Disconnected Graph Layout with polyomino packing

Roger Williams 08Jun09

The goal is to try to understand disconnected graph layout. The original example used numbers like this:

```
x = RandomReal[1, {50 + 1}]
```

```
{0.100423, 0.157237, 0.241073, 0.649718, 0.729848, 0.706181, 0.0837941, 0.860834, 0.669379,
 0.0228935, 0.54663, 0.470419, 0.996927, 0.148674, 0.907021, 0.220041, 0.54439, 0.13459,
 0.387416, 0.728711, 0.752726, 0.527584, 0.90652, 0.743122, 0.458572, 0.522537, 0.157689,
 0.909696, 0.844896, 0.799682, 0.991708, 0.905753, 0.729819, 0.922898, 0.661673,
 0.201751, 0.198726, 0.946205, 0.808666, 0.766001, 0.26061, 0.598094, 0.890975,
 0.28242, 0.602877, 0.822951, 0.412995, 0.436776, 0.947476, 0.357703, 0.0106058}
```

Then an artificial network is created thusly

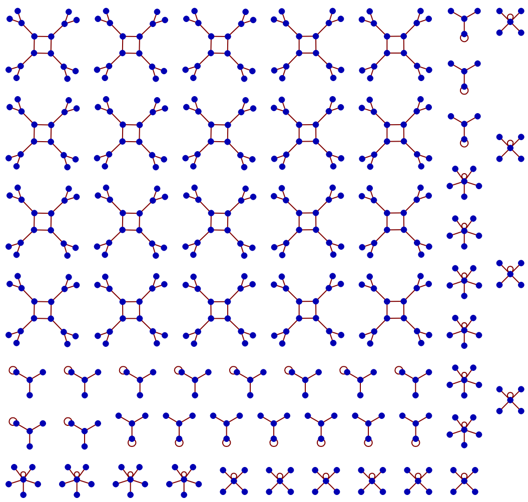
```
Table[x[[i + 1]] → x[[Mod[i^2, 50] + 1]], {i, 0, 50}]
```

```
{0.100423 → 0.100423, 0.157237 → 0.157237, 0.241073 → 0.729848,
 0.649718 → 0.0228935, 0.729848 → 0.54439, 0.706181 → 0.522537, 0.0837941 → 0.198726,
 0.860834 → 0.357703, 0.669379 → 0.907021, 0.0228935 → 0.905753, 0.54663 → 0.100423,
 0.470419 → 0.527584, 0.996927 → 0.602877, 0.148674 → 0.728711, 0.907021 → 0.412995,
 0.220041 → 0.522537, 0.54439 → 0.0837941, 0.13459 → 0.766001, 0.387416 → 0.458572,
 0.728711 → 0.470419, 0.752726 → 0.100423, 0.527584 → 0.598094, 0.90652 → 0.661673,
 0.743122 → 0.799682, 0.458572 → 0.157689, 0.522537 → 0.522537, 0.157689 → 0.157689,
 0.909696 → 0.799682, 0.844896 → 0.661673, 0.799682 → 0.598094, 0.991708 → 0.100423,
 0.905753 → 0.470419, 0.729819 → 0.458572, 0.922898 → 0.766001, 0.661673 → 0.0837941,
 0.201751 → 0.522537, 0.198726 → 0.412995, 0.946205 → 0.728711, 0.808666 → 0.602877,
 0.766001 → 0.527584, 0.26061 → 0.100423, 0.598094 → 0.905753, 0.890975 → 0.907021,
 0.28242 → 0.357703, 0.602877 → 0.198726, 0.822951 → 0.522537, 0.412995 → 0.54439,
 0.436776 → 0.0228935, 0.947476 → 0.729848, 0.357703 → 0.157237, 0.0106058 → 0.100423}
```

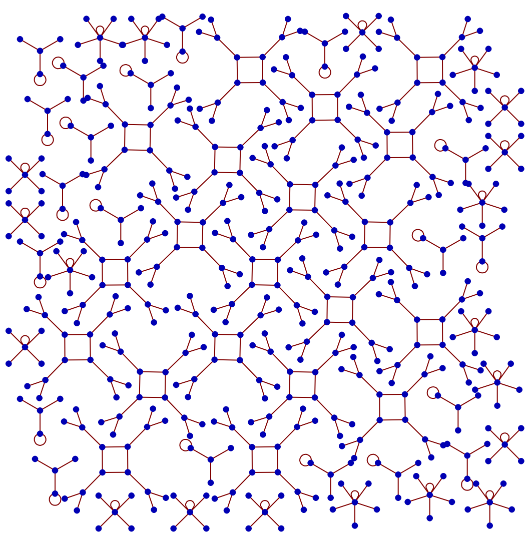
Then the graphs are created using the PackingMethod parameter

```
g = Flatten[Table[x = RandomReal[1, {50 + 1}];
    Table[x[[i + 1]] → x[[Mod[i^2, 50] + 1]], {i, 0, 50}], {10}]];
```

```
GraphPlot[g
 (*,PackingMethod→{"ClosestPackingCenter","Padding"→1} *)
]
```



```
GraphPlot[g
 , PackingMethod → {"ClosestPackingCenter", "Padding" → 1}
]
```



These graphs are a little abstract so I wanted to get a better feel for the nodes in these graphs...so I decided to use integers

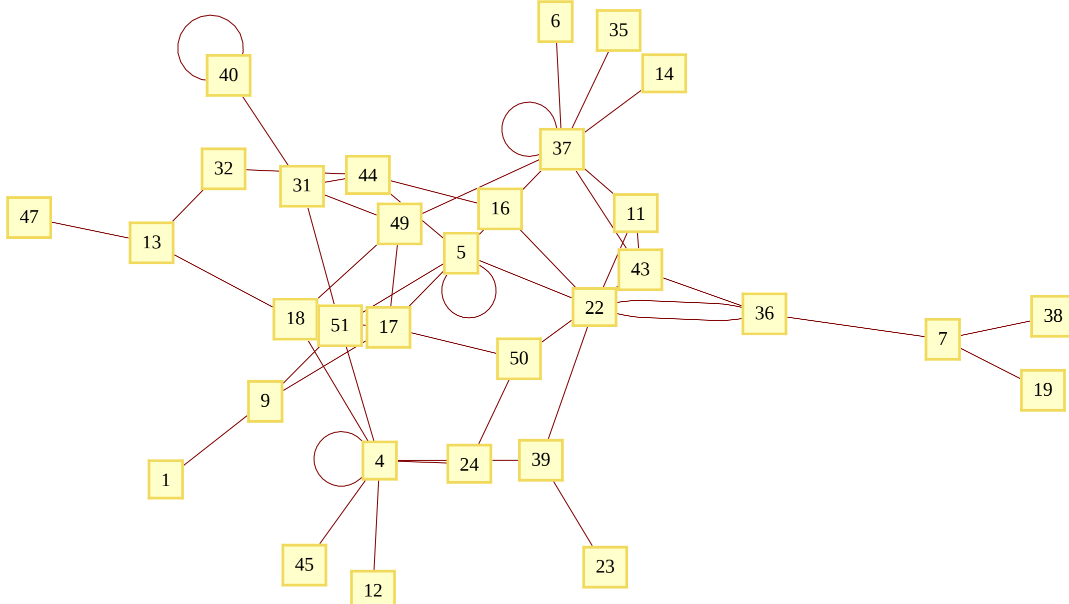
```
y = RandomInteger[{1, 51}, {50 + 1}]
```

```
{4, 40, 32, 23, 13, 49, 17, 51, 17, 39, 12, 36, 32, 38, 9, 14, 18, 24, 36, 7, 18, 43, 18, 43, 22, 37,
 5, 35, 31, 37, 24, 22, 16, 17, 49, 5, 5, 19, 16, 50, 45, 11, 1, 44, 44, 6, 51, 4, 47, 31, 51}
```

```
yRules = Table[y[[i + 1]] → y[[Mod[i^2, 50] + 1]], {i, 0, 50}]
```

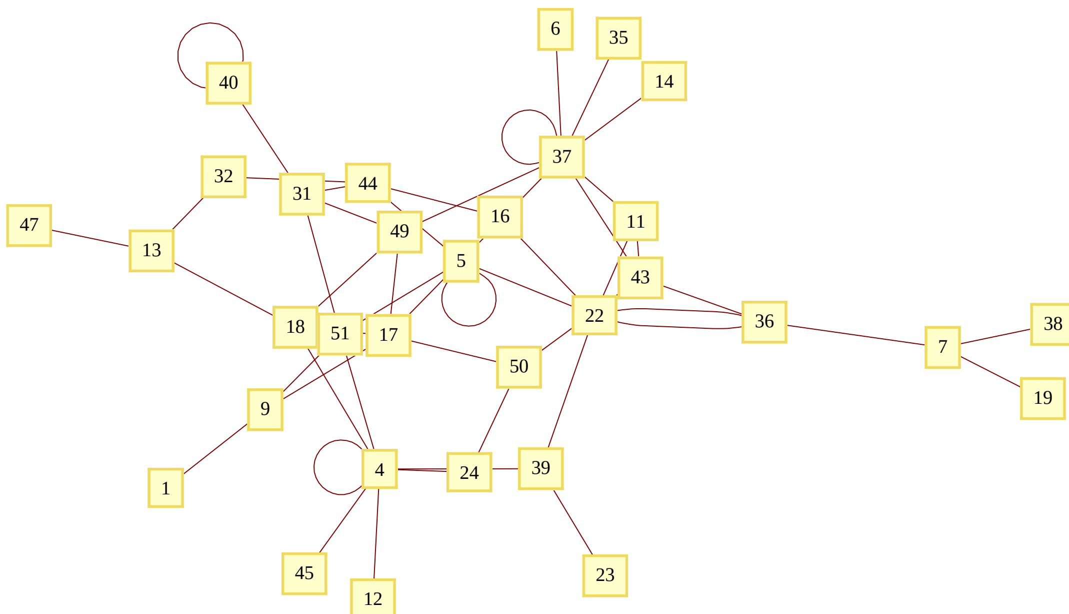
```
{4 → 4, 40 → 40, 32 → 13, 23 → 39, 13 → 18, 49 → 37, 17 → 5, 51 → 31, 17 → 9, 39 → 22,
 12 → 4, 36 → 43, 32 → 44, 38 → 7, 9 → 51, 14 → 37, 18 → 17, 24 → 50, 36 → 22, 7 → 36,
 18 → 4, 43 → 11, 18 → 49, 43 → 37, 22 → 5, 37 → 37, 5 → 5, 35 → 37, 31 → 49, 37 → 11,
 24 → 4, 22 → 36, 16 → 22, 17 → 50, 49 → 17, 5 → 37, 5 → 51, 19 → 7, 16 → 44, 50 → 43,
 45 → 4, 11 → 22, 1 → 9, 44 → 31, 44 → 5, 6 → 37, 51 → 18, 4 → 39, 47 → 13, 31 → 40, 51 → 4}
```

```
GraphPlot[yRules
 , VertexLabeling → True
 , ImageSize → 600
]
```



But trying to do packing without more than one graph is pointless...Doh!

```
GraphPlot[yRules
 , VertexLabeling → True
 , ImageSize → 600
 , PackingMethod → {"ClosestPackingCenter", "Padding" → 1}
]
```



So by adding a greater range of values, the same method gives some nice labelled graphs.

```
y = RandomInteger[{1, 51 000}, {50 + 1}]
```

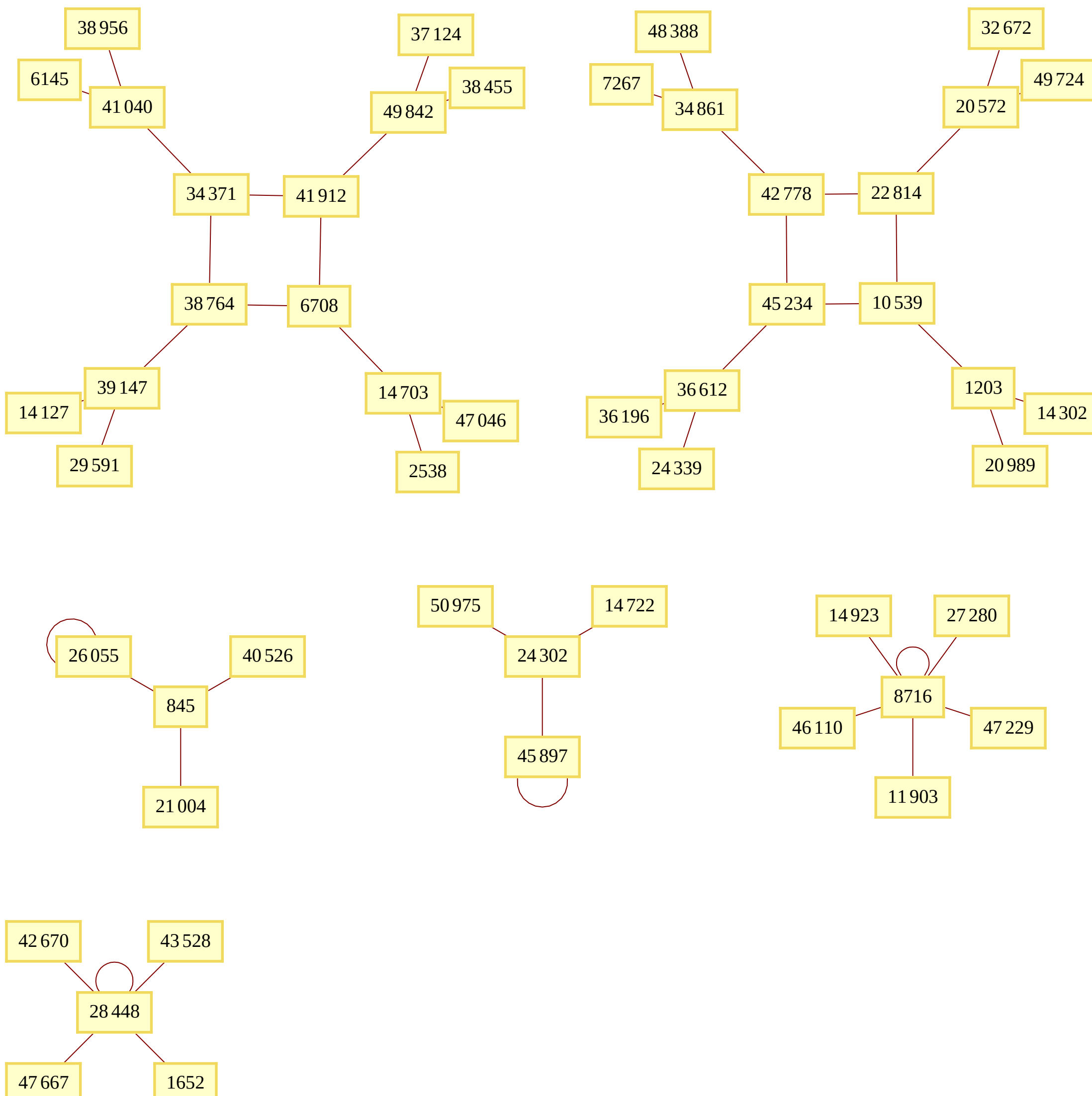
{8716, 26 055, 36 196, 37 124, 36 612, 42 670, 42 778, 21 004, 20 989, 49 842, 14 923, 6708, 32 672, 47 046, 1203, 47 667, 45 234, 14 127, 50 975, 14 703, 46 110, 38 764, 7267, 38 956, 24 302, 28 448, 45 897, 6145, 48 388, 41 040, 11 903, 41 912, 14 722, 29 591, 34 861, 1652, 22 814, 2538, 49 724, 39 147, 47 229, 34 371, 14 302, 40 526, 20 572, 43 528, 10 539, 38 455, 24 339, 845, 27 280}

```
yRules = Table[y⟦i + 1⟧ → y⟦Mod[i^2, 50] + 1⟧, {i, 0, 50}]
```

{8716 → 8716, 26 055 → 26 055, 36 196 → 36 612, 37 124 → 49 842, 36 612 → 45 234, 42 670 → 28 448, 42 778 → 22 814, 21 004 → 845, 20 989 → 1203, 49 842 → 41 912, 14 923 → 8716, 6708 → 38 764, 32 672 → 20 572, 47 046 → 14 703, 1203 → 10 539, 47 667 → 28 448, 45 234 → 42 778, 14 127 → 39 147, 50 975 → 24 302, 14 703 → 6708, 46 110 → 8716, 38 764 → 34 371, 7267 → 34 861, 38 956 → 41 040, 24 302 → 45 897, 28 448 → 28 448, 45 897 → 45 897, 6145 → 41 040, 48 388 → 34 861, 41 040 → 34 371, 11 903 → 8716, 41 912 → 6708, 14 722 → 24 302, 29 591 → 39 147, 34 861 → 42 778, 1652 → 28 448, 22 814 → 10 539, 2538 → 14 703, 49 724 → 20 572, 39 147 → 38 764, 47 229 → 8716, 34 371 → 41 912, 14 302 → 1203, 40 526 → 845, 20 572 → 22 814, 43 528 → 28 448, 10 539 → 45 234, 38 455 → 49 842, 24 339 → 36 612, 845 → 26 055, 27 280 → 8716}

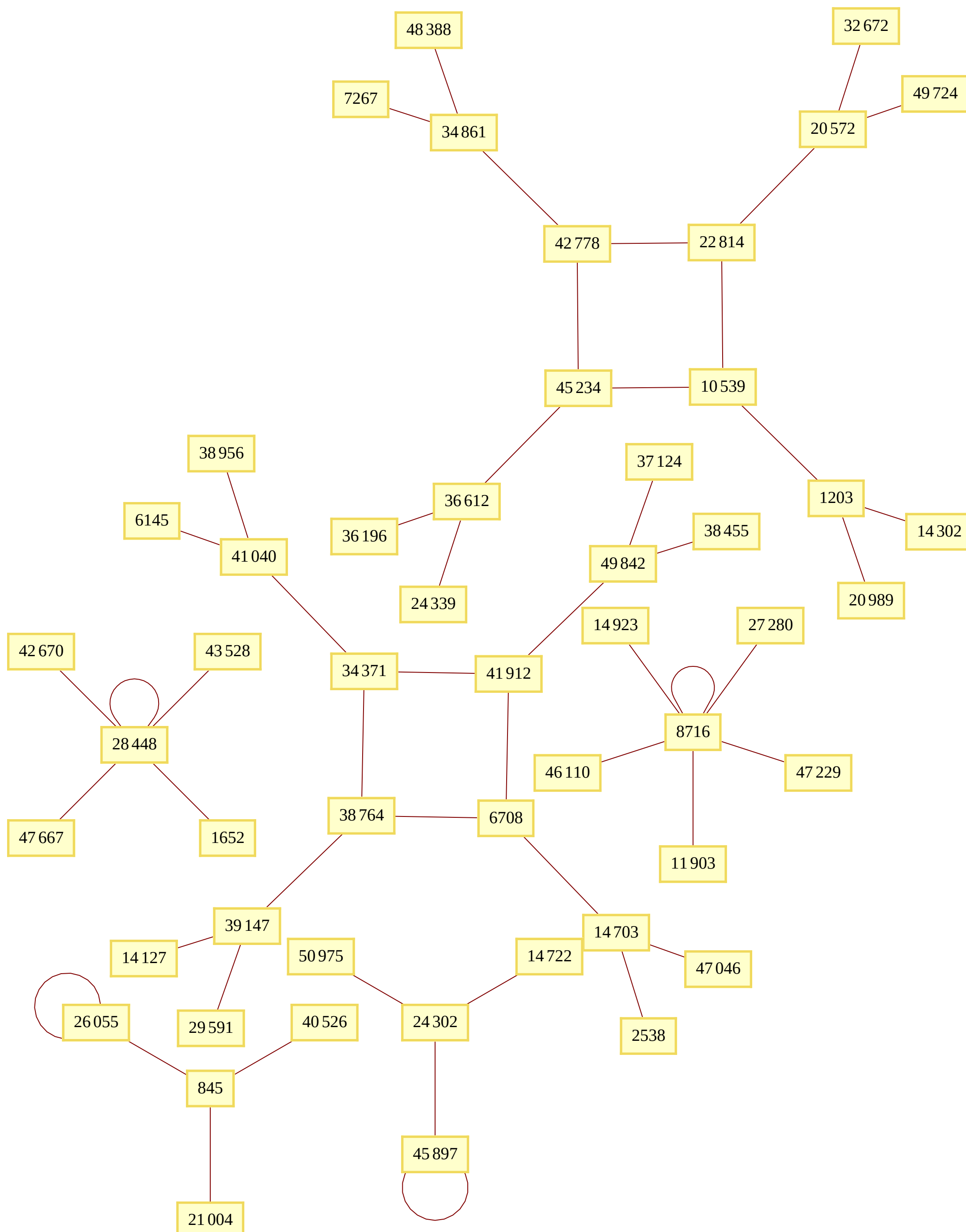
And they are disconnected now..

```
GraphPlot[yRules
 , VertexLabeling → True
 , ImageSize → 600
]
```



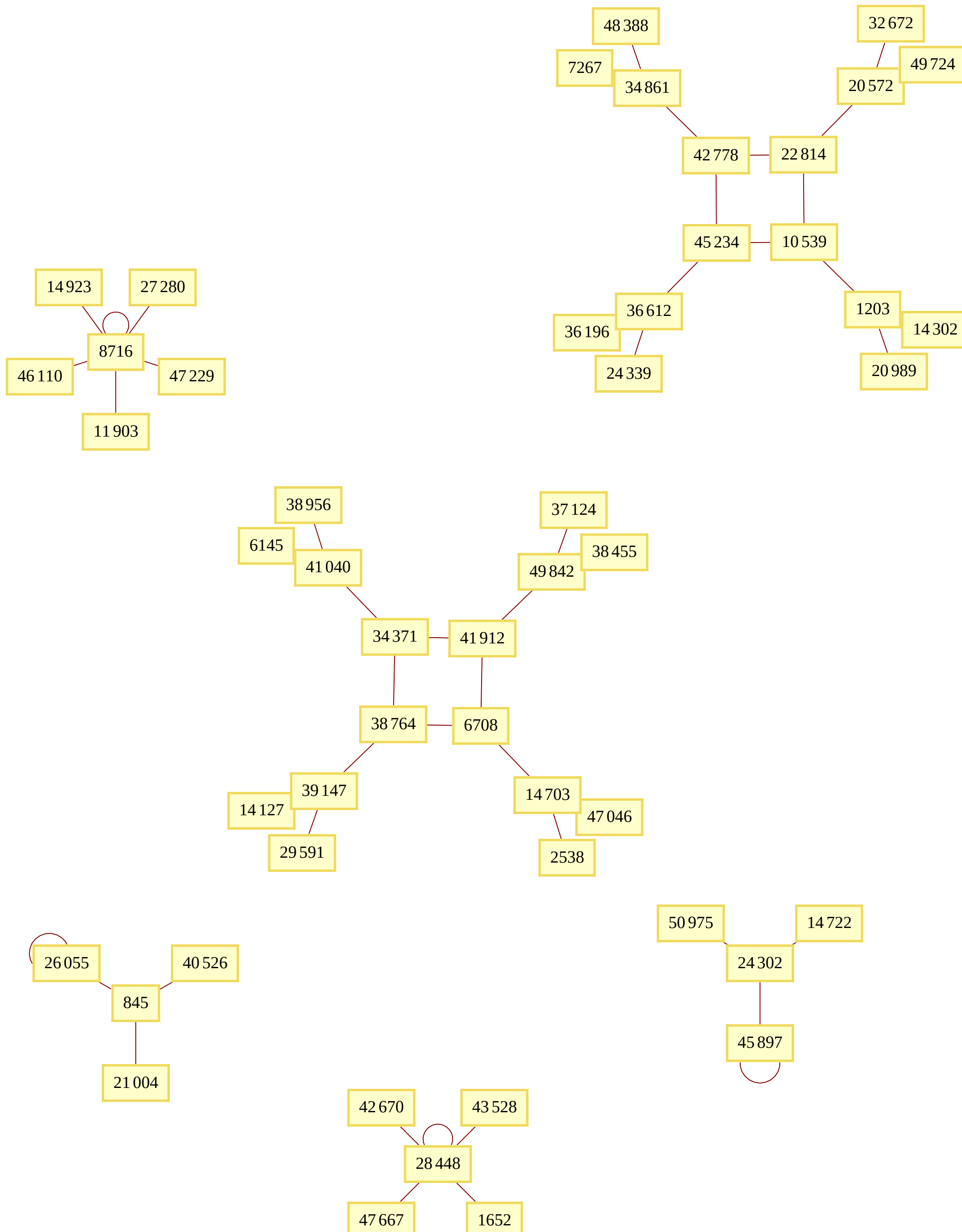
So now the disconnected options are more evident

```
GraphPlot[yRules
 , VertexLabeling → True
 , ImageSize → 600
 , PackingMethod → {"ClosestPackingCenter", "Padding" → 1}
]
```


48 388
32 672
7267
49 724
34 861
20 572
42 778
22 814
45 234
10 539
38 956
37 124
36 612
1203
6145
38 455
14 302
36 196
41 040
49 842
24 339
20 989
42 670
43 528
14 923
27 280
34 371
41 912
28 448
8716
46 110
47 229
38 764
6708
47 667
1652
11 903
39 147
14 703
14 127
50 975
14 722
47 046
26 055
29 591
40 526
24 302
2538
845
45 897
21 004

The polymonimo packing is definitely more evident now..

```
GraphPlot[yRules
 , VertexLabeling → True
 , ImageSize → 600
 , PackingMethod → {
   "ClosestPackingCenter"
   , "PolyominoNumber" → 5
   , "Padding" → 1}
]
```


48 388
32 672
7267
34 861
49 724
20 572
42 778
22 814
45 234
10 539
36 612
1203
36 196
14 302
24 339
20 989
14 923
27 280
8716
46 110
47 229
11 903
38 956
37 124
6145
41 040
38 455
49 842
34 371
41 912
38 764
6708
39 147
14 703
14 127
47 046
29 591
2538
50 975
14 722
26 055
40 526
24 302
845
45 897
21 004
42 670
43 528
28 448
47 667
1652