GrinPy Documentation

Release latest

David Amos, Randy Davila

May 30, 2019

CONTENTS

GrinPy is a NetworkX extension for calculating graph invariants. This extension imports all of NetworkX into the same interface as GrinPy for easy of use and provides the following extensions:

- extended functional interface for graph properties
- calculation of NP-hard invariants such as: independence number, domination number and zero forcing number
- calculation of several invariants that are known to be related to the NP-hard invariants, such as the residue, the annihilation number and the sub-domination number

Our goal is to provide the most comprehensive list of invariants. We will be continuing to add to this list as time goes on, and we invite others to join us by contributing their own implementations of algorithms for computing new or existing GrinPy invariants.

CHAPTER ONE

AUDIENCE

We envision GrinPy's primary audience to be professional mathematicians and students of mathematics. Computer scientists, electrical engineers, physicists, biologists, chemists and social scientists may also find GrinPy's extensions to the standard NetworkX package useful.

CHAPTER

TWO

HISTORY

Grinpy was originally created to aid the developers, David Amos and Randy Davila, in creating an ordered tree of graph databases for use in an experimental automated conjecturing program. It quickly became clear that a Python package for calculating graph invariants would be useful. GrinPy was created in November 2017 and is still in its infancy. We look forward to what the future brings!

CHAPTER THREE

FREE SOFTWARE

GrinPy is free software; you can redistribute it and/or modify it under the terms of the *3-clause BSD license*, the same license that NetworkX is released under. We greatly appreciate contributions. Please join us on Github.

CHAPTER

FOUR

DOCUMENTATION

4.1 Tutorial

This guide can help you start working with GrinPy. We assume basic knowledge of NetworkX. For more information on how to use NetworkX, see the NetworkX Documentation.

4.1.1 Calculating the Independence Number

For this example we will create a cycle of order 5.

```
>>> import grinpy as gp
>>> G = gp.cycle_graph(5)
```

In order to compute the independence number of the cycle, we simply call the `independence_number()` function on the graph:

```
>>> gp.independence_number(G)
2
```

It’s that simple!

Note: In this release (version latest), all methods are defined only for simple graphs. In future releases, we will expand to digraphs and multigraphs.

4.1.2 Get a Maximum Independent Set

If we are interested in finding a maximum independent set in the graph:

```
>>> gp.max_independent_set(G)
[0, 2]
```

4.1.3 Determine if a Given Set is Independent

We may check whether or not a given set is independent:

```
>>> gp.is_independent_set(G, [0, 1])
False
>>> gp.is_independent_set(G, [1, 3])
True
```

4.1.4 General Notes

The vast majority of NP-hard invariants will include three methods corresponding to the above examples. That is, for each invariant, there will be three methods:

- Calculate the invariant
- Get a set of nodes realizing the invariant
- Determine whether or not a given set of nodes meets some necessary condition for the invariant.

4.2 Reference

Release latest

Date May 30, 2019

4.2.1 Classes

Release latest

Date May 30, 2019

HavelHakimi

Overview

Methods

HavelHakimi.__init__
HavelHakimi.depth
HavelHakimi.get_elimination_sequence
HavelHakimi.get_initial_sequence
HavelHakimi.is_graphic
HavelHakimi.get_process
HavelHakimi.residue

4.2.2 Functions

Release latest

Date May 30, 2019

Degree

degree_sequence
min_degree
max_degree
average_degree
number_of_nodes_of_degree_k
number_of_degree_one_nodes
number_of_min_degree_nodes
number_of_max_degree_nodes
neighborhood_degree_list
closed_neighborhood_degree_list

Distance

distance

Graph Operations

contract_nodes

Neighborhoods

are_neighbors
closed_neighborhood
common_neighbors
neighborhood

4.2.3 Invariants

Release latest

Date May 30, 2019

Chromatic Number

chromatic_number

Clique Number

clique_number

Disparity

vertex_disparity
closed_vertex_disparity
disparity_sequence
closed_disparity_sequence
CW_disparity
closed_CW_disparity
inverse_disparity
closed_inverse_disparity
average_vertex_disparity
average_closed_vertex_disparity
k_disparity
closed_k_disparity
irregularity

Distance Measures

triameter

Domination

is_k_dominating_set
is_total_dominating_set
min_k_dominating_set
min_dominating_set
min_total_dominating_set
domination_number
k_domination_number
total_domination_number

DSI

sub_k_domination_number
slater
sub_total_domination_number
annihilation_number

Independence

is_independent_set
is_k_independent_set
max_k_independent_set
max_independent_set
independence_number
k_independence_number

Matching

`max_matching`
`matching_number`
`min_maximal_matching`
`min_maximal_matching_number`

Power Domination

`is_power_dominating_set`
`min_power_dominating_set`
`power_domination_number`

Residue

`residue`
`k_residue`

Topological Indices

`randic_index`
`generalized_randic_index`
`augmented_randic_index`
`harmonic_index`
`atom_bond_connectivity_index`
`sum_connectivity_index`
`first_zagreb_index`
`second_zagreb_index`

Zero Forcing

`is_k_forcing_vertex`
`is_k_forcing_active_set`
`is_k_forcing_set`
`min_k_forcing_set`
`k_forcing_number`
`is_zero_forcing_vertex`
`is_zero_forcing_active_set`
`is_zero_forcing_set`
`min_zero_forcing_set`
`zero_forcing_number`

4.3 License

GrinPy is distributed with the 3-clause BSD license. As an extension of the NetworkX package, we list the pertinent copyright information as requested by the NetworkX authors.

```
GrinPy
------
Copyright (C) 2017-2019, GrinPy Developers
David Amos <somacdivad@gmail.com>
Randy Davila <davilar@uhd.edu>

NetworkX
--------
Copyright (C) 2004-2019, NetworkX Developers
Aric Hagberg <hagberg@lanl.gov>
Dan Schult <dschult@colgate.edu>
Pieter Swart <swart@lanl.gov>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above
    copyright notice, this list of conditions and the following
    disclaimer in the documentation and/or other materials provided
    with the distribution.

  * Neither the name of the NetworkX Developers nor the names of its
    contributors may be used to endorse or promote products derived
    from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search