Enumerating Seating Arrangements that Obey Social Distancing

George SPAHN and Doron ZEILBERGER

In fond memory of Marko Petkovsek (1955-2023)

Abstract: We illustrate the power of symbolic computation and experimental mathematics by investigating maximal seating arrangements, either on a line, or in a rectangular auditorium with a fixed number of columns but an arbitrary number of rows, that obey any prescribed set of 'social distancing' restrictions. In addition to enumeration, we study the statistical distribution of the density, and give simulation algorithms for generating them.

Preface: How it all started

It all started when we came across the delightful article [PSZ] whose motivation had nothing to do with the now fashionable *social distancing*. Their starting point was an $r \times s$ housing development with rs building lots, each with room for a house. They wanted to enumerate the number of ways (out of the total of 2^{rs} possibilities, including building nothing, and building on all the lots) for which

- No house is 'blocked from the sun' (see below).

- You can't build any house on a currently empty lot without violating this condition.

They were not only interested in the number of such configurations, let's call it $T(r,s)$, but among those, the statistical distribution of the number of houses, or equivalently, the density (the number of houses divided by rs), in other words, the *generating function*, or *weight-enumerator*, of such maximal configurations according to the weight $z^{NnmberOfHouses}$.

It turns out that from an abstract point of view, this is nothing but enumerating **maximal** (w.r.t. the number of ones) $r \times s$ $0-1$ matrices that avoid the pattern

$$\begin{matrix} 1 & 1 & 1 \\ & 1 & \end{matrix} \quad .$$

In other words, $0-1$ matrices (a_{ij}) $(1 \leq i \leq r,\ 1 \leq j \leq s)$ where it is forbidden to have a location (i_0, j_0), where

$$a_{i_0,j_0-1} = 1 \quad , \quad a_{i_0,j_0} = 1 \quad , \quad a_{i_0,j_0+1} = 1 \quad , \quad a_{i_0+1,j_0} = 1 \quad ,$$

and changing any of the entries that are currently 0 to 1, would create this undesirable pattern.

But why just this particular pattern? The question makes sense for enumerating such $0-1$ matrices avoiding *any* pattern, and in fact, any *set of patterns*.

In particular this question has immediate relevance to *social distancing*. We have a rectangular classroom with r rows, each with s seats, and you can't have two students sitting next to each other in the same row, or having anyone immediately in front of you, or behind you (i.e. you can't have any two students sitting next to each other in the same column). In addition none of the currently empty seats can accommodate a newcomer without breaking this restriction. This is equivalent to the problem of weight-enumerating $r \times s$ **maximal** (with respect to the number of ones) $0-1$ matrices avoiding the patterns

$$\begin{matrix} 1 & 1 \end{matrix} \quad , \quad \begin{matrix} 1 \\ 1 \end{matrix} \quad .$$

The problem is already interesting for one row (equivalently one column), and one can impose lots of possible restrictions. Going back to two dimensions, one can think of prohibiting the pattern

$$\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} \quad .$$

and the possibilities are endless.

The famous problem of *maximal non-attacking kings* can be formulated as counting (and weight-counting according to the number of 1s) maximal $r \times s$ $0-1$ matrices avoiding the patterns

$$\begin{matrix} 1 & 1 \end{matrix} \quad , \quad \begin{matrix} 1 \\ 1 \end{matrix} \quad , \quad \begin{matrix} 1 & \\ & 1 \end{matrix} \quad , \quad \begin{matrix} & 1 \\ 1 & \end{matrix} \quad .$$

We will probably never know the exact number of doing it for a 100×100 board, and in general, for any fixed pattern, or set of patterns, the problem of enumerating such maximal $0-1$ **square** matrices, seems intractable. But thanks to the **transfer matrix** method one can efficiently find explicit bi-variate generating functions for such enumeration problems with a **fixed** number of columns, s (not too big, of course), but arbitrary number of rows r (or vice versa). So the problem of enumerating (and weight-enumerating) maximal configurations of non-attacking kings, say on a 1000×5 board can be found exactly.

To see their total number, and the number of these maximal placements with 668 kings (the smallest possible number), see the following output file:

`https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSDk5Story.txt` .

We will now describe how to do it.

A Finite State Machine for Maximal Seating

Suppose that we have r rows of seats. We wish to compute the sequence giving the number of maximal seating assignments as a function of the number of columns. As input we receive r, the number of rows, and $\mathcal{P}$, the set of patterns describing the shapes to be avoided.

There are 2 ways a seating arrangement can fail.

• It may fail to meet the social distancing requirement. That is there may be some people sitting in a place prohibited by one of the patterns. Call this a failure of type 1.

• It may fail to be maximal. That is there may be an empty seat which could be filled without violating any of the patterns. Call this a failure of type 2.

We note that both of these problems are *local.* We don't need to know the entire seating arrangement to determine with certainty that a possible assignment is not valid. This allows us to create a finite state machine with relative ease. We treat a seating arrangement as a sequence of columns. The possible columns will serve as inputs to our state machine. Here, we represent a column as a binary string of 0s and 1s of length r. The 1s correspond to occupied seats, and the 0s correspond to empty seats. There are thus 2^r possible symbols that our state machine must be able to process.

As we read in columns, we must determine whether the constraints have been satisfied *so far.* In order to detect whether the proposed next column causes a violation, we need to store some data about the preceding columns. In this case, we just need to store the contents of some number of previous columns.

Definition: For a given pattern, p, define it's width, $w(p)$, to be the largest x-value of any of its lattice coordinates minus the smallest x value. Define the width of our state machine, $W(r, \mathcal{P})$ or just W, to be the greatest width of all the pattern in $\mathcal{P}$.

To check whether the proposed next column causes a violation of type 1, we need to have stored the previous W columns. If we have that information stored, we can just loop through all the patterns and check that each new 1 does not add a violation of that pattern shape.

To check whether the proposed next column causes a violation of type 2, we must be a bit more clever. For each 0 that is added, there must be a group of nearby 1's that ensures it cannot be changed to a 1 without causing a violation. A 0 that has an appropriately shaped group of nearby 1s is called a satisfied 0. Since we do not know the contents of columns we might read in the future, we are unable to determine whether the 0s in the current column are satisfied. Sure, they may not be satisfied yet, but we should not consider them to be violations yet. If the current column is column j, then we check that the 0s in column $j - W$ are satisfied.

In total, we store the contents of the previous $2W$ columns in the state machine. Thus for column $j - W$ we know exactly the contents of the previous W columns and the next W columns so we can determine with certainty whether the 0s are satisfied.

Since we require a state for each of the possible $2W$ previous columns, this gives a total of a 2^{2Wr} possible states.

Now that we have constructed our set of states, it is simple to determine the transitions. For each state, we have a transition for each possible next column. If the transition causes no violations, it is valid, otherwise the state machine should immediately REJECT. Once a violation has been detected we do not need to look at any more columns.

When we are done reading our input, we must do a little more work to determine whether to ACCEPT or REJECT. Specifically, the 0s in the most recent $W-1$ columns have not been checked yet, so we must check that each is satisfied before accepting.

We also add some special initialization states for when the previous $2W$ columns do not have values yet. The details of this are implementation specific and omitted.

Constructing the Transfer Matrix

Once the finite state machine has been constructed, we seek to count the number of input strings of length n that cause the machine to ACCEPT. To do this we use the transfer matrix method. Each state corresponds to a row and column of the matrix, M. Then the i,j entry $M[i][j]$ is 1 if there is a valid transition from state i to state j, and 0 otherwise.

We can count the number of paths from state i to state j of length s by computing $M^s[i][j]$. Thus the number of maximal seating arrangements with s columns is given by an entry of the matrix M^s. We just set i to be the initial state and j to be the final state of our state machine.

Density of Seating arrangements

We can modify the transition matrix by replacing each 1 with a power of z. For a transition corresponding to adding a column with t 1s, the corresponding matrix entry will be z^t. Then $M^s[i][j]$ is a polynomial in z. The coefficient of z^k is the number of seating arrangements with a total of k seats occupied.

Getting the bi-variate generating function

The previous section explained how to compute the transfer matrix, let's call it $M_{\mathcal{P},r}(z)$, for *any* set of patterns $\mathcal{P}$, and any **fixed** number of rows r. Altogether it had, say, d, states, including the two special states, the initial, and the final, that our program labeled 1 and d, respectively.

We are interested, for an *arbitrary* number of columns, s, in the weight-enumerator, let's call it $W_{\mathcal{P},r,s}(z)$, according to the weight $z^{NumberOfOnes}$ of all maximal $r \times s$ $0-1$ matrices avoiding $\mathcal{P}$. Define the bi-variate generating function

$$f_{\mathcal{P},r}(z,x) := \sum_{s=0}^{\infty} W_{\mathcal{P},r,s}(z) x^s \quad .$$

Then we have

$$f_{\mathcal{P},r}(z,x) = (I \, - \, x\, M_{\mathcal{P},r}(z))^{-1}[1,d] \quad .$$

What's nice about computer algebra is that it can all be streamlined.

Using this generating function, we can automatically compute the *limiting average density* (among maximal seatings) for a fixed r and arbitrary $\mathcal{P}$, as s goes to infinity, namely

$$\frac{1}{r} \lim_{s \to \infty} \frac{W'_{\mathcal{P},r,s}(1)}{s\, W_{\mathcal{P},r,s}(1)} \quad .$$

We omit the details, since this is standard residue calculus, and the interested reader can look at the Maple source code of procedure `AsyAv(f,x,z)` in the Maple package `SD.txt`, accompanying this paper.

Random Sequential Adsorption

The polynomial $W_{\mathcal{P},r,s}(z)$ is the weight enumerator of maximal $0-1$ $r \times s$ matrices that avoid the patterns in $\mathcal{P}$. Hence the coefficient of z^m in that polynomial is the exact number of such matrices with exactly m ones (or equivalently the number of maximal seatings in an r by s classroom obeying the restrictions in $\mathcal{P}$ and having exactly m occupied chairs). It follows that the *probability generating function* (for the uniform distribution), let's call it $V_{\mathcal{P},r,s}(z)$, is:

$$V_{\mathcal{P},r,s}(z) = \frac{W_{\mathcal{P},r,s}(z)}{W_{\mathcal{P},r,s}(1)} \quad ,$$

and the expected number of occupied seats is $\frac{d}{dz} V_{\mathcal{P},r,s}(z)|_{z=1} = V'_{\mathcal{P},r,s}(1)$.

There is a quick way to generate random maximal seatings, described in the special case of T-avoiding seatings in [PSZ], but that makes sense in general, called *Random Sequential Adsorption*, that they abbreviated to RSA, but since for us RSA means 'Rivest-Shamir-Adleman', we will refrain from using this abbreviation.

In that random-generation process, one picks, *uniformly at random*, a permutation of the rs initial empty seats, representing rs people each having their favorite seat, all distinct from each other. When someone enters the classroom they attempt to occupy their favorite seat, and do so if they do not cause a violation. But if sitting there does create a violation, they are not allowed to sit elsewhere and must leave the room (and cut the class).

This process is very easy to simulate, and it generates a random maximal seating. Alas, it is not the uniform one. Thanks to our closed-form expressions for the probability distribution for $r \times s$ maximal seatings with a fixed (not too big) r but arbitrarily large s, we were able to compare notes, and estimate (via simulations) how close is random sequential adsorption to the uniform distribution.

The Maple package SD.txt

This article is accompanied by a Maple package `SD.txt`, available from

`https://sites.math.rutgers.edu/~zeilberg/tokhniot/SD.txt` .

The front of this article

https://sites.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/social.html ,

contains numerous sample output files, a selection of which we will describe next.

Output

Explicit Expressions for generating functions in One-Dimensional Seatings: Avoiding b consecutive occupied seats

Let $A_b(m,n)$ be the number of maximal $0-1$ vectors of length n with m ones (equivalently: sum m), avoiding b consecutive 1's, and let

$$f_b(z,x) := \sum_{n=0}^{\infty} \left(\sum_{m=0}^{n} A_b(m,n)\, z^m \right) x^n \quad ,$$

then we have

$$f_2(z,x) = -\frac{x^2z + xz + 1}{x^3z + x^2z - 1} \quad ,$$

and the limiting average density is $.4114955887\ldots$.

$$f_3(z,x) = -\frac{x^5z^3 - x^3z^2 - x^2z^2 + x^2z - xz - 1}{x^6z^3 - x^4z^2 - x^3z^2 - x^2z + 1} \quad ,$$

and the limiting average density is $0.5772029462\ldots$.

$$f_4(z,x) = -\frac{x^9z^6 + x^7z^5 - x^5z^4 + x^5z^3 - 2x^4z^3 - x^3z^3 + x^3z^2 - x^2z^2 - xz - 1}{x^{10}z^6 + x^8z^5 - x^6z^4 - 2x^5z^3 - x^4z^3 - x^3z^2 + 1}$$

and the limiting average density is $0.6686427921\ldots$.

$$f_5(z,x) =$$

$$-\frac{x^{14}z^{10} - x^{11}z^8 - 2x^9z^7 + x^9z^6 - 2x^8z^6 + x^6z^5 - x^6z^4 + 2x^5z^4 + x^4z^4 - 2x^4z^3 + x^3z^3 - x^3z^2 + x^2z^2 + xz + 1}{x^{15}z^{10} - x^{12}z^8 - 2x^{10}z^7 - 2x^9z^6 + x^7z^5 + 2x^6z^4 + x^5z^4 + x^4z^3 + x^3z^2 - 1}$$

and the limiting average density is $0.7269949175\ldots$.

For $f_6(z,x), f_7(z,x), f_8(z,x)$, see the output file

https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSD1.txt .

The limiting average densities are 0.7675902978, 0.7975140257, and 0.8205096203, respectively.

Explicit Expressions for generating functions in One-Dimensional Seatings: Stricter Social Distancing Restrictions

Suppose that, in a row of seats of length n, each person must have at least b empty seats on either side, and let $C_b(m,n)$ be the number of such maximal $0-1$ vectors of length n and m 1s, and define the bi-variate generating function:

$$g_b(z,x) := \sum_{n=0}^{\infty} \left(\sum_{m=0}^{n} C_b(m,n)\, z^m \right) x^n \quad .$$

To our pleasant surprise, there is an **explicit** expression not only in terms of z and x but also for symbolic b. We leave it as challenge to the reader to prove it for arbitrary b.

Conjecture: For $b \geq 1$,

$$g_b(z,x) \;=\; \frac{x^{2\,b+2}z - x^{b+1}z - x^{b+2}z + xz + (-1+x)^2}{(-1+x)\,(-x^{2\,b+2}z + x^{b+1}z + x - 1)} \quad .$$

We confirmed it for $b \leq 8$. See the output file

`https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSD2.txt` .

Avoiding Dimers in $3 \times s$, $4 \times s$, and $5 \times s$, $0-1$ matrices

To see explicit expressions, limiting average densities, and comparison with Random Sequential Adsorption simulation, for the $3 \times s$ case, see the output file

`https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSDd3.txt` .

In particular, the bi-variate generating function for $3 \times s$ maximal dimer-avoiding $0-1$ matrices is:

$$-\frac{\left(2x^5z^4 + 2x^3z^3 + 2x^2z^3 - 6x^2z^2 - x\,z^2 - xz - z - 1\right)xz}{x^5z^4 + 2x^4z^4 - x^4z^3 + x^3z^4 - 4x^3z^3 - x^2z^3 - xz + 1} \quad .$$

For similar information for the $4 \times s$ case, see the output file

`https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSDd4.txt` ,

in particular, the bi-variate generating function for $4 \times s$ maximal dimer-avoiding $0-1$ matrices: is,

$$\frac{\left(x^6z^6 - x^5z^6 + x^5z^5 - 2x^5z^4 - 3x^4z^4 + 2x^3z^4 - 7x^3z^3 + 2x^3z^2 - 4x^2z^3 + 7x^2z^2 - x\,z^2 + 4xz + 3\right)x\,z^2}{x^6z^6 + x^5z^6 + x^5z^5 + 2x^4z^5 - x^4z^4 + 2x^3z^5 - 4x^3z^4 - x^3z^3 - 2x^2z^3 - x^2z^2 - x\,z^2 + 1}$$

For similar information for the $5 \times s$ case, see the output file:

https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSDd5.txt .

Maximal Non-Attacking Kings

The generating function for maximal configurations of non-attacking kings on a $3 \times s$ chessboard is

$$-\frac{\left(x^5z^3+x^5z^2-x^3z^3+x^3z+2x^2z^2+x^2z+x\,z^2-x^2-3xz-2x-z-1\right)xz}{x^6z^4+x^6z^3-x^5z^4-x^5z^3+x^4z^3+x^4z^2+x^3z^3-x^3z^2-x^3z-x^2z^2-x^2z-xz+1} \quad .$$

For more details see: https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSDk3.txt .

The generating function for maximal configurations of non-attacking kings on a $4 \times s$ chessboard is

$$-\frac{\left(6x^6z^3+9x^5z^2-6x^4z^3+3x^4z^2-3x^3z^2+3x^3z+3x^2z^2+2x^2z-3x^2+3xz-12x-3\right)x\,z^2}{6x^7z^5-6x^6z^5+9x^6z^4-6x^5z^4+3x^4z^4+x^4z^3+3x^3z^3-6x^3z^2-4x^2z^2-xz+1} \quad .$$

For more details see: https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSDk4.txt .

For the bi-variate generating function of maximal configurations of non-attacking kings on a $5 \times s$ chessboard and more details, and results of simulations, see:

https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSDk5.txt .

See also the output files

https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSD22C3.txt ,

https://sites.math.rutgers.edu/~zeilberg/tokhniot/oSD22C4.txt ,

for avoiding a 2×2 block of *ones*.

Readers are welcome to generate more data, and experiment with other patterns (and set of patterns) to their heart's content, using the Maple package SD.txt. Enjoy!

The Uniform vs. the Random Sequential Adsorption Distributions

It seems much harder to get the *exact* probability generating functions for the density (equivalently, the number of 1s) by using Random Sequential Adsorption in general, and we have to resort to (very reliable!) simulations.

But for the very special case of seatings on a line with n seats, and avoiding the single pattern 11, i.e. non adjacent seatings, we can get exact, closed-form expression for **both** the uniform and the random sequential adsorption process.

Let $g_n(z)$ be the enumeration generating function, according to 'number of occupied seats' under the uniform distribution, then as we saw above

$$\sum_{n=0}^{\infty} g_n(z)\,x^n = -\frac{x^2z+xz+1}{x^3z+x^2z-1} \quad .$$

Equivalently the sequence $\{g_n(z)\}$ satisfies the **linear** recurrence

$$g_n(z) = z(g_{n-2}(z) + g_{n-3}(z)) \quad ,$$

subject to the initial conditions

$$g_0(z) = 1 \quad , \quad g_1(z) = z \quad , \quad g_2(z) = 2z \quad .$$

To turn it into a *probability generating function*, we have to divide by $g_n(1)$ and define

$$G_n(z) := \frac{g_n(z)}{g_n(1)} \quad .$$

Recall that in random sequential adsorption the input is a permutation π. If $\pi(1) = i$ then the persons that like locations $i-1$ and $i+1$ would not be able to sit, since i is already occupied. There are $(n-1)(n-2)$ ways where they can be placed. The remaining students independently seat in locations $1 \leq x \leq i-2$ and $i+2 \leq x \leq n$ and it follows that the probability generating function, let's call it $F_n(z)$ (after dividing by $n!$), satisfies the **non-linear** recurrence

$$F_n(z) = \frac{z}{n}\left(2\,F_{n-2}(z) + \sum_{i=2}^{n-1} F_{i-2}(z)F_{n-i-1}(z)\right) \quad , \quad n \geq 2 \quad ; \quad F_0(z) = 1 \quad , \quad F_1(z) = z \quad .$$

To get the expected number of occupied chairs under the uniform and the sequential adsorption distributions, we compute $G'_n(1)$ and $F'_n(1)$ respectively, or more informatively, the *densities* $G'_n(1)/n$ and $F'_n(1)/n$. They both converge to limits. The former can be found exactly. We have (as already stated above, with less precision, the number is a certain algebraic number)

$$\lim_{n\to\infty} \frac{G'_n(1)}{n} = 0.41149558866264576338190038133553194080060864935476581763580335693984 0\ldots$$

The other limit we estimated numerically. we have (by taking large n)

$$\lim_{n\to\infty} \frac{F'_n(1)}{n} \approx 0.4323\ldots \quad ,$$

agreeing with the exact value $(1-e^{-2})/2$ stated in Steven Finch's wonderful book [F] (section 5.3.1) and due to several people (see [F] for references).

Hence the average occupation densities, for large n, of the random sequential adsorption process is approximately 1.052 times those of the uniform distribution. Of course, the maximal density, obtained by the regular spacing 101010... is 0.5.

Conclusion

In this case study, we illustrated the power of *symbolic computation* and *experimental mathematics*, both so dear to Marko Petkovsek, to get insight about the statistics of maximal seating arrangements avoiding any set of social distancing restrictions.

References

[F] Steven R. Finch, *Mathematical Constants* (Encyclopedia of mathematics and its applications, v. 94) , Cambridge University Press, 2003.

[PSZ] Mate Puljiz, Stjepan Sebek, and Josip Zubrinic, *Packing density of combinatorial settlement planning models*, Amer. Math. Monthly **130**(10) (Dec. 2023).

[V] Vince Vatter, *Social Distancing, Primes, and Perrin Numbers*, , Math Horizons, **29**(1) (2022).
`https://sites.math.rutgers.edu/~zeilberg/akherim/vatter23.pdf` .

George Spahn and Doron Zeilberger, Department of Mathematics, Rutgers University (New Brunswick), Hill Center-Busch Campus, 110 Frelinghuysen Rd., Piscataway, NJ 08854-8019, USA.
Email: `gs828 at math dot rutgers dot edu` , `DoronZeil] at gmail dot com` .

Written: **Jan. 22, 2024**.