fraciso Documentation

Release 0.0.1-dev

Jeffrey Finkelstein

April 01, 2014

This package provides functions that decide whether two graphs are fractionally isomorphic, as well as functions that generate graphs fractionally isomorphic to a given initial graph.

Source code (github.com) · Packaging (pypi.python.org) · Issues (github.com)

CHAPTER 1

Installation

```
pip install fraciso
```

CHAPTER 2

Basic usage

```python
from fraciso import are_fractionally_isomorphic
from fraciso import coarsest_equitable_partition
from fraciso import random_fractionally_isomorphic_graph
from networkx import Graph

# Create some NetworkX graphs.
G = Graph()
H = Graph()

#
# Add some vertices and edges to graphs G and H here...
#

# Decide whether two graphs are fractionally isomorphic.
print('Isomorphic?', are_fractionally_isomorphic(G, H))

# Use linear programming to decide whether they are fractionally isomorphic.
#
# This requires PuLP and GLPK to be installed.
print('Isomorphic?', are_fractionally_isomorphic(G, H, algorithm='lp.pulp'))

# Generate a random graph that is fractionally isomorphic to G.
J = random_fractionally_isomorphic_graph(G)

# Compute the coarsest equitable partition of a graph.
P = coarsest_equitable_partition(G)
```

CHAPTER 3

Mathematical background

An undirected graph is a pair (V, E), where V is a set of vertices and E is a subset of $V \times V$. If we assume the graph has n vertices and $V = \{0, \ldots, n-1\}$, then the *adjacency matrix* of a graph is the matrix A in which entry A_{ij} is 1 if i is adjacent to j and 0 otherwise.

3.1 Graph isomorphism

Suppose A and B are the adjacency matrices of graphs G and H, respectively. The graphs G and H are *isomorphic* if the following linear program has a feasible solution.

$$\begin{aligned} AP &= PB \\ P1 &= 1 \\ 1^T P &= 1 \\ P &\in \{0,1\}^{n \times n} \end{aligned}$$

In other words, there is a permutation of the vertices of G that makes the adjacency matrices identical (any feasible P must be a permutation matrix).

3.2 Fractional graph isomorphism

The problem of determining whether two graphs are isomorphic is a computationally difficult problem. However, a relaxation of the graph isomorphism problem is computationally tractable. The *fractional graph isomorphism* problem is the problem of determining whether the following linear program has a feasible solution.

$$\begin{aligned} AS &= SB \\ S1 &= 1 \\ 1^T S &= 1 \\ S &\geq 0 \end{aligned}$$

The final inequality is an abbreviation for the requirement that all entries of S must be real and nonnegative (so any feasible S must be a doubly stochastic matrix). This sole relaxation permits the problem to be solved in polynomial time (by any efficient linear programming algorithm).

CHAPTER 4

Algorithms

Describe algorithms here.

4.1 Coarsest equitable partition

Describe coarsest equitable partition algorithm here.

4.2 Linear programming

Desrcibe linear programming algorithms here.

CHAPTER 5

API

f

fraciso, **??**