After using an SDP solver we obtain a numerical output with an optimal matrix X. With CSDP we obtain

$$X = \begin{pmatrix} 0.500000008 & & & & & \\ & 0.000000001 & & & & \\ & & 0.333333332 & & & \\ & & & 0.000000003 & & \\ & & & & 0.499999997 & -0.499999996 \\ & & & & -0.499999996 & 0.499999998 \end{pmatrix}$$

Looking at it, one can easily "guess" that it is possible to obtain the wanted bound $\frac{1}{2}$ using the rational matrix

$$Q = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Recall that this matrix was used in the inequality $[\![({}_{\circ}{}^{\bullet}, {}_{\circ}\!/{}^{\bullet})Q({}_{\circ}{}^{\bullet}, {}_{\circ}\!/{}^{\bullet})^T]\!] \geq 0$, which is exactly the inequality $\frac{1}{2}[\![({}_{\circ}\!/{}^{\bullet} - {}_{\circ}{}^{\bullet})^2]\!] \geq 0$ used in our proof of Mantel's theorem.

This explains why exactly this inequality was used in our proof. In some (small) applications this approach is enough as one can guess the rational matrix from the numerical output, but for larger instances it is not possible. Moreover, wrong approximation by rational numbers might break the positive semidefinite assumption. Fortunately, there are ways to perform the rounding step automatically.

2.3 Rounding

A naive approach to round the numerical matrix obtained by an SDP solver is to just round each entry to some very close rational number. There are two problems with such an approach. One is that the obtained bound can differ from what we want to prove (in our example we could not obtain 0.5, but 0.500000008). The second one is that the rounded matrix may not be positive semidefinite and the proof will not be valid anymore.

To overcome the first problem, we solve the linear system of equation (6) with the value of x_0 set to the wanted bound, and the values of x_i set to 0 whenever we expect the i-th graph $H \in \mathcal{G}_n$ to satisfy the inequality $x_0 \geq g_H + c_H$ as equality. Those graphs can be determined from the expected extremal constructions, as for any subgraph of any extremal construction the inequality must hold as equality.

In our working example, we set $x_0 = 0.5$, $x_1 = 0$ and $x_3 = 0$ as there are two graphs on 3 vertices that can appear as a subgraph of our expected extremal construction. We end up with the following system of equations:

$$\begin{cases} q_{00} = \dfrac{1}{2} \\ x_2 + \dfrac{1}{3}q_{00} + \dfrac{2}{3}q_{01} = \dfrac{1}{6} \\ \dfrac{2}{3}q_{01} + \dfrac{1}{3}q_{11} = -\dfrac{1}{6}. \end{cases} \tag{7}$$

We can now find a solution of the above system with values close to the numerical values coming from an SDP solver. This way we guarantee that the obtained bound is equal to the wanted bound. Unfortunately, it may happen that the obtained matrix is no longer positive semidefinite and so the proof is not valid. An example solution of the system of equalities (7) close to this matrix is $q_{00} = \frac{1}{2}$, $q_{01} = -\frac{1}{2} - \varepsilon$, $q_{11} = \frac{1}{2} + 2\varepsilon$ and $x_2 = \frac{1}{3} + \frac{2}{3}\varepsilon$ for some small ε. Unfortunately, the obtained matrix

$$\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} - \varepsilon \\ -\frac{1}{2} - \varepsilon & \frac{1}{2} + 2\varepsilon \end{pmatrix}$$

has determinant equal to $-\varepsilon^2$. In particular, if $\varepsilon \neq 0$ then this matrix is not positive semidefinite and our proof is not valid.

The problem is that the optimal matrix $\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$ has two eigenvalues 0 and 2. When we do even a tiny perturbation, we can change the value of the smaller eigenvalue to be below 0, which will make the matrix not positive semidefinite. In order to overcome this difficulty we need to somehow determine the subspace generated by the eigenvectors with eigenvalue 0 and round the matrix only in the subspace orthogonal to this space.

Let us call the eigenvectors with eigenvalue 0 as 0-*eigenvectors*. In general, we can change the basis of the matrix in order to exclude the space spanned by 0-eigenvectors and calculate on the reduced matrix of a smaller size. Formally speaking, if we want to obtain a matrix Q of size $n \times n$ having k linearly independent 0-eigenvectors forming $k \times n$ matrix V from a reduced matrix $\widetilde{Q}$ of size $(n-k) \times (n-k)$, we calculate the orthogonal complement $V^\perp$ and have $Q = (V^\perp)^T \cdot \widetilde{Q} \cdot V^\perp$. Then we apply the semidefinite optimization for the matrix $\widetilde{Q}$ and solve the system of linear equations close to the obtained numerical matrix. Since small perturbations are not changing the eigenvalues too much, the obtained matrix $\widetilde{Q}$ has still positive eigenvalues. Therefore the matrix $Q = (V^\perp)^T \cdot \widetilde{Q} \cdot V^\perp$ is positive semidefinite and our proof is valid. The only problem is to determine the eigenvectors with eigenvalue 0,

Note that in the extremal example we have equality in (4) and so also equality in (3). This means that if we choose any placement of the roots R in the extremal construction and calculate the vector v of densities of flags in $\mathcal{G}_m^R$, then we need to have $vQv^T = 0$. In particular, it implies that v is a 0-eigenvector of Q.

In Mantel's theorem the extremal construction is a balanced complete bipartite graph and the considered vector of flags is (,). For every placement of the root it is equal to $(\frac{1}{2}, \frac{1}{2})$. Thus, we know that the vector $(\frac{1}{2}, \frac{1}{2})$ is a 0-eigenvector of Q. An example vector spanning an orthogonal space is a vector $w = (1, -1)$. Therefore if we take $\widetilde{Q}$ as a 1×1 matrix consisting of a value q, we have

$$Q = w^T \cdot \widetilde{Q} \cdot w = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot (q) \cdot (1, -1) = \begin{pmatrix} q & -q \\ -q & q \end{pmatrix}.$$

Substituting $q_{00} = q$, $q_{01} = -q$ and $q_{11} = q$ to (7) we obtain

$$\begin{cases} q = \dfrac{1}{2} \\ x_2 - \dfrac{1}{3}q = \dfrac{1}{6} \\ -\dfrac{1}{3}q = -\dfrac{1}{6}, \end{cases} \tag{8}$$

which has a clear solution $q = \frac{1}{2}$ and $x_2 = \frac{1}{3}$. As $q \geq 0$ the matrix Q is positive semidefinite and gives a proper proof of Mantel's theorem.

In general, we obtain this way a usually overdetermined system of linear equations that has a solution close to the numerical values of $\widetilde{Q}$ obtained from an SDP solver. If we are lucky and there are no 0-eigenvectors linearly independent to the 0-eigenvectors coming from the extremal construction or rounding procedure didn't introduce negative eigenvalues, then the rounded matrix $\widetilde{Q}$ is positive definite, so Q is semidefinite positive, and we have a proper proof.

2.4 Flagmatic

Flagmatic is a software using the flag algebra calculus for computing bounds on densities of graphs, directed graphs and 3-uniform hypergraphs. It was written by Emil Vaughan (as a part of join work with Victor Falgas-Ravry) and is developed further by Jakub Sliacan. Its code is available at `https://github.com/jsliacan/flagmatic` together with many examples on its usage and User's Guide with detailed explanation of Flagmatic syntax to express graphs, constructions and problems. Here we just briefly present how one not familiar with Flagmatic (and programming in general) can use it to bound graph densities.

In order to prove Mantel's theorem considered earlier one need to execute the following code:

```
P = GraphProblem(3, density="2:12", forbid="3:121323")
P.solve_sdp()
P.set_extremal_construction(GraphBlowupConstruction("2:12"))
P.make_exact()
```

The first line is defining the considered problem as a graph problem (Flagmatic also works on oriented graphs and 3-uniform hypergraphs) using flags on 3 vertices, where we want to maximize the edge density among graphs not containing triangles. The next line is executing an SDP solver to obtain the numerical bound. The third line is setting the extremal construction to be the balanced blow-up of an edge, i.e., the balanced complete bipartite graph. This is provided to determine the wanted bound and to generate 0-eigenvectors needed in the rounding procedure described earlier. In some problems (for example in Goodman's bound) there might be more extremal examples. One should provide all possible constructions and sometimes additional 0-eigenvectors in order to properly round the matrices. If one is interested only in not necessarily tight bound, then providing the expected extremal construction or vectors is not needed. Finally, the last line is running the rounding script. If the rounding is successful, then we obtain a formal proof. One may generate a certificate file which contain the list of used graphs, flags and matrices. Such a certificate can be used to validate the proof independently on the Flagmatic software.

This approach is enough to prove many important results, even those that are not reachable by other methods. As an example, consider the Erdős pentagon conjecture from 1984, which states that the maximum number of pentagons in triangle-free graphs is attained in the balanced blow-up of a pentagon. The conjecture was open for almost 30 years and was solved using flag algebras. There is till no other proof known. One can solve this conjecture using Flagmatic by just executing the following lines:

```
P = GraphProblem(5, forbid="3:121323", density="5:1223344551")
P.solve_sdp()
P.set_extremal_construction(GraphBlowupConstruction("5:1223344551"))
P.make_exact()
```

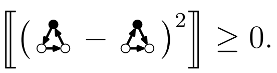
Instead of explicitly providing the extremal construction, one can just provide the target bound, vectors and tight graphs needed for the rounding procedure described in Section 2.3. As an example of such approach we prove that maximum possible density of induced directed stars on 3 vertices is equal to $2\sqrt{3}-3$ with an iterated construction being the extremal. This was proven by Falgas-Ravry and Vaughan, generalized to all one-way directed stars by Huang, and to all directed stars by Hu, Ma, Norin and Wu. Since the answer is not rational, we need to provide the target bound, vectors and graphs appearing in the extremal construction in an extension field of $\mathbb{Q}$. One can use extension fields in Flagmatic, because it is using Sage, where computations in other fields are implemented.

```
P = OrientedGraphProblem(3, density="3:1213")
P.solve_sdp()
x = polygen(QQ)
K.<x> = NumberField(x^2 + x - 1/2, embedding=0.366)
P.set_extremal_construction(field=K, target_bound=4*x - 1)
P.add_sharp_graphs(0, 2, 4, 5)
P.add_zero_eigenvectors(0, matrix(K, [[1-x, 0, x], [x*(1-x), 1-x, x^2]]))
P.make_exact()
```

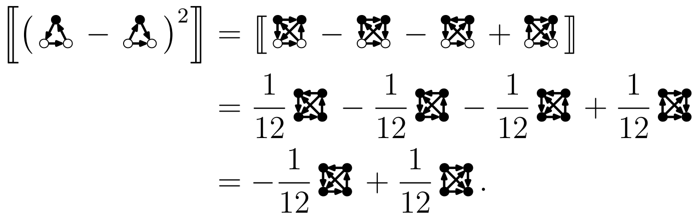
2.5 Applications in other settings

One can use flag algebras not only in the class of graphs, but also for many other discrete structures, including directed graphs, hypergraphs, or graphs with colors on edges or vertices.

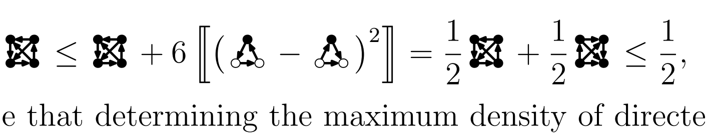
As an example we prove that the maximum possible density of 4-cycles in tournaments in bounded by $\frac{1}{2}$. The maximum is attained in a carousel tournament obtained by placing $2n+1$ vertices in a circle and joining each vertex with the proceeding n vertices. Because of this extremal construction we guess that a useful inequality on a rooted edge should be an equality whenever = , so we try with



Using the definition of multiplication and averaging we obtain



Therefore,



as wanted. We note that determining the maximum density of directed cycles of length $4k$ for $k \geq 3$ is still an open problem.