\big) \text{ and } \rho\big(\tau^{-1}(i)\big) < \rho\big(\tau^{-1}(j)\big) \Big\} \\
&= \Big\{ \big(\tau(i), \tau(j)\big) \in {}^2[\![n]\!] : \pi(i) < \pi(j) \text{ and } \rho(i) < \rho(j) \Big\} \\
&= \tau(\mathcal{S}_{\pi,\rho}) \\
&\subseteq \tau(\mathcal{S}).
\end{aligned}$$

A similar computation shows $\mathcal{W}_{\pi',\rho'} \subseteq \tau(\mathcal{W})$, and hence (π', ρ') is a sequence pair representing $(\tau(\mathcal{S}), \tau(\mathcal{W}))$. Furthermore, clearly (π, ρ) and (π', ρ') are structure-equivalent, and hence Lemma 4.5 implies $\operatorname{struc}\big(\pi', \rho'\big) = \operatorname{struc}(\pi, \rho) \in \Pi$. $\square$

n	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{M}\rvert$	$CR_n^{\mathrm{sym}}/n!$	$Baxter_n$	$T_{\mathrm{constr.}}$ [s]	T_{solve} [s]
1	1	1	1	1	0.00	0.00
2	2	2	2	2	0.00	0.00
3	6	6	6	6	0.00	0.00
4	22	22	22	22	0.00	0.00
5	96	100	92	92	0.00	0.00
6	478	556	422	422	0.00	0.00
7	2 624	3 670	2 074	2 074	0.02	0.00
8	15 550	28 012	10 754	10 754	0.12	0.02
9	98 036	242 470	58 202	58 202	0.64	0.17
10	650 464	2 345 814	326 240	326 240	6.43	3.85
11	4 504 774	25 079 566	1 882 960	1 882 960	76.12	99.97
12	32 356 774	293 608 226	11 140 560	11 140 560	1008.99	5085.03

Table 7.10: Symmetric set cover results. Column 1 gives the number of rectangles, columns 2 and 3 give the size of the set cover instance $(\mathcal{U}, \mathcal{M})$. Note that $\lvert\mathcal{U}\rvert$ equals the number of SP-equivalence classes of normalized tight configurations (cf. Table 7.6 (page 100)). Column 4 gives the size of an optimum solution of the set cover instance, i.e., the minimum cardinality CR_n^{sym} of a complete symmetric set of representations for n divided by $(n!)$. Column 5 gives the number of Baxter permutations on $[\![n]\!]$ and agrees with column 4. The last two columns give the running time of constructing and solving the set cover instance, respectively.

Lemmata 7.57 and 7.60 imply that we can compute CR_n^{sym} using *normalized* configurations only, eliminating the need to explicitly apply all $n!$ labelings to all normalized configurations. More precisely, recall that we can compute a set $\mathcal{C}_n^{\mathrm{T,SP,N}}$ of SP-equivalence representatives of normalized tight configurations. Then, we construct a set cover instance $(\mathcal{U}, \mathcal{M})$, where

$$\mathcal{U} := \mathcal{C}_n^{\mathrm{T,SP,N}}$$

and

$$\mathcal{M} := \{\, M_\sigma : \sigma \in \Pi_n \,\},$$

using

$$M_\sigma := \big\{\, (\mathcal{S}, \mathcal{W}) \in \mathcal{U} : \sigma \text{ structure-represents } (\mathcal{S}, \mathcal{W}) \,\big\}$$

for $\sigma \in \Pi_n$. If $\mathcal{N} \subseteq \mathcal{M}$ is an optimum solution of $(\mathcal{U}, \mathcal{M})$, then Lemmata 7.57 and 7.60 imply that $CR_n^{\mathrm{sym}} = n! \cdot \lvert\mathcal{N}\rvert$. Results are given in Table 7.10. Again, we can solve all set cover instances using reductions only.

Solving the largest case $n = 12$ required approximately 107 GB of memory. Recall that the running time of the set cover reductions primarily depends on the degrees of vertices in the bipartite set cover graph. For $n = 12$, the average degree is 5.30 (26.68 for elements $u \in \mathcal{U}$ and 2.94 for sets $M \in \mathcal{M}$), which is much larger than in the largest case $n = 8$ of Table 7.7 (page 104), where the average degree is 1.49. This explains the fact that solving the set cover instance for CR_{12}^{sym} took longer than solving the set cover instance for CR_8, despite both having much fewer elements and sets.

For all tested values of n, e.g., $n \leq 12$, we observe $CR_n^{\mathrm{sym}} = n! \cdot \mathit{Baxter}_n$. In particular, this implies $CR_n^{\mathrm{sym}} = CR_n$ for $n \leq 8$, that is, for $n \leq 8$ there is complete set of sequence pairs of minimum cardinality that indeed is symmetric.

Using Lemma 7.57, we conclude that for $n \leq 8$, there is a set of permutations $\Pi_n^{\mathrm{opt}} \subseteq \Pi_n$ such that

$$\mathcal{SP}_n^{\mathrm{opt}} := \left\{ (\pi, \sigma \circ \pi) : \pi \in \Pi_n, \sigma \in \Pi_n^{\mathrm{opt}} \right\}$$

is a complete set of sequence pairs of minimum cardinality, and we can determine the set Π_n^{opt} as the optimum solution of the solved set cover instances. Not surprisingly, Π_n^{opt} is the set of permutations avoiding a certain pattern.

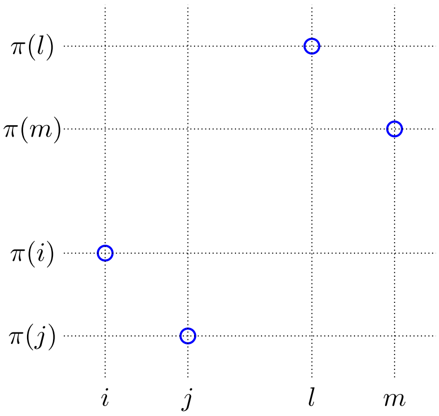
First, we need the auxiliary concept of *pseudo-plane* permutations. Before we define pseudo-plane permutations formally in Definition 7.61, recall that plane permutations π can be characterized as follows: Whenever there are indices $i < j < l < m$ with $j <_\pi i <_\pi m <_\pi l$ (forming a match of 2143), there must be an index k with $j < k < l$ and $i <_\pi k <_\pi m$ (forming a match of 21354). For *pseudo-plane* permutations, there are two more cases in which matches of 2143 are allowed: In both cases, we do not require an additional element k such that the relative order of i, j, k, l, m is pre-determined. Instead, in the first case, we require that the match i, j, l, m of 2143 can be turned into a match i, j', l', m of 2413 by replacing j by j' and l by l', where an element is allowed to be replaced by itself. Of course, at least one of j and l must be replaced by a different element in order to turn a match of 2143 into a match of 2413. In the second case, we replace i by i' and m by m' such that we obtain a match of 3142. For each possible replacement, the relative order of the replaced element and the replacing element is pre-determined:

Definition 7.61. *Let $n \in \mathbb{N}$ and $\pi \in \Pi_n$ be a permutation. We say that π is **pseudo-plane** if for all indices $i < j < l < m$ with $j <_\pi i <_\pi m <_\pi l$ (i.e., a match of 2143), one of the following conditions holds:*

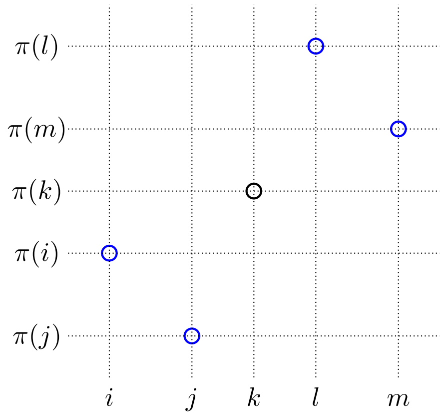
(i) There is an index k with $j < k < l$ and $i <_\pi k <_\pi m$ (ordinary plane case: match of 2154 embedded into match of 21354).

(ii) There are indices j', l' with $j \leq j'$, $j \leq_\pi j'$, $l' \leq l$, $l' \leq_\pi l$, $i < l' < j' < m$, and $j' <_\pi i <_\pi m <_\pi l'$ (forming a match of 2413).

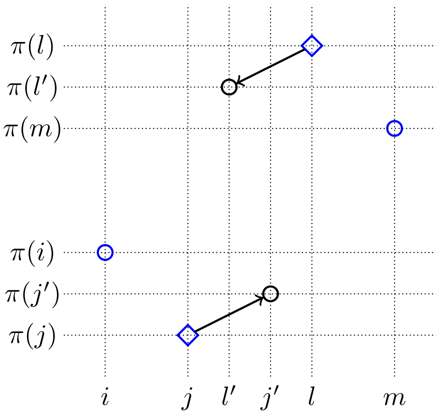
(iii) There are indices i', m' with $i \leq i'$, $i \leq_\pi i'$, $m' \leq m$, $m' \leq_\pi m$, $i' < j < l < m'$, and $j <_\pi m' <_\pi i' <_\pi l$ (forming a match of 3142).



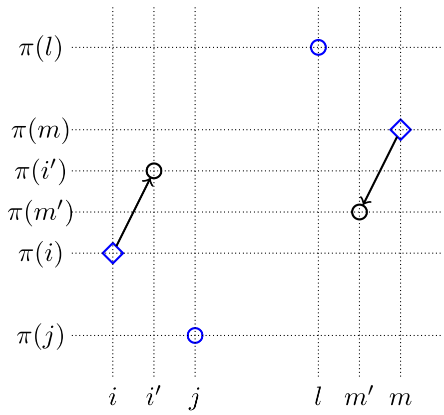
(a) Base setting: i, j, l, m form a match of 2143.



(b) Plane case (i): There is k such that i, j, k, l, m form a match of 21354.



(c) Example of case (ii): There are j', l' such that i, l', j', m form a match of 2413. Note that in this example we have $j < l'$ and $j' < l$, which is not required by the pattern.



(d) Example of case (iii): There are i', m' such that i', j, l, m' form a match of 3142. Note that in this example we have $i' <_\pi m$ and $i <_\pi m'$, which is not required by the pattern.

Figure 7.9: Pseudo-plane permutations: Whenever there is a match i, j, l, m of the pattern 2143 (see (a)), then one of the three cases (i), (ii) or (iii) must hold for π to be pseudo-plane. In all four figures, the elements i, j, l, m of the original match are drawn in blue, and elements replaced by other elements in the match (indicated by arrows, e.g., j is replaced by j' in (c)) are drawn as diamonds. These examples do not cover all possible cases, as elements are not necessarily replaced by *different* elements. For example, in example (c), $j' = j$ would be allowed if $l' < j$.

See Figure 7.9 for an illustration of pseudo-plane permutations. Note that we explicitly allow $j' = j$, $l' = l$, $i' = i$, and $m' = m$. Moreover, note that there are no constraints on the relative order of the pairs (j', l), (j, l'), $\big(\pi(i'), \pi(m)\big)$, and $\big(\pi(i), \pi(m')\big)$.

Definition 7.62. *Let $n \in \mathbb{N}$ and $\pi \in \Pi_n$ be a permutation.*
*We say that π is **pseudo-biplane** if both π and $-\pi$ are pseudo-plane.*

Note that plane permutations are in particular pseudo-plane, and hence biplane permutations are pseudo-biplane. As already mentioned, our computations show that the resulting set of structural permutations Π_n^{opt} is *exactly* the set of pseudo-biplane permutations. This is consistent with Observation 6.7 and Lemma 6.11, which predict that any complete set of sequence pairs contains all biplane sequence pairs. We summarize the empirical results:

Theorem 7.63. *Let $n \in [\![12]\!]$ and set $\mathcal{SP}_n^{opt}$ be the set of pseudo-biplane sequence pairs on $[\![n]\!]$, i.e.,*

$$\mathcal{SP}_n^{opt} := \big\{ (\pi, \sigma \circ \pi) : \pi, \sigma \in \Pi_n, \sigma \text{ is pseudo-biplane} \big\}.$$

Then, $\mathcal{SP}_n^{opt}$ is a complete set of sequence pairs with $\left|\mathcal{SP}_n^{opt}\right| = CR_n^{sym}$.
If $n \leq 8$, then we even have $\left|\mathcal{SP}_n^{opt}\right| = CR_n$, that is, $\mathcal{SP}_n^{opt}$ is a complete set of sequence pairs of minimum cardinality.

Our computational set cover experiments are restricted to $n \leq 12$ due to memory limitations. Still, we are able to check whether $\mathcal{SP}_n^{\text{opt}}$ is *complete* (but not necessarily of minimum cardinality) for even larger n: Recall that we are able to enumerate the set of normalized tight configurations up to $n = 14$, cf. Table 7.4 (page 97). Now, *while enumerating* the set of normalized tight configurations $(\mathcal{S}, \mathcal{W})$ in Algorithm 7.1, instead of storing $(\mathcal{S}, \mathcal{W})$ in a set, we check whether $(\mathcal{S}, \mathcal{W})$ is represented by a pseudo-biplane sequence pair, which in turn can be done by enumerating the set of sequence pairs representing $(\mathcal{S}, \mathcal{W})$. Using this idea, we have verified that the set of pseudo-biplane sequence pairs is indeed complete for all $n \leq 14$.

Finally, our experiments show that the number of pseudo-biplane permutations equals the number of Baxter permutations for all $n \leq 12$, which we verified to hold for all $n \leq 15$ using explicit enumeration.

We conclude this chapter by giving more specific conjectures which together imply Conjecture 7.55:

Conjecture 7.64. *Let $n \in \mathbb{N}$. Then, the number of pseudo-biplane permutations on $[\![n]\!]$ equals the number of Baxter permutations on $[\![n]\!]$.*

Conjecture 7.65. *Let $n \in \mathbb{N}$. Then, the set of pseudo-biplane sequence pairs on $[\![n]\!]$ is a complete set of sequence pairs of minimum cardinality for n.*

Summary

Axis-aligned rectangle packings can be characterized by the set of spatial relations that hold for pairs of rectangles (west, south, east, north). A representation of a packing consists of one satisfied spatial relation for each pair. We call a set of representations complete if it contains a representation of every packing of any n rectangles.

Both in theory and practice, fastest known algorithms for a large class of rectangle packing problems enumerate a complete set R of representations. The running time of these algorithms is dominated by the (exponential) size of R.

In this thesis, we have improved the best known lower and upper bounds on the minimum cardinality CR_n of complete sets of representations for n rectangles. The new upper bound implies theoretically faster algorithms for many rectangle packing problems, for example in chip design, while the new lower bound imposes a limit on the running time that can be achieved by any algorithm following this approach. The proofs of both results are based on pattern-avoiding permutations.

More precisely, the best known upper bound on CR_n is improved from $\mathcal{O}\Big(\frac{n!}{n^{4.5}} \cdot 32^n\Big)$ to $\mathcal{O}\Big(\frac{n!}{n^6} \cdot (\frac{11+5\sqrt{5}}{2})^n\Big)$, where $\frac{11+5\sqrt{5}}{2} \leq 11.091$. The previously best known lower bound of $n! \cdot 2^{n-1}$ is improved to $\Omega\Big(\frac{n!}{n^4} \cdot (4+2\sqrt{2})^n\Big)$, where $4+2\sqrt{2} \geq 6.828$.

Finally, we have empirically computed the minimum cardinality of complete sets of representations for small n. Our computations directly suggest two conjectures, connecting well-known Baxter permutations with the set of permutations avoiding an apparently new pattern, which in turn seem to generate complete sets of representations of minimum cardinality. Together, these conjectures would imply $CR_n = \Theta\Big(\frac{n!}{n^4} \cdot 8^n\Big)$.

Notation

$[\![n]\!]$	The first n integers: $[\![n]\!] := \{1, \ldots, n\}$.
id_S	Identity function on S: $\mathrm{id}_S \colon S \to S$ with $\mathrm{id}_S(i) = i$.
S^2	Ordered pairs of elements in a set S: $S^2 := \big\{ (i,j) : i,j \in S \big\}$.
${}^2S$	Ordered different-element pairs: ${}^2S := \big\{ (i,j) \in S^2 : i \neq j \big\}$.
$\overleftarrow{Q}$	Reversed relation of Q: $\overleftarrow{Q} := \big\{ (j,i) : (i,j) \in Q \big\}$.
$\mathrm{sym}(Q)$	Symmetric closure of relation Q: $\mathrm{sym}(Q) := Q \cup \overleftarrow{Q}$.
$\mathrm{tr}(Q)$	Transitive closure of relation Q.
$S + i$	The set S together with the element i: $S + i := S \cup \{ i \}$.
$S - i$	The set S without the element i: $S - i := S \setminus \{ i \}$.
$V(G)$	Vertices of graph G.
$E(G)$	Edges of graph G.
$\delta^-(v)$	Set of entering edges of vertex v in a directed graph.
$\delta^+(v)$	Set of leaving edges of vertex v in a directed graph.
$\Gamma_G(v)$	Set of neighbors of a vertex v in an undirected graph G.
$G + e$	Graph plus a new edge: $G + e := (V(G), E(G) + e)$.
$G - e$	Graph without the edge e: $G - e := (V(G), E(G) - e)$.
$G + F$	Graph plus a set of edges $G + F := (V(G), E(G) \cup F)$.
$G - F$	Graph without a set of edges $G - F := (V(G), E(G) \setminus F)$.
$\mathrm{tr}(G)$	Transitive closure of a directed graph G.
Π_n	Set of permutations $\pi \colon [\![n]\!] \to [\![n]\!]$.
$\mathcal{SP}_n$	Set of sequence pairs (π, ρ) on $[\![n]\!]$: $\mathcal{SP}_n := \Pi_n^2$.
$r_{\pi,\rho}$	Representation of a sequence pair (π, ρ), cf. Definition 4.1.
$\mathrm{struc}(\pi, \rho)$	Structural permutation of a sequence pair: $\mathrm{struc}(\pi, \rho) := \rho \circ \pi^{-1}$.
CR_n	Minimum cardinality of a complete set of representations.
$Plane_n$	Number of plane permutations, cf. Theorem 3.10 and Table 3.3.
$Biplane_n$	Number of biplane permutations, cf. Theorem 3.15 and Table 3.3.
$Baxter_n$	Number of Baxter permutations, cf. Theorem 3.17 and Table 3.3.
$<_\pi$	Total strict order induced by $\pi \in \Pi_n$: $i <_\pi j \iff \pi(i) < \pi(j)$.

$\big(\mathcal{S}_{\pi,\rho}, \mathcal{W}_{\pi,\rho}\big)$	Complementary biorder of a seq. pair (π, ρ), cf. Definition 4.9.
$(\mathcal{S}_P, \mathcal{W}_P)$	Biorder (or configuration) of a placement P, cf. Definition 4.6.
Q_I	Interval order of an interval placement I, cf. Definition 7.2.
G_{SW}	South-west digraph of a biorder $(\mathcal{S}, \mathcal{W})$, cf. Definition 4.11.
G_{SE}	South-east digraph of a biorder $(\mathcal{S}, \mathcal{W})$, cf. Definition 4.11.
$G_{\mathrm{SW}+}$	Augmented south-west digraph, cf. Definition 5.1.
$G_{\mathrm{SE}+}$	Augmented south-east digraph, cf. Definition 5.1.
G_Q	Interval constraint graph of a relation Q, cf. Definition 7.9.
G_π	Digraph of a permutation $\pi \in \Pi_n$, cf. Definition 3.6.

Index

Bibliography

[ABP06] E. Ackerman, G. Barequet, and R. Y. Pinter. “A bijection between permutations and floorplans, and its applications”. In: *Discrete Applied Mathematics* 154.12 (2006), pp. 1674–1684 (cit. on p. 22).

[Asi+13] A. Asinowski, G. Barequet, M. Bousquet-Mélou, T. Mansour, and R. Y. Pinter. “Orders induced by segments in floorplans and (2-14-3, 3-41-2)-avoiding permutations”. In: *Electronic Journal of Combinatorics* 20.2 (2013) (cit. on pp. 30, 31, 53).

[Bar+14] L. Barth, S. I. Fabrikant, S. G. Kobourov, A. Lubiw, M. Nöllenburg, Y. Okamoto, S. Pupyrev, C. Squarcella, T. Ueckerdt, and A. Wolff. “Semantic word cloud representations: hardness and approximation algorithms”. In: *Proceedings of the 11th Latin American Symposium on Theoretical Informatics*. LATIN (Montevideo, Uruguay). 2014, pp. 514–525 (cit. on p. 2).

[Bax64] G. Baxter. “On fixed points of the composite of commuting functions”. In: *Proceedings of the American Mathematical Society* 15.6 (1964), pp. 851–855 (cit. on p. 31).

[BB07] M. Bousquet-Mélou and S. Butler. “Forest-like permutations”. In: *Annals of Combinatorics* 11.3-4 (2007), pp. 335–354 (cit. on pp. 27, 29, 36).

[BCR80] B. S. Baker, E. G. Coffman Jr., and R. L. Rivest. “Orthogonal packings in two dimensions”. In: *SIAM Journal on Computing* 9.4 (1980), pp. 846–855 (cit. on p. 2).

[Bel58] R. Bellman. “On a routing problem”. In: *Quarterly of Applied Mathematics* (16 1958), pp. 87–90 (cit. on pp. 65, 67).

[BGRR18] M. Bouvel, V. Guerrini, A. Rechnitzer, and S. Rinaldi. "Semi-Baxter and strong-Baxter: two relatives of the Baxter sequence". In: *SIAM Journal on Discrete Mathematics* 32.4 (2018), pp. 2795–2819 (cit. on p. 29).

[BHHO15] U. Brenner, A. Hermann, N. Hoppmann, and P. Ochsendorf. "BonnPlace: a self-stabilizing placement framework". In: *Proceedings of the 2015 International Symposium on Physical Design*. ISPD (Monterey, California, USA). 2015, pp. 9–16 (cit. on p. 5).

[BL74] M. S. Bazaraa and R. W. Langley. "A dual shortest path algorithm". In: *SIAM Journal on Applied Mathematics* 26.3 (1974), pp. 496–501 (cit. on p. 65).

[Boo18] *Boost C++ Libraries*. Version 1.64. Available at `https://boost.org`. 2018 (cit. on p. 98).

[BSV08] U. Brenner, M. Struzyna, and J. Vygen. "BonnPlace: placement of leading-edge chips by advanced combinatorial algorithms". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27.9 (2008), pp. 1607–1620 (cit. on p. 5).

[BV04] U. Brenner and J. Vygen. "Legalizing a placement with minimum total movement". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23.12 (2004), pp. 1597–1613 (cit. on pp. 2, 6).

[BV08] U. Brenner and J. Vygen. "Analytical methods in placement". In: *Handbook of Algorithms for Physical Design Automation*. Ed. by C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar. CRC Press, 2008 (cit. on p. 7).

[Cal06] D. Callan. "A combinatorial interpretation of the eigensequence for composition". In: *Journal of Integer Sequences* 9.1 (2006) (cit. on p. 25).

[CC08] T.-C. Chen and Y.-W. Chang. "Packing floorplan representations". In: *Handbook of Algorithms for Physical Design Automation*. Ed. by C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar. CRC Press, 2008 (cit. on p. 21).

[CCWW00] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, and S.-W. Wu. "B*-trees: a new representation for non-slicing floorplans". In: *Proceedings of the 37th Design Automation Conference*. DAC (Los Angeles, California, United States). 2000, pp. 458–463 (cit. on pp. 3, 21).

[CFS70] A. V. Cabot, R. L. Francis, and M. A. Stary. "A network flow solution to a rectilinear distance facility location problem". In: *AIIE Transactions* 2.2 (1970), pp. 132–141 (cit. on pp. 3, 10).

[CGHK78] F. R. K. Chung, R. L. Graham, V. E. Hoggatt Jr., and M. Kleiman. "The number of Baxter permutations". In: *Journal of Combinatorial Theory, Series A* 24.3 (1978), pp. 382–394 (cit. on p. 31).

[Cod60] E. F. Codd. "Multiprogram scheduling: parts 1 and 2. introduction and theory". In: *Communications of the ACM* 3.6 (1960), pp. 347–350 (cit. on p. 2).

[Cun76] W. H. Cunningham. "A network simplex method". In: *Mathematical Programming* 11.1 (1976), pp. 105–116 (cit. on p. 10).

[Dij59] E. W. Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1.1 (1959), pp. 269–271 (cit. on p. 65).

[DLMT08] J. Dai, W. Lin, R. Moorthy, and C.-P. Teo. "Berth allocation planning optimization in container terminals". In: *Supply Chain Analysis.* Springer, 2008, pp. 69–104 (cit. on pp. 2, 3).

[DM41] B. Dushnik and E. W. Miller. "Partially ordered sets". In: *American Journal of Mathematics* 63.3 (1941), pp. 600–610 (cit. on pp. 38, 45).

[Eng13] C. Engels. "Algorithms for Macro Placement". Master's thesis. Research Institute for Discrete Mathematics, University of Bonn, 2013 (cit. on pp. 10, 12).

[FHS16] J. Funke, S. Hougardy, and J. Schneider. "An exact algorithm for wirelength optimal placements in VLSI design". In: *Integration, the VLSI Journal* 52 (2016), pp. 355–366 (cit. on pp. 3, 8, 10, 19, 44).

[Fis70] P. C. Fishburn. "Intransitive indifference with unequal indifference intervals". In: *Journal of Mathematical Psychology* 7.1 (1970), pp. 144–149 (cit. on p. 64).

[For56] L. R. Ford Jr. *Network flow theory.* Tech. rep. P-923. Rand Corporation, Santa Monica, 1956 (cit. on pp. 65, 67).

[FS97] S. P. Fekete and J. Schepers. "A new exact algorithm for general orthogonal d-dimensional knapsack problems". In: *Proceedings of the 5th Annual European Symposium on Algorithms.* ESA (Graz, Austria). 1997, pp. 144–156 (cit. on p. 66).

[Fun11] J. Funke. "Netzlängenoptimale Platzierung von VLSI-Chips". German. Diploma thesis. Research Institute for Discrete Mathematics, University of Bonn, 2011 (cit. on pp. 10, 12).

[Fur70] M. E. Furman. "Application of a method of rapid multiplication of matrices to the problem of finding the transitive closure of a graph". In: *Doklady Akademii Nauk*. Vol. 194. 3. Russian Academy of Sciences. 1970, pp. 524–524 (cit. on p. 71).

[FW91] M. Formann and F. Wagner. "A packing problem with applications to lettering of maps". In: *Proceedings of the Seventh Annual Symposium on Computational Geometry*. SCG (North Conway, New Hampshire, USA). 1991, pp. 281–288 (cit. on p. 2).

[GCY99] P.-N. Guo, C.-K. Cheng, and T. Yoshimura. "An O-Tree representation of non-slicing floorplan and its applications". In: *Proceedings of the 36th Design Automation Conference*. DAC (New Orleans, Louisiana, United States). 1999, pp. 268–273 (cit. on pp. 3, 21).

[GG65] P. C. Gilmore and R. E. Gomory. "Multistage cutting stock problems of two and more dimensions". In: *Operations Research* 13.1 (1965), pp. 94–120 (cit. on p. 2).

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractibility: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979 (cit. on p. 2).

[GN72] R. S. Garfinkel and G. L. Nemhauser. *Integer Programming*. Wiley, 1972 (cit. on p. 101).

[HKRV11] S. Held, B. Korte, D. Rautenbach, and J. Vygen. "Combinatorial optimization in VLSI design". In: *Combinatorial Optimization: Methods and Applications*. 2011, pp. 33–96 (cit. on pp. 2, 3, 5).

[HNR68] P. E. Hart, N. J. Nilsson, and B. Raphael. "A formal basis for the heuristic determination of minimum cost paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107 (cit. on p. 65).

[Hod82] T. J. Hodgson. "A combined approach to the pallet loading problem". In: *IIE Transactions* 14.3 (1982), pp. 175–182 (cit. on p. 2).

[Hon+04] X. Hong, S. Dong, G. Huang, Y. Cai, C.-K. Cheng, and J. Gu. "Corner block list representation and its application to floorplan optimization". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 51.5 (2004), pp. 228–233 (cit. on p. 21).

[HRT15] J. Holm, E. Rotenberg, and M. Thorup. "Planar reachability in linear space and constant time". In: *Proceedings of the 56th Annual Symposium on Foundations of Computer Science*. FOCS (Berkeley, California, USA). 2015, pp. 370–389 (cit. on p. 45).

[IN06] T. Ibaraki and K. Nakamura. “Packing problems with soft rectangles”. In: *Proceedings of the Third International Workshop on Hybrid Metaheuristics*. HM (Gran Canaria, Spain). 2006, pp. 13–27 (cit. on p. 20).

[Jer85] M. Jerrum. *Complementary partial orders and rectangle packing*. Tech. rep. University of Edinburgh, Department of Computer Science, 1985 (cit. on pp. 4, 20, 21, 33, 34, 37, 38, 42).

[Kah62] A. B. Kahn. “Topological sorting of large networks”. In: *Communications of the ACM* 5.11 (1962), pp. 558–562 (cit. on p. 104).

[Kam75] T. Kameda. “On the vector representation of the reachability in planar directed graphs”. In: *Information Processing Letters* 3.3 (1975), pp. 75–77 (cit. on pp. 45, 46).

[Knu68] D. E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*. Addison-Wesley, 1968 (cit. on p. 25).

[KRV07] B. Korte, D. Rautenbach, and J. Vygen. “BonnTools: mathematical innovation for layout and timing closure of systems on a chip”. In: *Proceedings of the IEEE* 95.3 (2007), pp. 555–572 (cit. on p. 5).

[KV08] B. Korte and J. Vygen. “Combinatorial problems in chip design”. In: *Building Bridges*. Springer, 2008, pp. 333–368 (cit. on pp. 3, 19).

[KV18] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. 6th ed. Springer, 2018. 698 pp. (cit. on pp. 16, 17, 20, 65, 66, 68, 98, 101).

[LC04] J.-M. Lin and Y.-W. Chang. “TCG-S: orthogonal coupling of P*-admissible representations for general floorplans”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23.6 (2004), pp. 968–980 (cit. on p. 21).

[LC05] J.-M. Lin and Y.-W. Chang. “TCG: a transitive closure graph-based representation for general floorplans”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 13.2 (2005), pp. 288–292 (cit. on p. 21).

[LCL03] J.-M. Lin, Y.-W. Chang, and S.-P. Lin. “Corner sequence – a P-admissible floorplan representation with a worst case linear-time packing scheme”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 11.4 (2003), pp. 679–686 (cit. on p. 21).

[LeG14] F. Le Gall. “Powers of tensors and fast matrix multiplication”. In: *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation.* ISSAC (Kobe, Japan). 2014, pp. 296–303 (cit. on p. 71).

[LLQ04] S. C. Li, H. W. Leong, and S. K. Quek. “New approximation algorithms for some dynamic storage allocation problems”. In: *Proceedings of the 10th International Computing and Combinatorics Conference.* COCOON (Jeju Island, South Korea). 2004, pp. 339–348 (cit. on p. 2).

[Lov79] L. Lovász. *Combinatorial Problems and Exercises.* North Holland Publishing Company, 1979 (cit. on p. 25).

[Lu+14] J. Lu, P. Chen, C.-C. Chang, L. Sha, D. J.-H. Huang, C.-C. Teng, and C.-K. Cheng. “ePlace: electrostatics based placement using Nesterov’s method”. In: *Proceedings of the 51st Design Automation Conference.* DAC (San Francisco, California, USA). 2014, Art. No. 121 (cit. on p. 7).

[Mac15] P. A. MacMahon. *Combinatory Analysis.* Vol. 1. Cambridge University Press, 1915 (cit. on p. 25).

[Mad79] O. B. G. Madsen. “Glass cutting in a small firm”. In: *Mathematical Programming* 17.1 (1979), pp. 85–90 (cit. on p. 2).

[McI96] R. J. McIntosh. “An asymptotic formula for binomial sums”. In: *Journal of Number Theory* 58.1 (1996), pp. 158–172 (cit. on p. 29).

[MFNK96] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. “VLSI module placement based on rectangle-packing by the sequence-pair”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 15.12 (1996), pp. 1518–1524 (cit. on pp. 2, 20, 21, 33–35, 42).

[MFWK97] H. Murata, K. Fujiyoshi, T. Watanabe, and Y. Kajitani. “A mapping from sequence-pair to rectangular dissection”. In: *Proceedings of the 2nd Asia and South Pacific Design Automation Conference.* ASP-DAC (Makuhari, Japan). 1997, pp. 625–633 (cit. on pp. 22, 23).

[Mic15] F. Michaelis. “Verbesserungen von Algorithmen für die Platzierung von Macros”. German. Bachelor’s thesis. Research Institute for Discrete Mathematics, University of Bonn, 2015 (cit. on p. 10).

[Moo59] E. F. Moore. “The shortest path through a maze”. In: *Proceedings of the International Symposium on Switching Theory.* Vol. II. 1959, pp. 285–292 (cit. on pp. 65, 67).

[Mun71] I. Munro. “Efficient determination of the transitive closure of a directed graph”. In: *Information Processing Letters* 1.2 (1971), pp. 56–58 (cit. on p. 71).

[NFMK96] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani. “Module placement on BSG-structure and IC layout applications”. In: *Proceedings of the 1996 IEEE/ACM International Conference on Computer-Aided Design*. ICCAD (San Jose, California, USA). 1996, pp. 484–491 (cit. on pp. 20, 21).

[Och19] P. Ochsendorf. “Timing-Driven Macro Placement”. PhD thesis. Research Institute for Discrete Mathematics, University of Bonn, 2019 (cit. on pp. 2, 3, 10).

[OTT91] H. Onodera, Y. Taniguchi, and K. Tamaru. “Branch-and-bound placement for building block layout”. In: *Proceedings of the 28th Design Automation Conference*. DAC (San Francisco, California, USA). 1991, pp. 433–439 (cit. on p. 3).

[Pud08] L. K. Pudwell. “Enumeration Schemes for Pattern-Avoiding Words and Permutations”. PhD thesis. Rutgers University, 2008 (cit. on p. 25).

[SC03] Z. C. Shen and C. C. N. Chu. “Bounds on the number of slicing, mosaic, and general floorplans”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 22.10 (2003), pp. 1354–1361 (cit. on pp. 4, 19, 23).

[Sil11] J. Silvanus. “Rechteckpackungen”. German. Bachelor’s thesis. Research Institute for Discrete Mathematics, University of Bonn, 2011 (cit. on pp. 4, 19).

[SKM03] K. Sakanushi, Y. Kajitani, and D. P. Mehta. “The quarter-state-sequence floorplan representation”. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 50.3 (2003), pp. 376–386 (cit. on p. 21).

[Slo18] N. J. A. Sloane. “The On-Line Encyclopedia of Integer Sequences”. Available at `https://oeis.org`. 2018 (cit. on pp. iii, 29, 31).

[SO80] A. A. Szepieniec and R. H. J. M. Otten. “The genealogical approach to the layout problem”. In: *Proceedings of the 17th Design Automation Conference*. DAC (Minneapolis, Minnesota, USA). 1980, pp. 535–542 (cit. on p. 21).

[SS85] R. Simion and F. W. Schmidt. “Restricted permutations”. In: *European Journal of Combinatorics* 6.4 (1985), pp. 383–406 (cit. on p. 25).

[SSJ08] P. Spindler, U. Schlichtmann, and F. M. Johannes. “Kraftwerk2 – a fast force-directed quadratic placement approach using an accurate net model”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27.8 (2008), pp. 1398–1411 (cit. on p. 7).

[SV17] J. Silvanus and J. Vygen. “Few sequence pairs suffice: representing all rectangle placements”. Submitted. 2017. arXiv: `1708.09779 [math.CO]` (cit. on p. 4).

[Tak00] T. Takahashi. “A new encoding scheme for rectangle packing problem”. In: *Proceedings of the 5th Asia and South Pacific Design Automation Conference*. ASP-DAC (Yokohama, Japan). 2000, pp. 175–178 (cit. on pp. 3, 21).

[Wes95] J. West. “Generating trees and the Catalan and Schröder numbers”. In: *Discrete Mathematics* 146.1-3 (1995), pp. 247–262 (cit. on p. 25).

[Woc17] D. M. Wochnik. “Verbesserte Algorithmen für die Platzierung von Makros”. German. Master’s thesis. Research Institute for Discrete Mathematics, University of Bonn, 2017 (cit. on pp. 2, 3, 10).

[Xu+17] B. Xu, S. Li, X. Xu, N. Sun, and D. Z. Pan. “Hierarchical and analytical placement techniques for high-performance analog circuits”. In: *Proceedings of the 2017 International Symposium on Physical Design (ISPD)*. ISPD. 2017, pp. 55–62 (cit. on p. 3).

[YCCG03] B. Yao, H. Chen, C.-K. Cheng, and R. Graham. “Floorplan representations: complexity and connections”. In: *ACM Transactions on Design Automation of Electronic Systems* 8.1 (2003), pp. 55–80 (cit. on pp. 21, 22).

[YCS03] E. F. Y. Young, C. C. N. Chu, and Z. C. Shen. “Twin binary sequences: a non-redundant representation for general non-slicing floorplan”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 22.4 (2003), pp. 457–469 (cit. on pp. 21, 23).

[You08] E. F. Y. Young. “Floorplan representations”. In: *Handbook of Algorithms for Physical Design Automation*. Ed. by C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar. CRC Press, 2008 (cit. on p. 21).

[ZSJK02] C. Zhuang, K. Sakanushi, L. Jin, and Y. Kajitani. “An enhanced Q-sequence augmented with empty-room-insertion and parenthesis trees”. In: *Proceedings of the 2002 Conference on Design, Automation and Test in Europe*. DATE (Paris, France). 2002, p. 61 (cit. on p. 23).