TECHNISCHE UNIVERSITÄT BERGAKADEMIE FREIBERG

The University of Resources. Since 1765.

Institute of Computer Science

An Algorithmic Approach for the Zarankiewicz Problem

Matthias Werner

September 20, 2012

10$^{\text{th}}$ International Workshop on Boolean Problems 2012

Table of Contents

- Fill a $m \times n$ Boolean matrix $k(m, n)$ with 0's and 1's
- ... without creating a submatrix $h(r, s)$ full of 1's
- Find the maximum number of 1's

- Fill a $m \times n$ Boolean matrix $k(m, n)$ with 0's and 1's
- ... without creating a submatrix $h(r, s)$ full of 1's
- Find the maximum number of 1's

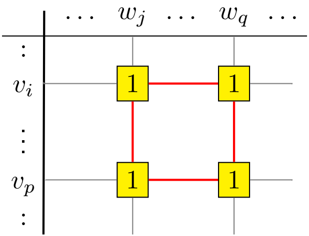


Fig.: Forbidden submatrix $h(2, 2)$

- Fill a $m \times n$ Boolean matrix $k(m, n)$ with 0's and 1's
- ... without creating a submatrix $h(r, s)$ full of 1's
- Find the maximum number of 1's

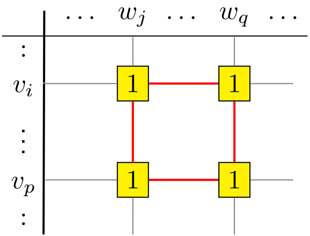


Fig.: Forbidden submatrix $h(2, 2)$

$v_i \setminus w_j$	w_1	w_2	w_3	w_4	w_5
v_1	1	1	1	1	.
v_2	1	.	.	.	1
v_3	.	1	.	.	1
v_4	.	.	1	.	1
v_5	.	.	.	1	1

Fig.: $k(5, 5)$ without any full $h(2, 2)$

Definition

Given $m \geq 3, n \geq 3$ and $r \geq 2, s \geq 2$. The **Zarankiewicz Function** $Z_{r,s}(m,n)$ determines the **least positive integer** such that **if** a Boolean matrix $k(m,n)$ contains $Z_{r,s}(m,n)$ 1's **then** it must have a submatrix $h(r,s)$ with r rows and s columns consisting entirely of ones.

- Focus on $r = s = 2$
- "Rectangle" := "$h(2,2)$"
- $\text{maxrf}(m,n) := Z_{2,2}(m,n) - 1$

How to construct maximum rectangle free $k(m,n)$ for a given (m,n)?

Definition

Given $m \geq 3, n \geq 3$ and $r \geq 2, s \geq 2$. The **Zarankiewicz Function** $Z_{r,s}(m,n)$ determines the **least positive integer** such that **if** a Boolean matrix $k(m,n)$ contains $Z_{r,s}(m,n)$ 1's **then** it must have a submatrix $h(r,s)$ with r rows and s columns consisting entirely of ones.

- Focus on $r = s = 2$
- "Rectangle" := "$h(2,2)$"
- $\mathrm{maxrf}(m,n) := Z_{2,2}(m,n) - 1$

How to construct maximum rectangle free $k(m,n)$ for a given (m,n)?

Definition

Given $m \geq 3, n \geq 3$ and $r \geq 2, s \geq 2$. The **Zarankiewicz Function** $Z_{r,s}(m,n)$ determines the **least positive integer** such that **if** a Boolean matrix $k(m,n)$ contains $Z_{r,s}(m,n)$ 1's **then** it must have a submatrix $h(r,s)$ with r rows and s columns consisting entirely of ones.

- Focus on $r = s = 2$
- "Rectangle" := "$h(2,2)$"
- $\text{maxrf}(m,n) := Z_{2,2}(m,n) - 1$

How to construct maximum rectangle free $k(m,n)$ for a given (m,n)?

Definition

Given $m \geq 3, n \geq 3$ and $r \geq 2, s \geq 2$. The **Zarankiewicz Function** $Z_{r,s}(m,n)$ determines the **least positive integer** such that **if** a Boolean matrix $k(m,n)$ contains $Z_{r,s}(m,n)$ 1's **then** it must have a submatrix $h(r,s)$ with r rows and s columns consisting entirely of ones.

- Focus on $r = s = 2$
- "Rectangle" := "$h(2,2)$"
- $\text{maxrf}(m,n) := Z_{2,2}(m,n) - 1$

How to construct maximum rectangle free $k(m,n)$ for a given (m,n)?

Will I live to see the result?

Computation of Zarankiewicz Problem:

- Bruteforce: $2^{m \cdot n}$ assignments
- Checking for rectangles $\Rightarrow \binom{n}{2}\binom{m}{2}$ for each assignment

Simplification:

- Permuting rows or columns yields an isomorphic solution.
- "Construct" only assignments that are distinct w.r.t. row and column permutations

Tab.: Time estimation for computing $\mathrm{maxrf}(m, n)$ on CPU 2.67 GHz, 1 instruction per cycle

$m \times n$	evaluate all grids	utilize permutations
8×8	1.71×10^{5} a	66 h 44 min
11×11	9.55×10^{22} a	2.83×10^{10} a
12×12	1.15×10^{30} a	4.29×10^{15} a

Computation of Zarankiewicz Problem:

- Bruteforce: $2^{m \cdot n}$ assignments
- Checking for rectangles $\Rightarrow \binom{n}{2}\binom{m}{2}$ for each assignment

Simplification:

- Permuting rows or columns yields an isomorphic solution.
- "Construct" only assignments that are distinct w.r.t. row and column permutations

Tab.: Time estimation for computing $\mathrm{maxrf}(m, n)$ on CPU 2.67 GHz, 1 instruction per cycle

$m \times n$	evaluate all grids	utilize permutations
8×8	1.71×10^{5} a	66 h 44 min
11×11	9.55×10^{22} a	2.83×10^{10} a
12×12	1.15×10^{30} a	4.29×10^{15} a

Computation of Zarankiewicz Problem:

- Bruteforce: $2^{m \cdot n}$ assignments
- Checking for rectangles $\Rightarrow \binom{n}{2}\binom{m}{2}$ for each assignment

Simplification:

- Permuting rows or columns yields an isomorphic solution.
- "Construct" only assignments that are distinct w.r.t. row and column permutations

Tab.: Time estimation for computing $\mathrm{maxrf}(m, n)$ on CPU 2.67 GHz, 1 instruction per cycle

$m \times n$	evaluate all grids	utilize permutations
8×8	1.71×10^{5} a	66 h 44 min
11×11	9.55×10^{22} a	2.83×10^{10} a
12×12	1.15×10^{30} a	4.29×10^{15} a

- $k(m,n)$: List of m numbers $a_i > 0$ in binary notation (n bits)
- $a_i \geq a_{i-1}$ (yields distinct assignments w.r.t. row permutations)
- $a_i \in \{1,\ldots,2^n-1\}$ is a bit pattern for row i
- If $a_i \equiv 2^n-1$ then increment a_{i-1} and reset a_k to a_{i-1}, $\forall k \geq i$

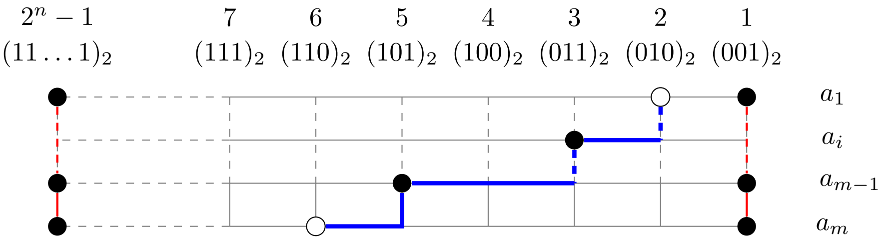
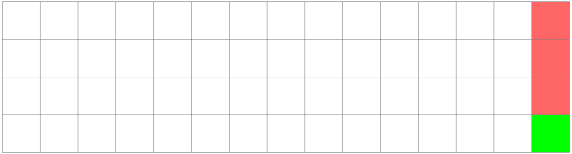


Fig.: Paths of patterns taken by the Overflow Algorithm

Overflow Algorithm on $k(4,4)$



$$a_1 = 1 = (0001)_2$$
$$a_2 = 1 = (0001)_2$$
$$a_3 = 1 = (0001)_2$$
$$a_4 = 1 = (0001)_2$$

Number of paths (assignments) passed by the algorithm:

$$N(m, n, a_1, T) = \binom{T - a_1 + m}{m}$$

where a_1 is start value and T is last value (a_1=1, T=$2^n - 1$ and $m \geq n$).

Tab.: Number of assignments modulo permutations

$m \times n$	None ($2^{m \cdot n}$)	Row	Row & Column
8×8	1.845×10^{19}	5.099×10^{14}	8.182×10^{11}
11×11	2.659×10^{36}	6.841×10^{28}	7.880×10^{23}
12×12	3.230×10^{43}	4.731×10^{34}	8.291×10^{28}

"..." modulo row permutations: $N(m, n, 0, 2^n - 1)$

"..." modulo row and column permutations: `http://oeis.org/A089006`.

Number of paths (assignments) passed by the algorithm:

$$N(m, n, a_1, T) = \binom{T - a_1 + m}{m}$$

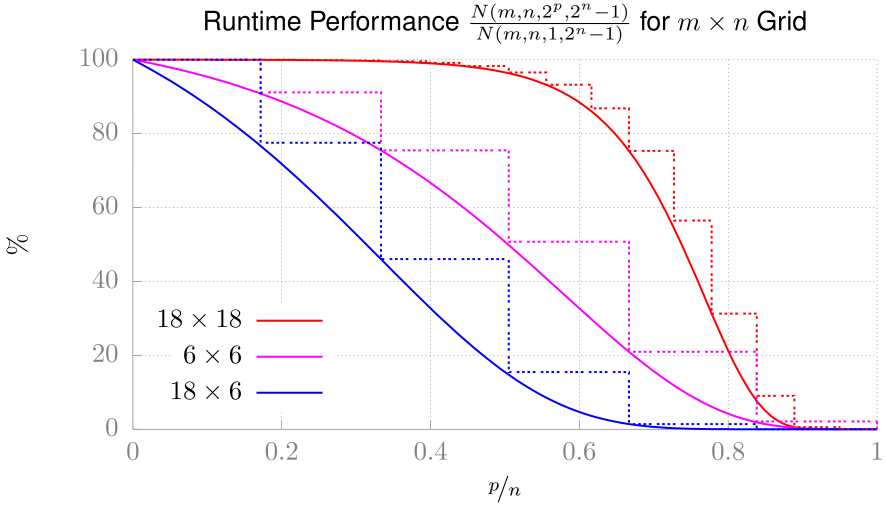
where a_1 is start value and T is last value (a_1=1, T=$2^n - 1$ and $m \geq n$).

Tab.: Number of assignments modulo permutations

$m \times n$	None ($2^{m \cdot n}$)	Row	Row & Column
8×8	1.845×10^{19}	5.099×10^{14}	8.182×10^{11}
11×11	2.659×10^{36}	6.841×10^{28}	7.880×10^{23}
12×12	3.230×10^{43}	4.731×10^{34}	8.291×10^{28}

"..." modulo row permutations: $N(m, n, 0, 2^n - 1)$

"..." modulo row and column permutations: `http://oeis.org/A089006`.


Runtime Performance $\frac{N(m,n,2^p,2^n-1)}{N(m,n,1,2^n-1)}$ for $m \times n$ Grid
100
80
60
40
20
0
%
0
0.2
0.4
0.6
0.8
1
p/n
18×18
6×6
18×6

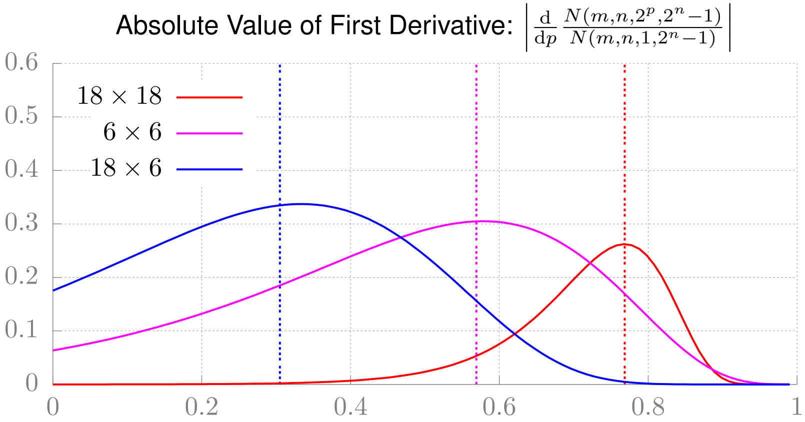


Fig.: Absolute value of first derivative with marker at $n - \log_2(m)$

Maximum can be approximated very well by: $n - \log_2(m) =: p_{\max}(m, n)$

The Overflow Algorithm

Improvements without loosing accuracy

- Rectangle Free Incrementing

$$
\begin{array}{l|cccccc|}
a_i & . & 1 & . & . & 1 & . \\
a_{i+1} & . & 1 & . & 1 & . & .
\end{array}
\qquad
\begin{array}{l|cccccc|}
a_i & . & 1 & . & 1 & . & 1 \\
a_{i+1} & . & 1 & 1 & . & . & .
\end{array}
$$

Fig.: Rectangle free incrementing of a_i and a_{i+1}

- Skipping branch when $a_i \geq (1100\ldots0)_2$, $i < m$
- Is the current branch able to improve the current maximum?

$$
\begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
\vdots & \vdots & \vdots
\end{bmatrix}
\leftarrow \text{current row}
$$

- Estimate or use precomputed maximum of lower submatrix
- #Ones(UpperMatrix) + Maximum*(LowerMatrix) > CurrentMaximum?

The Overflow Algorithm

Improvements without loosing accuracy

- Rectangle Free Incrementing

$$
\begin{array}{c|cccccc|}
a_i & . & 1 & . & . & 1 & . \\
a_{i+1} & . & 1 & . & 1 & . & .
\end{array}
\qquad
\begin{array}{c|cccccc|}
a_i & . & 1 & . & 1 & . & 1 \\
a_{i+1} & . & 1 & 1 & . & . & .
\end{array}
$$

Fig.: Rectangle free incrementing of a_i and a_{i+1}

- Skipping branch when $a_i \geq (1100\ldots0)_2$, $i < m$
- Is the current branch able to improve the current maximum?

$$
\begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
\vdots & \vdots & \vdots
\end{bmatrix}
$$

← current row

- Estimate or use precomputed maximum of lower submatrix
- $\#$Ones(UpperMatrix) + Maximum*(LowerMatrix) $>$ CurrentMaximum?

Tab.: Runtime results of Overflow Algorithm with optimizations (X5650 2.67 GHz)

	RFI	Upper Bound & RFI	Exact & RFI
$\mathrm{maxrf}(8,8)$	160 s	3 s	1 s
$\mathrm{maxrf}(9,9)$	12 h 6 min	5 min 49 s	23 s
$\mathrm{maxrf}(10,10)$	–	37 h 19 min	18 min

RFI = using Rectangle Free Incrementing

Upper bound (overestimation ≤ 1 for $m, n < 15$):

$$\mathrm{maxrf}(m,n) \leq \frac{1}{2}\left(m + \sqrt{m^2 + 4 \cdot mn(n-1)}\right)$$

1	·	·	1	·	1	1	1	·	46
2	·	·	1	1	·	·	·	1	49
3	·	1	·	·	1	·	·	1	73
4	·	1	·	1	·	1	·	·	84
5	·	1	1	·	·	·	·	·	96
6	1	·	·	·	·	1	·	1	133
7	1	·	·	1	1	·	·	·	152
8	1	·	1	·	·	·	·	·	160
9	1	1	·	·	·	·	1	·	194

1	·	·	1	·	1	1	1	·	·	92
2	·	·	1	1	·	·	·	1	1	99
3	·	1	·	·	·	1	·	1	·	138
4	·	1	·	·	1	·	·	·	1	145
5	·	1	·	1	·	·	1	·	·	164
6	1	·	·	·	·	·	1	1	·	262
7	1	·	·	·	·	1	·	·	1	265
8	1	·	·	1	1	·	·	·	·	304
9	1	1	1	·	·	·	·	·	·	448

Fig.: Last obtained optimum for 9×8 and 9×9

- Last optimum generated by the algorithm
- Automatically produces ordered partitions
- MSB(a_1) = 5, $p_{\max}(9,8) = 4.830\ldots$ (correct: $4.852\ldots$)
- MSB(a_1) = 6, $p_{\max}(9,9) = 5.830\ldots$ (correct: $5.841\ldots$)

·	·	·	1	1	1	1	15
·	·	1	·	·	1	·	18
·	·	1	·	1	·	·	20
·	·	1	1	·	·	·	24
·	1	·	·	·	1	·	34
·	1	·	·	1	·	·	36
·	1	·	1	·	·	·	40
·	1	1	·	·	·	1	49
1	·	·	·	·	·	1	65
1	·	·	·	·	1	·	66
1	·	·	·	1	·	·	68
1	·	·	1	·	·	·	72
1	·	1	·	·	·	·	80
1	1	·	·	·	·	·	96

·	·	·	1	·	1	1	·	22
·	·	·	1	1	·	·	1	25
·	·	1	·	·	1	·	1	37
·	·	1	·	1	·	1	·	42
·	1	·	·	·	·	1	1	67
·	1	·	·	1	1	·	·	76
·	1	1	1	·	·	·	·	112
1	·	·	·	·	·	·	1	129
1	·	·	·	·	·	1	·	130
1	·	·	·	·	1	·	·	132
1	·	·	·	1	·	·	·	136
1	·	·	1	·	·	·	·	144
1	·	1	·	·	·	·	·	160
1	1	·	·	·	·	·	·	192

Fig.: Last obtained optimum for 14×7 and 14×8

$\mathsf{MSB}(a_1) = 4$, $p_{\max}(14, 8) = 4.192\ldots$ (correct: $4.228\ldots$)

The Overflow Algorithm

Key Ideas for Improvements

- a_1: Fix pattern and MSB at $\lfloor p_{\max}(m, n) \rfloor$
- a_2: Same pattern size and MSB (latest valid configuration)
- a_3: Init with MSB(a_3) = MSB(a_1)+1
- Overflow Algorithm shall take only THE latest, important "branch"

a_1	.	.	**1**	.	1	.
a_2	.	.	1	1	.	.
a_3	.	1	.	.	.	.
⋮						

a_1	.	.	**1**	.	1	1	.	.
a_2	.	.	1	1	.	.	1	.
a_3	.	1	.	.	.	.	.	.
⋮								

a_1	.	.	**1**	.	1	1	1	.	.	.
a_2	.	.	1	1	.	.	.	1	1	.
a_3	.	1	.	.	.	.	.	.	.	.
⋮										

Fig.: Configurations of first three rows where a_1 is a fixed pattern

Question

Suppose MSB(a_1) is fixed at $\lfloor n - \log_2(m) \rfloor$.
Does the algorithm always find an optimal solution with this limitation?

- a_1: Fix pattern and MSB at $\lfloor p_{\max}(m,n) \rfloor$
- a_2: Same pattern size and MSB (latest valid configuration)
- a_3: Init with MSB(a_3) = MSB(a_1)+1
- Overflow Algorithm shall take only THE latest, important "branch"

a_1	.	.	**1**	.	1	.
a_2	.	.	1	1	.	.
a_3	.	1	.	.	.	.
$\vdots$						

a_1	.	.	**1**	.	1	1	.	.
a_2	.	.	1	1	.	.	1	.
a_3	.	1	.	.	.	.	.	.
$\vdots$								

a_1	.	.	**1**	.	1	1	1	.	.	.
a_2	.	.	1	1	.	.	.	1	1	.
a_3	.	1	.	.	.	.	.	.	.	.
$\vdots$										

Fig.: Configurations of first three rows where a_1 is a fixed pattern

Question

Suppose MSB(a_1) is fixed at $\lfloor n - \log_2(m) \rfloor$.

Does the algorithm always find an optimal solution with this limitation?

Tab.: MSB(a_1) of last found optimum and $\left[\lfloor n - \log_2(m) \rfloor - \text{MSB}(a_1)\right]$

	4	5	6	7	8	9	10
4	2						
5	2 [-1]	3 [-1]					
6	–	2	3				
7	–	2	3	4			
8	–	2	3	4	5		
9	–	2 [-1]	3 [-1]	4 [-1]	5 [-1]	6 [-1]	
10	–	–	3 [-1]	4 [-1]	4	5	6
11	–	–	2	3	4	5	6
12	–	–	2	3	3 [+1]	4 [+1]	6
13	–	–	2	3	4	4 [+1]	6
14	–	–	2	3	4	4 [+1]	?

Results

Tab.: Runtime Overflow Algorithm (Intel X5650 2.67 GHz)

$\mathrm{maxrf}(\cdot,\cdot)$		Full Run		a_1 fixed	
$\mathrm{maxrf}(7,7)$	=21	3 ms	(3 ms)	0.5 ms	(0.1 ms)
$\mathrm{maxrf}(14,7)$	=31	11 s	(2 s)	400 ms	(1.7 ms)
$\mathrm{maxrf}(8,8)$	=24	1 s	(72 ms)	14 ms	(1 ms)
$\mathrm{maxrf}(10,10)$	=34	18 min	(34 s)	1.6 s	(1.6 s)
$\mathrm{maxrf}(11,11)$	=39	7 d 4 h	(9 h)	34 min	(6 min)
$\mathrm{maxrf}(12,12)$	=45	–	–	12 h 28 min	(5 h 7 min)

"(. . .)" : Time after first optimal solution was found.
Overflow Algorithm with "Rectangle Free Incrementing" and usage of exact maxima of subgrids.

Results

Tab.: Number of optimal solutions found by Overflow Algorithm

$\text{maxrf}(\cdot,\cdot)$	FULL RUN	a_1 FIXED
$\text{maxrf}(7,7)$	1	1
$\text{maxrf}(14,7)$	27 641	282
$\text{maxrf}(8,8)$	16 596	128
$\text{maxrf}(9,9)$	4464	144
$\text{maxrf}(10,10)$	32 838	1
$\text{maxrf}(11,11)$	1 168 996	432

[BF02] Bryant, Darryn E. and Fu, Hung-Lin. "C_4-saturated bipartite graphs". In: *Discrete Mathematics* 259. 1-3 (2002), pp. 263–268. ISSN: 0012-365X. DOI: 10.1016/S0012-365X(02)00371-0. URL: http://dx.doi.org/10.1016/S0012-365X(02)00371-0.

[GRID] Fenner, Stephen, Gasarch, William, Glover, Charles, and Purewal, Semmy. "Rectangle Free Coloring of Grids". In: *ArXiv e-prints* (May 2010). eprint: 1005.3750. URL: http://arxiv.org/abs/1005.3750.

[Guy69] Guy, Richard. "A many-facetted problem of Zarankiewicz". In: *The Many Facets of Graph Theory*. Ed. by Chartrand, G. and Kapoor, S. Vol. 110. Lecture Notes in Mathematics. 10.1007/BFb0060112. University of Calgary Canada: Springer Berlin / Heidelberg, 1969, pp. 129–148. ISBN: 978-3-540-04629-5. URL: http://dx.doi.org/10.1007/BFb0060112.

[Rei58] Reiman, I. “Ueber ein Problem von K. Zarankiewicz”. In: *Acta Mathematica Academiae Scientiarum Hungaricae* 9 (3 1958), pp. 269–279. DOI: 10.1007/BF02020254.

[Rom75] Roman, Steven. “A problem of Zarankiewicz”. In: *Journal of Combinatorial Theory, Series A* 18. 2 (1975), pp. 187–198. ISSN: 0097-3165. DOI: 10.1016/0097-3165(75)90007-2. URL: http://www.sciencedirect.com/science/article/pii/0097316575900072.

[SP12a] Steinbach, B. and Posthoff, C. “Extremely Complex 4-Colored Rectangle-Free Grids: Solution of Open Multiple-Valued Problems”. In: *Proceedings of the IEEE 42nd International Symposium on Multiple-Valued Logic (ISMVL 2012)* (2012), pp. 37–44. ISSN: 0195-623X/12. DOI: 10.1109/ISMVL.2012.12.

[SP12b] Steinbach, B. and Posthoff, C. "Search Space Restriction for Maximal Rectangle-Free Grids". In: *10th International Workshop on Boolean Problems* (2012).

[SP12c] Steinbach, B. and Posthoff, C. "The Solution of Ultra Large Grid Problems". In: *21st International Workshop on Post-Binary ULSI Systems (ULSI 2012)* (2012), pp. 1–10.

[Wer12] Werner, M. *Algorithmische Betrachtungen zum Zarankiewicz Problem*. Seminararbeit. May 2012. URL: http://11235813tdd.blogspot.de/2012/02/zarankiewicz-problem.html.