Deterministic Greedy Algorithm for Maximum Independent Set Problem in Graph Theory

Joshua C. Ballard-Myer

December 18, 2019

Abstract

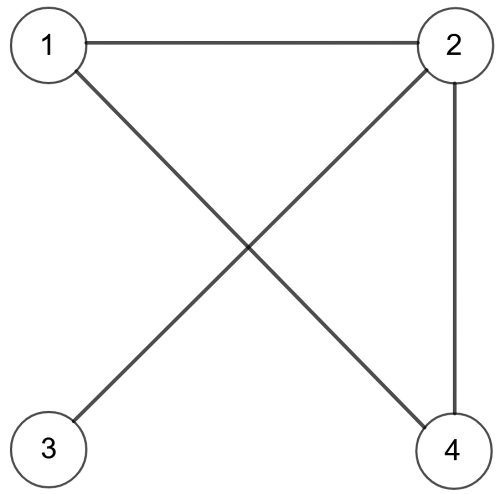
The Maximum Independent Set (MIS) problem in graph theory is the task of finding the largest independent set in a graph, where an independent set is a set of vertices such that no two vertices are adjacent. There is currently no known efficient algorithm to find maximum independent sets. We will present a deterministic greedy algorithm that is an improvement on the general greedy algorithm for MIS. This algorithm is not valid for all graphs, but conditions where the algorithm fails will be discussed. We will also briefly discuss the extension of the algorithm to coloring and hypergraphs.

1 Introduction

Definition: A **Graph** $G = (V, E)$ is a collection of vertices (or nodes) and edges (or lines), where edges are defined as a set of two vertices.
Note: In this paper we will limit our scope to **simple graphs**, which only allow one edge between vertices and vertices in an edge must be unique.

Example 1. Let $G = (V, E)$.

Let $V = \{1, 2, 3, 4\}$, $E = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{1, 4\}\}$



Definition: A vertex v and edge e are **incident** to each other if $v \in e$.

Definition: Vertices u, v are **adjacent** if they are incident to the same edge.

Definition: The **order** of a graph, denoted $o(G)$, is the number of vertices in V.

Definition: The **size** of a graph, denoted $s(G)$, is the number of edges in E.

Definition: The **degree** of a vertex v, denoted $\deg(v)$, is the number of vertices adjacent to v, or equivalently, the number of edges incident to v. The maximum degree among all $v \in V$ is denoted by $\Delta(G)$ and the minimum by $\delta(G)$.

Definition: An **induced subgraph** of a graph G is the graph obtained by removing a set of vertices S and all edges incident to those vertices from G. We will denote this $G - S$.

<u>Definition:</u> An **independent set** is a set of vertices such that no two vertices in the set are adjacent. An independent edge set, or **matching**, is a set of edges such that no two edges in the set are incident to the same vertex.

<u>Definition:</u> A **maximal independent set** is an independent set such that if any vertex not in the set is added to the set it will no longer be an independent set.

<u>Definition:</u> A **maximum independent set** (MIS) is an independent set such that no other independent sets of G are larger. The number of elements in a MIS is denoted $\alpha(G)$.

<u>Definition:</u> An **algorithm** is a set of steps used to process a computation.

<u>Definition:</u> A **deterministic** algorithm is one that always yields the same output given the same input. In this context this means that there would be no "ties" when the algorithm is choosing the next vertex to add to the MIS, unless the choice will not affect the output.

However, a **non-deterministic** algorithm may not always have a rule to decide between given vertices. For example, the algorithm may assign probability to determine how to proceed.

<u>Definition:</u> A **greedy algorithm** is an algorithm that attempts to find a global optimum by making the locally optimal choice in each step according to a given rule. The general greedy algorithm for MIS is non-deterministic. [5]

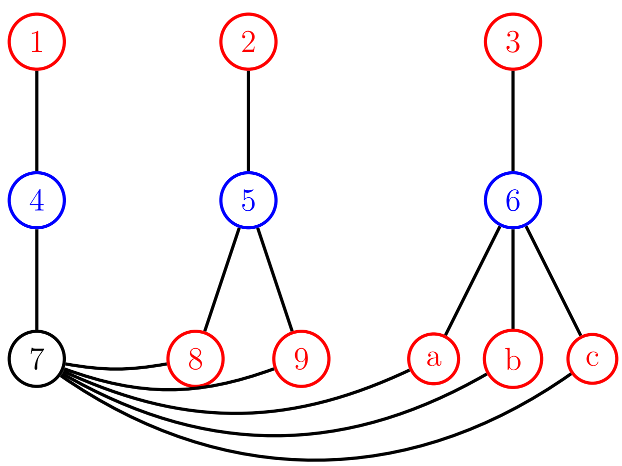


Figure 1: Here the blue vertices form a maximal independent set and the red vertices form a maximum independent set.

2 Using greedy algorithm to find the MIS

While there is no efficient algorithm to find the MIS, it can be found by brute force. The following algorithm finds all maximal independent sets by exhaustion then takes the largest.

- Select a vertex v_1 and add it to a set S_1. Take the induced subgraph $G - N_1$, where N_1 is the set of v_1 and all vertices adjacent to v_1.
- Select a vertex v_2 and add it to S_1. Take the induced subgraph $G - N_1 - N_2$, where N_2 is the set of v_2 and all vertices adjacent to v_2.
- Select a vertex v_3 and add it to S_1. Take the induced subgraph $G - N_1 - N_2 - N_3$, where N_3 is the set of v_3 and all vertices adjacent to v_3.

  $\vdots$
- Continue until the induced subgraph is the null graph. Then S_1 is a maximal independent set.
- Return to beginning and repeat for $S_2, S_3, ..., S_n$ until all permutations of vertices have been exhausted. Then $\{S_1, S_2, ...S_n\}$ contains all maximal independent sets.
- It follows that the MIS of G is $\max(S_1, S_2, ...S_n)$.

The greedy algorithm for MIS is similar to the one to construct a single maximal independent set, but we want to choose vertices in each step such that the resulting set is not just maximal, but maximum.

- Select a vertex of least degree v_1 and add it to a set S. Take the induced subgraph $G - N_1$, where N_1 is the set of v_1 and all vertices adjacent to v_1.
- Select a vertex of least degree v_2 and add it to S. Take the induced subgraph $G - N_1 - N_2$, where N_2 is the set of v_2 and all vertices adjacent to v_3.
- Select a vertex of least degree v_3 and add it to S. Take the induced subgraph $G - N_1 - N_2 - N_3$, where N_3 is the set of v_3 and all vertices adjacent to v_3.

  $\vdots$

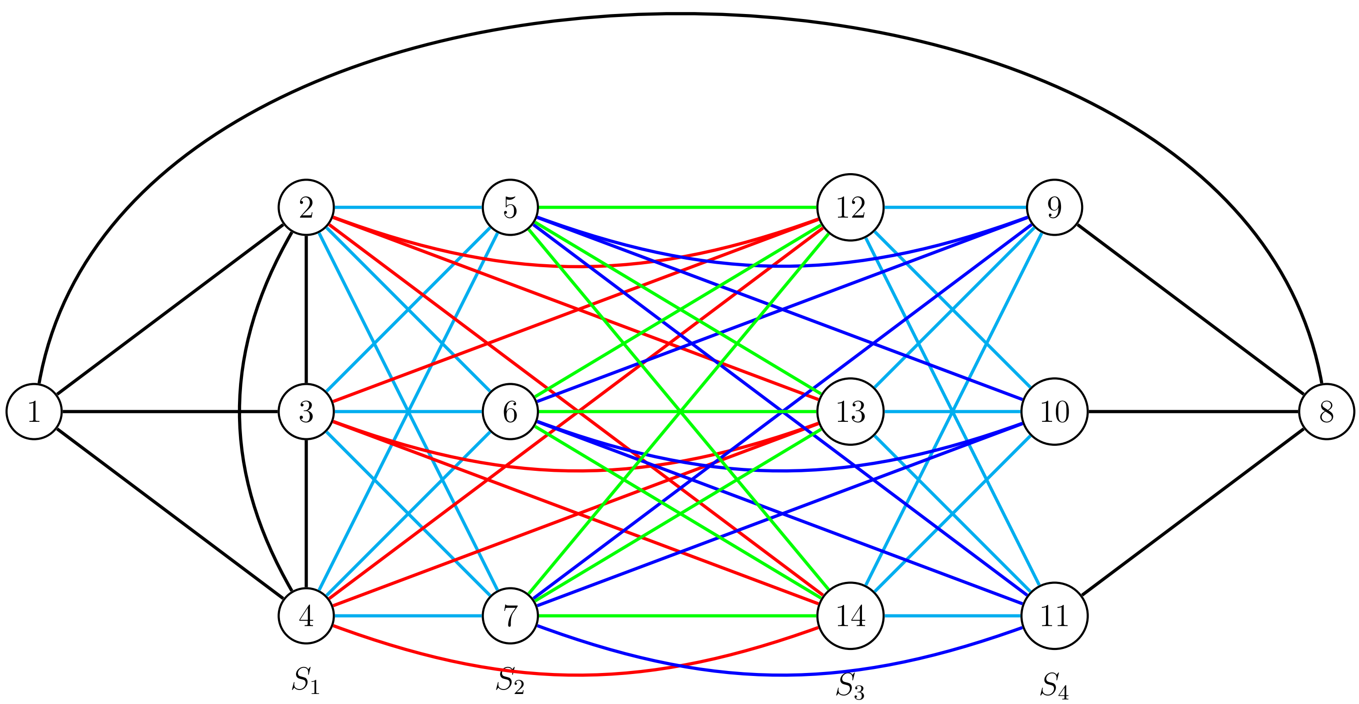
- Continue until no viable vertices remain. Then S is a maximum independent set (hopefully).

A vertex of least degree (VLD) will have the fewest adjacent vertices, so choosing it in each step gives us the most possible paths for a single iteration of the algorithm. In other words, by choosing the VLD we will eliminate the least possible vertices in each step, meaning that we have more vertices in the induced subgraph and thus more vertices able to be in the MIS. Least degree is a very strong property, since it directly affects the size of our induced subgraphs in each step, and eventually our MIS itself. This leads us to our first conjecture.

Conjecture 1. *The MIS must contain a vertex of least degree.*

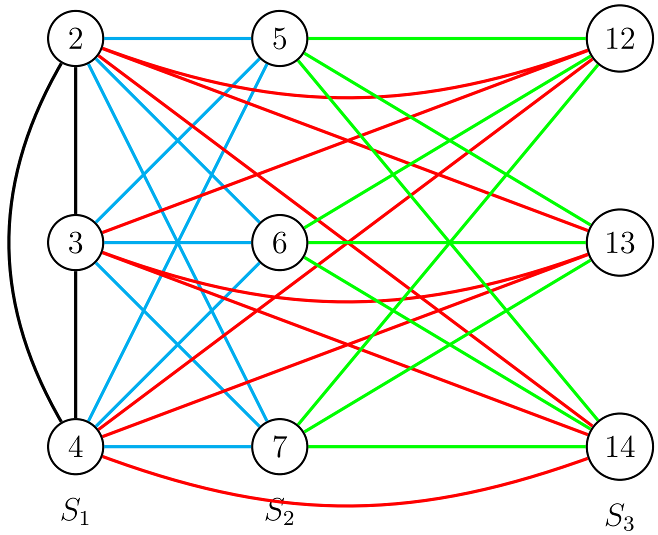
However, now we are left with the natural question: What if two (or more) vertices in the same step have the same degree? Does it matter which one we choose

In the following graph we will see that it does matter which one we choose.



We can see that the VLD's are 1 and 8, so greedy algorithm dictates that we choose one of them to start.

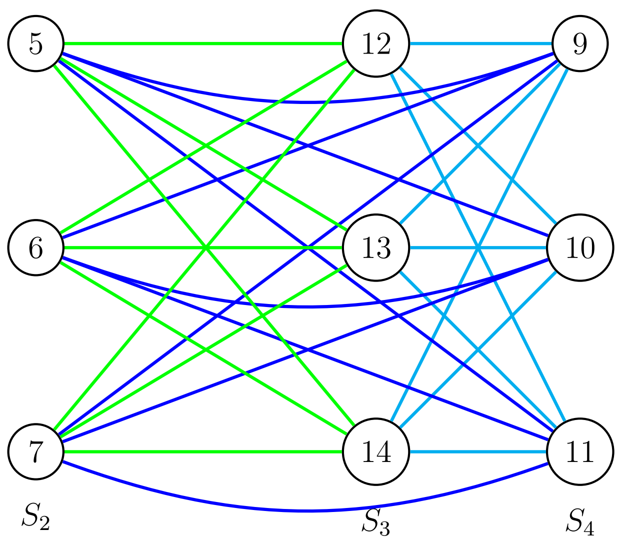
If we choose 8, we have the following induced subgraph.



The MIS for this subgraph is $\{12, 13, 14, (2 \text{ xor } 3 \text{ xor } 4)\}$.

So we have an MIS of total size 5.

If we choose 1, we have the following induced subgraph.



The MIS for this subgraph is $\{5, 6, 7, 9, 10, 11\}$.

So we have an MIS of total size 7.

Now we have a new question: How do we decide which vertex to pick?

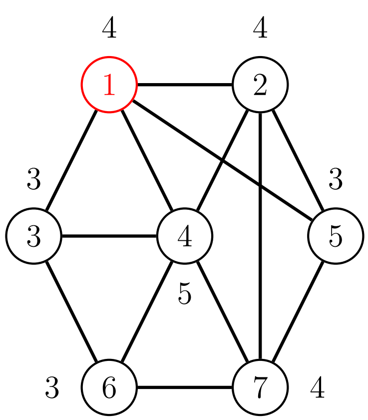
<u>Define</u>: the **secondary degree** of a vertex v, denoted $\deg^2(v)$, as the sum of the degrees of each vertex adjacent to v.

<u>Define</u>: the **tertiary degree** of a vertex v, denoted $\deg^3(v)$, as the sum of the degrees of each vertex adjacent to a vertex adjacent to v, not including vertices already considered in a lesser degree.

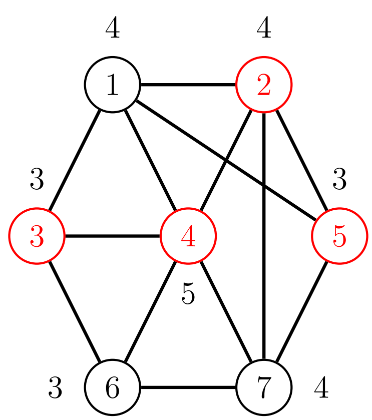
<u>Define</u>: in general, the $n^{\mathbf{th}}$ **degree** of a vertex v, denoted $\deg^n(v)$, as the sum of the degrees of vertices adjacent to the vertices considered for the $(n-1)^{\text{th}}$ degree, not including vertices already considered in a lesser degree.

Note: We will retroactively rename the degree of a vertex to be the **primary degree**.
Note: If there are no viable vertices adjacent to the ones considered in the $(n-1)^{\text{th}}$ degree, then $\deg^k(v) = 0$ for all $k \geq n$.

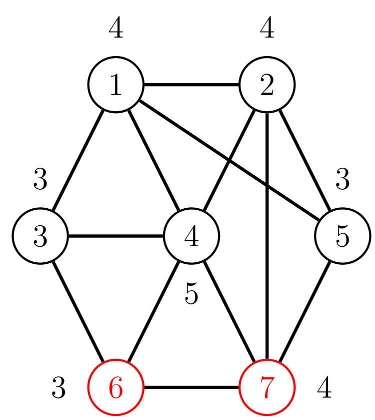
The following is a demonstration of these definitions. The red vertices are the vertices whose degrees are being considered in each step and the labels next to the vertices are their primary degrees.



$\deg(1) = 4$



$\deg^2(1) = 4+3+5+3 = 15$

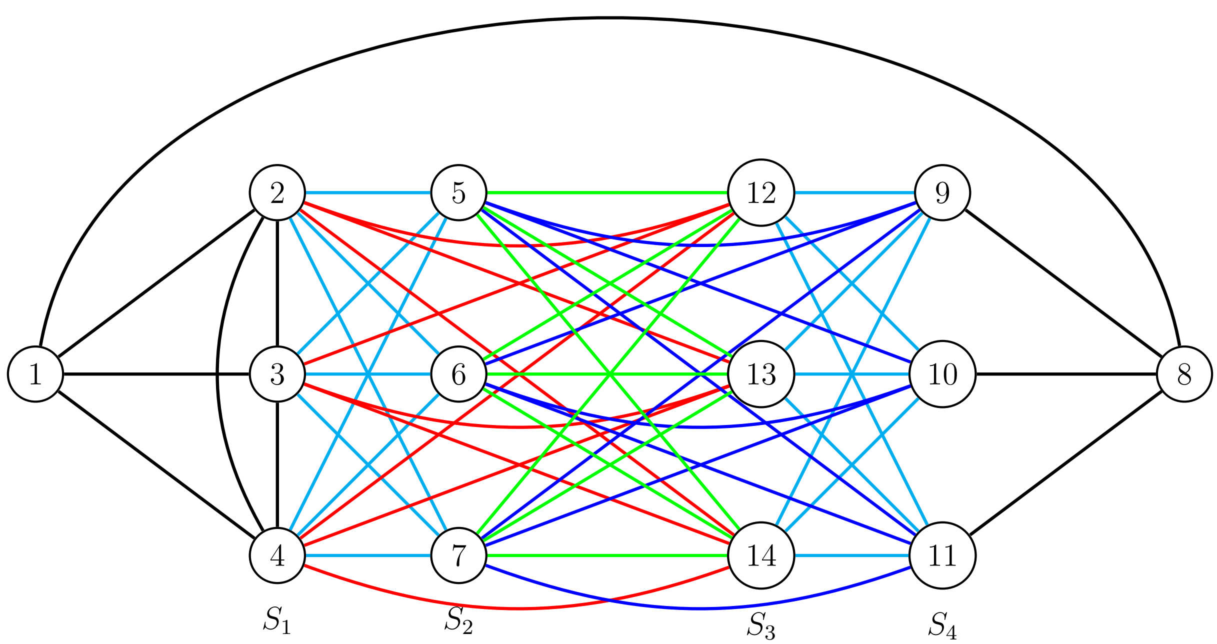


$\deg^3(1) = 3+4 = 7$

3 Improving the greedy algorithm

Now that we have defined these terms, we are ready to improve the rule of our greedy algorithm. We will modify it so that the algorithm always knows which vertex is best to choose, or if they are equivalent choices. Now we will return to our previous question: How do we decide which vertex to pick?

If two vertices, u and v have the same degree, but $\deg^2(u) > \deg^2(v)$, then choosing u will eliminate the same number of vertices as choosing v. However, by eliminating the vertices with greater total degree, the induced subgraph will be less connected, that is, this subgraph will have less edges. This implies that the the sum of the degrees of the remaining vertices will be lower, so in general they will be better candidates to be chosen in the next step than the vertices that would remain if we had chosen v. These vertices are "better" candidates because of the strength of the property of lower degree sum.
Now we can return to the previous graph and we see that 1 was the better candidate because it had a greater secondary degree, meaning S_1 is more connected than S_4, so by eliminating S_1 rather than S_4 the induced subgraph is less connected in a critical area.



Note: It is inconsequential whether or not we include the degree of v itself in the definition of secondary degree; it will only be applied in the case that two vertices have equal degree and we are only comparing the sizes of each, so adding or subtracting the same to each will not affect the outcome.

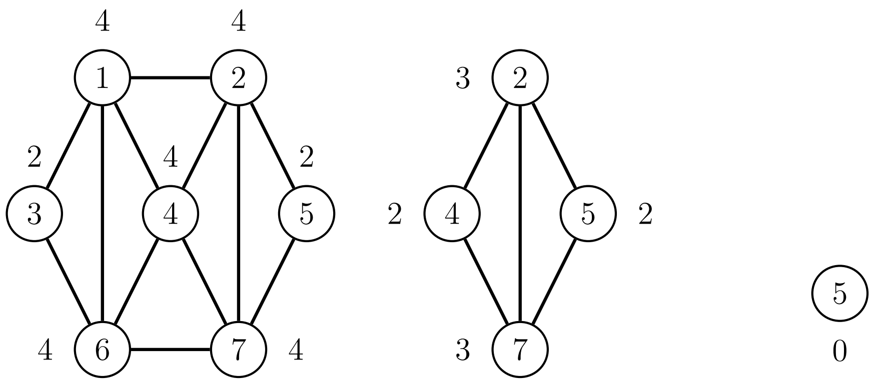
Now we can create our improved, deterministic, greedy algorithm. It will be the same as the non-deterministic greedy algorithm except for the improved rule.

- Eliminate vertex of least degree from graph and add to MIS. Remove all vertices adjacent to the VLD and all edges incident to removed vertices.
  - If there is more than one vertex of least degree, choose the vertex with the greatest secondary degree.
    - If multiple of those vertices have the same secondary degree, choose the vertex with the least tertiary degree.
    - In general, if $d^n(v_1) = d^n(v_2)$, choose the vertex of greatest $(n+1)^{\text{th}}$ degree if $n+1$ is even and the least if $n+1$ is odd.
    - If two vertices have the same n^{th} degree for all n then either may be chosen.
- Return to step 1 and repeat algorithm on the induced subgraph.
- Once the induced subgraph is the null graph, the algorithm ends and we have our MIS.

Conjecture 2. *Suppose two vertices have the same n^{th} degree for all n. Then choosing either vertex will yield an MIS of the same size.*

Note: The MIS's may contain different vertices.

In the next example we will see the algorithm demonstrated. Note that the above conjecture holds for this graph. The labels next to each vertex indicate the degree of the vertex.



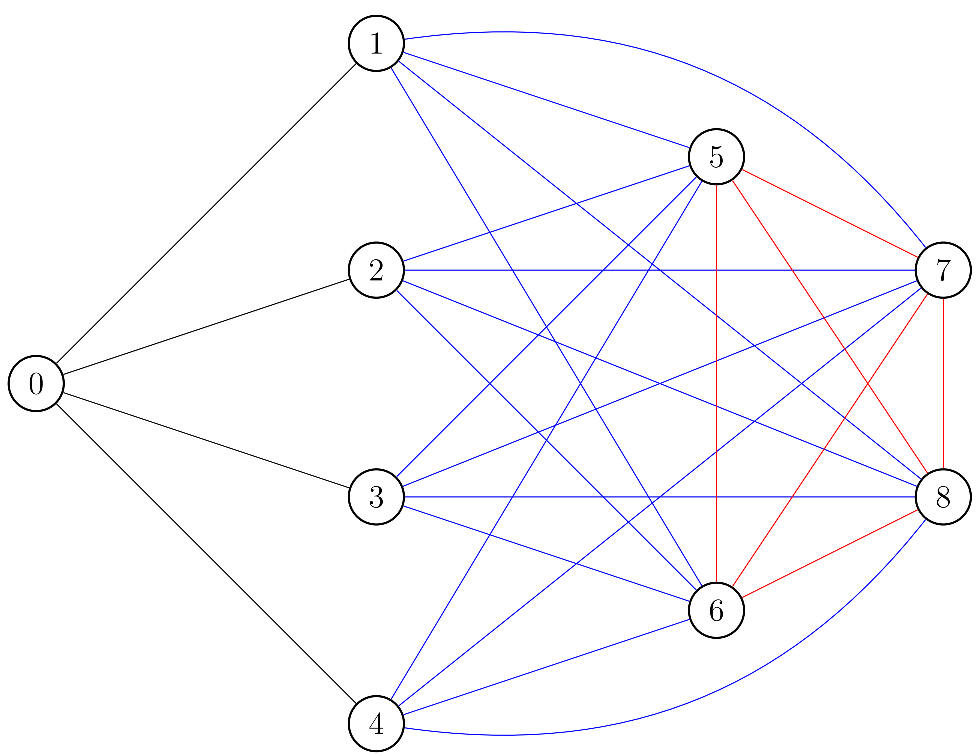
Add {3} to MIS.
Then MIS={3}

Add {4} to MIS.
Then MIS={3, 4}

Add {5} to MIS.
Then MIS={3, 4, 5}

We can see from this example that it was irrelevant which vertex we started with because both of them yielded the same MIS.

Unfortunately, this algorithm does not always work [3] [5]. Observe the graph below:



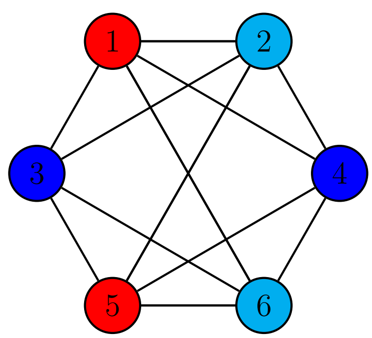
Theorem 1. *The MIS does not necessarily contain a vertex of least degree.*

We can see that the algorithm fails on the very first step, because the vertex of least degree is not in the maximum independent set. In fact, selecting the VLD eliminates the MIS from the graph. So now we know that our earlier conjecture is false, and we have the above theorem. Since the greedy algorithm must necessarily fail for this type of counterexample, we suspect that:

Conjecture 3. *This algorithm fails only if greedy algorithm must fail.*

4 Extensions

Definition: A graph **coloring** is a partition of the vertices of a graph into subsets such that no vertices in the same subset are adjacent. The number of colors needed to color a graph G is called the **chromatic number**, denoted $\chi(G)$. The class of vertices of a certain color is called a **color class**.

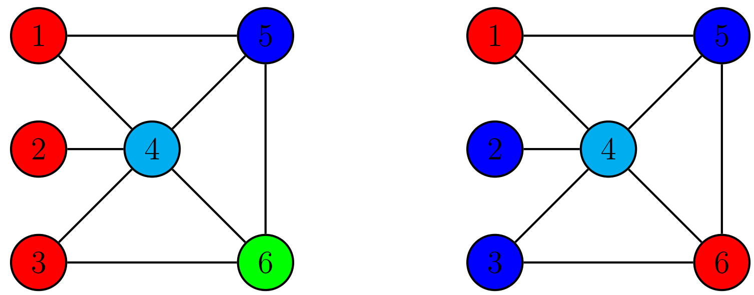


Note that color classes are a similar concept to independent sets, since a color class is necessarily independent. This idea leads us to our next conjecture:

Conjecture 4.
Let G be a graph.

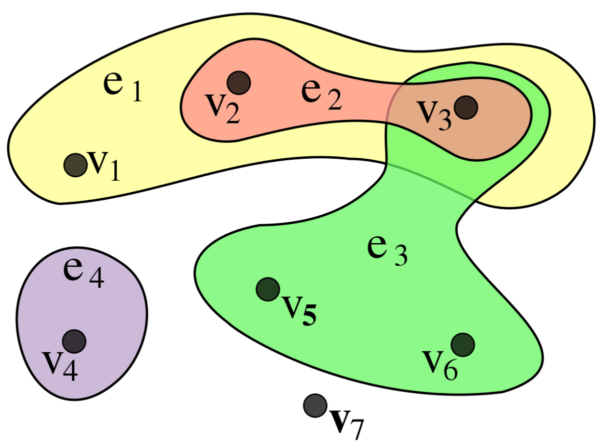
- *Take the MIS of G, denote it M_1 and add it to a collection C.*
- *Take the MIS of the induced subgraph $G - M_1$, denote it M_2 and add it to C. Continue until the induced subgraph is the null graph.*
- *Then C is a partition of G and each $M_n \in C$ is a color class of G such that we have an optimal coloring of G.*

Counterexample [2] [4]:



Thus we know that our above conjecture is false. Applying the algorithm to this graph we obtain the coloring on the left, but the coloring on the right is more optimal. Note that the right graph can also be archived by a different method of MIS extraction, but an example of a graph where MIS extraction must necessarily fail is [2].

__Definition:__ A **hypergraph** is a generalization of a graph where an edge need not contain exactly two vertices.



$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

$E = \{e_1, e_2, e_3, e_4\}$

$e_1 = \{v_1, v_2, v_3\},$

$e_2 = \{v_2, v_3\},$

$e_3 = \{v_3, v_5, v_6\},$

$e_4 = \{v_4\}$

Since edges in a hypergraph are represented by shapes, if we want to find an independent set we need to find vertices such that none of them are in the same shape as any others.

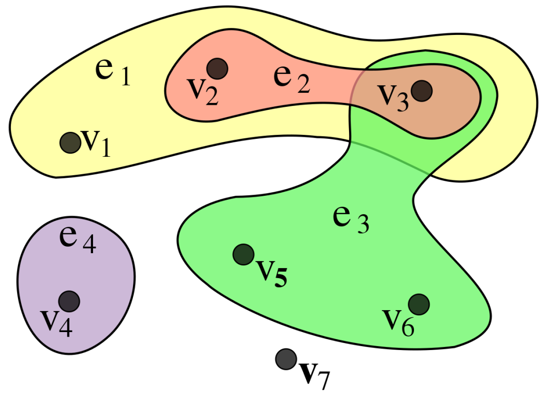


Figure 2: MIS's: $\{v_7, v_4, v_5, v_1\}, \{v_7, v_4, v_5, v_2\}, \{v_7, v_4, v_6, v_1\}, \{v_7, v_4, v_6, v_2\}$

Note that for hypergraphs we must define the secondary degree more precisely as only the number of vertices adjacent to a vertex and not the number of edges incident to it, since in hypergraphs they will not necessarily be the same. Since the properties of edges are so different in hypergraphs it is likely that to apply the algorithm to hypergraphs we will need to alter it further.

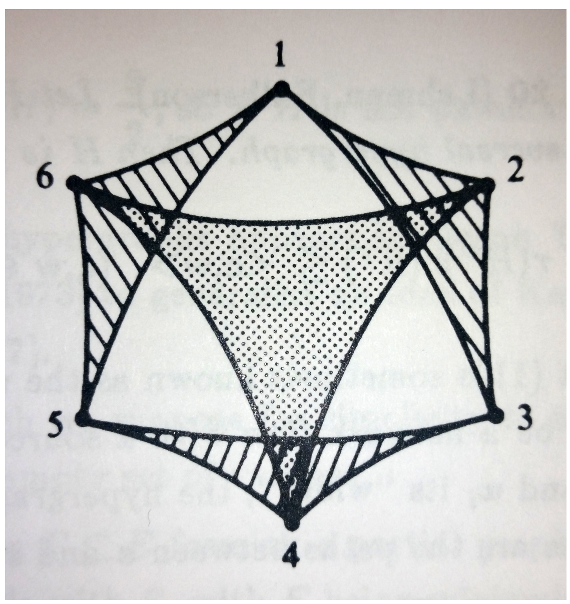


Figure 3: $V = \{1, 2, 3, 4, 5, 6\}$, $E = \{\{1, 2, 3\}, \{3, 4, 5\}, \{5, 6, 1\}, \{2, 4, 6\}\}$
MIS's: $\{1, 4\}, \{2, 5\}, \{3, 6\}$ [1]

References

[1] Claude Berge. *Hypergraphs : combinatorics of finite sets / Claude Berge.* North Holland : Distributors for the U.S.A. and Canada, Elsevier Science Pub. Co., 1989.

[2] Alberto Rivelli (https://mathoverflow.net/users/26449/alberto rivelli). Coloring a graph by maximum independent set extraction. MathOverflow. URL:https://mathoverflow.net/q/118143 (version: 2013-01-05).

[3] Mathieu Mari. Study of greedy algorithm for solving maximum independent set problem. Technical report, University of Liverpool, 2017.

[4] P.M. Pardalos S.I. Butenko, C.W. Commander. On the complexity of the broadcast scheduling problem. Technical report, University of Florida, 2004.

[5] Douglas West. *Introduction to Graph Theory.* Prentice Hall, 2nd edition, 1996.