Rose Conjecture: Total Acquisition in Diameter Two Graphs

Grant Shirley [1] Lybitina Koene [2]

[1]East Tennessee State University

[2]Virginia Tech

September 4, 2024

Outline

- Background
- Past Attempts
- Probability Approach
- Induction Approach
- Edge Swap Algorithm
- Locking Algorithm
- Digraph Approach
- Next Steps

Total Acquisition and Diameter

Total Acquisition Process

The total acquisition process works as follows:

Total Acquisition and Diameter

Total Acquisition Process

The total acquisition process works as follows:

- Given a graph G, every vertex is given a weight of 1, $w(v) = 1$.
- A total acquisition move may be performed between two adjacent vertices (u, v) if $w(u) \geq w(v)$.
- This total acquisition move transfers all the weight of v to u.
- This process continues until no more moves are possible.

Total Acquisition and Diameter

Total Acquisition Process

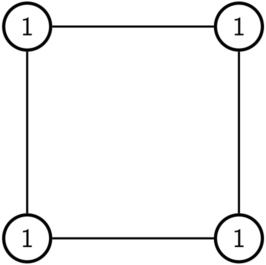
The total acquisition process works as follows:

- Given a graph G, every vertex is given a weight of 1, $w(v) = 1$.
- A total acquisition move may be performed between two adjacent vertices (u, v) if $w(u) \geq w(v)$.
- This total acquisition move transfers all the weight of v to u.
- This process continues until no more moves are possible.

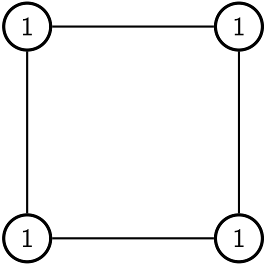
Definition

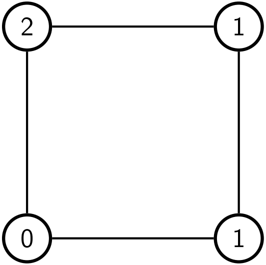
The total acquisition number denoted as $a_t(G)$ is the minimum number of vertices with nonzero weight after a total acquisition process has been performed.

Total Acquisition Example - C_4

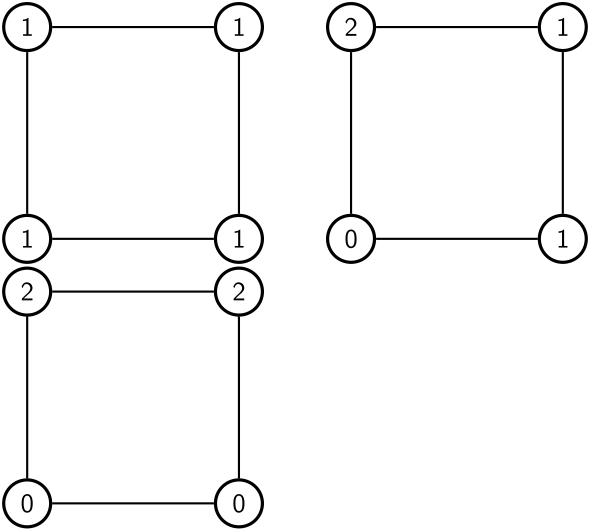


Total Acquisition Example - C_4

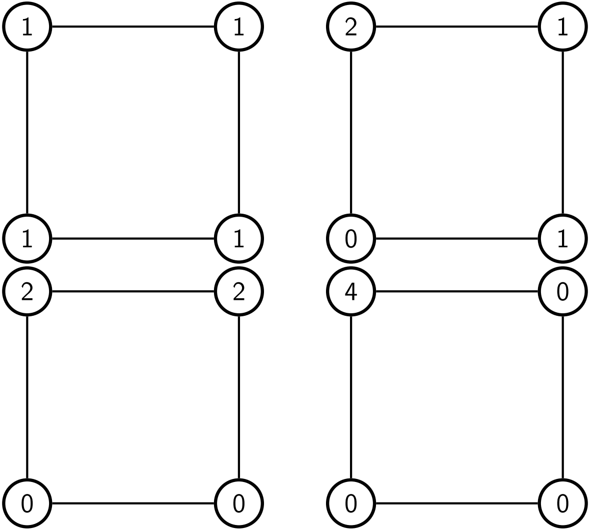




Total Acquisition Example - C_4



Total Acquisition Example - C_4



Background

- LeSaulnier, et al. proved in 2013 that $a_t(G) \leq 32 \ln(n) \ln(\ln(n))$ for any diameter two graph, and if it contains no four-cycles, $a_t(G)$ must be equal to 1 or 2.

Background

- LeSaulnier, et al. proved in 2013 that $a_t(G) \leq 32 \ln(n) \ln(\ln(n))$ for any diameter two graph, and if it contains no four-cycles, $a_t(G)$ must be equal to 1 or 2.
- They conjectured that there is some constant upper bound on $a_t(G)$ for diameter two graphs with any number of vertices

Background

- LeSaulnier, et al. proved in 2013 that $a_t(G) \leq 32 \ln(n) \ln(\ln(n))$ for any diameter two graph, and if it contains no four-cycles, $a_t(G)$ must be equal to 1 or 2.
- They conjectured that there is some constant upper bound on $a_t(G)$ for diameter two graphs with any number of vertices
- To date, no diameter 2 graph is known with $a_t(G) > 2$.

Background

- LeSaulnier, et al. proved in 2013 that $a_t(G) \leq 32\ln(n)\ln(\ln(n))$ for any diameter two graph, and if it contains no four-cycles, $a_t(G)$ must be equal to 1 or 2.
- They conjectured that there is some constant upper bound on $a_t(G)$ for diameter two graphs with any number of vertices
- To date, no diameter 2 graph is known with $a_t(G) > 2$.
- In 2017, Eastham, Kay, McCarty and Spencer proved in an unpublished work from this REU (detailing work that took place in 2015) that $a_t(G)$ for all diameter 2 graphs is at most 4 if every n-regular bipartite multigraph with n vertices in each partition contains a subgraph such that, for every integer between one and n, there exists a vertex in each partition whose degree is that integer.

The Rose Conjecture

Rose Conjecture

Each n-regular bipartite multigraph with n vertices in each partition, which we call a Rose Graph, contains a subgraph such that, for every integer between one and n, there exists a vertex in each partition whose degree is that integer.

We can also state this in terms of matrices:

The Rose Conjecture

Rose Conjecture

Each n-regular bipartite multigraph with n vertices in each partition, which we call a Rose Graph, contains a subgraph such that, for every integer between one and n, there exists a vertex in each partition whose degree is that integer.

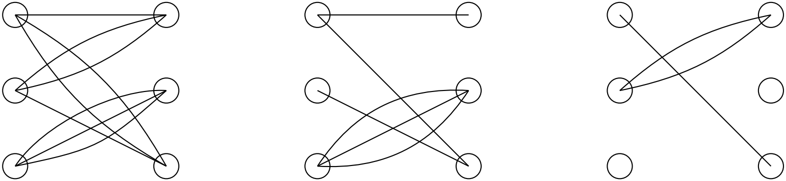
We can also state this in terms of matrices:

Rose Conjecture

Each $n \times n$ matrix A with non-negative integer entries in which each row and column has entries summing to n, can be decomposed into the sum of two matrices, $B + C$, where B and C both have non-negative integer entries, B has row and column sums equal to $\{1, 2, \ldots, n\}$ and C has row and column sums equal to $\{0, 1, \ldots, n-1\}$.

Example

$$\begin{bmatrix} 1 & 0 & 2 \\ 2 & 0 & 1 \\ 0 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 3 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



Past Attempts

Many attempts have been made from this REU to prove the Rose Conjecture.

Past Attempts

Many attempts have been made from this REU to prove the Rose Conjecture.
In the summer of 2020, Austin Allen, Sebastian Papanikolau, Heidi Perez, Doel Rivera and Thea Rugg made the following attempts:

- Linear Programming
- Combinatorial Nullstellensatz
- Edge Swap Algorithm
- Lovász Local Lemma

Past Attempts

Many attempts have been made from this REU to prove the Rose Conjecture.
In the summer of 2020, Austin Allen, Sebastian Papanikolau, Heidi Perez, Doel Rivera and Thea Rugg made the following attempts:

- Linear Programming
- Combinatorial Nullstellensatz
- Edge Swap Algorithm
- Lovász Local Lemma

In the summer of 2023, the "Roseate Spoonbill" team, consisting of Alex Morales, Grant Shirley and Steven Vasquez made the following attempts:

- Counting Argument + Compatibility
- Weak Compositions

Past Attempts

Many attempts have been made from this REU to prove the Rose Conjecture.
In the summer of 2020, Austin Allen, Sebastian Papanikolau, Heidi Perez, Doel Rivera and Thea Rugg made the following attempts:

- Linear Programming
- Combinatorial Nullstellensatz
- Edge Swap Algorithm
- Lovász Local Lemma

In the summer of 2023, the "Roseate Spoonbill" team, consisting of Alex Morales, Grant Shirley and Steven Vasquez made the following attempts:

- Counting Argument + Compatibility
- Weak Compositions

In the spring of 2024, Robert Beeler and Grant Shirley begun working on a different edge swap algorithm, which we have adopted during this attempt.

FKG Inequality

The Harris inequality, a specific adaptation of the Fortuin–Kasteleyn–Ginibre inequality, is a correlation inequality used in statistical mechanics and probabilistic combinatorics that is particularly important in the probabilistic method and random graphs, which we are now attempting to implement in the probabilistic approach to the Rose Conjecture.

FKG Inequality

The Harris inequality, a specific adaptation of the Fortuin–Kasteleyn–Ginibre inequality, is a correlation inequality used in statistical mechanics and probabilistic combinatorics that is particularly important in the probabilistic method and random graphs, which we are now attempting to implement in the probabilistic approach to the Rose Conjecture. In simplest terms, the Harris inequality states that for two events $A(x)$ and $B(x)$ whose probabilities are monotonically increasing with x, $P(A, B) \geq P(A)P(B)$.

FKG Inequality

The Harris inequality, a specific adaptation of the Fortuin–Kasteleyn–Ginibre inequality, is a correlation inequality used in statistical mechanics and probabilistic combinatorics that is particularly important in the probabilistic method and random graphs, which we are now attempting to implement in the probabilistic approach to the Rose Conjecture. In simplest terms, the Harris inequality states that for two events $A(x)$ and $B(x)$ whose probabilities are monotonically increasing with x, $P(A, B) \geq P(A)P(B)$.

Example

For example, consider a random graph G with n vertices and x edges, and let A be the event that G is Hamiltonian, and B be the event that G has diameter 2.

FKG Inequality

The Harris inequality, a specific adaptation of the Fortuin–Kasteleyn–Ginibre inequality, is a correlation inequality used in statistical mechanics and probabilistic combinatorics that is particularly important in the probabilistic method and random graphs, which we are now attempting to implement in the probabilistic approach to the Rose Conjecture. In simplest terms, the Harris inequality states that for two events $A(x)$ and $B(x)$ whose probabilities are monotonically increasing with x, $P(A, B) \geq P(A)P(B)$.

Example

For example, consider a random graph G with n vertices and x edges, and let A be the event that G is Hamiltonian, and B be the event that G has diameter 2. Since both of these increase as the number of edges increases, we can see that the Harris inequality applies here.

FKG Inequality

The Harris inequality, a specific adaptation of the Fortuin–Kasteleyn–Ginibre inequality, is a correlation inequality used in statistical mechanics and probabilistic combinatorics that is particularly important in the probabilistic method and random graphs, which we are now attempting to implement in the probabilistic approach to the Rose Conjecture. In simplest terms, the Harris inequality states that for two events $A(x)$ and $B(x)$ whose probabilities are monotonically increasing with x, $P(A, B) \geq P(A)P(B)$.

Example

For example, consider a random graph G with n vertices and x edges, and let A be the event that G is Hamiltonian, and B be the event that G has diameter 2. Since both of these increase as the number of edges increases, we can see that the Harris inequality applies here.

Breakdown

The "Gallons *n* Roses" team found a flaw in last summer's proof of the breakdown lemma, which resulted in a cascading effect on our induction approach.

Breakdown

The "Gallons *n* Roses" team found a flaw in last summer's proof of the breakdown lemma, which resulted in a cascading effect on our induction approach. We found a new way to prove a stronger statement using Hall's Marriage Theorem, however this approach is currently on the bench until we can think of some clever induction step.

Breakdown

The "Gallons *n* Roses" team found a flaw in last summer's proof of the breakdown lemma, which resulted in a cascading effect on our induction approach. We found a new way to prove a stronger statement using Hall's Marriage Theorem, however this approach is currently on the bench until we can think of some clever induction step.

Lemma

Each $n \times n$ k-regular bipartite multigraph has k distinct perfect matchings, for any non-negative integers n and k.

Edge Swap Algorithm

The "Gallons *n* Roses" team has also adopted the edge swap approach mentioned earlier.

Edge Swap Algorithm

The "Gallons n Roses" team has also adopted the edge swap approach mentioned earlier.

1. The premise of this approach is that one would label each vertex in partition X x_1 through x_n, and for each x_i in X, select i edges to be in A.

Edge Swap Algorithm

The "Gallons *n* Roses" team has also adopted the edge swap approach mentioned earlier.

1. The premise of this approach is that one would label each vertex in partition X x_1 through x_n, and for each x_i in X, select i edges to be in A.
2. Then, one would label the Y vertices by their degree.

Edge Swap Algorithm

The "Gallons *n* Roses" team has also adopted the edge swap approach mentioned earlier.

1. The premise of this approach is that one would label each vertex in partition X x_1 through x_n, and for each x_i in X, select i edges to be in A.
2. Then, one would label the Y vertices by their degree.
3. *Starting with a vertex in Y with a non-desired degree, one would create an A-alternating path to another vertex in Y with a non-desired degree.*

Edge Swap Algorithm

The "Gallons *n* Roses" team has also adopted the edge swap approach mentioned earlier.

1. The premise of this approach is that one would label each vertex in partition X x_1 through x_n, and for each x_i in X, select i edges to be in A.
2. Then, one would label the Y vertices by their degree.
3. *Starting with a vertex in Y with a non-desired degree, one would create an A-alternating path to another vertex in Y with a non-desired degree.*
4. *Then, they would swap the roles of the edges in the path. This changes the degree of the start and endpoints by one.*
5. If the resultant graph does not have the desired labels on Y, repeat the italicized steps.

Edge Swap Algorithm

The "Gallons *n* Roses" team has also adopted the edge swap approach mentioned earlier.

1. The premise of this approach is that one would label each vertex in partition X x_1 through x_n, and for each x_i in X, select i edges to be in A.
2. Then, one would label the Y vertices by their degree.
3. *Starting with a vertex in Y with a non-desired degree, one would create an A-alternating path to another vertex in Y with a non-desired degree.*
4. *Then, they would swap the roles of the edges in the path. This changes the degree of the start and endpoints by one.*
5. If the resultant graph does not have the desired labels on Y, repeat the italicized steps.

So far, we've seen that it works for each connected graph we've tried it on; the next steps would be showing that it works on disconnected graphs and proving that an alternating path always exists.

Locking Rows and Columns

- **Locking a row/column** Locking a row or column in a matrix means that cells in that row/column may not have their values changed for the remainder of the algorithmic process.
- **Free Cell** A free cell is one which has both its column and row unlocked.

Example of the Algorithm

$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

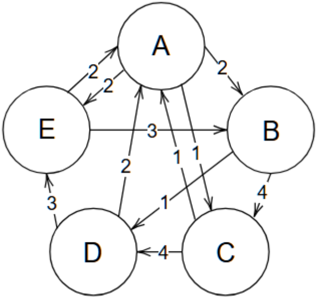
$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Example of the Algorithm

$$
\begin{array}{cc}
\begin{bmatrix}
0 & 1 & 1 & 0 & 2 \\
0 & 0 & 4 & 1 & 0 \\
0 & 0 & 0 & 3 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0
\end{bmatrix}
&
\begin{matrix} 4 \\ 5 \\ 3 \\ 2 \\ 1 \end{matrix}
\\
\begin{matrix} 1 & 2 & 5 & 4 & 3 \end{matrix} &
\end{array}
$$

Defining the Analogy

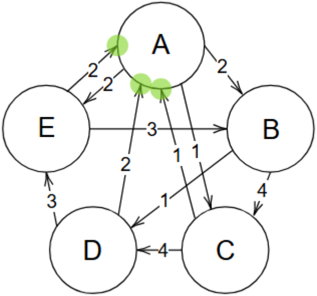
Matrix/Digraph Conversion is the process of converting an $n \times n$ matrix of nonnegative integers to a degree n digraph or vice versa.



$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Defining the Analogy

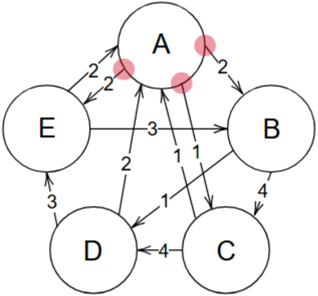
Matrix/Digraph Conversion is the process of converting an $n \times n$ matrix of nonnegative integers to a degree n digraph or vice versa.



$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

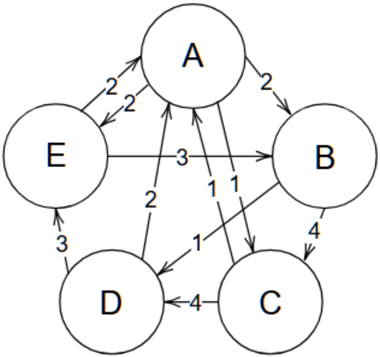
Defining the Analogy

Matrix/Digraph Conversion is the process of converting an $n \times n$ matrix of nonnegative integers to a degree n digraph or vice versa.



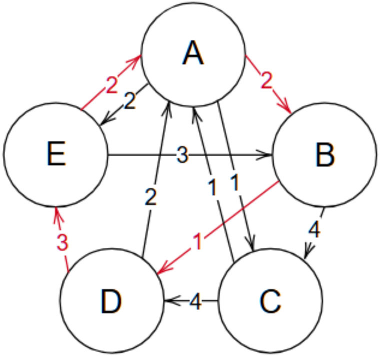
$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Previous Attempt at an Algorithm



$$\begin{bmatrix} 0 & 2 & 1 & 0 & 2 \\ 0 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 \end{bmatrix}$$

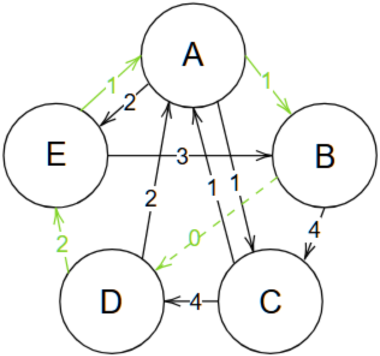
Previous Attempt at an Algorithm



$$\begin{bmatrix} 0 & \boxed{2} & 1 & 0 & 2 \\ 0 & 0 & 4 & \boxed{1} & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & \boxed{3} \\ \boxed{2} & 3 & 0 & 0 & 0 \end{bmatrix}$$

$ABDEA$

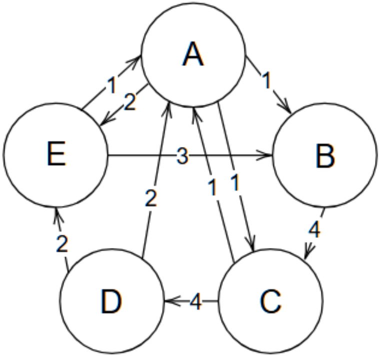
Previous Attempt at an Algorithm



$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 0 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 2 \\ 1 & 3 & 0 & 0 & 0 \end{bmatrix}$$

$ABDEA$

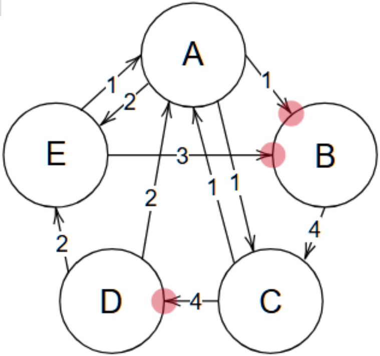
Previous Attempt at an Algorithm



$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 0 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 2 \\ 1 & 3 & 0 & 0 & 0 \end{bmatrix}$$

$ABDEA$

Previous Attempt at an Algorithm

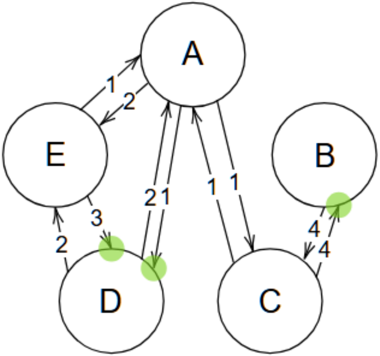


$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 4 & 0 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 2 & 0 & 0 & 0 & 2 \\ 1 & 3 & 0 & 0 & 0 \end{bmatrix}$$

$ABDEA$

$\text{SwapInputs}(B, D)$

Previous Attempt at an Algorithm

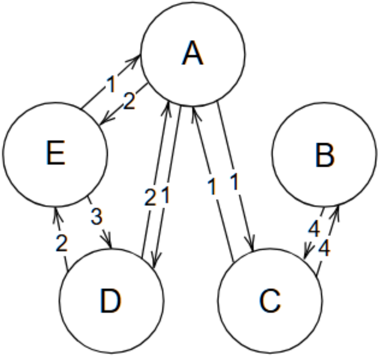


$$\begin{bmatrix} 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 4 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 3 & 0 \end{bmatrix}$$

$ABDEA$

$\text{SwapInputs}(B, D)$

Previous Attempt at an Algorithm



$$\begin{bmatrix} 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 4 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 3 & 0 \end{bmatrix}$$

$ABDEA$

$\text{SwapInputs}(B, D)$

Current Status

We conjecture that an algorithm could be produced by this method. It is a very short proof to show that this implies the Rose Conjecture.

Proof.

Assume the Weighted Digraph Conjecture. This conjecture declares that a Solution Digraph exists for any given Rose Digraph. Using Matrix/Digraph Conversion, we will revert these digraphs into a Rose Matrix and Solution Matrix. Using cell subtraction, you can find a third matrix, C such that $A = B + C$. Thus, if the Weighted Digraph Conjecture is true, so is the Rose Conjecture. $\square$

Next Steps

- Generalize the Janson/FKG approach to all n in order to prove that a solution exists
- Continue to refine algorithmic approaches in order to learn more about this class of matrices and graphs
- Explore further implications of the problem

References

A lot of the literature we've used in this project is unpublished, however we do have the following:

- Alon, Noga, and Joel H. Spencer. The Probabilistic Method. Wiley, 2016.
- C. M. Fortuin, J. Ginibre, P. W. Kasteleyn. "Correlation inequalities on some partially ordered sets." Communications in Mathematical Physics, 22(2) 89-103 1971.
- LeSaulnier, Timothy D., et al. “Total acquisition in graphs.” SIAM Journal on Discrete Mathematics, vol. 27, no. 4, Jan. 2013, pp. 1800–1819, https://doi.org/10.1137/110856186.

Questions?