Optimisation Coursework

* A clique is a subset of the vertices of a graph such that the graph induced by these vertices is complete – that is, every pair of vertices in the clique are connected
* The empty set is a clique, and every singleton is a clique, every pair of connected vertices is a clique.
* A clique cover is a set of cliques such that every vertex belongs to at least one clique. The clique cover number of D, denoted , is the minimum number of cliques in any clique cover of D.
* A fraction al clique cover of a digraph D is a family of cliques, each of which is allocated a weight, such that for every vertex in the graph, the sum of the weights of the cliques is greater than or equal to 1.
* is the collection of all cliques of G. xs is the variable given to the subset of vertices S
* in our notation is 0 for any subset of vertices that is not a clique of the graph. Therefore is a number only given to cliques. Therefore when we are summing over all the possible subsets of vertices of our graph, we need only to sum over the cliques of our graph. here is a *weight* applied to our cliques, and the set of all forms a **fractional clique cover** as the

A fractional clique of a graph is a nonnegative real function on the vertices of G such that the sum of the values on the vertices of any independent set is at most

A *fractional clique cover* of a graph *G* is a weighting of the cliques of *G* with real numbers from the unit interval [0,1] with the property that, for any vertex *v* of *G*, the sum of the weights of the cliques containing *v* is at least 1; it is *regular* if the sum is equal to 1 for every vertex. The *fractional clique cover number* is the minimum, over all fractional clique covers, of the sum of the weights of all the cliques. It can be shown that the same minimum value is obtained if we restrict to regular fractional clique covers.

Assign a weight from the unit interval [0,1] to every clique of the graph. For every vertex in the graph, the sum of the weights of all the cliques containing the vertex is at least 1. If it is exactly 1 for all vertexes, the we say the cover is regular. The fractional clique cover number is the minimum, over all fractional clique covers, of the sum of the weights of all the cliques. It can be shown that the same minimum value is obtained if we restrict to regular fractional clique covers.

Finite Dynamical Systems and Notation:

* Network of entities has a **state**  (i.e it is an n-length vector of type q), and each takes a value from , and evolves according to a deterministic function where represents the update of the local state . The architecture of the update function can be represented via ints interaction graph - has as its vertex set, and there is an arc from if depends on . The study of **fixed points** on an FDS is crucial, as they represent stable states.
* The **q-guessing number** of a digraph (directed graph) is the logarithm of the maximum number of fixed points over all FDSs *f* whose interaction graph is a subgraph of . A feedback vertex set of a graph is a set of vertices whose removal leaves a graph without any cycles. The guessing number is always upper bounded by the size of a minimum feedback vertex set of D; if equality holds, we say that D is **solvable** over an *alphabet of size q* and the FDS that reaches this upper bound is called a **solution.**
* For a given digraph D, the supremum over all *q* of the *q-ary* guessing numbers of D is referred to as the **entropy** of D.
* A **bipartite graph** is a graph whose vertices can be divided into two disjoint and independent sets, such that every edge connects a vertex in U to a vertex in V – we denote it (A, B, E’), where A and B are the two sets and E’ is the set of edges that joins the vertices in the two graphs.
* A **simple graph** is a graph that has no multiple edges or loops.
* The **neighborhood** of a vertex *u* in G is denoted or just simply .
* If then the **interaction graph** of *f* is the graph on/of *n* vertices where
* The set of all functions whose interaction graph is contained in a digraph D is defined as .
* The guessing number of is defined as i.e the guessing number of (the update) function is the log to base q of size of the vector f, with the fix (round to the nearest integer) function applied to each function? The **q-guessing number** of D is then defined as
* The **entropy** of D is defined as . i.e the entropy is the supremum across all q of the q-guessing number of D.
* If D’ is a subgraph of D, then the guessing number of D’ is less than or equal to the guessing number of D. If two sets of vertex sets are disjoint, then the entropy of the union of these two sets is equal to the sum of the individual entropies – CHECK IF THE TWO GRAPHS ARE BIPARTITE OR SPLIT UP NICELY, MAY BE ABLE TO MORE QUICKLY COMPUTE THE ENTROPY
* The **Shannon entropy** of a graph D is defined as where the supremum is taken over all function (the power set of V – the set of all subsets of V) that satisfy the following properties:

i.e the function, which is defined on all subsets of vertices of the graph, must give a value less than or equal to 1 for every vertex in V

i.e the function must allocate a number to a subset that is smaller than the number allocated to the set containing the subset

i.e the sum of the value the function gives to two subsets must be greater than the sum of the values the function gives to the intersection and union of these two subsets

i.e the value the function gives to the the set of vertices

xV is a value given to every subset of matrices. We want the maximum value of this, given that:

* the value 0 is given to the empty set
* the value given to every individual vertex is less than 1
* the value given to the set of neighbours of a vertex is equal to the value given to the set of neighbours of the vertex plus the vertex itself
* the value given to any set must be greater than or equal to the value given to any of its subsets
* the value given to any subset is greater than or equal to 0

Model Using:

Xi = Value given to the ith subset of V

Xj st j e [0, n], n := number of vertices, are the subsets containing the single n vertices – MAY NEED TO OFFSET BY ONE as x[empty set] = 0

Jk

S\* = list of subsets

S = current subset

A = list of allowed vertices

Adding condition is when len(S) + len(X) = total number of vertice

Initially S = [0], A = [0, 1, 2, 3, 4]

S = current subgraph

N = Neighbours

X = forbidden

Subgraphs(S, A)

If len(S) + len(A) == numVert

Add S to A\*

For v in A:

N = Neighbours[v]

Subgraphs[S union v, N]

First constrain: x\_0 = 0

To Do:

* Greater than 0 constraints – add if needed?? - BOUNDS
* Remove notes dotted all over

To Talk About:

* Inverting constraints
* NEED TO MAKE SURE INPUT TO THE COMMAND IS IN THE RIGHT FORMAT – it needs to be python3 assignment.py graphs/GRAPH NAME

+ x10 - x38 = 0 (0,3), (0,3,4)

+ x15 - x44 = 0 (1,3), (1,2,3)

+ x24 - x62 = 0

+ x36 - x66 = 0

+ x51 - x71 = 0

+ x33 - x73 = 0

+ x58 - x95 = 0 (2,4,6) (2,3,4,6)