Hajós’ cycle conjecture for small graphs

Irene Heinrich, Marco Natale and Manuel Streicher

Abstract

Hajós’ conjecture states that an Eulerian graph of order $n$ can be decomposed into at most $\left\lfloor \frac{n-1}{2} \right\rfloor$ edge-disjoint cycles. We describe preprocessing steps, heuristics and integer programming techniques that enable us to verify Hajós’ conjecture for all Eulerian graphs with up to twelve nodes.

1 Introduction

In 1968 Hajós conjectured that the edge set of an Eulerian graph of order $n$ can be decomposed into at most $\left\lfloor \frac{n-1}{2} \right\rfloor$ edge-disjoint cycles (cf. [6]). Granville and Moisiadis [5] showed that the conjecture holds true for graphs of maximum degree 4. Furthermore, Seyffarth [9] verified the conjecture for planar graphs and, even more general, Fan and Xu [3] proved it for projective planar graphs. Moreover, Fuchs, Gellert and Heinrich [4] showed that Hajós’ conjecture holds true for all graphs of pathwidth at most 6.

The main contribution of this paper is to verify Hajós’ conjecture for small graphs:

Theorem 1. Every graph $G$ of order at most 12 fulfills Hajós’ conjecture.

The basic idea behind the verification procedure is as simple as it gets: We run through all graphs of order at most 12 and test if they fulfill Hajós’ conjecture. Nevertheless, the large number of non-isomorphic Eulerian graphs (there are 87723296 such graphs of order 12, cf. [7]) makes this task harder than it seems. We use two different approaches for the tests – minimum counterexamples and explicit computation. Figure 1 illustrates the verification procedure.

In Chapter 2.1, essential properties of a minimum counterexample are discussed. Chapter 2.2 gives an insight to various heuristics which explicitly compute cycle decompositions. Another way of explicit calculation via integer programming techniques is then presented in Chapter 2.3 and Appendix A. Finally, the interaction of all the techniques and further implementation details as well as the computational results are demonstrated in Chapter 3.

2 Methods

2.1 Minimum counterexamples to Hajós’ conjecture

The following theorem summarizes all results on minimum counterexamples which we use to verify Hajós’ conjecture.
Input $G$

Section 2.1

Properties of minimum counterexamples
Fulfils all conditions of Theorem 2?

Section 2.2

Random heuristics
Found desired cycle decomposition?

Output: No minimum counterexample

IP-solver.
Found feasible solution?

Sections 2.3, 2.4

start parallel

$Random long cycle.$
Found desired cycle decomposition?

Output: Found counterexample $G$

Figure 1: Algorithmic scheme of the verification procedure
Theorem 2 (Minimum counterexamples to Hajós’ conjecture [3–5]). A counterexample to Hajós’ conjecture is minimum if it is of minimum size amongst all counterexamples of minimum order. A minimum counterexample $G$ to Hajós’ conjecture is biconnected and has the following properties:

(i) $G$ contains at most one node of degree 2 or 4.

(ii) Neighbours of a degree-2 node in $G$ are adjacent.

(iii) The neighbourhood of a degree-4 node in $G$ induces a regular graph.

(iv) If $v$ is a degree-6 node in $G$ with neighbours $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ and $\{x_1, x_2, x_3, x_4\}$ is a 4-clique in $G$, then $x_5x_6$ is an edge in $G$.

(v) If $G$ contains two adjacent degree-6-nodes $u$ and $v$ with $|N(u) \cap N(v)| = 5$. Then, $N(u) \cap N(v)$ is an independent set in $G$.

(vi) Assume that $G$ contains two adjacent degree-6-nodes $u$ and $v$ with $|N(u) \cap N(v)| = 5$. Then, $G - \{u, v\}$ does not contain a node with at least three neighbours in $|N(u) \cap N(v)|$.

(vii) Assume that $G$ contains two adjacent degree-6-nodes $u$ and $v$ with $|N(u) \cap N(v)| = 4$. Let $\{x_u\} = N(u) \setminus (N(v) \cup \{v\})$ and $\{x_v\} = N(v) \setminus (N(u) \cup \{u\})$. If $G - \{u, v\} - (N(u) \cap N(v))$ contains a path from $x_u$ to $x_v$ then, $N(u) \cap N(v)$ is an independent set in $G$.

Property (i) was discovered by Fan and Xu [3]. Granville and Moisiadis [5] proved (ii) and (iii). Fuchs, Gellert and Heinrich [4] showed the properties (iv), (v), (vi) and (vii).

2.2 Heuristics

We use the following heuristics to verify Hajós’ conjecture. Each of which is applied recursively to the biconnected components of the given graph. When a cycle $C$ is found, we remove the edge set of $C$ from the given graph, add it to the cycle decomposition, and apply the heuristic again.

**Random cycle (RC)** Randomly walk through the graph until a cycle is closed. Return this cycle.

**Random long cycle (RLC)** This is a variation of RC. Whenever a cycle is closed, save its length, but (if possible) continue walking through the graph by choosing another neighbour. If every current neighbour closes a cycle, return the longest cycle seen on the walk.

**Longest distance (LD)** Choose two nodes with maximum distance at random and find two node-disjoint paths connecting them. This can be done by a single maximum flow computation (cf. [8]).

**High degree first (HDF)** If there is exactly one node $v$ of maximum degree, use the Random Long Cycle heuristic starting at $v$ to find the desired cycle. If there are exactly two vertices of maximum degree, use the Longest Distance heuristic to find a cycle containing both vertices. In all other cases, choose any node $v$ of maximum degree and use the Random Long Cycle heuristic starting at $v$. 

3
2.3 An IP-formulation for graphs with nodes of high degree

Regard an Eulerian graph \( G = (V, E) \), with \( n = |V| \) containing a node \( \tilde{v} \in V \) with \( \text{deg}(\tilde{v}) \in \{n - 2, n - 1\} \). If \( G \) fulfils Hajós’ conjecture, there is a cycle decomposition of \( G \) in which every cycle contains \( \tilde{v} \). We can use this fact for a formulation as an integer linear program (IP-HD). For \( S \subset V \) denote by \( \delta(S) \) the set of edges having at least one endpoint in \( S \). Further set \( I := \{1, ..., \lfloor (n - 1)/2 \rfloor \} \) and regard the following constraints.

\[
\text{(IP-HD)} \quad \sum_{i \in I} x_{e,i} = 1 \quad \forall e \in E \quad (1)
\]
\[
\sum_{e \in \delta(v)} x_{e,i} = 2 \cdot y_{v,i} \quad \forall v \in V, i \in I \quad (2)
\]
\[
y_{v,i} = 1 \quad \forall i \in I \quad (3)
\]
\[
n \cdot \sum_{e \in \delta(S)} x_{e,i} \geq (n + 1) \cdot \sum_{v \in S} y_{v,i} \quad \forall i \in I, S \subset V \setminus \{\tilde{v}\} \quad (4)
\]
\[
x_{e,i} \in \{0, 1\} \quad \forall e \in E, i \in I \quad (5)
\]
\[
y_{v,i} \in \{0, 1\} \quad \forall v \in V, i \in I \quad (6)
\]

Variable \( x_{e,i} \) has the meaning that it is equal to 1 when edge \( e \) lies on cycle \( i \). Similarly, \( y_{v,i} \) has the meaning that it is equal to 1 if and only if the node \( v \) lies on the \( i \)th cycle of the decomposition. We now show that a solution to the integer linear problem corresponds to a cycle decomposition of the desired size and vice versa.

Let \( x, y \) be a solution to the constraints above. Set \( C_i := \{e \in E : x_{e,i} = 1\} \). We show that \( C := \{C_i : i \in I\} \) corresponds to a cycle decomposition of \( G \). Constraints (1) ensure that any edge in \( G \) is contained in exactly one set \( C_i \). Thus, it suffices to show that the sets \( C_i \) correspond to cycles in \( G \). To this end fix \( i \in I \) and regard equations (2). If \( y_{v,i} = 1 \) for some \( v \in V \) we must have exactly to edges \( e_1, e_2 \in \delta(v) \) with \( x_{e_1,i} = x_{e_2,i} = 1 \). On the other hand if \( x_{e,i} = 1 \) for some \( e \in E \) we must have \( y_{u,i} = y_{v,i} = 1 \) for \( (u, v) = e \). Thus, each \( C_i \) is empty or corresponds to a 2-regular graph. We now show that equations (3) and (4) ensure that \( C_i \) corresponds to a single cycle or to the empty set. To see this, assume there exists \( C' \subseteq C_i \) corresponding to a cycle. Since \( C_i \) corresponds to a 2-regular graph there also exists \( C \subseteq C_i \) corresponding to a cycle which does not run through \( \tilde{v} \). Define by \( V(C) \) the nodes of this cycle. Since \( \tilde{v} \notin V(C) \) equation (4) must hold for \( V(C) \). On the other hand, with (1) we calculate

\[
n \sum_{e \in \delta(V(C))} x_{e,i} = n \cdot |V(C)| < (n + 1) \cdot |V(C)| = (n + 1) \sum_{v \in V(C)} y_{v,i}.
\]

This contradicts the feasibility of \( x, y \) and we may conclude that \( C_i \) corresponds to a cycle or is empty. Thus, any feasible solution gives a cycle decomposition of \( G \) into at most \( \lfloor (n - 1)/2 \rfloor \) cycles.

Now regard any cycle decomposition \( C \) of \( G \) into at most \( \lfloor (n - 1)/2 \rfloor \) cycles, and let \( C_i \) for \( 1 \leq i \leq \lfloor (n - 1)/2 \rfloor \) be the edge sets of the cycles in \( C \). Note that we allow \( C_i \) to be the empty set. We assign values to the variables in the obvious
way, \( x_{e,i} = 1 \) if and only if \( e \in C_i \) and similarly \( y_{e,i} = 1 \) if and only if \( v \in V(C_i) \). It is straightforward to see that equations (1) and (2) hold. Equations (3) hold because \( \tilde{v} \) must be contained in all cycles in the decomposition. Fix some \( i \in I \) and \( S \subset V \setminus \{ \tilde{v} \} \). If \( S \cap V(C_i) = \emptyset \) both sides of equation (4) equal 0 and it is thereby fulfilled. Let now \( U := S \cap V(C_i) \neq \emptyset \). Since \( \tilde{v} \in C_i \) we have \( U \subseteq V(C_i) \). Thus, we get

\[
\sum_{e \in \delta(S)} x_{e,i} \geq 1 + \sum_{v \in S} y_{v,i}
\]

With this equation we calculate

\[
n \sum_{e \in \delta(S)} x_{e,i} \geq n \left( 1 + \sum_{v \in S} y_{v,i} \right) = n + \sum_{v \in S} y_{v,i} \geq |S| + n \sum_{v \in S} y_{v,i} \geq (n + 1) \sum_{v \in S} y_{v,i},
\]

which shows that all equations (4) are fulfilled.

Summing up we can test Hajós’ conjecture on graphs containing a node of degree \( n - 1 \) or \( n - 2 \) by finding a solution to equations (1) to (4).

### 2.4 General IP-formulation

Since in our experiment the general IP-formulation never managed to find a solution faster than Random long cycle, we skip the description of the general IP here. It can be found in Appendix A.

### 3 Implementation and results

In the previous section we presented the algorithms we used to verify Hajós’ conjecture for all simple Eulerian graphs with twelve or less nodes. We may restrict the verification to the class of all simple and biconnected Eulerian graphs by Theorem 2. We use the programs \texttt{geng} and \texttt{pickg} from McKay and Piperno [7] to create all non-isomorphic, biconnected, Eulerian graphs. All procedures introduced in Section 2 are implemented in \texttt{python} version 2.7.13 [2]. In order to solve the IP-formulations, we used \texttt{Gurobi} version 7.0.2 [1].

Let \( G \) be the graph of order \( n \) in the current iteration. If \( G \) does not fulfill the criteria from Theorem 2, we go on to the next graph. Otherwise, we let all heuristics presented in Section 2.2 run once and check if any heuristic finds a cycle decomposition of size less or equal than \( \lceil \frac{(n - 1)}{2} \rceil \). If we still cannot confirm the conjecture for \( G \) we start an IP-solver and at the same time, we let Random long cycle run repeatedly. If the conditions for the IP in section 2.3 do not apply, we use the general IP-formulation (cf. Appendix A). We abort the program either when the heuristic finds a suitable cycle decomposition or when the IP-solver finishes. Tables 1, 2 and 3 give an overview about which procedure verified the conjecture on how many graphs.

The whole computation took roughly two days in total on a standard PC equipped with 16 GB RAM. However, the memory usage was always less than 2 GB. We stress here that the main challenge in raising the bar to graphs of higher order is not the verification for a single graph but rather the gigantic number of non-isomorphic graphs.
Table 1: $\mathcal{G}_n$ is the set of non-isomorphic Eulerian, biconnected, simple, order $n$ graphs. Columns 4-6 show on how many graphs Hajós’ Conjecture was verified using the according technique.

| $n$ | $|\mathcal{G}_n|$ | Fulfilling Hajós Theorem 2 | Heuristic IP-HD |
|-----|-------------------|-----------------------------|----------------|
| 3   | 1                 | 100%                        | 1              | 0 |
| 4   | 1                 | 100%                        | 1              | 0 |
| 5   | 3                 | 100%                        | 3              | 0 |
| 6   | 7                 | 100%                        | 7              | 0 |
| 7   | 30                | 100%                        | 30             | 0 |
| 8   | 162               | 100%                        | 162            | 0 |
| 9   | 1648              | 100%                        | 1645           | 1 |
| 10  | 30054             | 100%                        | 30030          | 11 |
| 11  | 1136467           | 100%                        | 1134576        | 1846 |
| 12  | 86265865          | 100%                        | 85691978       | 553679 |

Table 2: Detailed overview on usage of the criteria (i) - (vii) from Theorem 2.

| $n$ | Theorem 2 | (i) | (ii) | (iii) | (iv) | (v) | (vi) | (vii) |
|-----|-----------|-----|------|-------|------|-----|------|-------|
| 3   | 1         | 1   | 0    | 0     | 0    | 0   | 0    | 0     |
| 4   | 1         | 1   | 0    | 0     | 0    | 0   | 0    | 0     |
| 5   | 1         | 1   | 0    | 0     | 0    | 0   | 0    | 0     |
| 6   | 30        | 7   | 0    | 0     | 0    | 0   | 0    | 0     |
| 7   | 162       | 150 | 1    | 0     | 0    | 0   | 0    | 0     |
| 8   | 1645      | 1617| 1    | 7     | 8    | 3   | 0    | 9     |
| 9   | 30030     | 2942| 46   | 282   | 83   | 22  | 5    | 150   |
| 10  | 1134576   | 1095272 | 1663 | 23557 | 5673 | 713 | 75   | 7623  |
| 12  | 85691978  | 79468073 | 230553 | 4051363 | 960457 | 54770 | 10047 | 916715 |

Table 3: Detailed overview on usage of the heuristics from section 2.2.

| $n$ | Total | RC | RLC | LD | HDF |
|-----|-------|----|-----|----|-----|
| 9   | 1     | 0  | 1   | 0  | 0   |
| 10  | 11    | 0  | 11  | 0  | 0   |
| 11  | 1846  | 0  | 1700| 11 | 135 |
| 12  | 553679| 4  | 534851 | 687 | 18137 |
A The IP-formulation

We present here the IP-Formulation used to solve the general problem. In order to avoid the introduction of dozens of auxiliary variables and constraints we made use of so called general constraints, which were introduced to Gurobi with version 7.0 (cf. [1]). Among other things, they make it possible to use conjunctions, disjunctions and conditional expressions (see conditions (10)-(12)). Let $G = (V, E)$ be a given Eulerian graph of order $n$. Set $I := \{1, 2, \ldots, [n-1/2]\}$. The set of feasible cycle decompositions can be described as follows:

(IP-Gen)

$$\sum_{e \in I} x_{e,i} = 1 \quad \forall e \in E$$
$$\sum_{e \in \delta(v)} x_{e,i} = 2y_{v,i} \quad \forall v \in V, i \in I$$
$$\bigvee_{v \in S} y_{v,i} = \beta_{S,i} \quad \forall S \subset V, i \in I$$
$$\bigvee_{v \in V \setminus S} y_{v,i} = \gamma_{S,i} \quad \forall S \subset V, i \in I$$
$$\beta_{S,i} \land \gamma_{S,i} = z_{S,i} \quad \forall S \subset V, i \in I$$

If $z_{S,i} = 1$ then

$$x_{e,i} \in \{0, 1\} \quad \forall e \in E, i \in I$$
$$y_{v,i} \in \{0, 1\} \quad \forall v \in V, i \in I$$
$$\beta_{S,i}, \gamma_{S,i}, z_{S,i} \in \{0, 1\} \quad \forall S \subset V, i \in I$$

As in Section 2.3 the variables $x_{e,i}$ and $y_{v,i}$ are set to one if and only if the edge $e$ and the vertex $v$ lie on the cycle corresponding to $C_i$, respectively. Again, for a feasible solution $x, y, z, \beta, \gamma$ let $C_i := \{e \in E : x_{e,i} = 1\}$. Constraints (7) and (8) are the same as (1) and (2), and ensure that each edge is contained in exactly one $C_i$ and that the graph corresponding to $C_i$ is 2-regular. The constraints (9) to (12) guarantee that $C_i$ indeed corresponds to a cycle. To this end, suppose that the graph corresponding to $C_i$ consists of at least two distinct cycles, say $C_i'$ and $C_i''$. Then we have $y_{v,i} = 1$ for each vertex $v \in S := V(C_i')$ and $\beta_{S,i} = 1$. Moreover, it holds that $y_{v,i} = 1$ for each $v \in V \setminus S$ and therefore $\gamma_{S,i} = 1$. Consequently, from (11) it follows that $z_{S,i} = 1$. Constraint (12) now yields that there exist at least two outgoing edges in $S = C_i'$, contradicting the fact that $C_i'$ corresponds to a cycle.

If $C = \{C_i : i \in I\}$ corresponds to a cycle decomposition as in Section 2.3 we set $x_{e,i}$ and $y_{v,i}$ to one if and only if $e \in C_i$ and $v \in V(C_i)$, respectively. Again, constraint (7) and (8) are fulfilled. Let now $S \subset V$. We define $\beta_{S,i}$, $\gamma_{S,i}$ and $z_{S,i}$ as in (9), (10) and (11), respectively. If $S \cap V(C_i) \neq \emptyset$ and $(V \setminus S) \cap V(C_i) \neq \emptyset$ for some $C_i \in C$ then both $\beta_{S,i}$ and $\gamma_{S,i}$ are equal to 1. Thus, their conjunction is 1. Since $S$ as well as $V \setminus S$ contain a vertex of the cycle corresponding to $C_i$, the cut $(S, V \setminus S)$ contains at least two edges of $C_i$. Hence, constraint (12) is fulfilled. On the other hand if $S$ or $V \setminus S$ do not have any vertex in common with the cycle corresponding to $C_i$ then $z_{S,i} = 0$ and (12) is trivially fulfilled.
References

[1] Gurobi. http://www.gurobi.com/.
[2] Python. https://www.python.org/.
[3] G. Fan and B. Xu. Hajós’ conjecture and projective graphs. Discrete Mathematics, 252(1):91 – 101, 2002.
[4] E. Fuchs, L. Gellert, and I. Heinrich. Cycle decompositions of pathwidth-6 graphs. arxiv preprint, 2017.
[5] A. Granville and A. Moisiadis. On Hajós’ conjecture. Congressus Numerantium, 56:183–187, 1987.
[6] L. Lovász. On covering of graphs. Theory of Graphs, pages 231–236, 1968.
[7] B. D. McKay and A. Piperno. Practical graph isomorphism, {II}. Journal of Symbolic Computation, 60(0):94 – 112, 2014.
[8] A. Schrijver. Combinatorial optimization: polyhedra and efficiency. Springer, 2003.
[9] K. Seyffarth. Hajos’ conjecture and small cycle double covers of planar graphs. Discrete mathematics, 101(1):291–306, 1992.