PRÁCTICA 3

a) Problema de las N-Torres (descripción del problema e interpretación de los resultados obtenidos al ejecutar el archivo Torres.m)

El problema de las N torres consiste en poner en un tablero NxN N torres de forma que ninguna torre se ataque a otra.

```
>> Torres

S =

     1     0     0     0
     0     0     0     1
     0     0     1     0
     0     1     0     0


S =

     1     0     0     0
     0     0     0     1
     0     0     1     0
     0     1     0     0


S =

     1     0     0     0
     0     0     0     1
     0     0     1     0
     0     1     0     0


S =

     1     0     0     0
     0     0     0     1
     0     0     1     0
     0     1     0     0
```

Imagen 1. 4X4 con Si = zeros(N,N)

```
>> Torres

S =

     0     1     1     1
     1     1     1     0
     1     1     0     1
     1     0     1     1


S =

     0     0     1     1
     0     1     1     0
     1     1     0     0
     1     0     0     1


S =

     0     0     1     1
     0     1     1     0
     1     1     0     0
     1     0     0     1


S =

     0     0     1     1
     0     1     1     0
     1     1     0     0
     1     0     0     1
```

Imagen 2. 4X4 con Si = ones(N,N)

```
S =

     0     1     0     0
     0     0     0     0
     0     1     1     0
     1     0     1     1


S =

     0     1     0     0
     0     0     0     0
     0     1     1     0
     1     0     1     1


S =

     0     1     0     0
     0     0     0     0
     0     1     1     0
     1     0     0     1


S =

     0     1     0     0
     0     0     0     0
     0     1     1     0
     1     0     0     1


S =

     0     1     0     0
     0     0     0     0
     0     1     1     0
     1     0     0     1
```

```
S =

     1     1     1     1
     1     0     0     0
     0     0     1     1
     1     0     0     1


S =

     0     1     1     1
     1     0     0     0
     0     0     1     1
     1     0     0     1


S =

     0     1     1     1
     1     0     0     0
     0     0     1     0
     1     0     0     1


S =

     0     1     0     1
     1     0     0     0
     0     0     1     0
     1     0     0     1


S =

     0     1     0     1
     1     0     0     0
     0     0     1     0
     1     0     0     1
```

Imagen 3. Dos 4X4 con Si aleatorios

Como podemos ver en las imágenes, solo con la entrada inicial con todos los valores a 0 sale a la primera una solución al problema. Los demás seguramente se hayan quedado en un mínimo local, por lo que probablemente nunca lleguen a una solución si se sigue computando. Sin embargo, en la ejecución del programa el W no es igual que como sale en las diapositivas, ya que no pone -2 en las entradas Wijik con k!=j y Wijrj con r!=i y Wijij=0.

b) Bipartición de un grafo (proponer un problema y resolverlo con distintas soluciones iniciales y distintos valores del parámetro lambda)

Queremos dividir en dos clústeres un sistema que se puede transpolar a un grafo.

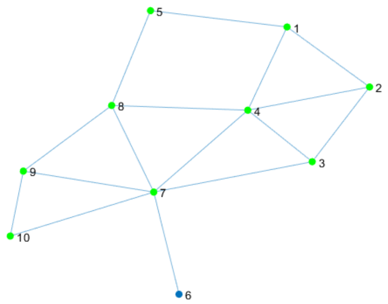

Imagen 4. Bipartición con landa=0.1

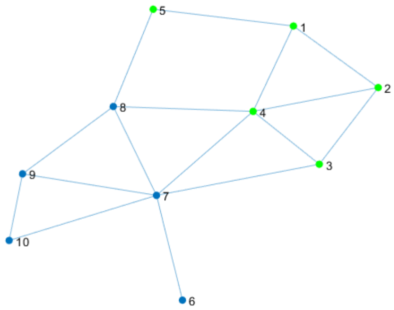

Imagen 5. Bipartición con landa=0.5

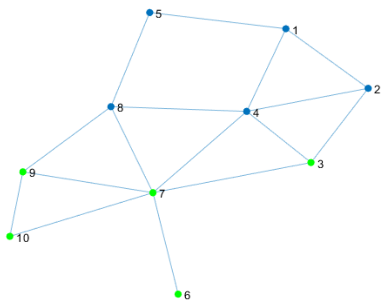

Imagen 6. Bipartición con landa=0.9

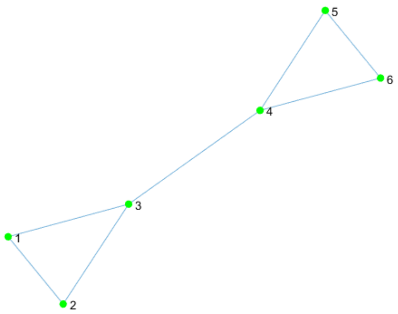

Imagen 7. Bipartición con landa=0.1

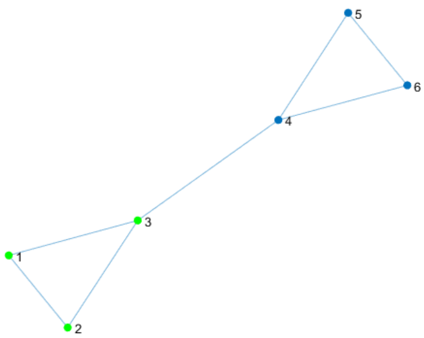

Imagen 8. Bipartición con landa=0.5

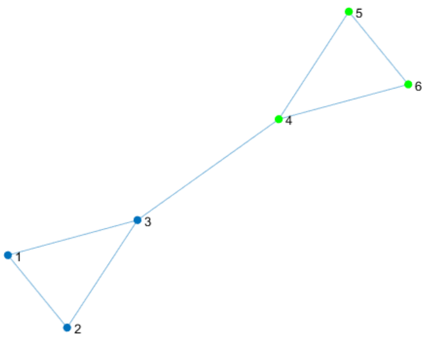

Imagen 9. Bipartición con landa=0.9

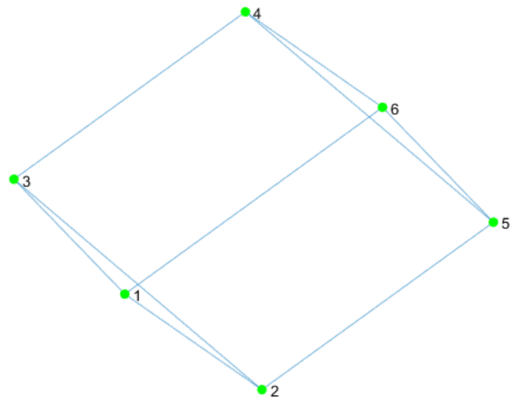

Imagen 10. Bipartición con landa=0.1

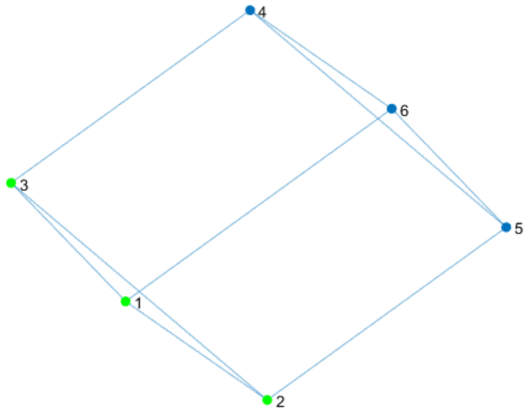

Imagen 11. Bipartición con landa=0.5

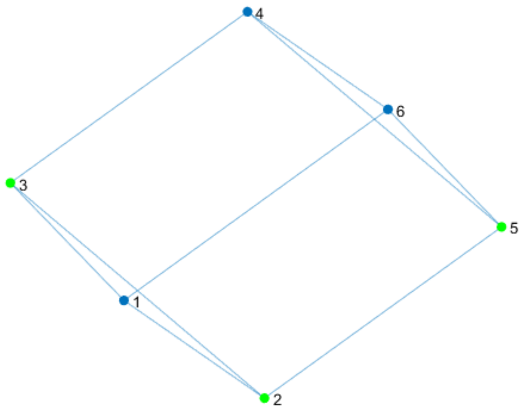

Imagen 12. Bipartición con landa=0.9

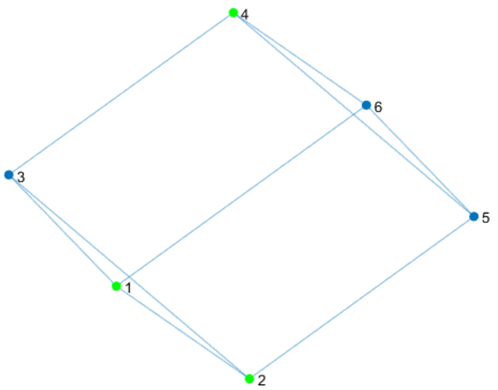

Imagen 13. Bipartición con landa=50

Como vemos en las imágenes, cuando menor es landa, mas agrupa a un grupo que al otro, y cuando mas alto, mejor separa. Sin embargo si están muy co-enlazados sus vértices, como podemos apreciar para las imágenes 10, 11, 12 y 13, pues puede a llegar a darnos un resultado erróneo, por lo que puede llegar a un estado mínimo local.