Isoparametric Problems on Graphs and Posets

Jerad Devries, Math and Computer Science
Dr. Sergei L. Bezrukov, Math and Computer Science

ABSTRACT

We consider extremal graph-theoretic problems on regular structures. The problems ask to maximize a certain parameter of the structures in question under the condition that the values of some other parameter(s) are constant. This is where the term isoparametric is taken from. The graph problems we consider are edge-isoperimetric problems, and the poset problems are dealing with the notion of Macaulay posets. The research is split into two parts - theoretical and practical. The practical part of research is developing computer programs that are used to find examples of structures satisfying certain properties. The theoretical part involves proving that the found examples build infinite series of new graph/poset classes.

Part I: Introduction and Summary of Results

1 Edge-isoperimetric Problems in Graphs

Let $G = (V_G, E_G)$ be a graph and $A \subseteq V_G$. Denote

$$\begin{aligned}
\theta_G(A) &= \{(u,v) \in E_G \mid u \in A,\ v \notin A\} \\
E_G(A) &= \{(u,v) \in E_G \mid u,v \in A\} \\
\theta_G(m) &= \{\min_{\substack{A\subseteq E_G\\ |A|=m}} \theta_G(A)\} \\
E_G(m) &= \{\max_{\substack{A\subseteq E_G\\ |A|=m}} |E(A)|\}.
\end{aligned}$$

The *Edge-Isoperimetric Problem (EIP)* consists of finding for a given m, $1 \leq m \leq |V_G|$, a set $A \subseteq E_G$ such that $|E_G(A)| = E_G(m)$. Such sets are called *isoperimetric*. There are two versions of EIP: maximizing $|E_G(A)|$ and minimizing $|\theta_G(A)|$. In this paper we are most concerned with maximizing $|I_G(A)|$. For k-regular graphs both version of EIP are equivalent due to the following identity:

$$2 \cdot |E_G(A)| + |\theta_G(A)| = k \cdot |A|. \tag{1}$$

We say that a graph G with $|V_G| = p$ admits an *isoperimetric order* if there exists a numbering of V_G by numbers $0, 1, \ldots, p$ such that for every m, $1 \leq m \leq p$ the initial segment I_m of length m of this numbering constitutes an isoperimetric set. In other words, $|E_G(I_m)| = E_G(m)$. If an isoperimetric order exists for a graph G then G is called *isoperimetric*.

For graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ we define ***cartesian product*** $G \times H$ as the graph with the vertex set $V_\times$ and the edge set $E_\times$ defined as follows

$$\begin{aligned} V_\times &= V_G \times V_H \\ E_\times &= \{((u,v),(u',v)) \mid (u,u') \in E_G\} \cup \{((u,v),(u,v')) \mid (v,v') \in E_H\}. \end{aligned}$$

Our main interest is to find new isoperimetric graph classes which are cartesian products of some isoperimetric graphs. For this purpose we study for odd values of p the graphs $G_p = K_p - C_p$ obtained from the complete graph K_p by removing from it a spanning cycle. All graphs obtained this way will be isomorphic. However, we assume throughout the paper that the vertices of K_p are labeled by $\{0, 1, 2 \ldots, p-1\}$ and the cycle $0 \to 1 \to 2 \to \cdots \to p-1 \to 0$ is removed from K_p to obtain G_p.

2 Poset Problems

2.1 Maximum weight ideals

Let $P = (S, \leq_P)$ be a finite poset. We say that P is ***ranked*** if there exists a function $r_P : S \mapsto I\!N$ such that $\min_{x \in S} r_P(x) = 0$ and $r_P(x) + 1 = r_P(y)$ whenever $x \leq_P y$ and there exists no $z \in S$ with $x \leq_P z \leq_P y$.

A subset $I \subseteq S$ is called ***ideal*** if for every $x \in I, y \leq_P x$ implies $y \in I$. The maximum weight ideal ***(MWI)*** problem is to find an ideal $A \in S$ such that $|A| = m$ for some given fixed natural number m, $1 \leq m \leq |P|$.

2.2 Macaulay posets

Macaulay posets are posets for which there is an analogue of the classical Kruskal-Katona theorem for finite sets ([6]). For $i = 0, 1, \ldots, r(P)$ let $P_i = \{x \in P \mid r_P(x) = i\}$ and for $X \subseteq P_i$ introduce the shadow of X

$$\Delta(X) = \{x \in P_{i-1} \mid x < y \text{ for some } y \in X\}.$$

A ranked poset P is said to be a *Macaulay poset* if there exists a total order $\preceq$ of its elements such that for every $i = 1, 2, \ldots, r(P)$ and any $X \subseteq P_i$

$$\Delta(C_i(X)) \subseteq C_{i-1}(\Delta(X))$$

where $C_i(X)$ is the set of the first $|X|$ elements of P_i in the order $\preceq$.

In other words, for Macaulay posets any initial segment of P_i of the induced order $\preceq$ has minimum shadow among all subsets of P_i of its size.

3 Importance of Study

This research combines some computational and theoretical results. Although only theoretical results are of greatest interest, computational experiments help to come up with conjectures. This way this research follows a pattern started in [9] with respect to vertex-isoperimetric problems (VIP). It is also of interest to study EIP for cartesian products of graphs and Macaulay posets, since they have a lot of applications, some of which are listed below.

Edge-isoperimetric problems are applicable to many areas including mathematics, computer science, physics, etc. Below we sketch some of the most important applications.

3.1 Applications to Graph Theory

Graph-theoretic applications of edge-isoperimetric problems mostly involve minimum cut problems. However, for regular graphs our version of the EIP is equivalent to the cut minimization problem. Minimum cuts are critical for flows in networks, for graph partitioning and a number of related problems.

Existence of an isoperimetric order provides:

- easy way to construct an optimal set of size m
- solution to the Bandwidth problem

$$bd(G) = \max_{m} \Gamma(m)$$

- solution to the Profile problem

$$pf(G) = \sum_{m} \Gamma(m)$$

- solution to the Bisection Width problem

$$bw(G) = \Theta(|V_G|/2)$$

- solution to the Cutwidth problem

$$cw(G) = \max_{m} \Theta(m)$$

- solution to the Wirelength problem

$$wl(G) = \sum_{m} \Theta(m)$$

- estimate for graph Isoperimetric number(s)

$$i(G) = \min_{0<|A|\leq|V_G|/2} |\Theta(A)|/|A|$$

- Partitioning of graphs in k parts

$$\nabla(G,k) \geq \frac{k}{2} \min\left\{\Theta(\lfloor\frac{|V_G|}{k}\rfloor),\ \Theta(\lceil\frac{|V_G|}{k}\rceil)\right\}$$

More details concerning those problems can be found in survey [2].

Many graph-theoretic problems mentioned above are equivalent or follow from some poset problems [3]. Poset technique brings an additional insight into the graph problems and is in many cases very helpful.

3.2 Applications to Computer Science

Computer Science applications of edge-isoperimetric problems are mostly based on graph partitioning. If we study a process described by differential equations, a common approach is to solve those equations numerically. For this the finite elements methods is usually applied. This method consists in creating a network over the domain by partitioning it into simple figures like triangles or squares. Each node of this network involves an estimation of the solution to the differential equations at this point.

To solve the finite elements method equations efficiently, one uses multiprocessor computers. In order to equate the loads of processors in a multiprocessor system one has to split the underlying network into subnetworks. Each processor will handle computations in the nodes of its subnetwork.

However, inter-processor communications is the bottleneck of modern parallel computers and communication between different processors should be minimized by making all computations as local as possible. Minimization of information exchange between the processors leads to partitioning the network in equal-size parts by breaking the minimum number of network edges. Thus we naturally come to the edge-isoperimetric problems.

4 Overview of Published Results

EIP for cartesian products of graphs has been intensively studied and several surveys are written on that (see, e.g., [2]). The greatest interest for applications is the case of regular graphs. For regular graphs the following two publications are the most relevant ones.

In [9] the authors studied the vertex-isoperimetric problem (VIP) for graphs of a small order. Computational results helped to discover new a class of graphs that admit nested solutions property. This paper is currently in preparation for a journal publication.

In [4] the authors studied EIP on the cartesian products of regular graphs and, in particular, regular bipartite graphs. They proved that removal of a certain number of perfect matchings from a complete graph preserves the nested solutions property for the cartesian powers of obtained graphs. However, in order to obtain a regular graph after removal of a factor from a clique, the clique must be of an even order. This is a restriction of approach in [9].

In our paper we study the complementary case. Namely, we also remove edges from a complete graph. However, the starting graph must be of an odd order. Since such graphs do not possess perfect matchings, the graphs we consider are new and not studied before.

Poset technique as it is mentioned above allows us to look at graphs problems from a different perspective. It is proved in [3] that EIP on graphs is equivalent to MWI problem in a related poset. Namely, it turns out that for every graph one can construct a representing poset such that the existence of nested solutions in EIP for that graph is equivalent to the existence of nested solutions for the obtained poset. The reverse is not true, in general. However, as it is shown in [5], for every poset there is a representing hypergraph.

Therefore, poset problems are, in a sense, more difficult than graph problems. Concerning posets, it is known [2, 6] that if a poset P is Macaulay then it admits nested solutions in MRI problem (the reverse is not true, in general). Furthermore, if we find a graph G admitting the nested solutions property in VIP that satisfies one more reasonable condition (all known graphs classes do satisfy it) then the related poset P is Macaulay. Starting from P one can construct many graphs admitting nested solutions in EIP. The graph studied in [8] nicely correlates without results.

Part II: Computer Programs

5 Software Distribution

Two programs were created to aid in our research: Delta and CoverRelation. Both Delta and CoverRelation are free software: everyone is free to use them and free to redistribute them under the terms of the GNU General Public License published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version (see http://www.gnu.org/licenses/gpl.html).

6 Delta Sequence Program

The δ-sequences (see (2)) of the graphs in question were needed to determine if the cartesian product of a graph with itself admits nested solutions in EIP or not. The computer program Delta (written in C++) was created to compute all δ-sequences of given length that have self-dual posets. Once computed, these sequences were sent as the input to another program for maximizing the number of inner edges found at http://mcs.uwsuper.edu/sb/posets/eip2.php.

Let's look at this excerpt from Delta:

```
void Delta::calcSequences_r(IntVector2D & result,
   IntVector & currentSequence, unsigned int n, bool selfDualOnly)
{
  if (currentSequence.size() >= n)
  {
    ... // add sequence to result
  }

  // one plus the value of the last number in the sequence
  int maxVal = currentSequence.back() + 1;

  for (int i=1; i <= maxVal; i++)
  {
    currentSequence.push_back(i); // add i to end of sequence

    // recursive call
    calcSequences_r(result, currentSequence, n, selfDualOnly);

    currentSequence.pop_back(); // remove i from end of sequence
  }
}
```

This method recursively computes all δ-sequences of length n. The sequences that are not self-dual are filtered out if the parameter *selfDualOnly* is set to true. The parameter *currentSequence* holds the currently computed sequence. It is initialized to the one-element sequence 0. The resulting sequences are stored in the reference parameter *result*.

When the size of *currentSequence* reaches n, *currentSequence* is added to *result*. If *selfDualOnly* is set to true and the sequence is not self-dual, then the method simply returns without adding *currentSequence* to *result*.

Computing the δ-sequence is based on property (3). In the case where the size of *currentSequence* is less than n, the next number in the sequence is added to *currentSequence*. The next number in the sequence is at least 1 and is no more than one plus the current number in the sequence. To compute each sequence the method loops over every possible value for the next number in sequence. The rest of the sequence is computed recursively in the same manner.

7 Covering Relation Program

Definition 1 (see [6]) *Let $P = (S, \leq)$ be a poset. For $x, y \in P$, we say that y covers x, written $x <\cdot\, y$, if $x \leq y$ and there is no $z \in S$ such that $x < z < y$.*

Let Q and R be ranked posets. In order to study the cartesian product of Q and R the computer program CoverRelation (written in Java) was created. This program takes the covering relation of Q and R as input and outputs the covering relation of $Q \times R$. The result of this program was used as input for a program that tests if a given ranked poset is Macaulay found at http://mcs.uwsuper.edu/sb/posets/macaulay.php.

CoverRelation reads two ASCII text file, containing the cover relation of posets P and Q. Empty lines in the file are ignored, and a portion of a line beginning with a # is considered a comment and is ignored. For each element w in the poset the corresponding line must have the following structure:

label $label_1$ $label_2$ ... $label_k$

where *label* is the label for w (a string consisting of letters, digits or underscores only) and $label_2$ ... $label_k$ is a list of labels of vertices covered by w. If w does not cover any elements, then the list is empty. Each label may be separated by one or more spaces.

Let's look at this excerpt from CoverRelation:

```
public PosetCoverRelation cross(PosetCoverRelation rhs)
{
  // The resulting poset
  PosetCoverRelation result = new PosetCoverRelation();
```

```
// get the vertices of the invoking object
Set<String> vertices = coverRelations.keySet();

// get the vertices of the rhs PosetCoverRelation
Set<String> rhsVertices = rhs.coverRelations.keySet();

Iterator<String> i = vertices.iterator();
while (i.hasNext()) // loop over vertices in invoking object
{
  String vertex = i.next();

  // the vertices covered by the current
  // vertex in invoking object
  String[] cover = coverRelations.get(vertex);

  Iterator<String> j = rhsVertices.iterator();
  while (j.hasNext()) // loop over vertices in rhs
  {
    String rhsVertex = j.next();

    String resultVertex = "(" + vertex + ","
                          + rhsVertex + ")";

    // the vertices covered by the current
    // vertex in rhs
    String[] rhsCover = rhs.coverRelations.get(rhsVertex);

    // create array for covered vertices in result
    String[] resultCover
            = new String[cover.length + rhsCover.length];

    int ind = 0; // index in resultCoverVertices

    // add (cover[i],rhsVertex) to resultCover
    for (int k=0; k < cover.length; k++)
    {
      resultCover[ind] = "(" + cover[k] + ","
                         + rhsVertex + ")";
      ind++;
    }
```

```
        // add (vertex,rhsCover[i]) to resultCover
        for (int k=0; k < rhsCover.length; k++)
        {
           resultCover[ind] = "(" + vertex + ","
                              + rhsCover[k] + ")";
           ind++;
        }

        // add the cover relation to the result
        result.coverRelations.put(resultVertex, resultCover);
      }
    }

    return result;
  }
```

The previous method consists of two nested while loops. The outer while loop runs over all vertices in the invoking object (poset P), while the inner while loop runs over the vertices of poset Q (referred as *rhs* in the code). For each vertex $i \in P$ and each vertex $j \in Q$ the element (i, j) in the resulting poset $P \times Q$ covers the elements $(i, j_1), (i, j_2), \ldots, (i, j_m)$ and $(i_1, j), (i_2, j), \ldots, (i_n, j)$ where $j_1, \ldots, j_m$ are the elements covered by j in Q and $i_1, \ldots, i_n$ are the elements covered by i in P.

Part III: Results

8 Brief Overview of Results

The results of this research can be split in two parts: computational and theoretical. Computational results are done with the help of computer to discover new graph and poset classes satisfying some properties. Since computations can be applied only to structures with relatively small number of vertices (below 20), a generalization of computational results to arbitrary number of vertices can only be done theoretically.

To obtain computational results several new programs has to be developed. Since we emphasize on regular graphs, a program for generating regular graphs was needed. As it was mentioned above, we do not distinguish between graphs of the same equivalence class (in the sense of [3]), we only need a program for generating δ-sequences of graphs, which corresponds to the equivalence classes. This is much easier and faster approach. Upon getting the list of all δ-sequences for a given graph order, we apply an existing tool available from the advisor's web site
`http://mcs.uwsuper.edu/sb/posets`
to check them for the nested solutions property. This way we result in just a few δ-sequence for each order.

After getting a list of good δ-sequences we try to find a graph class for them. Many sequences from the obtained list lead to already studied graphs. However, a number of them lead to new graph classes, see below for details.

The theoretical part of the research was to generalize the obtained computational results for graphs with a small number of vertices to arbitrary graphs. We only had time to prove a result for one particular new graph family. However, the results looks promising to a further generalization.

9 New Graph Classes in EIP

Here we present some computational results on symmetric δ-sequences of length 9 - 13. The tables below present along with the δ-sequence one of graph represented by that sequence. The graphs marked with stars are new and not previously studied.

$n = 9$

For $n = 9$ there are 10 symmetric δ-sequences, out of which only 5 satisfy the nested solutions property.

δ-sequence	graph
$(0,1,1,2,2,2,3,3,4)$	interesting new graph*
$(0,1,2,1,2,3,2,3,4)$	$K_3 \times K_3$ or $K_9 - 2C_9^*$ or $K_9 - K_3 \times K_3^*$
$(0,1,2,2,3,4,4,5,6)$	$K_{3,3,3}$ or $K_9 - 3C_3^*$
$(0,1,2,3,3,3,4,5,6)$	$K_9 - C_9^*$
$(0,1,2,3,4,5,6,7,8)$	K_9

Table 1: δ-sequences for $n = 9$

$n = 10$

For $n = 10$ there are 36 symmetric δ-sequences, out of which only 11 satisfy the nested solutions property. However, only one of them leads to a new graph.

δ-sequence	graph
$(0,1,1,1,2,1,2,2,2,3)$	Petersen graph
$(0,1,1,2,1,2,1,2,2,3)$	$C_5 \times P_1$
$(0,1,1,2,2,2,2,3,3,4)$	$K_{5,5} - M$
$(0,1,1,2,2,3,3,4,4,5)$	$K_{5,5}$
$(0,1,2,2,2,3,3,3,4,5)$	$K_{10} - 4M$
$(0,1,2,2,3,3,4,4,5,6)$	$K_{10} - 3M$
$(0,1,2,3,3,4,4,5,6,7)$	$K_{10} - 2C_5^*$
$(0,1,2,3,4,1,2,3,4,5)$	$K_5 \times K_1$
$(0,1,2,3,4,3,4,5,6,7)$	$K_{10} - 2M$
$(0,1,2,3,4,4,5,6,7,8)$	$K_{10} - M$
$(0,1,2,3,4,5,6,7,8,9)$	K_{10}

Table 2: δ-sequences for $n = 10$

$n = 11$

For $n = 11$ there are 28 symmetric δ-sequences, out of which only 5 satisfy the nested solutions property.

δ-sequence	graph
$(0, 1, 2, 2, 2, 3, 4, 4, 4, 5, 6)$	$K_{11} - 2C^*$
$(0, 1, 2, 2, 3, 3, 3, 4, 4, 5, 6)$	previously unknown graph*
$(0, 1, 2, 3, 3, 4, 5, 5, 6, 7, 8)$	previously unknown graph*
$(0, 1, 2, 3, 4, 4, 4, 5, 6, 7, 8)$	$K_{11} - C^*$
$(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$	$K_1 1$

Table 3: δ-sequences for $n = 11$

10 Theoretical Results

10.1 Auxiliary Graph Statements

Denote

$$\delta_G(m) = E_G(m) - E_G(m-1) \quad \text{with} \quad \delta_G(0) = 0. \tag{2}$$

As it is easily seen, $E(m) = \sum_{i=0}^{m-1} \delta_G(i)$. The function δ introduced above plays an important role in finding of isoperimetric graphs. It is easily seen that the function δ satisfies the following property

$$\delta_G(i+1) \leq \delta_G(i) + 1 \quad \text{for } i = 0, 1, \ldots, |V_G| - 2. \tag{3}$$

We need to examine how it relates to the cartesian product operation. Assume graphs G and H are isoperimetric and number their vertices with $0, 1, \ldots$ in accordance with the corresponding isoperimetric orders. Then $V_G \times V_H$ can be considered as the set of pairs of natural numbers. For a set $A \subseteq V_G \times V_H$ we say A is *compressed* if for every $(u, v) \in A$ the vertices $(u-1, v)$ and $(u, v-1)$ (if they exist) are also in A.

Lemma 1 *For a compressed set $A \subseteq V_G \times V_H$ one has*

$$|E_{G\times H}(A)| = \sum_{(u,v)\in A} (\delta_G(u) + \delta_H(v)). \tag{4}$$

Proof.
We prove the lemma by induction on $m = |A|$. For $m = 1$ the number of inner edges spanned by A is 0. On the other hand, the sum in (4) consists of a single term only, whose value is $\delta_G(u)+\delta_H(v) = 0$. This establishes the induction basis.

Assume (4) is valid for any compressed set of size m and let A be a compressed set of size $m+1$. Denote $b = \max_{(0,y)\in A} y$ and $a = \max_{(x,b)\in A} x$. Then $w = (a, b) \in A$ and the set $A' = A \setminus \{w\}$ is compressed. By induction, $|E_{G\times H}(A')| = \sum_{(u,v)\in A'} (\delta_G(u) + \delta_H(v))$. Note that $(a, b+1) \notin A$ and $(a+1, b) \notin A$. Since A is compressed, the sets $\{(x, y) \in A \mid y = b\}$ and $\{(x, y) \in A \mid x = a\}$ are initial segments in G and H respectively. Therefore, $|E_{G\times H}(A) \setminus E_{G\times H}(A')| = \delta_G(b) + \delta_H(a)$. One has

$$\begin{aligned} |E_{G\times H}(A)| &= |E_{G\times H}(A')| + |E_{G\times H}(A) \setminus E_{G\times H}(A')| \\ &= |E_{G\times H}(A')| + (\delta_G(b) + \delta_H(a)) \\ &= \sum_{(u,v)\in A} (\delta_G(u) + \delta_H(v), \end{aligned}$$

which completes the proof. □

Note that the assumption that A is compressed in Lemma 1 is essential. Without this assumption the lemma is not true, in general. The importance of this lemma is provided by the next statement.

Lemma 2 *Let $A \subseteq V_G \times V_H$. Then there exists a compressed set $B \subseteq V_G \times V_H$ such that $|B| = |A|$ and $|E_{G\times H}(B)| \geq |E_{G\times H}(A)|$.*

Proof.
For $A \subseteq G \times H$ and $x = 0, 1, \ldots, |V_G| - 1$ denote

$$\begin{aligned} V_x(a) &= \{(x, y) \in V_G \times V_H \mid x = a\}, \\ V_y(b) &= \{(x, y) \in V_G \times V_H \mid y = b\}, \\ A_x(a) &= A \cap V_x(a), \\ A_y(b) &= A \cap V_y(b). \end{aligned}$$

Note that the subgraph of $G \times H$ induced by the vertex set $V_x(a)$ is isomorphic to H and the one induced by the vertex set $V_y(b)$ is isomorphic to G. We denote those subgraphs by H_a and G_b respectively.

For $A \subseteq G \times H$ denote by $C = \mathcal{C}_x(A)$ the set obtained from A by replacing for every $a = 0, 1, \ldots, |V_G| - 1$ the set $A_x(a)$ with the initial segment of the same size in the subgraph H_a. We show that $|E_{G\times H}(C) \geq |E_{G\times H}(A)|$.

Note that if $(u, v) \in E_{G\times H}$ then either the x-coordinates of u and v are equal, or their y-coordinates are equal. In the first case we call the edge vertical and horizontal otherwise. Hence, all edges induced by the sets A and C are partitioned into horizontal and vertical ones. We first show that the number of vertical edges in C is not small smaller than the one for A. Indeed, since initial segments in every graph H_a induces the maximum number of edges, the number of vertical edges in each component cannot decrease.

Concerning the horizontal edges, if $(a', a'') \notin E_G$ then the number of horizontal edges between $A_x(a')$ and $A_x(a'')$ is 0. Otherwise, the number of those edges does not exceed $\min\{|A_x(a')|, |A_x(a'')|\}$. However, the number of horizontal edges between $C_x(a')$ and $C_x(a'')$ is exactly $\min\{|C_x(a')|, |C_x(a'')|\} = \min\{|A_x(a')|, |A_x(a'')|\}$, since both $C_x(a')$ and $C_x(a'')$ are initial segments in the graph H. Repeating the argument for every $e = (a', a'') \in E_G$ and noting that the horizontal edges corresponding to distinct e's are disjoint, we conclude that $|E_{G\times H}(C) \geq |E_{G\times H}(A)|$.

Finally, we repeat the same argument with respect to the horizontal compression. Similarly, the number of inner edges of the obtained set cannot decrease. After two compressions we obtain a compressed set, which completes the proof. $\square$

Therefore, in order to construct optimal sets we can search for an optimal set in the class of compressed sets only. This reduction significantly reduces the number of candidates to go through.

We introduce one more transformation defined on the vertex set of $G \times G$. Informally, this transformation is just a reflection about the main diagonal. For $A \subseteq V_G \times V_G$ and $(u, v) \in A$ denote

$$\begin{aligned} M(u,v) &= \begin{cases} (v,u), & \text{if } v \geq u \text{ and } (v,u) \notin A \\ (u,v), & \text{otherwise} \end{cases} \\ M(A) &= \bigcup_{(u,v)\in A} M(u,v). \end{aligned}$$

Lemma 3 *Let G be an isoperimetric graph. For any compressed set $A \subseteq V_G \times V_G$ the set $M(A)$ is compressed and $|E_{G\times H}(A)| = |E_{G\times H}(M(A))|$.*

Proof.
For the first statement we need to prove that for any $(u, v) \in M(A)$ the points $(u-1, v)$ and $(u, v-1)$, if they exist, are in $M(A)$. So let $(u, v) \in M(A)$. Assume $u < v$. If $(u, v) \in A$ then $\{(u-1, v), (u, v-1)\} \subseteq A$ since A is compressed. In this case we also have $\{(u-1, v), (u, v-1)\} \subseteq M(A)$. If $(u, v) \notin A$ then $(v, u) \in A$. Since A is compressed, $\{(v-1, u), (v, u-1)\} \subseteq A$. This implies $\{(u-1, v), (u, v-1)\} \subseteq M(A)$. Finally, if $u \geq v$ then $(v, u) \in A$. Since A is compressed, $\{(v-1, u), (v, u-1)\} \subseteq A$. Thus, also in this case we have $\{(u-1, v), (u, v-1)\} \subseteq M(A)$.

For the second statement let $(u, v) \in A$ and $(u', v') = M(u, v)$. We show that $\delta_G(u) + \delta_H(v) = \delta_G(u') + \delta_H(v')$. Indeed, if $u > v$ then $(u', v') = (v, u)$ if $(v, u) \notin A$ and $(u', v') = (u, v)$ otherwise. Then $\delta_G(u) + \delta_H(v) = \delta_G(u') + \delta_H(v')$. If $u \leq v$ then $(u, v) = (u', v')$ and the equality in question holds. Applying this equality for all $(u, v) \in A$ and taking into account Lemma 1 completes the proof. $\square$

The next auxiliary statement relates to the graphs $G_p = K_p - C_p$ that we study. We will often use it in the proofs.

Lemma 4 *For every odd $p \geq 5$ the graph $G_p = K_p - C_p$ is isoperimetric and*

$$\delta_{G_p}(m) = \begin{cases} m, & \text{for } 0 \leq m \leq (p-1)/2 \\ (p-3)/2, & \text{for } m = (p+1)/2 \\ m-3, & \text{for } (p+3)/2 \leq m \leq p-1. \end{cases} \tag{5}$$

Proof.
To prove that G_p is isoperimetric we show that the order $\mathcal{O}_p$ on V_{G_p}, which first labels all even

vertices followed by the odd ones taken in increasing order, is isoperimetric. For example, for $p = 9$ the order $\mathcal{O}_9$ is as follows: $(0, 2, 4, 6, 8, 1, 3, 5, 7)$. Let I_m denote the initial segment of order $\mathcal{O}_p$ of length m. Note that if $m \leq (p-1)/2$ then the subgraph of G_p induced by the vertex set $\{0, 1, \ldots, m-1\}$ is isomorphic to K_m. This follows from the fact that the difference between any two vertices (taken modulo p) is even. Therefore, this subgraph spans the maximum number of edges. Now consider the case $m > (p-1)/2$. First note that for any regular graph G, if $A \subseteq V_G$ is an optimal set then $\overline{A} = V_G \setminus A$ is also optimal. Also note that the graph spanned by I_m is isomorphic to the one spanned by F_m, where F_m is the final segment of order $\mathcal{O}_p$ of length m. Therefore, for $m > (p-1)/2$ the set $\overline{I_m}$ is isomorphic to $I_{m'}$ for $m' = p - m$. However, since $m' \leq (p-1)/2$, the optimality of $\overline{I_m}$ follows from the optimality of $I_{m'}$ which is established above. This proves the isoperimetricity of order $\mathcal{O}_p$.

Concerning the function δ, as it follows from above, for $m \leq (p-1)/2$ the graph spanned by I_m is isomorphic to K_m. Therefore, $\delta_{G_p}(i) = \delta_{K_m}(i) = i - 1$. For $i = (p-1)/2 + 1 = (p+1)/2$ the contribution of the vertex number i in G_p (in order $\mathcal{O}_p$) is one less than in K_i, that is $i-2 = (p-3)/2$. Finally, for $i \geq (p+3)/2$ the contribution of the vertex number i in G_p (in order $\mathcal{O}_p$) is two less than in K_i, that is $i - 3 = (p-3)/2$. $\square$

We put the values of δ_G in a vector δ_G with $|V_G|$ components and call this vector δ-*sequence* of G. According to the last statement, the δ-sequence of G_p is non-decreasing and the middle value is repeated three times. For example, for $p = 9$ we have $\delta_{G_p} = (0, 1, 2, 3, 3, 3, 4, 5, 6)$.

10.2 Main Result

In this section we show that for every $n \geq 2$ the n-th cartesian power of the graph G_p is isoperimetric for $p \geq 7$. As counterexamples show this is not true for $p = 5$ and $n = 2$. The case $p = 3$ is not interesting because the graph $K_3 - C_3$ is empty (i.e. has no edges).

To formulate the main result we need to introduce the *lexicographic order* and formulate a known statement which will reduce our problem to the case $n = 2$ only. Recall, that we associate the set V_{G_p} with the set $[p] = \{0, 1, \ldots, p-1\}$ of vertex labels in an isoperimetric order of G_p, in accordance with Lemma 4.

The lexicographic order $\leq_{\text{lex}}^n$ is defined recursively on n. For $n = 1$ it is identical to the natural order on the set $[p]$. For $n \geq 2$ and $\mathbf{u} = (u_1, dots, u_n) \in [p]^n$ and $(v) = (v_1, \ldots, v_n) \in [p]^n$ we write $\mathbf{u} \leq_{\text{lex}}^n \mathbf{v}$ if and only if

(*i*) $u_1 < v_1$, or

(*ii*) $u_1 = v_1$ and $(u_2, \ldots, u_n) \leq_{\text{lex}}^{n-1} (v_2, \ldots, v_n)$.

Our main result is the following theorem.

Theorem 1 *For every $n \geq 2$ and odd $p \geq 9$ or $p = 5$ the lexicographic order on G_p^n is isoperimetric order.*

It turns out that in order to prove Theorem 1 it is sufficient to consider the case $n = 2$ only. It is true due to the following result known as the ***Local-Global Principle*** for the lexicographic order.

Lemma 5 (Ahlswede, Cai [1])
For $n \geq 2$ the lexicographic order is isoperimetric for G^n iff it is isoperimetric for G^2.

Counterexamples show that Theorem 1 is not true for $p = 7$.

Proof of Theorem 1
Due to Lemma 5 we consider the case $n = 2$. The validity of theorem in case $n = 5$ follows from [8], so we assume $p \geq 9$. The proof is by induction on p. The validity of the theorem for $p = 9$ (induction basis) is provided by software. Assuming that it is true for some $p' \geq 9$ we show that it is also true for $p = p' + 2$.

Let $A \subseteq V_{G_{p+2}}$ be an optimal set. We introduce a series of transformations that convert A into the initial lexicographic segment of the same size without decreasing the number of inner edges. Due to Lemmas 2 and 3 we can assume that A is compressed. This implies, in particular, that $\{(x, y) \in A \mid y = 0\} = \{(0, 0), (1, 0), \ldots, (t, 0)\}$ for some $t \geq 0$.

Case 1. Assume $(0, p - 1) \in A$. Denote

$$\begin{aligned} V' &= \{1, 2, \ldots, p-2\} \times \{1, 2, \ldots, p-2\} \subset V_{G_p \times G_p} \\ A' &= A \cap V' \end{aligned}$$

and let G' be the subgraph of G_p induced by the vertex set V'. It is easily seen that G' is isomorphic to G_{p-2}.

Case 1a. Assume $\{(x, y) \in A \mid x = p - 1\} = \emptyset$. Taking into account Lemma 1, one has

$$|E_{G_p \times G_p}(A)| = \sum_{(x,y)\in A'} (\delta_{G_p}(x) + \delta_{G_p}(y)) + \sum_{(x,y)\in A\setminus A'} (\delta_{G_p}(x) + \delta_{G_p}(y)). \tag{6}$$

Note that $\delta_{G_p}(x) = 1 + \delta_{G'_p}(x - 1)$ for $1 \leq x \leq p - 2$. Therefore, denoting $x' = x - 1$ we get

$$\begin{aligned} \sum_{(x,y)\in A'} (\delta_{G_p}(x) + \delta_{G_p}(y)) &= \sum_{(x,y)\in A'} ((1 + \delta_{G'_p}(x-1) + (1 + \delta_{G'_p}(y-1)) \\ &= \sum_{(x',y')\in A'} (\delta_{G'_p}(x') + \delta_{G'_p}(y')) + 2|A'|. \end{aligned} \tag{7}$$

The sum in the RHS of (7) can be considered as $|E_{G_{p-2}\times G_{p-2}}(A')|$, which along with (6) implies

$$\sum_{(x,y)\in A'} (\delta_{G_p}(x) + \delta_{G_p}(y)) = |E_{G_{p-2}\times G_{p-2}}(A')| + 2|A'|.$$

and

$$|E_{G_p \times G_p}(A)| = \sum_{(x,y)\in A\setminus A'} (\delta_{G_p}(x) + \delta_{G_p}(y)) + \left(|E_{G_{p-2}\times G_{p-2}}(A')| + 2|A'|\right). \tag{8}$$

Denote by B the set obtained from A by replacing its part A' with the initial lexicographic segment of the same size in V'. Also denote $B' = B \cap V'$. Note that $\{(x, y) \in B \mid x = p-1\} = \emptyset$. We show that B is compressed. First observe that B' is compressed since it is an initial lexicographic segment in V'.

To show that B is compressed we need to verify two following conditions. First note that for every element $(x, 1) \in B'$ the element $(x, 0) \in B$. This follows from the fact that $|\{x \mid (x, 1) \in B'\}| = \lceil |A'|/(p-2) \rceil \leq |\{x \mid (x, 1) \in A'\}|$. Next, note that for every element $(x, p-1) \in B$ the element $(x, p-2) \in B'$. This follows from the fact that $|\{x \mid (x, p-2) \in A'\}| \leq \lfloor |A'|/(p-2) \rfloor = |\{x \mid (x, p-2) \in B'\}|$.

Taking into account that $|B'| = |A'|$ we have

$$|E_{G_p \times G_p}(B)| = \sum_{(x,y) \in A \setminus A'} (\delta_{G_p}(x) + \delta_{G_p}(y)) + \Big(|E_{G_{p-2} \times G_{p-2}}(B')| + 2|A'| \Big) . \tag{9}$$

Since $|E_{G_{p-2} \times G_{p-2}}(A')| \leq |E_{G_{p-2} \times G_{p-2}}(B')|$ by induction, (8) and (9) imply $|E_{G_p \times G_p}(A)| \leq |E_{G_p \times G_p}(B)|$. Therefore, B is an optimal set. Denote $m' = |A'|$, $a = \lfloor m'/(p-2) \rfloor$ and $b = m' \bmod (p-2)$. Then $m' = a(p-2) + b$. Further denote $u = \max_{(x,p-1) \in B} x$. Note that $u \leq a+1$ since B is compressed. Let $s = t-a-2$ if $b \neq 0$ and $s = t-a-1$ otherwise.

Case 1aa. Assume $s \leq a+1-u$. In this case we replace the set $S = \{(x, 0) \in B \mid t-s-1 \leq x \leq t\}$ with the set $U = \{(x, p-1) \mid u+1 \leq x \leq u+s\}$ and obtain a set C. The assumptions above imply that $|C| = |B|$ and C is compressed. We show that $|E_{G_p \times G_p}(B)| \leq |E_{G_p \times G_p}(C)|$. Taking into account Lemma 1 it suffices to show that $\sum_{(x,y) \in S} (\delta_{G_p}(x) + \delta_{G_p}(y)) \leq \sum_{(x,y) \in U} (\delta_{G_p}(x) + \delta_{G_p}(y))$. To show this, assume some vertex $(x, 0) \in S$ is replaced with $(x', p-1) \in U$ for some $x' < x$. Since $\delta(G_p) = (0, 1, \ldots, p-3)$, one has $\delta(x) - \delta(x') \leq p-3$. Hence,

$$\begin{aligned}
\delta_{G_p}(x) + \delta_{G_p}(0) &= \delta_{G_p}(x) + 0 \\
&\leq \delta_{G_p}(x') + (p-3) \\
&= \delta_{G_p}(x') + \delta_{G_p}(p-1).
\end{aligned}$$

Therefore, the contribution of each vertex $v \in S$ into the sum (4) does not exceed the contribution of the image of v in U. Thus, C is an optimal set. If C is an initial segment, we are done. If not, consider the set U as above and let T be the set of all vertices of C with maximum x-coordinate, which denote it by d.

If $|T| \leq |U|$, consider the set $D = (C \setminus T) \cup U$. It is easily shown that D is compressed. Denoting $t = |T|$ one has

$$\begin{aligned}
\sum_{(x,y) \in T} (\delta_{G_p}(x) + \delta_{G_p}(y)) &= t \cdot \delta_{G_p}(d) + \sum_{i=0}^{t-1} \delta_{G_p}(i) \\
&\leq t \cdot \delta(p-1) + \sum_{i=0}^{t-1} \delta_{G_p}(i) \\
&\leq \sum_{(x,y) \in U} (\delta_{G_p}(x) + \delta_{G_p}(y)).
\end{aligned}$$

Hence, $|E_{G_p \times G_p}(C)| \leq |E_{G_p \times G_p}(D)|$.

If $|T| > |U|$, we move the vertex $(t-i, d)$ to $(d-i, p-1)$ for $i = 1, 2, \ldots, u = |U|$, and obtain the set F. The desired inequality $|E_{G_p \times G_p}(C)| \leq |E_{G_p \times G_p}(F)|$ would follow from the inequalities

$$\delta_{G_p}(d) + \delta_{G_p}(t-i) \leq \delta_{G_p}(d-i) + \delta_{G_p}(p-1)$$

for $i = 1, 2, \ldots, u$. Since $\delta_{G_p}(t-i) \leq \delta_{G_p}(p-1-i)$ we only need to consider the case $t = p-1$. Hence, we only need to verify the inequalities

$$\delta_{G_p}(d) + \delta_{G_p}(p-1-i) \leq \delta_{G_p}(d-i) + \delta_{G_p}(p-1)$$

for $i = 1, 2, \ldots, u$, which we rewrite as

$$\delta_{G_p}(d) - \delta_{G_p}(d-i) \leq \delta_{G_p}(p-1) - \delta_{G_p}(p-1-i). \tag{10}$$

To show (10) we consider several cases. Assume $d \leq (p-3)/2$. In this case $\delta_{G_p}(d) - \delta_{G_p}(d-i) = i = \delta_{G_p}(p-1) - \delta_{G_p}(p-1-i)$. If $d = (p-1)/2$ then $\delta_{G_p}(d) - \delta_{G_p}(d-i) = i-1 \leq \delta_{G_p}(p-1) - \delta_{G_p}(p-1-i)$. Now assume $d \geq (p+1)/2$ and $d - i < (p-1)/2$. Then we have $\delta_{G_p}(d) - \delta_{G_p}(d-i) = i-2 \leq \delta_{G_p}(p-1) - \delta_{G_p}(p-1-i)$. In the case where $d - i = (p-1)/2$ and $d \geq (p+1)/2$ we have $\delta_{G_p}(d) - \delta_{G_p}(d-i) = i-1 \leq \delta_{G_p}(p-1) - \delta_{G_p}(p-1-i)$. Finally, if $d - i > (p-1)/2$ and $d \geq (p+1)/2$ then $\delta_{G_p}(d) - \delta_{G_p}(d-i) = i = \delta_{G_p}(p-1) - \delta_{G_p}(p-1-i)$.

Case 1ab. Assume $s > a+1-u$. In this case we replace the set $S = \{(x,0) \in B \mid t-u-1 \leq x \leq t-1\}$ with the set $U = \{(x, p-1) \mid u+1 \leq x \leq u+s\}$ and obtain a set C. The assumptions above imply that $|C| = |B|$ and C is compressed. We show that $|E_{G_p \times G_p}(B)| \leq |E_{G_p \times G_p}(C)|$. Indeed, one has

$$\begin{aligned}
\sum_{(x,y) \in S} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right) &\leq \sum_{i=1}^{u} \delta_{G_p}(p-1-i) \\
&\leq u \cdot (p-1) + \sum_{i=0}^{u} \delta_{G_p}(i) \\
&\leq \sum_{(x,y) \in U} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right).
\end{aligned}$$

Case 1b. Assume $\{(x,y) \in A \mid x = p-1\} \neq \emptyset$. We consider several cases depending on the size of the set $L = \{(x,y) \in A \mid x = p-1\}$. Our objective is to replace the elements of L with elements of $(V_{G_p} \times V_{G_p}) \setminus A$ whose x-coordinate is less than $p-1$. This way this case will be reduced to case 1a. Denote by (a,b) the smallest element of $\overline{A}$ in the lexicographic order and let $U = \{(a,y) \mid b \leq y < p\}$. Without loss of generality we assume that the set A has minimum sum of lexicographic numbers of its elements among all compressed optimal sets of its size.

Case 1ba. Assume $|L| \geq (p+1)/2$. Denote

$$\begin{aligned}
V'' &= \{(p+1)/2, \ldots, p-1\} \times \{(p+1)/2, \ldots, p-1\} \subset V_{G_p \times G_p} \\
A'' &= A \cap V''
\end{aligned}$$

and let G'' be the subgraph of G_p induced by the vertex set V''. It is easily seen that G'' is isomorphic to $K_{(p-1)/2}$. Due to a known result surveyed in [2], we can replace the set A'' with the initial lexicographic segment in V'' without losing the number of inner edges in A. It is easy to see that the resulting set is compressed.

If $a > (p+1)/2$ we replace the top $|U|$ of the set L with U. It can be shown that the resulting set is compressed and the number of inner edges in it does not decrease.

If $a = (p+1)/2$ and $|U| = (p-1)/2$ (in which case $A'' = \emptyset$) then we replace the set $\{(i, (p-1)/2) \mid (p-5)/2 \leq i \leq p-1\}$ with the lowest $|U| - 1$ elements of U. The number of inner edges in this case would increase, which contradicts the optimality of A.

Finally, if $a = (p+1)/2$ and $|A''| \neq \emptyset$ we replace the set $\{(i, (p-1)/2) \mid p - |U| \leq i \leq p-1\}$ with U. The resulting set is compressed and has the same number of inner edges as A.

Therefore, the set obtained in this subcase has a smaller sum of lexicographic numbers of its elements, which contradicts our assumption above.

Case 1bb. Assume $|L| = (p-1)/2$. If $|U| \leq |L|$ then we can replace the $|U|$ top elements of L with U without decreasing the number of inner edges in the resulting set and leaving the class of compressed sets. The same argument also works if $a > (p+1)/2$. The only two cases when the replacement of L with U leads to a set with a smaller number of inner edges is $a = b = (p-1)/2$ or $a = (p+1)/2 = b+1$. The set E is compressed for $p \geq 9$.

Denote $r = (p+1)/2$, $M = \{(p-2, j) \mid 0 \leq j < b\}$, and $J = \{(a+1, j) \mid b \leq j < p-1\}$. We replace the set L with the top $|L|$ elements of U. Then replace the set M with J. Finally, move the element $(p-3, (p-1)/2)$ with (a, b). Denote the obtained set by E. We show that $|E_{G_p \times G_p}(A)| = |E_{G_p \times G_p}(E)|$. Indeed, one has

$$\begin{aligned}
\sum_{(x,y)\in E} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right) &= \sum_{(x,y)\in A} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right) + \sum_{(x,y)\in U\setminus\{(a,b)\}} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right) \\
&\quad + \sum_{(x,y)\in J} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right) + \left(\delta(a) + \delta(b)\right) \\
&\quad - \sum_{(x,y)\in L} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right) - \sum_{(x,y)\in M} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right) \\
&\quad - \left(\delta(p-3) + \delta(r-1)\right) \\
&\geq \sum_{(x,y)\in A} \left(\delta_{G_p}(x) + \delta_{G_p}(y)\right).
\end{aligned}$$

The set E has no elements in the last column (whose x-coordinate is $p-1$), so in order to reduce E to the initial segment we can use transformations described in case 1a.

Case 1bc. Assume $|L| < (p-1)/2$. If $((p-1)/2, (p-1)/2) \in A$ then denote

$$\begin{aligned}
V''' &= \{(p+1)/2, \ldots, p-1\} \times \{0, \ldots, (p-3)/2\} \subset V_{G_p \times G_p} \\
A''' &= A \cap V'''
\end{aligned}$$

If we replace A''' with the initial lexicographic segment in V''' the resulting set will be compressed and the number of inner edges in the resulting set does not decrease. Denote

$$\begin{aligned} V'''' &= \{0,\ldots,(p-3)/2)\} \times \{(p+1)/2,\ldots,p-1\}\} \subset V_{G_p \times G_p} \\ A'''' &= A \cap V'''' \end{aligned}$$

Note the subgraphs of $G_p \times G_p$ induced by the vertex sets V''' and V''' are isomorphic to the products of complete graphs $K_{(p-1)/2} \times K_{(p-1)/2}$. Therefore, if $|A''''| < (p-1)^2/4$ then applying the additivity property of those cartesian products (see [2]) we can pump the vertices of the last column of A''' to A'''' without decreasing the number of inner edges. The obtained set, however, will have a smaller sum of the lexicographic numbers of its elements.

If $|A''''| = (p-1)^2/4$ (that is, $|A''''| = |V''''|$) then we can move the vertices from the last column of A''' to fill the first unfilled column of A. One can show that this leads to a set with a non-smaller number of inner edges. This way we will end up in a set with a smaller value of the sum of lex numbers of its elements (which contradicts our assumption) or obtain a compressed optimal set satisfying assumptions of case 1a. This way we can apply the transformations of case 1a to convert A into the initial lex segment of its size.

Finally, if $((p-1)/2,(p-1)/2) \notin A$ then this vertex is in $\overline{A}$ (the complement of A). Since the optimality of A is equivalent to the optimality of $\overline{A}$, and $\overline{A}$ satisfies the assumptions of case 1bc, we can apply to $\overline{A}$ the above presented arguments of this case.

Case 2. Assume now that $(0,p-1) \notin A$. Due to Lemma 3 we can assume that $M(A) = A$. In this case $(p-1,0) \notin A$. Indeed, if $(p-1,0) \in A$ then the condition $M(A) = A$ implies $(0,p-1) \in A$. It follows from (1) that $\overline{A} = V_{G_p} \setminus A$ is also an optimal set since maximizing $|E_G(A)|$ is equivalent to minimizing $|\theta_G(A)| = |\theta_G(\overline{A})|$ which is equivalent to maximizing $|E_G(\overline{A})|$. However, $(0,p-1) \in \overline{A}$ and $(p-1,0) \in \overline{A}$. Therefore, we can apply the transformations described in *case 1* to $\overline{A}$ to convert it into the initial lexicographic segment of its size without increasing the number of inner edges. This way the set A will be transformed into a set which is isomorphic to an initial lexicographic segment. Applying (1) again, we deduce that A is optimal. $\square$

11 Further Research Directions

We conjecture that the obtained theoretical results can be generalized to the case when more than one cycle is removed from a complete graph of an odd degree. Actually, we conjecture that up to $|V_G|/4$ can be removed without losing the nested solutions property for the cartesian products. This conjecture agrees with the results proved in [4] where up to $|V_G|/2$ perfect matchings were removed from a clique. Note that removing A corresponds to removing 2 matchings. This conjecture was supported for all orders of complete graphs up to 18. Computations for the product of graphs of degree 20 takes about 30 minutes. Therefore, it looks reasonable to attempt to prove the above conjecture.

It is also possible to remove cycle from complete graphs of an even order. The resulting graphs will be regular. We hope that the graph family of that kind that we experienced for $n = 10$ is infinite and the proof can be obtained by using a similar approach as in Theorem 1.

This way we have completed study of small graphs with nested solutions in EIP. We could only do some preliminary results for posets, so this direction should be further studied. This would form a basis for a good student project.

In this research and in [4] we removed perfect matchings and cycles from complete graphs to obtain a regular graph. Our computational experiments show that for small graph orders (up to 18) those are the only graph families of regular graphs admitting the nested solutions property. We suspect, however, that for larger graph orders there might be other families. Thus, a reasonable question to investigate would be what regular graphs one can remove from a clique so that the cartesian powers of the obtained graphs satisfy the nested solutions property in EIP.

12 Conclusion

We have studied several graph classes for admitting nested solutions in the edge-isoperimetric problem and found several new ones with the number of vertices up to 20. This way we discovered a new, hopefully infinite, two-parametric class of such graphs.

We proved that if we fix one parameter for the above mentioned family (namely if we remove a cycle from a complete graph of an odd degree), then any cartesian power of the obtained graphs admits nested solutions in EIP. We conjecture that the same holds for removing up to $p/4$ cycles from a complete graph K_p.

References

[1] R. Ahlswede, N. Cai, General edge-isoperimetric inequalities, Part II: A local-global principle for lexicographic solution, *Europ. J. Combin.* **18** (1997), 479–489.

[2] S.L. Bezrukov, Edge Isoperimetric Problems on Graphs, in: *Graph Theory and Combinatorial Biology*, Bolyai Soc. Math. Stud. **7**, 1. Lovasz, A. Gyarfas, G.O.H Katona, A. Recski, 1. Szekely eds., Budapest 1999, 157–197.

[3] S.L. Bezrukov, On an Equivalence in Discrete Extremal Problems, *Discr. Math.* **203** (1999), no. 1, 9–22.

[4] S.L. Bezrukov, R. Elsässer, Edge isoperimetric problems for cartesian powers of regular graphs, *Theor. Comput. Sci.* **307** (2003), 473–492.

[5] S.L. Bezrukov, R. Battiti, On partitioning of hypergraphs, *Discrete Math.* **307** (2007), no. 14, 1737–1753.

[6] S.L. Bezrukov, U. Leck, Macaulay Posets, *The Electronic Journal of Combinatorics*, DS12, (2004).

[7] S.L. Bezrukov, T.J. Pfaff, V.P. Piotrowski, A new approach to Macaulay posets, *Journal of Comb. Theory*, **A-105** (2004), no. 2, 161–184.

[8] S.L. Bezrukov, M. Rius, O. Serra, The vertex-isoperimetric problem for the powers of the diamond graph, *Discrete Math.* **308** (2008), No. 11, 2059–2342.

[9] S.L. Bezrukov, O. Serra, Vertex-isoperimetric graphs of order 5 - 20, *in preparation*.