AmbitiousClique

March 18, 2018

This one is going for $k = 36$.

In [1]:
```
from RRR import *
from rankProject import *
from cliqueProject import *

n = 5

H = np.array(
    [[0, 1, 1, 1, 1],
    [1, 0, 1, 0, 0],
    [1, 1, 0, 0, 0],
    [1, 0, 0, 0, 0],
    [1, 0, 0, 0, 0]])

A = semiDefProject(np.random.rand(n, n))

print A
```

```
[[ 0.89689362  0.65275567  0.20386811  0.23239795  0.47663747]
 [ 0.65275567  0.41932745  0.35422516  0.43831328  0.61255364]
 [ 0.20386811  0.35422516  0.49815384  0.43824569  0.76018794]
 [ 0.23239795  0.43831328  0.43824569  0.48549436  0.77710255]
 [ 0.47663747  0.61255364  0.76018794  0.77710255  0.72948761]]
```

In [2]:
```
n = 125
from graphReader import getGraph
H = getGraph()
```

In [3]:
```
def verifyClique(c, H):
    r = 1
    for i in range(len(c)):
        for j in range(i+ 1, len(c)):
            r = r*H[c[i], c[j]]
    return r
```

In [4]:
```
k = 36
```

```
        A = semiDefProject(np.random.rand(n, n))

        Y, errors, sols = RRR(A,
                              lambda x: cliqueProject(x, k, H),
                              lambda x: rankProject(x, 1, False),
                              0.5, 1e-12, 10, True)
```

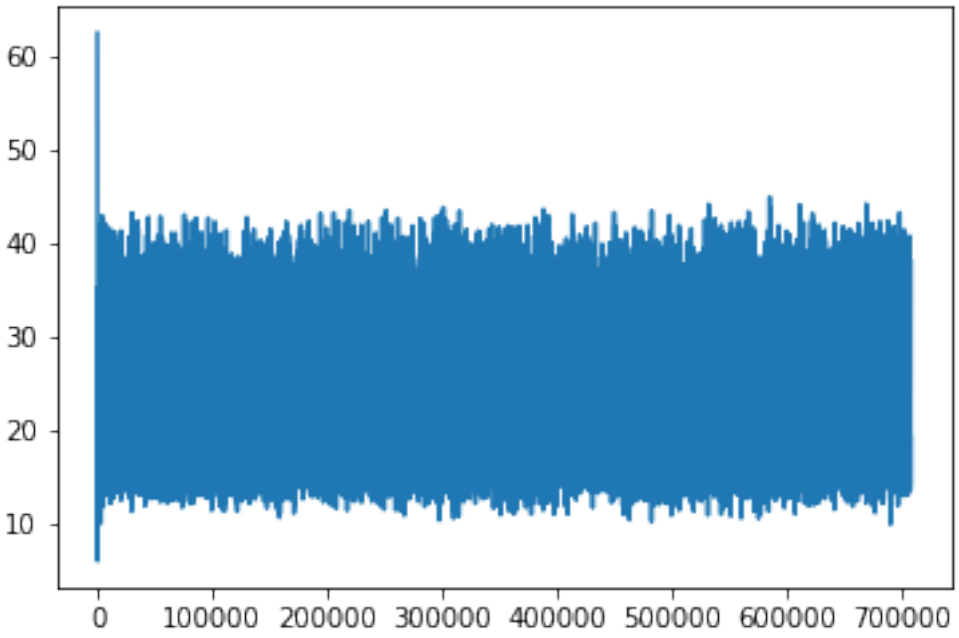
```
Warning: maximum iterations exceeded, no convergence
```

```
In [9]: while(True):
            Y, errors2, sols = RRR(Y,
                              lambda x: cliqueProject(x, k, H),
                              lambda x: rankProject(x, 1, False),
                              0.5, 1e-12, 1000, True)
            errors += errors2
            if errors[-1] < 1e-4:
                break
```

```
        ---------------------------------------------------------------------------

        KeyboardInterrupt                         Traceback (most recent call last)

        <ipython-input-9-413345873064> in <module>()
          3                     lambda x: cliqueProject(x, k, H),
          4                     lambda x: rankProject(x, 1, False),
    ----> 5                     0.5, 1e-12, 1000, True)
          6     errors += errors2
          7     if errors[-1] < 1e-4:


        /home/atbolsh/VeitElserModule/HW4/RRR.pyc in RRR(v, proj1, proj2, beta, cutoff, maxIter,
         14     while norm(error) > cutoff and i < maxIter:
         15         i += 1
    ---> 16         error = RRR_error(v, proj1, proj2)
         17         v = v + beta*error
         18         if errors:


        /home/atbolsh/VeitElserModule/HW4/RRR.pyc in RRR_error(v, proj1, proj2)
          5 def RRR_error(v, proj1, proj2):
          6     b = proj2(v)
    ----> 7     a = proj1(2*b - v)
          8     return a - b
          9
```

```
    <ipython-input-9-413345873064> in <lambda>(x)
      1 while(True):
      2     Y, errors2, sols = RRR(Y,
----> 3                     lambda x: cliqueProject(x, k, H),
      4                     lambda x: rankProject(x, 1, False),
      5                     0.5, 1e-12, 1000, True)


    /home/atbolsh/VeitElserModule/HW4/cliqueProject.pyc in cliqueProject(A, k, H)
     39     #Switch to a line
     40     B = B.reshape(s[0]*s[1])
---> 41     ind = np.argsort(B)
     42     rev = np.argsort(ind)
     43     #Switch to ordered


    /home/atbolsh/anaconda2/lib/python2.7/site-packages/numpy/core/fromnumeric.pyc in argsor
    905
    906     """
--> 907     return _wrapfunc(a, 'argsort', axis=axis, kind=kind, order=order)
    908
    909


    /home/atbolsh/anaconda2/lib/python2.7/site-packages/numpy/core/fromnumeric.pyc in _wrapf
     55 def _wrapfunc(obj, method, *args, **kwds):
     56     try:
---> 57         return getattr(obj, method)(*args, **kwds)
     58
     59     # An AttributeError occurs if the object does not have


    KeyboardInterrupt:
```

In [8]:
```
plt.plot(errors)
plt.show()
```



It's worth noting that the error trace for $k = 36$ and $k = 35$ look very different.

$k = 35$ stays quite close to 0, while $k = 36$ seems bounded away from it. Perhaps there is no solution at $k = 36$? Or just, most sets are far from being cliques?