Points and Combinatorics

Oswin Aichholzer, Franz Aurenhammer, Hannes Krasser

Institute for Theoretical Computer Science
Graz University of Technology
Graz, Austria
e-mail: {oaich,auren,hkrasser}@igi.tu-graz.ac.at

Which point sets exist, anyway?

This intuitive question sketches the topic of the article at hands. Its relevance is apparent: a configuration of n points is the underlying structure for countless problems in computational and combinatorial geometry.

In fact, a point set in the Euclidean plane is among the simplest geometric objects that lead to non-trivial questions – in a geometrical, combinatorial, and algorithmic sense. Not surprisingly, most basic data structures in computational geometry have first been developed for point sets, and have been later generalized to more general objects, like line segments, circles, polygons, etc. Examples include geometric search trees, convex hulls, Voronoi diagrams (an accompaning article is devoted to this versatile structure), and many others. During the past two years, a good share of effort of our research group went into the systematic investigation of point sets of small size. Let us report on this original and challenging topic at this place.

> *"There is a point in pointing out points, in order to order them by order type."*
> The Realizing Enumerators O.F.&H.

Of course, there exist infinitely many different sets of n points, for a given number n. However, quite a large class of problems is determined already by the *combinatorial* properties of an n-point set, rather than by its metric properties. To be more specific, look at all the $\binom{n}{2}$ straight-line segments spanned by a given n-point set, as in Figure 1. The way these line segments cross each other turns out to be important: point sets with identical crossing properties give rise to equivalent geometric structures. This is valid for many popular structures, like spanning trees, triangulations, polygonalizations, so-called k-sets, and many more.

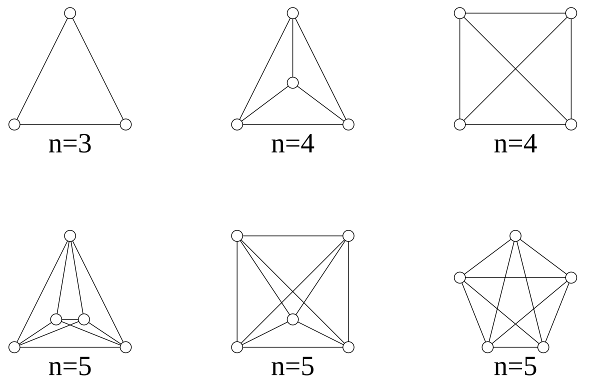

Figure 1: Inequivalent sets of small size. There is only one (combinatorial) way to place 3 points, two ways to place 4 points, and three to place 5. The situation changes drastically for larger size.

Several of these structures lead to notoriously hard problems. To gain insight, the study of typical (or extremal) instances is very fruitful. E.g., detecting a counter-example prevents us from constructing hopeless proofs. Conversely, the non-existence of counter-examples of small size gives some evidence for the truth of a conjecture. This motivates the complete enumeration of all problem instances of small size. In our case, this means investigating all *inequivalent* sets of points, where equivalence is with respect to their crossing properties. Let us ask again and more specifically, "Which inequivalent sets of, say 10, points do exist?"

The answer is surprisingly difficult, for two major reasons. Firstly, the number of inequivalent point sets of size 10 is already in the millions (14 309 547, to be precise). Secondly, there seems to be no simple way to enumerate all these sets. Each increase in set size gives rise to types which cannot be obtained directly from sets of smaller size. This may explain why it took until recently that an enumeration has been successful. We are proud to be the first having established (see [3]) and made available a complete and reliable data base[1] of all inequivalent 10-point sets (in December 2000). Meanwhile – in fact a few weeks ago – a further milestone was put. We completed the generation of all inequivalent *11-point sets* (in implicit form). The present article describes the inherent difficulties of such a project, along with numerous applications and results we obtained from our point set data base.

The theory behind

Order types An appropriate tool to encode equivalence of point configurations has been developed by combinatorial geometers [15] quite a while ago. The *order type* of a set $\{p_1, ..., p_n\}$ of points is a mapping that assigns to each (ordered) index triple i, j, k the orientation – clockwise or counter-clockwise – of the point triple p_i, p_j, p_k. (We require that no three points lie on a common straight line.) Two point sets S_1 and S_2 are considered *equivalent* if they exhibit the same order types. That is, there is a one-to-one correspondence between S_1 and S_2 such that each triple in S_1 agrees in orientation with the corresponding triple in S_2. It is not hard to see that two line segments spanned by S_1 cross if and only if their counterparts in S_2 do so. Our aim is to enumerate all order types of given size n, for $n \leq 11$.

Duality We base our approach on a well-known *duality* between points and straight lines in the plane. By this duality, a set of points maps to a *line arrangement*, i.e., a dissection of the plane induced by a set of n straight lines. As no direct way to enumerate line arrangements is known, we first produce all (non-isomorphic) arrangements of so-called pseudolines. *Pseudolines* are simple curves that pairwise cross at exactly one point, just as non-parallel straight lines do. Handling pseudolines is relatively easy in view of their equivalent description by *wiring diagrams*; consult Figure 2. We can read off a corresponding *pseudo order type* from each wiring diagram: the order in which the wires cross each other determines the orientations for all index triples. Back in the primal setting, where each wire potentially corresponds to a point, this leads to a list of candidates guaranteed to contain all different order types.

Realizability We are left with the problem of identifying all the *realizable* order types in the obtained list, that is, those for which corresponding point coordinates do exist. Here we enter the realm of *oriented matroids*, an axiomatic (combinatorial) abstraction of geometric objects, introduced in the late 1970s. As a known phenomenon, a wiring diagram need not be *stretchable* to straight. In fact, there exist non-stretchable diagrams already[2] for 8 wires [10]. As a consequence, our candidate list will contain *non-realizable* pseudo order types for size $n \geq 9$. Moreover, even if realizability has been recognized for a particular candidate, how can we find a corresponding point set?

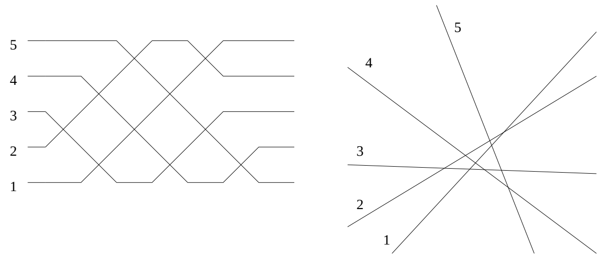

Figure 2: A wiring diagram that can be stretched

Projectivity Surprisingly at first glance, the situation gets easier (conceptually and computationally) on the 3D sphere, that is, in the projective setting. To visualize *projective order types*, imagine all possible ways n stars can be arranged in the sky. Watching stars from a fast moving space vessel changes their geometry, but not their projective order type. Unlike before, order types now directly correspond to arrangements of great circles by duality (and isomorphism classes of pseudo-circle arrangements coincide with reorientation classes of rank 3 oriented matroids). Moreover, the desired order types in the plane can be nicely grouped into *projective classes*, and in every class either each or no order type is realizable.

Basic strategy In a nutshell, our approach works as follows. It produces a candidate list of planar pseudo order types (from wiring diagrams), clusters them into projective classes, and tries to decide realizability for each class. A collection of all yet unrealized projective classes is maintained.

For problem sizes $n = 9$ and $n = 10$, the process could be terminated after the collection had shrunk to 1 and 242 members, respectively. We know from the literature that exactly that many classes are non-realizable [10, 14]. (The unique class for $n = 9$ derives

[1] `http://www.igi.TUGraz.at/oaich/triangulations/ordertypes.html`

[2] Two of the authors won a pizza due to this peculiarity; the third was paying ...

from the well-known Pappus theorem.) For the case $n = 11$, however, where information about projective classes is completely lacking, a means for disproving realizability had to be applied.

Deciding realizability (for order types in particular and oriented matroids in general) is an intriguing problem, which is known to be NP-hard. Several heuristics have been developed; see [14] and references therein. For our case, a singly exponential algorithm exists but turned out to be too slow, already for problem size 10. Instead, we used a combination of methods with success. Even for size 11, we could decide upon the status of *all* the projective classes, i.e., distinguish the realizables from the non-realizables.

Coordinate recovering We applied an obvious though powerful *insertion strategy* in a first step: obtain realizations of size n from realizations of size $n-1$. Suppose that, for each (realizable) order type T of size $n-1$, a corresponding point set $S(T)$ is available. Consider the arrangement $A(T)$ formed by $\binom{n-1}{2}$ lines, each passing through a pair of points in $S(T)$. Point sets of size n are now generated from $S(T)$, by placing an additional point in a cell of $A(T)$, for all possible cells. Carrying out this process for all existing order types of size $n-1$ leads to realizations for a majority of order types of size n.

However, the *geometry* of an $(n-1)$-point set $S(T)$ critically affects the *set* of order types of size n which actually get realized. (This principal difficulty makes order type enumeration a hard problem.) To increase effectiveness, we restarted the insertion method after random (but order type preserving) perturbations of the $(n-1)$-point sets.

All but a vanishing fraction of the realizable order types could be detected in this way, in fact, over 99.99% even for $n = 11$. The rest (the 'hard core' which contains the unrealizables, of course) we tried to realize directly – starting from scratch with tailor-made *simulated annealing* strategies.

Detecting non-realizability Clearly, certain pseudo order types notoriously denied themselves to realization. An obvious method to disprove their realizability is by detecting some non-realizable sub-order type as a *witness*. Though being powerful like the insertion strategy before (as being its reverse), the method has similar drawbacks: not all cases are covered. We therefore had to unmask certain non-realizables by different means. Following [10], we utilized the so-called *Grassmann-Plücker* relations, a determinant relation fulfilled by each 5-tuple of points in the plane. For a given pseudo order type of size n, these $\binom{n}{5}$ relations lead to a linear program, whose infeasibility disproves its realizability.

> *" ... one notices a peculiar asymmetry between the realizable and the non-realizable cases. There exists a simple and systematic representation for a realizability proof, namely a sample point configuration. Even the longest and most complicated derivation of a realizability proof can be discarded once we have found some coordinate matrix. This means that we only need a few real algebraic numbers to finally encode the proof."*
> From: Oriented matroids [10].

The data base

Output Table 1 gives a quick overview of our results (in bold numbers), which significantly extend all relevant known results [14]. The tremendous growth of order types, especially in the planar case, becomes apparent. By the nature of our approach, we computed a combinatorial description for all the objects counted in Table 1, along with a geometric representation if the object is realizable. An obvious exception are the size-11 planar order types which – unless we are willing to spend some 160 CDs – have to be represented implicitly (by their projective classes).

Size 10 To get an impression of the computation time required, we give some details for the case $n = 10$. The generation of all pseudo order types, plus partial realization using insertion, took 36 hours on a 500 MHz Pentium III. Some 200 000 pseudo order types were left unrealized. Most of the corresponding projective classes got some member realized, however, and could be completed quickly by applying a projection technique. Only 251 projective classes of undecided status remained. For these, we had to invoke our simulated annealing routines, as we had no information on which are the 242 classes known to be non-realizable. We were successful for 9 classes within 60 hours which completed this task.

Size 11 As expected, the situation turned out to be much more complex for $n = 11$. Although we further improved the implementation, the total time spent on *projective classes* was more than three months on several processors. Their planar members

n	3	4	5	6	7	8	9	10	11
Projective Pseudo Order Types	1	1	1	4	11	135	4 382	312 356	**41 848 591**
– *thereof non-realizable*							1	242	**155 214**
= *Projective Order Types*	1	1	1	4	11	135	4 381	312 114	**41 693 377**
Planar Pseudo Order Types	1	2	3	16	135	3 315	**158 830**	**14 320 182**	**2 343 203 071**
– *thereof non-realizable*							**13**	**10 635**	**8 690 164**
= Planar Order Types	1	2	3	16	135	3 315	**158 817**	**14 309 547**	**2 334 512 907**

Table 1: Numbers of different order types of size n

(which we are mainly interested in) can be derived on-line and independently for each particular class, and thus need not be stored. We point out once more that the time-consuming routines (simulated annealing and Grassmann-Plücker) were used only in 'emergency cases' where the insertion technique (and its reverse) were no help any more.

Whenever computing realizations, care was taken to avoid large coordinates. In addition, the simulated annealing routines were used to post-process the point sets. In their final form, our point sets enjoy several user-friendly properties, including a 16 bit integer representation. This should be contrasted with known negative results on the efficient grid embeddability of order types [16].

Reliability Despite the theoretical correctness of the algorithms used, the question arises of how reliable are our implementations. The wiring diagram based calculation of planar pseudo order types, including their grouping into projective classes, is of purely combinatorial nature. Evidence of correctness is gained from the correctly computed total numbers of projective pseudo order types for $n \leq 10$ (which have been known before; see Table 1).

The critical part are the geometric routines for both, proving and disproving realizability. As realizabilty is witnessed by the point sets (in nice representation), it can be re-checked in a reliable way. Non-realizability, on the other hand, may have no direct witness. We therefore put particular emphasis on the clean implementation of the geometric part of the Grassmann-Plücker routine, which basically is a linear program.

Applicability In summary, a data base has been obtained that is complete, user-friendly, and reliable. Small integer coordinates ensure that all calculations based on our data base are efficient and numerically stable. Due to storage limitations, the point sets for sizes 10 and 11 are not accessible on-line, but rather have been stored on CDs which are available from us upon request. We believe our data base to be of value to many researchers who wish to examine their conjectures on small point configurations.

The applicability of the data base is broad. In principle, any geometric straight-line structure is a candidate. Let us describe – for a few prominent questions – the progress achieved when using our data base. A more complete collection of our results is [5]. The concepts skipped here for space limitations include convex polygon covering, convex decomposition, pseudo-triangulations, k-sets (partitioning lines), as well as the crossing-free variants of polygonalizations (Hamiltonian cycles), spanning trees, and matchings.

Crossing numbers

The (rectilinear) *crossing number* of a graph G is the least number of edge crossings attained by a straight-line drawing of G in the plane. (To avoid trivialities, no three vertices of G are allowed to be collinear in the drawing.) It is easy to prove that drawings which differ in their numbers of crossings stem from vertex sets of different order type. Our interest is in $\overline{cr}(K_n)$, the rectilinear crossing number of the *complete graph* K_n on n vertices.

"... Our work was to bring out bricks from the ovens where they were made ... each oven was connected by rail with each store. ... trouble arose at the crossing of two rails. Here the cars jumped out, the bricks fell down; a lot of extra work and loss of time arose. Having this experience a number of times it occured to me why on earth did they build the rail system so uneconomically; minimizing the number of crossings the production could be made much more economical."
From [18], Turan's brick factory problem

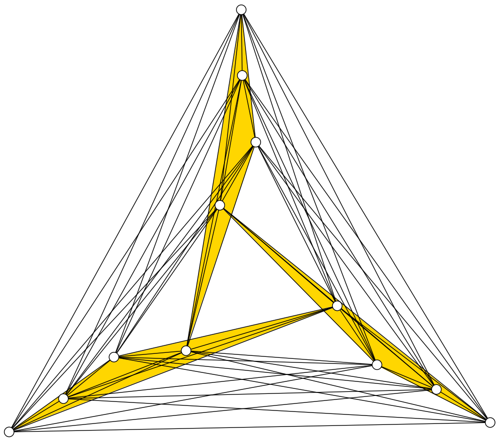

Figure 3: Minimal drawing of K_{12}

Determining $\overline{cr}(K_n)$ is commonly agreed to be a difficult task. From an algorithmic point of view, deciding whether $\overline{cr}(G) \leq k$ for a given graph G and parameter k, is NP-hard [9]. Despite the remarkable hunt for crossing numbers – initiated in [17] in the 1960s – only for very small n the exact values of $\overline{cr}(K_n)$ have been known, until recently. The instances $n \leq 9$ have been settled quite a time ago, but no progress has been made until in 2001 two groups of researchers ([11] and the authors [3], respectively) independently found $\overline{cr}(K_{10}) = 62$. In [11] the goal was reached by a purely combinatorial argument, while our result came as a byproduct of the exhaustive enumeration of all order types of size 10.

Encouraged by this achievement, we tailored our approach to generate inequivalent drawings of K_n for larger n. The effect was surprising. Not only could we completely resolve the situation for $n = 11$ (to be precise, $\overline{cr}(K_{11})$ is 102, and this value is attained by exactly 374 inequivalent drawings), we also could break new ground for $n = 12$. More specifically, $\overline{cr}(K_{12})$ is 153, and the only known drawing that attains this value is Figure 3. These findings also implied the best known lower bound on the limit

$$\lim_{n\to\infty} \overline{cr}(K_n) / \binom{n}{4} > 0.311507$$

which describes the 'asymptotic gain' when drawing K_n optimally. In fact, this is a geometric quantity of separate interest, by a connection to Silvester's four point problem [21].

For $n \geq 13$, we constructed various new drawings of K_n with few crossings, by insertion into the small optimal examples. These drawings led to improved *upper* bounds for the quantity above. The tantalizing question of determining $\overline{cr}(K_{13})$ is left open. By our calculations, the latest ra(n)ge is $\{221, 223, 225, 227, 229\}$; our conjecture is $\overline{cr}(K_{13}) = 229$.

Compatible triangulations

Can any two planar point sets (within the obvious size and convex hull restrictions) be triangulated in the same manner? Surprisingly, this basic question – which has applications in computer graphics (morphing) and in cartography (PLH maps) – is still unsettled. A *triangulation* of a set S of points is a maximal collection of non-crossing edges (line segments) spanned by S. Two point sets of the same order type allow for the same variety of triangulations.

Triangulations are flexible constructs, which play a rôle in various areas and have important applications [8]; an obvious one is mesh generation. Their appealing structural properties won our interest right away. Beside Voronoi diagrams, triangulations have been the favorite structures of our research group; 15 publications arose in the last few years. Our work on this topic is partially reported in the survey article on optimal triangulations [7].

Returning to the question above, let S_1 and S_2 be two point sets in the plane. Two triangulations T_1 of S_1 and T_2 of S_2 are called *compatible* if there is a one-to-one correspondence $\varphi : S_1 \to S_2$ with the following property: for each empty triangle ijk in T_1, the points $\varphi(i)\varphi(j)\varphi(k)$ define an empty triangle in T_2; see Figure 4. Compatible triangulations show isomorphic graphs of edges; the converse is not true, in general.

The problem of triangulating two given point sets compatibly comes in two flavors, namely where the correspondence between their points is either fixed in advance, or variable. The case of *fixed correspondence* is a known problem. In particular, compatible triangulations do not always exist [19], because edges can be forced to cross (which is prohibited in a triangulation). The related decision problem is neither known to be NP-complete nor to be polynomially solvable. However, the problem becomes easier if the two point sets represent the vertices of two simple n-gons [6], namely decidable in $O(n^2)$ time.

We asked the different question, "under which conditions *does there exist* a correspondence which allows for compatible triangulations?" Clearly, in both sets S_1 and S_2 the number n of points must be the same; as has to be the number h of *extreme* points (points on their convex hulls), by the well-known Eulerian for-

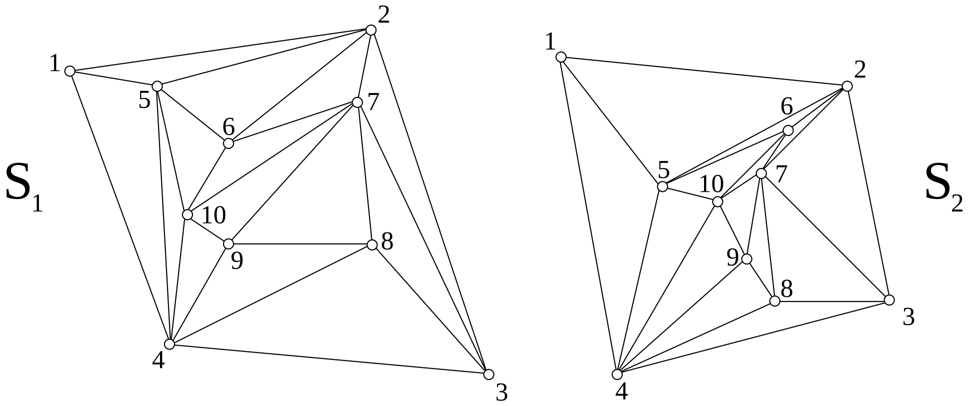

Figure 4: These two triangulations are compatible

mula $e = 3n - h - 3$ for the number of triangulation edges. Extreme points necessarily map to extreme points, and their cyclic order on the convex hulls is preserved. (More generally, for compatible triangulations any two corresponding triangles ijk and $\varphi(i)\varphi(j)\varphi(k)$ must match in orientation. That is, their point sets locally exhibit the same order type.) Finally, both point

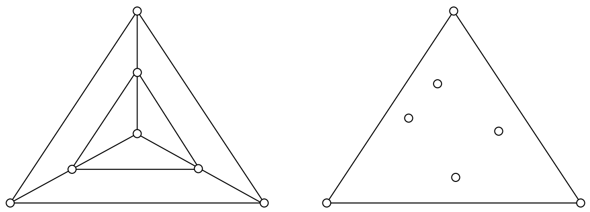

Figure 5: No compatible triangulations exist

sets have to be in *general position*, i.e., no three points on the same line are allowed. Counter-examples exist, otherwise; see Figure 5.

These observations led us to state the following conjectures (which have been included in the recently founded open problems list in computational geometry[3]).

Main Conjecture Any two n-point sets that agree in the number of extreme points, and both are in general position, admit a compatible triangulation.

The status of the following extended problem is unresolved, too.

Extended Conjecture The Main Conjecture still holds if a correspondence between the extreme points is prescibed (by a cyclically shifted labeling).

The extended conjecture is relevant for a possible induction proof, which naturally will lead to assuming the stronger version as a hypothesis.

In [2] we undertook first steps towards proving the conjectures. For instance, they are true if the number of interior points in either set is three or less. Obtaining an affirmative answer would be a deep theoretical result, showing that all planar point sets of the same size and hull size are 'topologically equivalent' in this very sense. The result, if true, gains in importance in view of the huge number of different order types for n points; cf. Table 1.

In order to encourage and support future research on this topic, we are glad to announce the reward below. (A related playing tool[4] may turn out helpful.)

ATTENTION — EuroPrizes!

€ 30 for proving the Conjectures
+ an original Austrian Wiener Schnitzel ...
€ 7 for a counter-example to the Main
€ 5 for a counter-example to the Extended

No reward for proving the Conjectures and providing a counter-example at the same time!

[3] `http://www.cs.smith.edu/~orourke/TOPP/node39.html#Problem.38`

[4] `http://www.igi.TUgraz.at/oaich/triangulations/compatible.html`

Using our data base, we tested the Extended Conjecture for small point set sizes, and could verify it for up to 8 points. The lack of good heuristics for checking compatibility made the calculations time-consuming, however; they are still incomplete for size 9. Determining the theoretical complexity of computing compatible triangulations, if they exist, is an open problem. On the other hand, adding a small number of extraneous points (so-called Steiner points), which can be found efficiently, will always do the job [2].

Triangulation counting

Efficiently counting all possible triangulations of a given n-point set (in general position) is an interesting open problem. To give a motivation, the question relates to computing *random* triangulations – a concept of importance for testing triangulation-based algorithms.

The fastest counting method to date is based on a concept we called triangulation path [1]. Its running time still shows exponential growth, however, and computations are limited to $n \leq 40$. On the other hand, no good theoretical bounds on the number of triangulations are known. The current best maximizing examples yield $8^{n-\Theta(\log n)}$ triangulations, whereas the strongest upper bound is much larger, $59^{n-\Theta(\log n)}$; see [20].

Not even for small values of n, *exact numbers* have been known. But point sets of the same order type admit equally many triangulations, and so we could compute these numbers for $n \leq 10$. Table 2 lists their extremum values. The currently best examples for $n =$ 11 through 20 can be found on-line[5].

n	minimum	maximum
3	1	1
4	1	2
5	2	5
6	4	14
7	11	42
8	30	150
9	89	780
10	250	4550

Table 2: Extremum numbers of triangulations

So, for example, every set of 8 points must have at

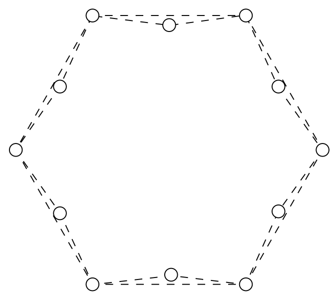

Figure 6: Double-circle configuration

least 30 triangulations, and there actually exists a set having that few triangulations. This shows that point sets in convex position, whose number of triangulations is given by the so-called Catalan numbers $C(n-2)$ (yielding 132 for $n = 8$), do not lead to the minimum. This conjecture would have been plausible, as it is true for $n \leq 7$, as well as for other structures, namely for crossing-free matchings and crossing-free spanning trees [13]. Though the conjecture has been resolved before, the results in Table 2 may be useful in proving or disproving the extremal behavior of other point configurations. For instance, we could observe that the point sets leading to the lower bounds in Table 2 obey a rather special structure. We conjecture that these 'double-circle' configurations – shown in Figure 6 – indeed yield the minimum for general n.

A legitimate question is, "How many triangulations must *every* n-point set have?" Surprisingly little is known in this respect. We therefore seek for functions $t(n)$ such that every set of n points exhibits $\Omega(t(n))$ triangulations. A quick answer is $t(n) = 2^{(n-4)/6}$, because every triangulation on n points contains at least $(n-4)/6$ edges that can be flipped (i.e., replaced by a crossing edge within the same quadrilateral) in an independent way [12]. A substantial improvement is based, among other things, on a new result on *crossing families* (pairwise crossing edges), which we found out by checking order types: every set of 10 points already spans a crossing family of size 3. (This drastically improves on the previous best bound of 37, in [22].) We exploited this result [4] to obtain the relation

$$t(n) \geq 3 \cdot t(n_1) \cdot t(n_2)$$

for $n_1 + n_2 = n + 2$. For any base α, the function $t(n) = \frac{1}{3} \cdot \alpha^{n-2}$ is a solution. To get the recurrence started, values of α as large as possible are sought by examining all instances of small constant size. This is done by utilizing Table 2, in combination with other

[5] `http://www.igi.tugraz.at/oaich/triangulations/counting/counting.html`

methods. The best value we achieved is $\alpha = 2 + \varepsilon$ for $\varepsilon < \frac{1}{80}$, that is,

$$t(n) > \frac{1}{13} \cdot (2 + \varepsilon)^n.$$

This constitutes the first bound that asymptotically beats the magic threshold 2^n.

> *The enumeration of combinatorial objects clearly is limited to small object size. Still, proving a general property may boil down to a case analysis of constant-size problems. Proofs that require a non-trivial induction base are an example ... The point is that the* quality *of the induction base affects the* asymptotics *of the result. In this way, 'constant' knowledge may well lead to insight concerning 'infinity and beyond'...*

References

[1] O.Aichholzer, *The path of a triangulation.* Proc. 15^{th} Ann. ACM Symp. on Computational Geometry, 1999, 14-23.

[2] O.Aichholzer, F.Aurenhammer, F.Hurtado, H.Krasser, *Towards compatible triangulations.* Theoretical Computer Science (invited).

[3] O.Aichholzer, F.Aurenhammer, H.Krasser, *Enumerating order types for small point sets with applications.* Proc. 17th Ann. ACM Symp. on Computational Geometry, 2001, 11-18.

[4] O.Aichholzer, F.Hurtado, M.Noy, *On the number of triangulations every planar point set must have.* Proc. 13th Ann. Canadian Conf. on Computational Geometry, 2001, 13-16.

[5] O.Aichholzer, H.Krasser, *The point set order type data base: a collection of applications and results.* Proc. 13th Ann. Canadian Conf. on Computational Geometry, 2001,17-20.

[6] B.Aronov, R.Seidel, D.Souvaine, *On compatible triangulations of simple polygons.* Computational Geometry: Theory and Applications 3 (1993), 27-35.

[7] F.Aurenhammer, Y.-F.Xu, *Optimal triangulations.* In: Encyclopedia of Optimization 4, P.M.Pardalos and C.A.Floudas (eds), Kluwer Academic Publishing, 2001, 160-166.

[8] M.Bern, D.Eppstein, *Mesh generation and optimal triangulation.* In: D.-Z.Du, F.K.Hwang (eds.), Computing in Euclidean Geometry, Lecture Notes Series in Computing 4, World Scientific, Singapore, 1995, 47-123.

[9] D.Bienstock, *Some provably hard crossing number problems.* Discrete & Computational Geometry 6 (1991), 443-459.

[10] A.Björner, M.Las Vergnas, B.Sturmfels, N.White, G.Ziegler, *Oriented Matroids.* Cambridge University Press, 1993.

[11] A.Brodsky, S.Durocher, E.Gether, *The rectilinear crossing number of K_{10} is 62.* The Electronic J. of Combinatorics 8 (2001), Research Paper 23.

[12] J.Galtier, F.Hurtado, M.Noy, S.Perennes, J.Urrutia, *Simultaneous edge flipping in triangulations.* Manuscript, Universitat Politecnica de Catalunya, Barcelona, Spain, 2000.

[13] A.García, M.Noy, J.Tejel, *Lower bounds on the number of crossing free subgraphs of K_n*, Computational Geometry: Theory and Applications 16 (2000), 211-221.

[14] J.E.Goodman, *Pseudoline arrangements.* In J.E.Goodman, J.O'Rourke (eds.), *Handbook of Discrete and Computational Geometry.* CRC Press LLC, Boca Raton, NY, 1997.

[15] J.E.Goodman, R.Pollack, *Multidimensional sorting.* SIAM J. Computing 12 (1983), 484-507.

[16] J.E.Goodman, R.Pollack, B.Sturmfels, *Coordinate representation of order types requires exponential storage.* Proc. 21^{st} Ann. ACM Sympos. Theory of Computing, 1989, 405-410.

[17] R.K.Guy, *A combinatorial problem.* Nabla (Bulletin of the Malayan Mathematical Society) 7 (1960), 68-72.

[18] R.K.Guy, *The decline and fall of Zarankiewicz's theorem.* In: Proof Techniques in Graph Theory (Proc. 2^{nd} Ann Arbor Graph Theory Conf.), 63-69, Academic Press, New York, 1969.

[19] A.Saalfeld, *Joint triangulations and triangulation maps.* Proc. 3^{rd} Ann. ACM Sympos. Computational Geometry, 1987, 195-204.

[20] F.Santos, R.Seidel, *On the number of triangulations.* Manuscript, Universität des Saarlandes, Saarbrücken, Germany, 2000.

[21] E.R.Scheinerman, H.S.Wilf, *The rectilinear crossing number of a complete graph and Sylvester's "four point problem" of geometric probability.* Amer. Math. Monthly 101 (1994), 939-943.

[22] G.Tóth, P.Valtr, *Note on Erdös-Szekeres theorem.* Discrete & Computational Geometry 19 (1998), 457-459.