exposicion

June 30, 2024

0.1 Cargamos los imports.

```
[1]: import init

[2]: import grafo as grafo
  import experimento as exp
  import modelo as modelo
  import pandas as pd
  import numpy as np
  import matplotlib.pyplot as plt
```

0.2 Estudio de la propagación y dinámica de precios a partir de la Matriz de Insumo-Producto (MIP)

Grupo: 6

Integrantes: Miguel De Lillo, Manuel Fernandez, Augusto Kielbowicz, Mariano Oca

Repositorio

1 Motivación

La idea de este modelo es estudiar, ante un shock de precios en un sector dado, cómo afectan las dinámicas propuestas entre los sectores productivos de la Argentina en la inflación global calculada a partir de las variaciones en los precios de los mismos. (Utilizando las relaciones dadas por la matriz de Insumo-Producto).

TODO: Agregar imagenes acá

- Introducción
 - Preguntas a responder
 - Suposiciones (y limitaciones) del modelo
- A
- B
- C
- Conclusiones
- Apéndices
 - Código

2 Introducción

2.1 Preguntas a responder

- ¿Cómo se propaga el aumento de precios a través de la red definida por la MIP?
- ¿Cómo impacta el aumento de precio en un producto/sector sobre otros productos/sectores? ¿Se mantiene en la misma cadena productiva?
- ¿Qué métricas se pueden computar sobre todo el sistema para describir el estado global del sistema?
- ¿Cómo influyen las dinámicas de comportamiento de los agentes en la variación de la inflación global?
- ¿Cuál es la sensibilidad del sistema respecto a variaciones de precio en nodos específicos? ¿Cuáles son los nodos que propagan de mayor forma la variación de precios?

2.2 Suposiciones (y limitaciones) del modelo

- Cada agente es un sector productivo de la Argentina. Éste representaría a todos los productos (y productores) del sector.
- Un cambio de precios >0 en los insumos provoca un cambio de precios saliente del agente (producción). Sólo vamos a estudiar variaciones positivas en los precios para atenernos a las preguntas a investigar con el modelo.
- La economía es cerrada. Esto es, no se traen productos de otros países (importaciones) ni se vende nada a ellos (exportaciones) durante la evolución del sistema.
- Rige la Lev de Say: la oferta es igual a la demanda.

3 Matriz Insumo Producto y Análisis

Acá mostramos un poco el análisis del grafo y lo visualizamos

```
[3]:
    mip = pd.read_csv('../resources/MIP_normalizada.csv', index_col=0)
[4]: mip
[4]:
                                                           Cultivo de cereales,
     oleaginosas y forrajeras
     Cultivo de cereales, oleaginosas y forrajeras
     0.010319
     Cultivo de hortalizas, legumbres, flores y plan...
     0.000000
     Cultivo de frutas y nueces
     0.000000
     Cultivos industriales
     0.000000
     Producción de semillas
     0.783146
```

```
Servicios sociales
0.000000
Servicios de saneamiento
0.000870
Actividad de asociaciones
0.003092
Servicios de cine, radio y televisión
0.000536
Servicios personales, de reparación, actividade...
0.006876
                                                     Cultivo de hortalizas,
legumbres, flores y plantas ornamentales \
Cultivo de cereales, oleaginosas y forrajeras
0.000000
Cultivo de hortalizas, legumbres, flores y plan...
Cultivo de frutas y nueces
0.000000
Cultivos industriales
0.000000
Producción de semillas
0.016374
Servicios sociales
0.000000
Servicios de saneamiento
0.000193
Actividad de asociaciones
0.000686
Servicios de cine, radio y televisión
Servicios personales, de reparación, actividade...
0.000019
                                                     Cultivo de frutas y nueces
Cultivo de cereales, oleaginosas y forrajeras
                                                                        0.001542
Cultivo de hortalizas, legumbres, flores y plan...
                                                                      0.009565
Cultivo de frutas y nueces
                                                                        0.000000
Cultivos industriales
                                                                        0.000000
Producción de semillas
                                                                        0.029820
```

Servicios sociales

0.000000

Servicios de saneamiento Actividad de asociaciones Servicios de cine, radio y televisión Servicios personales, de reparación, actividade…	0.000127 0.000453 0.000069 0.000158
Cultivo de cereales, oleaginosas y forrajeras Cultivo de hortalizas, legumbres, flores y plan Cultivo de frutas y nueces Cultivos industriales Producción de semillas	Cultivos industriales \
Servicios sociales Servicios de saneamiento Actividad de asociaciones Servicios de cine, radio y televisión Servicios personales, de reparación, actividade	0.000000 0.000219 0.000777 0.000121 0.001907
Cultivo de cereales, oleaginosas y forrajeras Cultivo de hortalizas, legumbres, flores y plan Cultivo de frutas y nueces Cultivos industriales Producción de semillas	Producción de semillas \
Servicios sociales Servicios de saneamiento Actividad de asociaciones Servicios de cine, radio y televisión Servicios personales, de reparación, actividade	0.000000 0.000107 0.000379 0.000055 0.000225
de leche, lana y pelos \ Cultivo de cereales, oleaginosas y forrajeras 0.181660 Cultivo de hortalizas, legumbres, flores y plan 0.000000 Cultivo de frutas y nueces 0.000000 Cultivos industriales 0.000000 Producción de semillas 0.127875	Cría de ganado y producción
Servicios sociales 0.000000	

Servicios 0.000952	de	saneam	nient	0		
Actividad 0.003384	de	asocia	cion	es.	;	
Servicios 0.000569	de	cine,	radi	0	y televisió	n
Servicios 0.006695	рез	rsonale	es, d	.e	reparación,	actividade

	Producción de granja \
Cultivo de cereales, oleaginosas y forrajeras	0.012263
Cultivo de hortalizas, legumbres, flores y plan	0.00000
Cultivo de frutas y nueces	0.000000
Cultivos industriales	0.00000
Producción de semillas	0.002291
***	***
Servicios sociales	0.00000
Servicios de saneamiento	0.000202
Actividad de asociaciones	0.000985
Servicios de cine, radio y televisión	0.000564
Servicios personales, de reparación, actividade	0.000461
account processes, as copinione, accountant	
	Servicios agropecuarios \
Cultivo de cereales, oleaginosas y forrajeras	0.000000
Cultivo de hortalizas, legumbres, flores y plan	0.00000
Cultivo de frutas y nueces	0.00000
Cultivos industriales	0.00000
Producción de semillas	0.00000
Servicios sociales	0.00000
Servicios de saneamiento	0.000148
Actividad de asociaciones	0.000526
Servicios de cine, radio y televisión	0.000076
Servicios personales, de reparación, actividade	0.006604
borviorob porbonarob, do roparación, actividadom	0.00001
	Caza \
Cultivo de cereales, oleaginosas y forrajeras	0.00000e+00
Cultivo de hortalizas, legumbres, flores y plan	0.00000e+00
Cultivo de frutas y nueces	0.000000e+00
Cultivos industriales	0.000000e+00
Producción de semillas	0.000000e+00
Servicios sociales	0.000000e+00
Servicios de saneamiento	1.460399e-06
Actividad de asociaciones	4.849340e-06
Servicios de cine, radio y televisión	7.971898e-07
Servicios de cine, fadio y televisión Servicios personales, de reparación, actividade	
betvictos personates, de reparación, accividade	1.1001216-04

Silvicultura y extracción de

```
Cultivo de cereales, oleaginosas y forrajeras
Cultivo de hortalizas, legumbres, flores y plan...
0.000000
Cultivo de frutas y nueces
0.000000
Cultivos industriales
0.000000
Producción de semillas
0.000000
Servicios sociales
0.000000
Servicios de saneamiento
0.000078
Actividad de asociaciones
0.000276
Servicios de cine, radio y televisión
0.000423
Servicios personales, de reparación, actividade...
0.000856
                                                     ... Enseñanza pública \
Cultivo de cereales, oleaginosas y forrajeras
                                                                  0.000038
Cultivo de hortalizas, legumbres, flores y plan...
                                                               0.005679
Cultivo de frutas y nueces
                                                                 0.003827
Cultivos industriales
                                                                  0.000000
Producción de semillas
                                                                  0.000000
Servicios sociales
                                                                  0.000000
Servicios de saneamiento
                                                                  0.000000
Actividad de asociaciones
                                                                  0.000000
Servicios de cine, radio y televisión
                                                                  0.001680
Servicios personales, de reparación, actividade... ...
                                                               0.003071
                                                     Enseñanza privada \
Cultivo de cereales, oleaginosas y forrajeras
                                                              0.000080
Cultivo de hortalizas, legumbres, flores y plan...
                                                            0.012030
Cultivo de frutas y nueces
                                                               0.008099
Cultivos industriales
                                                               0.000000
Producción de semillas
                                                               0.000000
Servicios sociales
                                                               0.000000
```

Servicios de saneamiento	0.00000
Actividad de asociaciones	0.00000
Servicios de cine, radio y televisión	0.004810
Servicios personales, de reparación, actividade	0.006645
	Salud humana pública \
Cultivo de cereales, oleaginosas y forrajeras	0.000007
Cultivo de hortalizas, legumbres, flores y plan	
Cultivo de frutas y nueces	0.011067
Cultivos industriales	0.000000
Producción de semillas	0.00000
<u></u>	
Servicios sociales	0.00000
Servicios de saneamiento	0.000000
Actividad de asociaciones	0.00000
Servicios de cine, radio y televisión	0.000517
Servicios personales, de reparación, actividade	e 0.005508
	Salud humana privada \
Cultivo de cereales, oleaginosas y forrajeras	0.000042
Cultivo de hortalizas, legumbres, flores y plan	
Cultivo de frutas y nueces	0.005904
Cultivos industriales	0.00000
Producción de semillas	0.00000
rioduccion de bemiliab	0.00000
 Servicios sociales	0.00000
Servicios de saneamiento	0.00000
Actividad de asociaciones	0.482561
Servicios de cine, radio y televisión	0.007592
Servicios personales, de reparación, actividade	
•	
	Servicios veterinarios \setminus
Cultivo de cereales, oleaginosas y forrajeras	0.000012
Cultivo de hortalizas, legumbres, flores y plan	
Cultivo de frutas y nueces	0.000000
Cultivos industriales	0.000000
Producción de semillas	0.000000
Servicios sociales	0.000000
Servicios de saneamiento	0.00000
Actividad de asociaciones	0.005531
Servicios de cine, radio y televisión	0.000178
Servicios personales, de reparación, actividade	0.000005
	Servicios sociales \
Cultivo de cereales, oleaginosas y forrajeras	0.00000
Cultivo de hortalizas, legumbres, flores y plan	
<u> </u>	

Cultivo de frutas y nueces Cultivos industriales Producción de semillas	0.001638 0.000000 0.000000
Servicios sociales Servicios de saneamiento Actividad de asociaciones Servicios de cine, radio y televisión Servicios personales, de reparación, actividade	0.000000 0.000000 0.000000 0.000856 0.002243
Cultivo de cereales, oleaginosas y forrajeras Cultivo de hortalizas, legumbres, flores y plan Cultivo de frutas y nueces Cultivos industriales Producción de semillas Servicios sociales Servicios de saneamiento Actividad de asociaciones Servicios de cine, radio y televisión Servicios personales, de reparación, actividade	Servicios de saneamiento 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.001262 0.002969
Cultivo de cereales, oleaginosas y forrajeras Cultivo de hortalizas, legumbres, flores y plan Cultivo de frutas y nueces Cultivos industriales Producción de semillas Servicios sociales Servicios de saneamiento Actividad de asociaciones Servicios de cine, radio y televisión Servicios personales, de reparación, actividade	Actividad de asociaciones \
televisión \ Cultivo de cereales, oleaginosas y forrajeras 1.735054e-09 Cultivo de hortalizas, legumbres, flores y plan 1.237709e-06 Cultivo de frutas y nueces 1.472092e-07 Cultivos industriales 0.000000e+00 Producción de semillas	Servicios de cine, radio y

```
0.000000e+00
Servicios sociales
0.000000e+00
Servicios de saneamiento
0.000000e+00
Actividad de asociaciones
3.831347e-03
Servicios de cine, radio y televisión
8.850829e-02
Servicios personales, de reparación, actividade...
5.858829e-02
                                                     Servicios personales, de
reparación, actividades deportivas y de esparcimiento
Cultivo de cereales, oleaginosas y forrajeras
Cultivo de hortalizas, legumbres, flores y plan...
0.079292
Cultivo de frutas y nueces
0.00006
Cultivos industriales
0.000000
Producción de semillas
0.000000
Servicios sociales
0.000000
Servicios de saneamiento
0.000000
Actividad de asociaciones
Servicios de cine, radio y televisión
0.017336
Servicios personales, de reparación, actividade...
0.054336
[123 rows x 123 columns]
```

- [5]: # Visualización del grafo en spring layout con centralidad de autovalores.

 # La centralidad de autovalores mide la influencia de un nodo sobre la redu

 basado en la influencia de sus vecinos.
- [6]: # Visualización del grafo con todos los nodos distribuidos y legibles.

```
[7]: # Métricas del grafo. Distribución de grado de salida y de entrada. Coeficiente...

de clustering, etc. El output de esto deberían ser los nodos sobre los...

cuales vamos a hacer los experimentos y por qué. Eigenvector centrality.
```

4 Los experimentos

```
[8]: def plot inflaciones(inflaciones, aumento, umbral label=0.5):
         fig, (ax,ax2) = plt.subplots(1,2,figsize=(22, 10))
         ax.set_title(f"Valor de la inflación en el tiempo, a partir de un shock de⊔

¬{aumento}% \n para todos los sectores.")
         ax.set_xlabel("Tiempo")
         ax.set_ylabel("Porcentaje de inflación")
         ax2.set_title(f"Variación de la inflación en el tiempo, a partir de un∪
      ⇒shock de {aumento}%, \n para todos los sectores.")
         ax2.set_xlabel("Tiempo")
         ax2.set_ylabel("Porcentaje de inflación")
         for i, inflacion in enumerate(inflaciones):
             tiempo = range(0,len(inflacion))
             if (max(inflacion) > umbral_label):
                 ax.plot(tiempo, inflacion, label=sectores[i])
                 ax2.plot(tiempo[1:],np.diff(inflacion), label=sectores[i])
             else:
                 ax.plot(tiempo, inflacion)
                 ax2.plot(tiempo[1:],np.diff(inflacion))
         ax.legend()
         ax2.legend()
         plt.show()
```

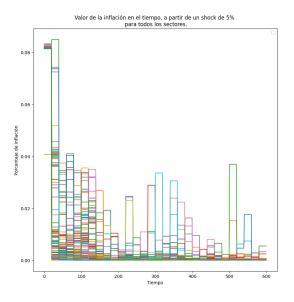
Esqueleto de experimento: - visualizacion del nodo desde donde se lanza el shock con su vecindario (dos niveles de separación como mucho?) ó de su clúster. - codigo de shock de inflación - visualizacion de la inflacion en funcion de t - grafo total pintado por la inflacion.

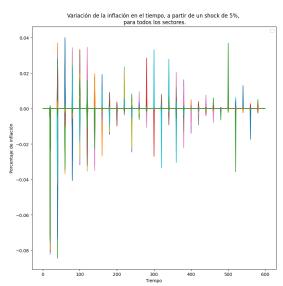
4.1 Experimentación con la dinámica 1.

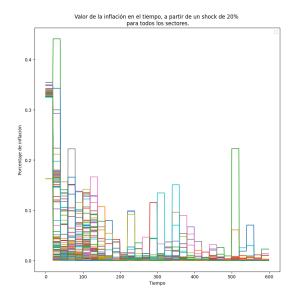
```
inflacion = experimento.metricas_evaluadas['inflacion']
inflaciones.append(inflacion)
plot_inflaciones(inflaciones,aumento)
```

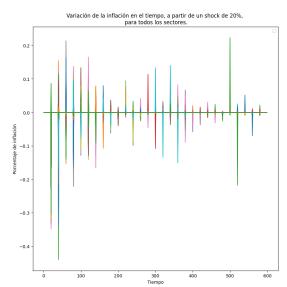
C:\Users\augus\AppData\Local\Temp\ipykernel_14720\2569110641.py:17: UserWarning:
No artists with labels found to put in legend. Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
ax.legend()

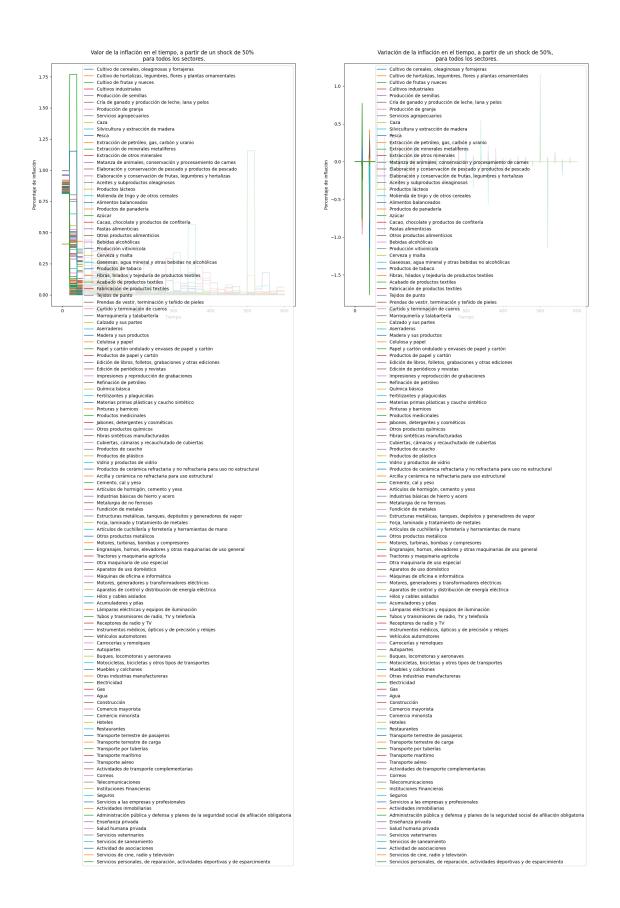
C:\Users\augus\AppData\Local\Temp\ipykernel_14720\2569110641.py:18: UserWarning:
No artists with labels found to put in legend. Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
ax2.legend()











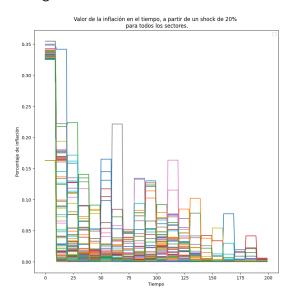
5 Experimentos con la dinámica 3.

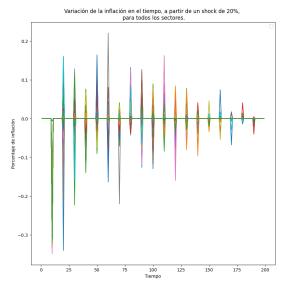
5.0.1 alpha = 0 (Igual que la dinámica 1)

```
[10]: aumentos = [20]
      alpha = 0
      mip_grafo = grafo.armar_grafo(mip,precios_random=False)
      sectores = list(mip_grafo.nodes)
      for j, aumento in enumerate(aumentos):
          inflaciones = []
          for sector in sectores:
              experimento = exp.Experimento(mip_grafo,
                                       dinamica=modelo.dinamica_3,
                                       calcular_inflacion=modelo.calcular_inflacion,
                                       alpha=alpha)
              experimento.shock(sector,aumento)
              experimento.step(200)
              inflacion = experimento.metricas_evaluadas['inflacion']
              inflaciones.append(inflacion)
          plot_inflaciones(inflaciones, aumento)
```

C:\Users\augus\AppData\Local\Temp\ipykernel_14720\2569110641.py:17: UserWarning:
No artists with labels found to put in legend. Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
ax.legend()

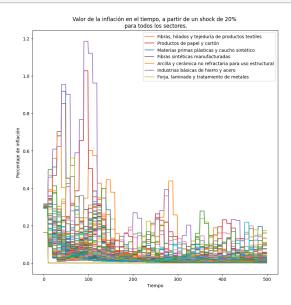
C:\Users\augus\AppData\Local\Temp\ipykernel_14720\2569110641.py:18: UserWarning: No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument. ax2.legend()

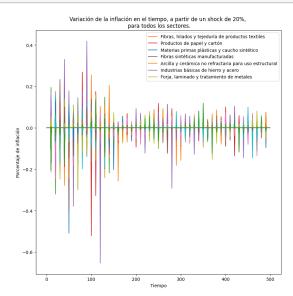




5.0.2 Alpha = 0.2

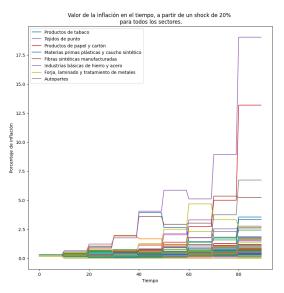
```
[11]: aumentos = [20]
      umbral_visualizacion = [1000,2000,2000]
      alpha = 0.2
      mip_grafo = grafo.armar_grafo(mip,precios_random=False)
      sectores = list(mip_grafo.nodes)
      for j, aumento in enumerate(aumentos):
          inflaciones = []
          for sector in sectores:
              experimento = exp.Experimento(mip_grafo,
                                      dinamica=modelo.dinamica_3,
                                      calcular_inflacion=modelo.calcular_inflacion,
                                      alpha=alpha)
              experimento.shock(sector,aumento)
              experimento.step(500)
              inflacion = experimento.metricas_evaluadas['inflacion']
              inflaciones.append(inflacion)
          plot_inflaciones(inflaciones,aumento,0.5)
```

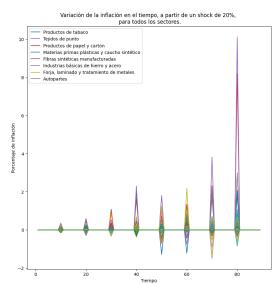




5.0.3 Alpha 0.3

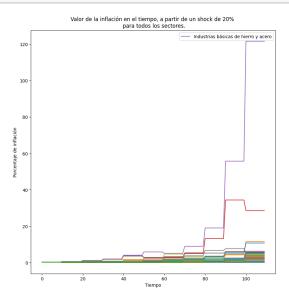
```
\lceil 12 \rceil: aumento = 20
      # umbral_visualizacion = [1000]
      alpha = 0.3
      mip_grafo = grafo.armar_grafo(mip,precios_random=False)
      sectores = list(mip_grafo.nodes)
      inflaciones = []
      for sector in sectores:
          experimento = exp.Experimento(mip_grafo,
                                   dinamica=modelo.dinamica_3,
                                   calcular_inflacion=modelo.calcular_inflacion,
                                   alpha=alpha)
          experimento.shock(sector,aumento)
          experimento.step(90)
          inflacion = experimento.metricas_evaluadas['inflacion']
          inflaciones.append(inflacion)
      plot_inflaciones(inflaciones, aumento,3)
```

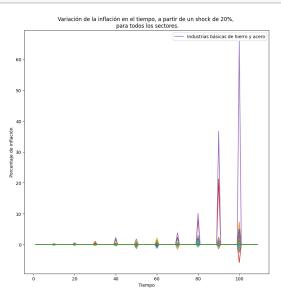




El mismo experimento pero con más pasos.

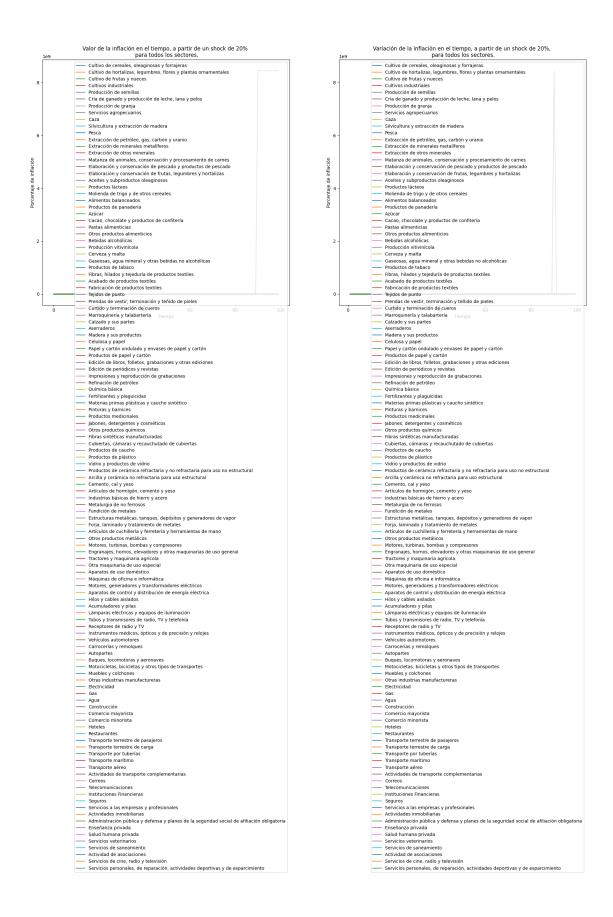
```
[13]: aumento = 20
# umbral_visualizacion = [1000]
alpha = 0.3
mip_grafo = grafo.armar_grafo(mip,precios_random=False)
sectores = list(mip_grafo.nodes)
inflaciones = []
for sector in sectores:
    experimento = exp.Experimento(mip_grafo,
```





5.0.4 Alpha 0.4

```
[14]: aumento = 20
      # umbral_visualizacion = [1000]
      alpha = 0.4
      mip_grafo = grafo.armar_grafo(mip,precios_random=False)
      sectores = list(mip_grafo.nodes)
      inflaciones = []
      for sector in sectores:
          experimento = exp.Experimento(mip_grafo,
                                  dinamica=modelo.dinamica_3,
                                  calcular_inflacion=modelo.calcular_inflacion,
                                  alpha=alpha)
          experimento.shock(sector,aumento)
          experimento.step(100)
          inflacion = experimento.metricas_evaluadas['inflacion']
          inflaciones.append(inflacion)
      plot_inflaciones(inflaciones, aumento)
```



6 Conclusiones

6.1 Sobre los experimentos y las preguntas respondidas

• Los nodos 112, 113 y 6 son los más influyentes en la red según la centralidad de autovalores.

6.2 Próximos pasos y preguntas no respondidas

•

6.2.1 Mejoras para hacer al modelo

- Se podría implementar algo que permita una suerte de oferta-demanda (nodo consumidor final?) AMPLIAR
 - * Romper ley de Say
- Se podría taggear a los sectores productivos por "rubro" (por ej, electricidad, gas y agua serían rubro "servicios esenciales") y estudiar qué sucede dentro de cada rubro. También qué interacciones inflacionarias tienen ese rubro con otros rubros.

6.3 Julio Olivera

Inflación estructural: La inflación estructural se refiere a un tipo de inflación causada por desajustes sectoriales que afectan a productos específicos, lo que resulta en aumentos de precios individuales que luego se generalizan. Estos desajustes pueden ser el resultado de la rigidez de la estructura productiva y la imperfección de los mercados, y son ajenos a las decisiones de las autoridades monetarias. La inflación estructural se origina en desequilibrios sectoriales que afectan a productos específicos y no en un desajuste global entre la oferta y la demanda monetaria.

Olivera, J. H. G. (1965), "Inflación estructural y política financiera".

Podemos notar que este modelo apoya la noción de que la inflación tiene un origen estructural. El hecho de que exista un alfa tal que si el aumento local de precios en un sector productivo estuviese dado por alfa*inflación_global, podría ser interpretado como una descoordinación entre los agentes. Estos hitos en la red estarían dando lugar a una espiralización de la inflación del modelo, que podría ser interpretada como la generalización de aumentos de precios individuales.

7 Apéndices

7.1 Código

- Experimento
- Bla

7.1.1 Experimento

```
[15]: exp.Experimento??

Init signature:
exp.Experimento(
    grafo,
    dinamica,
    duracion_periodo=10,
    metricas=None,
    calcular_inflacion=None,
    alpha=0.5,
)
Docstring: <no docstring>
Source:
```

```
class Experimento:
  def __init__(self, grafo, dinamica, duracion_periodo = 10, metricas=None,__

¬calcular_inflacion=None, alpha = 0.5):
    0.00
    Inicializa una instancia de la clase Experimento.
    Parámetros
    grafo : DiGraph
      Grafo que representa el MIP
    dinamica : Callable
      Función con la siguiente signatura : (aumento, peso_arista, inflacion, u
 →alpha)
    duracion_periodo : int, opcional
        La duración de cada período en pasos para actualizar la inflacion, poru
 ⇔defecto 10 pasos.
    metricas : Dict(string,Callable), opcional
        Diccionario de con metricas a computar sobre el grafo, ej,

→{"cantidad_nodos", lambda grafo: len(grafo)}
    calcular_inflacion : Callable, opcional
        Función para calcular la inflación en un período, recibe dos arrays deu
 →precios. ej, calc(precios_actual, precios_pasado)
    alpha: float, opcional
        Parametro de ponderacion
        El alpha es CUANTO ve la inflación.
        Alpha == 1 es lo mismo que la dinamica 2.
        Alpha == 0 es lo mismo que la dinamica 1.
    0.00
    self.grafo = grafo.copy()
    self.dinamica = dinamica
    self.metricas = metricas or {}
    self._curr_step = 0
    self.metricas_evaluadas = {nombre:[] for nombre in self.metricas}
    self.metricas_evaluadas['inflacion'] = []
    self.queue = queue.Queue() #Fila de tuplas (Nodo, aumento)
    self.duracion_periodo = duracion_periodo
    self.precios_periodo_pasado = {}
    for nodo in self.grafo.nodes:
```

```
c:\users\augus\dev\gh\akielbowicz\tp-msscae-2024\src\experimento.
      ∽ру
     Type:
                     type
     Subclasses:
     7.1.2 Dinámicas
[16]: modelo.dinamica_1??
     Signature: modelo.dinamica_1(aumento, peso_arista, inflacion=0.0, alpha=1.0)
     Docstring: <no docstring>
     Source:
     def dinamica_1(aumento, peso_arista, inflacion=0.0, alpha=1.0):
         aumento_vecino = aumento * peso_arista
         return aumento vecino
                c:\users\augus\dev\gh\akielbowicz\tp-msscae-2024\src\modelo.py
     File:
     Type:
                function
[17]: modelo.dinamica_2??
     Signature: modelo.dinamica_2(aumento, peso_arista, inflacion, alpha=1.0)
     Docstring: <no docstring>
     Source:
     def dinamica_2(aumento, peso_arista, inflacion, alpha=1.0):
         aumento_vecino = inflacion
         return aumento_vecino
                c:\users\augus\dev\gh\akielbowicz\tp-msscae-2024\src\modelo.py
     File:
     Type:
                function
[18]: modelo.dinamica_3??
     Signature: modelo.dinamica_3(aumento, peso_arista, inflacion, alpha)
     Docstring: <no docstring>
     Source:
     def dinamica_3(aumento, peso_arista, inflacion, alpha):
         aumento_vecino = (alpha * inflacion) + ((1 - alpha) * ((aumento) *__
      →peso_arista))
         return aumento_vecino
                c:\users\augus\dev\gh\akielbowicz\tp-msscae-2024\src\modelo.py
     File:
     Type:
                function
```

File: