Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Maestría en Ciencias

Algoritmo determinista para encontrar el Conjunto Máximo Independiente (MIS) para un grafo cáctus y extendido para grafos de mayor grado

Análisis y Diseño de Algoritmos: Otoño 2024

Alhely González Luna

Septiembre 30, 2024

1 Introducción

En teoría de grafos el problema del conjunto máximo independiente (MIS) consiste en encontrar el conjunto independiente más grande en un grafo, donde un conjunto independiente es un conjunto de vértices en el cual no existen dos nodos que sean adyacentes[1]. Actualmente no existe un algoritmo eficiente para encontrar el MIS [3] y además posee una historia reelevante en el desarrollo para problemas NP completos[2].

En este documento se presenta un algoritmo determinista para encontrar el MIS de un grafo tipo cáctus y se muestra también un ejemplo para un grafo de grado mayor.

El documento se organiza como sigue; en el capitulo 2 se presenta una breve introducción sobre grafos, conjuntos independientes y máximo conjunto independiente así como una descripción general de un grafo tipo cactus, en el capítulo 3 se presenta el código implementado en python y una descripción de su funcionamiento, finalmente en el capitulo 4 se presentan varios ejemplos del funcionamiento del algoritmo y en el capitulo 5 mostramos conclusiones.

2 Grafos, IS y MIS

2.1 Grafos

Un grafo consiste en dos conjuntos V y A donde: G=(V, A) [4]

- V es un conjunto finito no vacío de vértices o nodos.
- A es un conjunto de aristas o arcos, tales que cada arista a_i de A está identificada por un único par (v_j, v_k) de nodos de V, denotada por $a_i = (v_j, v_k)$

Algunas definiciones importantes sobre grafos:

- **Un grafo dirigido** es aquel en el que los arcos tienen un único sentido. En este caso, un arco se dirige desde el nodo origen hasta el nodo destino. Se dice que el nodo origen precede al nodo destino, y que éste sucede al origen. Los arcos de un grafo dirigido se representan gráficamente con flechas.[5]

- **Un grafo no dirigido** es un grafo donde los arcos conectan a los nodos en ambos sentidos.[5]

- Los vértices o nodos v1 y v2 son **adyacentes** o **vecinos** si (v1,v2) es una arista en A(G), la cual diremos que es **incidente** sobre ambos vértices [4].

- La **vecindad** de un vértice v en un grafo G es el subgrafo de G inducido por todos los vértices adyacentes av, es decir, el grafo compuesto por los vértices adyacentes avy todas las aristas que conectan los vértices adyacentes av.[6]

- Un **camino** desde v_p hasta e v_q en el grafo G es una secuencia de vértices $v_p, v_1, ..., v_n v_q$ tal que $(v_p, v_1)(v_1, v_2), ..., (v_n, v_q)$ son aristas de A(G)[4].

- Un **camino** desde v_p hasta e v_q en el grafo G es una secuencia de vértices $v_p, v_1, ..., v_n v_q$ tal que $(v_p, v_1)(v_1, v_2), ..., (v_n, v_q)$ son aristas de A(G) [4].

- Se llama **camino simple** a aquél en el que todos los vértices excepto el primero y el último, son distintos

- Un **ciclo** es un camino simple en el que el vértice primero y último coinciden [4].

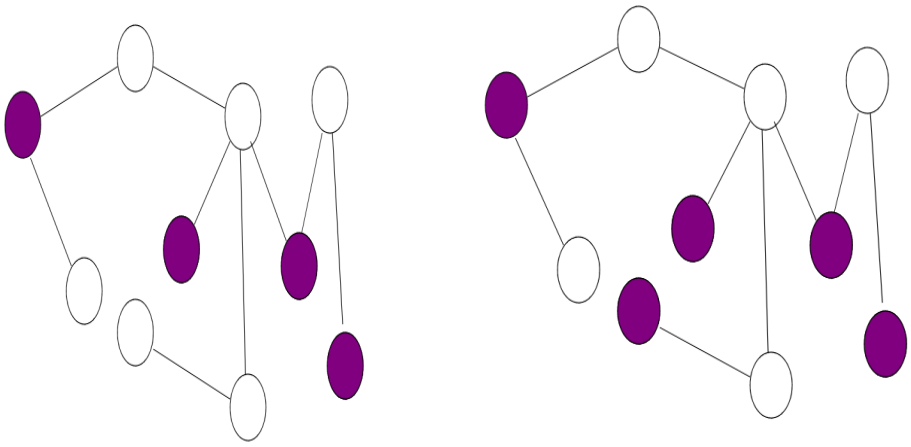
2.2 Conjunto Independiente (IS)

Es un conjunto de nodos o vértices en un grafo tal que ninguno de sus vértices es adyacente a otro. Cara arista contiene a lo más un vértice [2]

2.3 Máximo Conjunto Independiente (MIS)

Un conjunto independiente de *máxima* cardinalidad es llamado Máximo Conjunto Independiente [1]. Es decir, es un conjunto independiente tal que, al agregar un vértice más deja de ser independiente.

A continuacion se muestran ejemplos de un MIS vs un IS



(a) Conjunto Independiente (b) Máximo Conjunto Independiente [8]

Figure 1: IS vs MIS

2.4 Grafo Cáctus

Un grafo cáctus es un grafo $G = (V, A)$ en el que cada arista pertenece a lo sumo a un ciclo. Cualquier par de ciclos comparten, como mucho, un vértice.

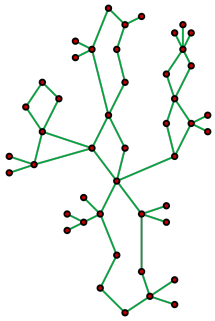


Figure 2: Ejemplo de un grafo cáctus

3 Algoritmo para encontrar el MIS de un grafo cáctus

3.1 Descripción

Se diseñó el siguiente algoritmo simple determinista

Process: Encontrar MIS

INPUT conjunto de aristas de un grafo G
SET MIS = conjunto
SET queue = nodos(G)
while $len(queue) > 0$:
for nodo in queue:
MSI.add(nodo)
for vecino in vecindad(nodo):
if vecino in queue:
queue.remove(vecino)

Se inicia con todos los nodos del grafo como *no visitados*, después asumimos que el primer nodo pertenece al MIS, así que lo agregamos al conjunto, como este nodo pertenece al MIS, ninguno de sus vecinos puede pertenecer a él, por lo tanto se remueven de la lista de nodos no visitados y se continua al siguiente nodo, se itera sobre todos los nodos no visitados actualizados y se finaliza con la obtención del MIS. Este algoritmo es determinista ya que siempre regresa el mismo MIS dado un grafo. Puede usarse para cualquier grafo, no solo uno de cáctus. La implementación en python se muestra en la Figura 3.

```python
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt


relacion_str = input("Please enter something: ")
relacion_set = eval(relacion_str)

G = nx.DiGraph()
G.add_edges_from(list(relacion_set))



nodos = list(G.nodes)

queue = nodos


print('-------------´Calculando MSI-----------------')


maximal_independent_set = set()
while len(queue)>0:
    for node in queue:
        maximal_independent_set.add(node) ## add to msi
        queue.remove(node) ## remove from nodes to visit
        for neighbor in nx.neighbors(G, node):
            if neighbor in queue:
                queue.remove(neighbor)

print('-------------MSI = '+str(maximal_independent_set)+'-----------------')
```

Figure 3: Código de python que implementa el algoritmo descrito arriba

4 Ejemplos

Se muestran tres grafos de diferente orden y el MIS encontrando por el algoritmo descrito en la sección anterior.

4.1 Ejemplo 1

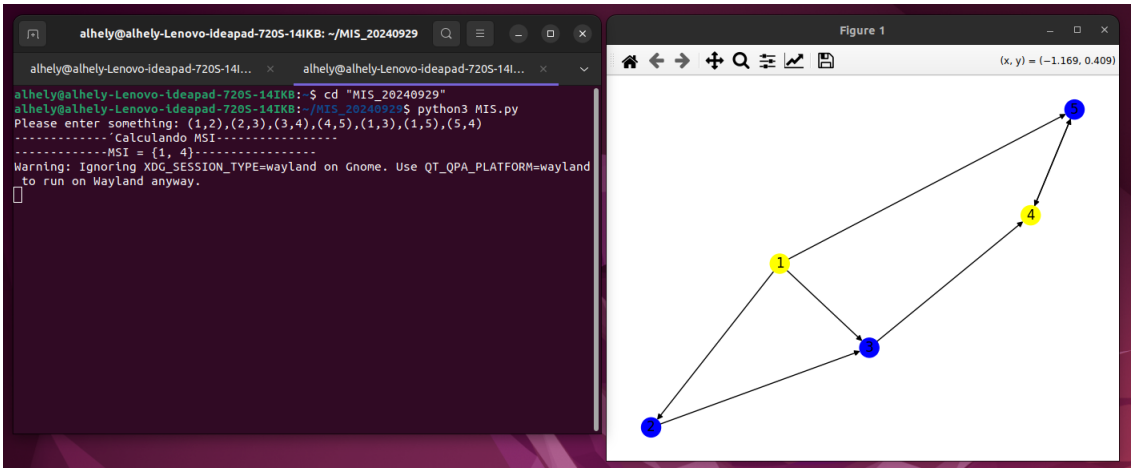


Figure 4: MIS encontrado por el algoritmo, los nodos en amarillo pertenecen al MIS

4.2 Ejemplo 2

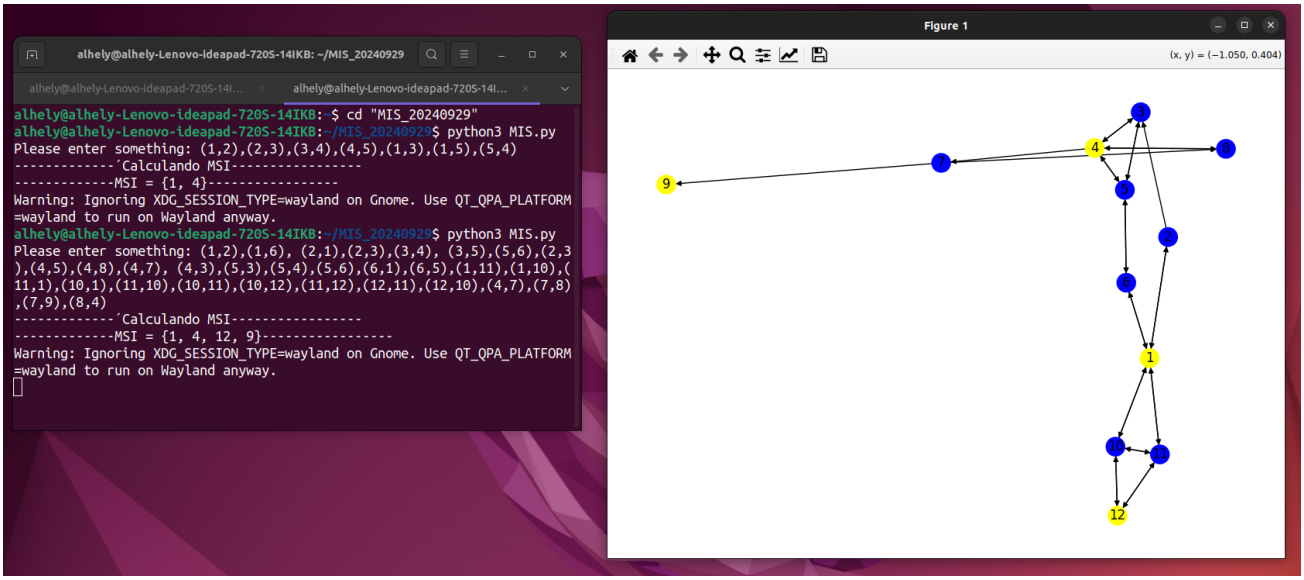


Figure 5: MIS encontrado por el algoritmo, los nodos en amarillo pertenecen al MIS

4.3 Ejemplo 3

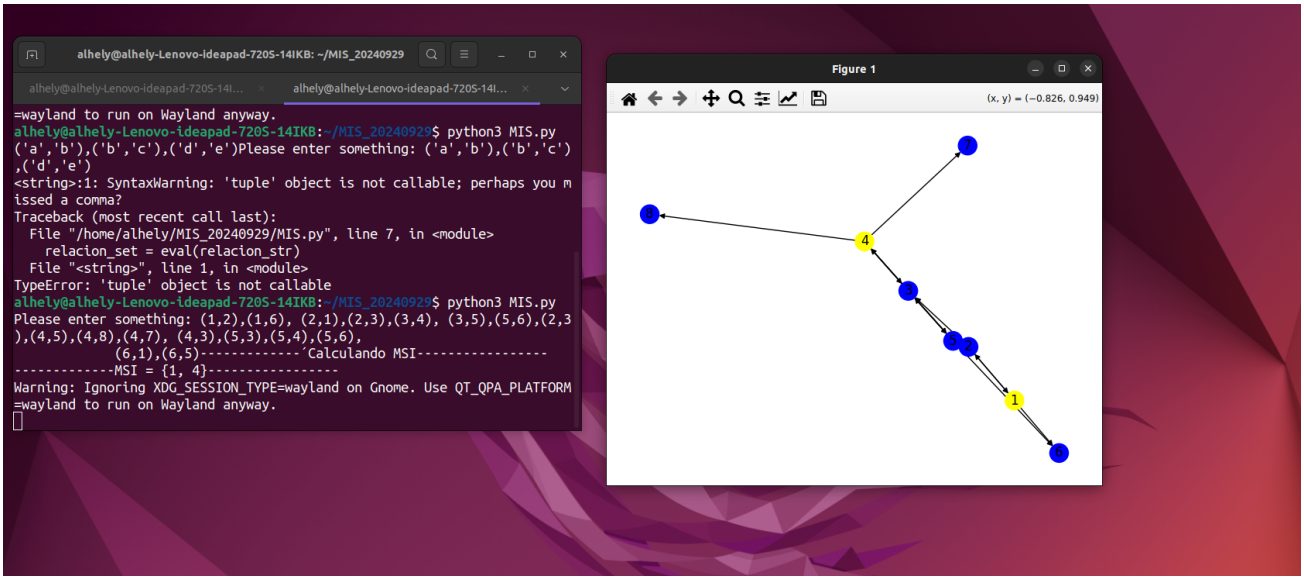


Figure 6: MIS encontrado por el algoritmo, los nodos en amarillo pertenecen al MIS

5 Conclusiones

Se mostró un algoritmo determinista capaz de identificar el MIS de un grafo, es un algoritmo determinista porque siempre regresa el mismo conjunto. Sin embargo, puede haber distintos MIS de misma cardinalidad pero distintos elementos, este algoritmo no es capaz de encontrarlos todos, ya que depende del orden en que se le proveen los nodos.

References

[1] Chapter 6: Maximal Independent Set https://ac.informatik.uni-freiburg.de/teaching/ss_12/netalg/lectures/chapter6.pdf

[2] Moctezuma Pascal, Luis Alfredo, Un algoritmo exacto para el máximo conjunto independiente https://es.slideshare.net/slideshow/conjunto-independiente-mximo/60995568

[3] Ballard-Myer, Joshua. Deterministic Greedy Algorithm for Maximum Independent Set Problem in Graph Theory. 2019. https://www.gcsu.edu/sites/files/page-assets/node808/attachments/ballardmyer.pdf

[4] TEMA 5 El Tipo Grafo PROGRAMACIÓN Y ESTRUCTURAS de DATOS. https://rua.ua.es/dspace/bitstream/10045/16037/12/ped-09_10-tema5.pdf

[5] 5.2 Grafos Dirigidos Y No Dirigidos. https://www6.uniovi.es/usr/cesar/Uned/EDA/Apuntes/TAD_apUM_07.pdf

[6] Vecindad (Teoría de Grafos) https://academia-lab.com/enciclopedia/vecindad-teoria-de-grafos/

[7] Martin Peñalver, Anabek. Propagación En Grafos. June 201 https://oa.upm.es/55597/1/TFG_ANABEL_MARTIN_PE%C3%91ALVER.pdf

[8] Sarathi, Partha, et al. Maximal Independent Set. https://www.iitg.ac.in/gkd/aie/slide/MIS-PSM.pdf