Generators of Jacobian Groups of Graphs

Aurash Vatan
Mentor: Xiaomeng Xu

May 19-20, 2018
MIT PRIMES Conference

What are Graphs?

Definition (Graph)

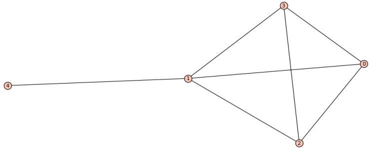
A graph G is an ordered pair (V, E) where V is a set of vertices and E is a collection of two-element subsets of V with repetition allowed.

What are Graphs?

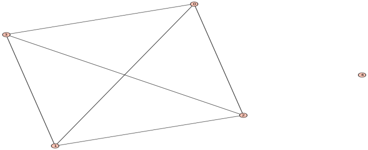
Definition (Graph)

A graph G is an ordered pair (V, E) where V is a set of vertices and E is a collection of two-element subsets of V with repetition allowed.

Connected graph:



Unconnected graph:

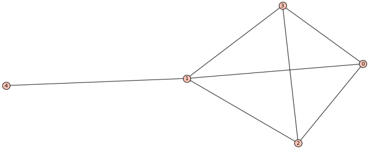


What are Graphs?

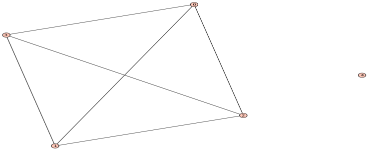
Definition (Graph)

A graph G is an ordered pair (V, E) where V is a set of vertices and E is a collection of two-element subsets of V with repetition allowed.

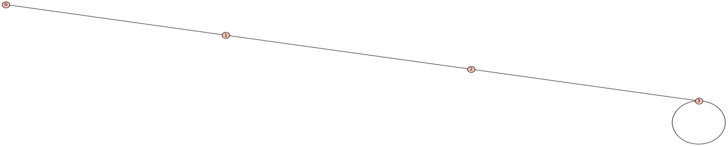
Connected graph:



Unconnected graph:



Graph with loops:



Motivation

1. To study geometric objects, useful to assign algebraic structures to them
2. Eg. symmetry groups of shapes, groups on elliptic curves, etc.
3. Inspired by that, we do the same for graphs

Motivation

1. To study geometric objects, useful to assign algebraic structures to them
2. Eg. symmetry groups of shapes, groups on elliptic curves, etc.
3. Inspired by that, we do the same for graphs
4. The elements of these groups will be called divisors

Divisors

Definition (Divisor)

A divisor D of a graph G is an assignment of integers $D(v_i)$ to its vertices v_i.

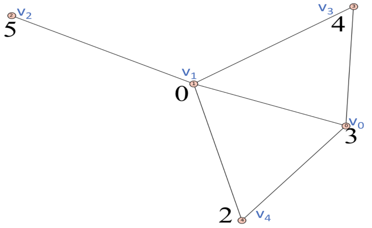
Divisors

Definition (Divisor)

A divisor D of a graph G is an assignment of integers $D(v_i)$ to its vertices v_i.

Here are some examples:

$D_1(G) = 3v_0 + 0v_1 + 5v_2 + 4v_3 + 2v_4$:



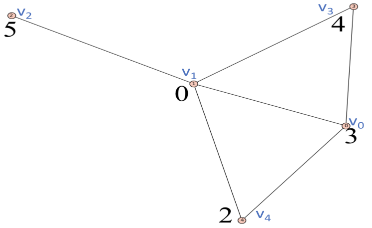
Divisors

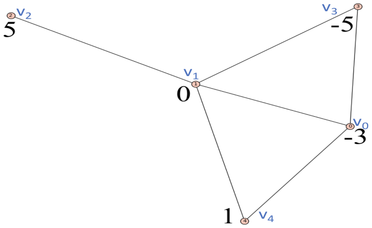
Definition (Divisor)

A divisor D of a graph G is an assignment of integers $D(v_i)$ to its vertices v_i.

Here are some examples:

$D_1(G) = 3v_0 + 0v_1 + 5v_2 + 4v_3 + 2v_4$: $D_2(G) = -3v_0 + 0v_1 + 5v_2 - 5v_3 + 1v_4$:





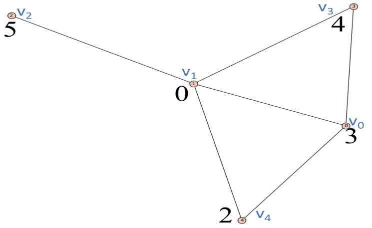
Divisors

Definition (Divisor)

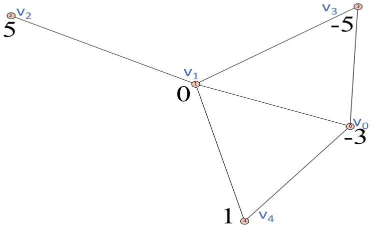
A divisor D of a graph G is an assignment of integers $D(v_i)$ to its vertices v_i.

Here are some examples:

$D_1(G) = 3v_0 + 0v_1 + 5v_2 + 4v_3 + 2v_4$:



$D_2(G) = -3v_0 + 0v_1 + 5v_2 - 5v_3 + 1v_4$:



We can add the divisors as follows:

$D_1 + D_2 = 0v_0 + 0v_1 + 10v_2 - 1v_3 + 3v_4$.

Divisors

Theorem

The set of divisors of a graph G form an Abelian group $\mathrm{Div}(G)$ under this operation.

There is an identity, inverses exist, the operation is closed, associative, and commutative.

Divisors

Theorem

The set of divisors of a graph G form an Abelian group $\mathrm{Div}(G)$ *under this operation.*

There is an identity, inverses exist, the operation is closed, associative, and commutative.

Definition

The degree of a divisor D is the sum $\sum_i D(v_i)$.

Divisors

Theorem

The set of divisors of a graph G form an Abelian group $\mathrm{Div}(G)$ under this operation.

There is an identity, inverses exist, the operation is closed, associative, and commutative.

Definition

The degree of a divisor D is the sum $\sum_i D(v_i)$.

Theorem

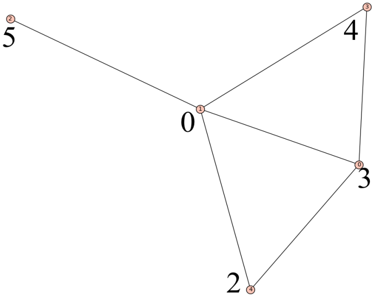
The set of divisors with degree 0 form a subgroup $\mathrm{Div}_0(G)$ of $\mathrm{Div}(G)$.

Sum of degree zero divisors must have degree zero. Inverses of degree zero divisors must have degree zero.

Chip Firing

- Think of a divisor as assigning a number of "chips" to each node
- When a node "fires," it sends one chip along each edge
- Note that this operation preserves the degree of the divisor

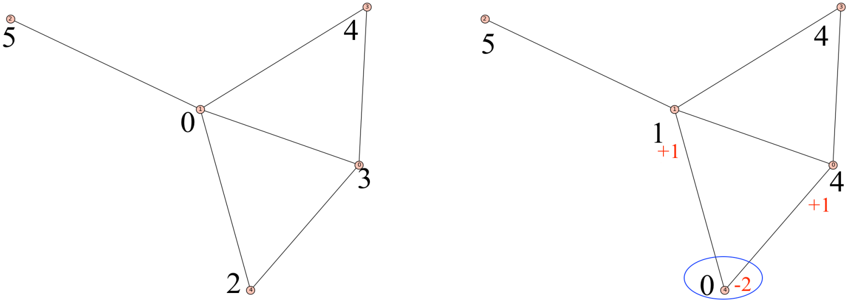
Example:



Chip Firing

- Think of a divisor as assigning a number of “chips” to each node
- When a node “fires,” it sends one chip along each edge
- Note that this operation preserves the degree of the divisor

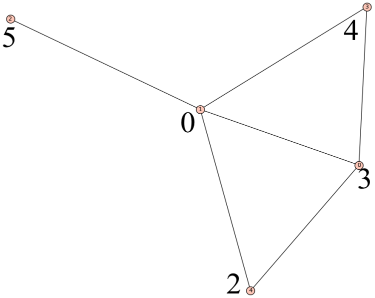
Example:

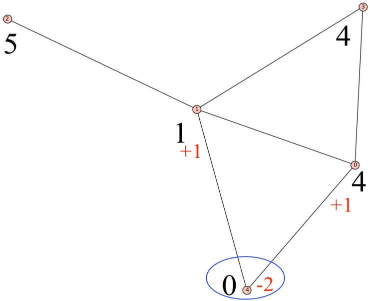


Chip Firing

- Think of a divisor as assigning a number of "chips" to each node
- When a node "fires," it sends one chip along each edge
- Note that this operation preserves the degree of the divisor

Example:





Definition (Firing Script)

A firing script σ is an integer vector $\sigma \in \mathbb{Z}^n$ whose entries specify the number of times each node of a divisor should be fired.

Chip Firing as an Equivalence Relation

Definition

Let A and B be divisors on a graph G. Then $A \sim B$ if and only if there exists a firing script σ that takes A to B.

Chip Firing as an Equivalence Relation

Definition

Let A and B be divisors on a graph G. Then $A \sim B$ if and only if there exists a firing script σ that takes A to B.

1. For a graph G, the set $\mathrm{Prin}(G)$ of all divisors equivalent to 0 is a subgroup of $\mathrm{Div}(G)$.

Chip Firing as an Equivalence Relation

Definition

Let A and B be divisors on a graph G. Then $A \sim B$ if and only if there exists a firing script σ that takes A to B.

1. For a graph G, the set $\text{Prin}(G)$ of all divisors equivalent to 0 is a subgroup of $\text{Div}(G)$.
2. It follows that $\text{Prin}(G)$ is a subgroup of $\text{Div}_0(G)$.

Chip Firing as an Equivalence Relation

Definition

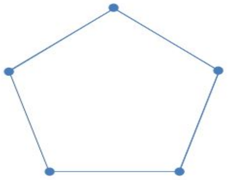
Let A and B be divisors on a graph G. Then $A \sim B$ if and only if there exists a firing script σ that takes A to B.

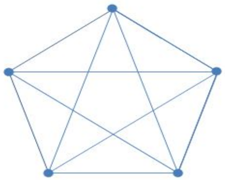
1. For a graph G, the set $\mathrm{Prin}(G)$ of all divisors equivalent to 0 is a subgroup of $\mathrm{Div}(G)$.
2. It follows that $\mathrm{Prin}(G)$ is a subgroup of $\mathrm{Div}_0(G)$.

Definition (Jacobian Group)

Let G be a graph. The Jacobian group Jac(G) is defined as $\mathrm{Div}_0(G)/\,\mathrm{Prin(G)}$.

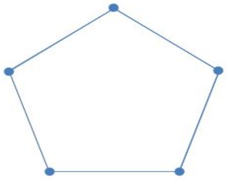
Examples of Jacobians

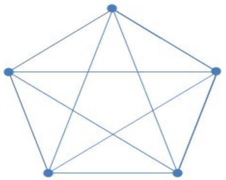




1. The linear graph L_n has a trivial Jacobian.

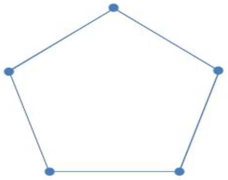
Examples of Jacobians

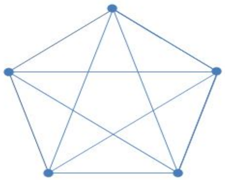




1. The linear graph L_n has a trivial Jacobian.
2. The cyclic graph C_n has $\mathrm{Jac(K_n)} = \mathbb{Z}_n$

Examples of Jacobians





1. The linear graph L_n has a trivial Jacobian.
2. The cyclic graph C_n has $\mathrm{Jac(K_n)} = \mathbb{Z}_n$
3. The complete graph K_n has $\mathrm{Jac(K_n)} = \mathbb{Z}_n^{n-2}$

The Laplacian

Definition

Let G be a graph. Let Δ be a diagonal matrix where $\Delta_{(i,i)}$ equals the number of edges incident to vertex i. Let A be the adjacency matrix of G. Then the Laplacian $L := \Delta - A$.

1. Properties of the Jacobian can be derived from the Laplacian, and so it is key to computation and proofs.

The Laplacian

Definition

Let G be a graph. Let Δ be a diagonal matrix where $\Delta_{(i,i)}$ equals the number of edges incident to vertex i. Let A be the adjacency matrix of G. Then the Laplacian $L := \Delta - A$.

1. Properties of the Jacobian can be derived from the Laplacian, and so it is key to computation and proofs.
2. If we take the zero-divisor and fire the nodes by σ, the resulting divisor is $D = L\sigma$.

The Laplacian

Definition

Let G be a graph. Let Δ be a diagonal matrix where $\Delta_{(i,i)}$ equals the number of edges incident to vertex i. Let A be the adjacency matrix of G. Then the Laplacian $L := \Delta - A$.

1. Properties of the Jacobian can be derived from the Laplacian, and so it is key to computation and proofs.
2. If we take the zero-divisor and fire the nodes by σ, the resulting divisor is $D = L\sigma$.
3. $\det(L) = |\operatorname{Jac}(G)|$.

The Laplacian

Definition

Let G be a graph. Let Δ be a diagonal matrix where $\Delta_{(i,i)}$ equals the number of edges incident to vertex i. Let A be the adjacency matrix of G. Then the Laplacian $L := \Delta - A$.

1. Properties of the Jacobian can be derived from the Laplacian, and so it is key to computation and proofs.
2. If we take the zero-divisor and fire the nodes by σ, the resulting divisor is $D = L\sigma$.
3. $\det(L) = |\operatorname{Jac}(G)|$.
4. The entries of the SNF of the Laplacian are the invariants of Jac(G).

Previous Results

1. We want to relate properties of G to properties of $\mathrm{Jac}(G)$

Previous Results

1. We want to relate properties of G to properties of $\mathrm{Jac}(G)$
2. Lorenzini determined effect of adding or removing an edge of G on the size of its minimal generating sets

Theorem (Lorenzini 1989)

Let G be a connected graph. Let G' be a connected graph formed by removing an edge of G. Then the size of the minimal generating set of $\mathrm{Jac}(G')$ differs from that of $\mathrm{Jac}(G)$ by at most 1.

Previous Results

1. Lorenzini does not provide a method for finding generators

Previous Results

1. Lorenzini does not provide a method for finding generators
2. Natural to start with simplest non-trivial divisors, δ_{xy}

Previous Results

1. Lorenzini does not provide a method for finding generators
2. Natural to start with simplest non-trivial divisors, δ_{xy}

Definition (δ_{xy} divisors)

The δ_{xy} divisor is $D(v_x) = 1$, $D(v_y) = -1$, and $D(v_i) = 0$ for all $i \neq 1, -1$.

Previous Results

1. Lorenzini does not provide a method for finding generators
2. Natural to start with simplest non-trivial divisors, δ_{xy}

Definition (δ_{xy} divisors)

The δ_{xy} divisor is $D(v_x) = 1$, $D(v_y) = -1$, and $D(v_i) = 0$ for all $i \neq 1, -1$.

Theorem (Brandfonbrener et. al. 2017)

Let G be a graph and G_1 be a connected graph formed by adding/removing the edge between x and y. The divisor δ_{xy} is a generator of $\mathrm{Jac}(G)$ *if and only if*

$$gcd(|\mathrm{Jac}(G)|, |\mathrm{Jac}(G_1)|) = 1.$$

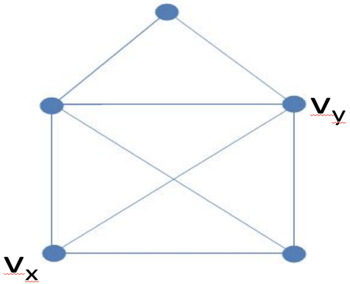
Previous Results

Theorem (Brandfonbrener et. al. 2017)

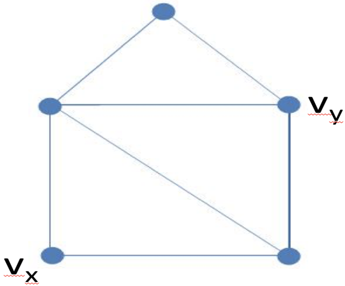
Let G be a graph and G_1 be a connected graph formed by adding/removing the edge between x and y. The divisor δ_{xy} is a generator of $\mathrm{Jac}(G)$ *if and only if*

$$gcd(|\,\mathrm{Jac}(G)|, |\,\mathrm{Jac}(G_1)|) = 1.$$

G with $|\,\mathrm{Jac}(G)| = 40$:



G_1 with $|\,\mathrm{Jac}(G_1)| = 21$:



New Procedure

1. One ultimate goal is to produce minimal generating sets of divisors

New Procedure

1. One ultimate goal is to produce minimal generating sets of divisors
2. Working with δ-divisors is a place to start

New Procedure

1. One ultimate goal is to produce minimal generating sets of divisors
2. Working with δ-divisors is a place to start
3. We developed a procedure which we conjecture produces a smallest generating set consisting of only δ-divisors

New Procedure

1. Choose a $\delta_{x_1 y_1}$ that generates a largest subgroup of $\mathrm{Jac}(G)$

New Procedure

1. Choose a $\delta_{x_1 y_1}$ that generates a largest subgroup of $\mathrm{Jac}(G)$
2. If x and y are connected, remove edge xy to form G_1. Otherwise, add edge xy

New Procedure

1. Choose a $\delta_{x_1 y_1}$ that generates a largest subgroup of $\mathrm{Jac}(G)$
2. If x and y are connected, remove edge xy to form G_1. Otherwise, add edge xy
3. Choose a $\delta_{x_2 y_2}$ that generates a largest subgroup of $\mathrm{Jac}(G_1)$

New Procedure

1. Choose a $\delta_{x_1 y_1}$ that generates a largest subgroup of Jac(G)
2. If x and y are connected, remove edge xy to form G_1. Otherwise, add edge xy
3. Choose a $\delta_{x_2 y_2}$ that generates a largest subgroup of Jac(G_1)
4. If x and y are connected, remove edge xy to form G_2. Otherwise, add edge xy

New Procedure

1. Choose a $\delta_{x_1 y_1}$ that generates a largest subgroup of $\mathrm{Jac}(G)$
2. If x and y are connected, remove edge xy to form G_1. Otherwise, add edge xy
3. Choose a $\delta_{x_2 y_2}$ that generates a largest subgroup of $\mathrm{Jac}(G_1)$
4. If x and y are connected, remove edge xy to form G_2. Otherwise, add edge xy
5. Repeat until $\langle \delta_{x_n y_n} \rangle = \mathrm{Jac}(G_{n-1})$

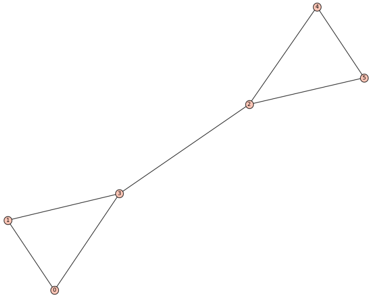
New Procedure

1. Choose a $\delta_{x_1 y_1}$ that generates a largest subgroup of $\mathrm{Jac}(G)$
2. If x and y are connected, remove edge xy to form G_1. Otherwise, add edge xy
3. Choose a $\delta_{x_2 y_2}$ that generates a largest subgroup of $\mathrm{Jac}(G_1)$
4. If x and y are connected, remove edge xy to form G_2. Otherwise, add edge xy
5. Repeat until $\langle \delta_{x_n y_n} \rangle = \mathrm{Jac}(G_{n-1})$

Conjecture

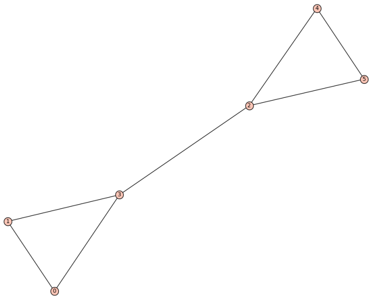
If the above procedure terminates, then $\langle \delta_{x_1 y_1}, \ldots, \delta_{x_n y_n} \rangle = \mathrm{Jac}(G)$.

New Procedure Example



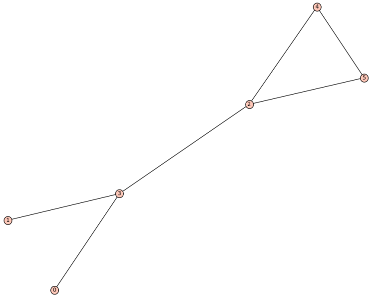
1. $|\operatorname{Jac}(G)| = 9$. The largest subgroups generated by a δ_{xy} have order 3. Set δ_{xy1} equal to one such divisor, for example δ_{01}.

New Procedure Example



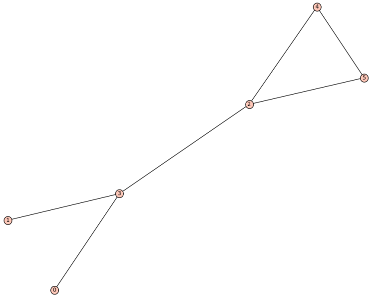
1. $|\operatorname{Jac}(G)| = 9$. The largest subgroups generated by a δ_{xy} have order 3. Set δ_{xy1} equal to one such divisor, for example δ_{01}.
2. G contains a $0 - 1$ edge, so remove it to form G_1.

New Procedure Example



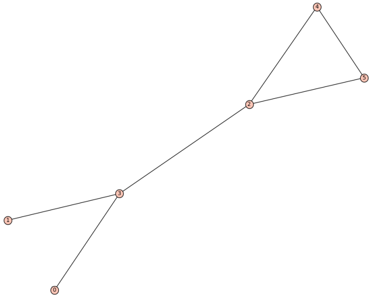
1. $|\operatorname{Jac}(G)| = 9$. The largest subgroups generated by a δ_{xy} have order 3. Set δ_{xy1} equal to one such divisor, for example δ_{01}.
2. G contains a $0-1$ edge, so remove it to form G_1.
3. $|\operatorname{Jac}(G_1)| = 3$. The largest subgroups generated by a δ_{xy} have order 3. Set δ_{xy2} equal to one such divisor, for example δ_{45}.

New Procedure Example



1. $|\operatorname{Jac}(G)| = 9$. The largest subgroups generated by a δ_{xy} have order 3. Set δ_{xy1} equal to one such divisor, for example δ_{01}.
2. G contains a $0-1$ edge, so remove it to form G_1.
3. $|\operatorname{Jac}(G_1)| = 3$. The largest subgroups generated by a δ_{xy} have order 3. Set δ_{xy2} equal to one such divisor, for example δ_{45}.
4. $|\operatorname{Jac}(G1)| = |\langle\delta_{45}|\rangle$, so the process terminates.

New Procedure Example



Example

Computationally, we can check that $\langle \delta_{01}, \delta_{45} \rangle = \mathrm{Jac}(G)$.

1. $|\mathrm{Jac}(G)| = 9$. The largest subgroups generated by a δ_{xy} have order 3. Set δ_{xy1} equal to one such divisor, for example δ_{01}.
2. G contains a $0 - 1$ edge, so remove it to form G_1.
3. $|\mathrm{Jac}(G_1)| = 3$. The largest subgroups generated by a δ_{xy} have order 3. Set δ_{xy2} equal to one such divisor, for example δ_{45}.
4. $|\mathrm{Jac}(G1)| = |\langle \delta_{45}| \rangle$, so the process terminates.

Computational Support

1. Wrote software to apply procedure to randomly generated graphs.
2. About 1000 across graphs of $4 - 10$ nodes.
3. In 99 percent of trials, the process terminated.
4. All terminated trials resulted in a generating set for the original graph.

Future Research

1. Prove that when the procedure terminates, it produces a generating set for the Jacobian of the original graph.
2. With what probability does the procedure terminate?
3. With what probability does it produce a generating set with minimal order

Acknowledgements

I'd like to thank

1. My mentor, Dr. Xiaomeng Xu
2. Dr. Dhruv Ranganathan
3. The PRIMES program and MIT Math Department
4. Dr. Tanya Khovanova
5. My parents