

# *Grph*: list of algorithms

Luc Hogue (CNRS), Aurélien Lancin (INRIA), Issam Tahiri (INRIA),  
Nathan Cohen (INRIA), David Coudert (INRIA)

April 5, 2012

## Contents

<b>1</b>	<b>Traversal</b>	<b>4</b>
1.1	Breadth First Search . . . . .	4
1.2	Depth First Search . . . . .	4
1.3	Random Search . . . . .	4
1.3.1	Single visit . . . . .	4
1.4	Parallelization . . . . .	4
<b>2</b>	<b>Shortest paths</b>	<b>4</b>
2.1	BFS-based . . . . .	5
2.2	Dijkstra . . . . .	5
2.3	Bellman-Ford . . . . .	5
2.3.1	David mod . . . . .	5
2.4	Floyd Warshall . . . . .	5
<b>3</b>	<b>Clustering</b>	<b>5</b>
3.1	Kernighan-Lin . . . . .	5
3.2	Number of triangles . . . . .	5
3.3	Chromatic number . . . . .	5
<b>4</b>	<b>Distances</b>	<b>6</b>
4.1	Distance matrix . . . . .	6
4.2	Eccentricity . . . . .	6
4.3	Radius . . . . .	6
4.4	Diameter . . . . .	6
4.5	Center vertex . . . . .	6
4.5.1	Minimum eccentricity . . . . .	6
4.5.2	Two sweeps . . . . .	6
4.5.3	Distance matrix-based diameter . . . . .	6
4.5.4	Two sweeps . . . . .	6
4.5.5	Four sweeps . . . . .	6
<b>5</b>	<b>Connected components</b>	<b>7</b>
5.1	Set of connected components . . . . .	7
5.2	Connected component containing a given vertex . . . . .	7
<b>6</b>	<b>Cuts</b>	<b>7</b>
6.1	Test if two vertex sets form a cut in the graph . . . . .	7
6.2	Test if a set of vertex sets form a cut in the graph . . . . .	7
6.3	Computes the set of edges of a given cut . . . . .	7
6.4	Computes the size of given cut . . . . .	7

<b>7</b>	<b>Navigation</b>	<b>7</b>
7.1	Adjacency matrix . . . . .	7
7.2	Incidence matrix . . . . .	7
7.3	Degrees . . . . .	7
7.4	Clustering coefficient . . . . .	8
7.5	Set of edges connecting two given vertices . . . . .	8
7.6	Set of edges incident to two given vertices . . . . .	8
7.7	Headless/tailless edges . . . . .	8
7.8	In/out neighbors . . . . .	8
7.9	Isolated vertices . . . . .	8
7.10	Loops . . . . .	8
7.11	Set of vertices at a given distance . . . . .	8
7.12	Set of vertices incident to a given edge . . . . .	8
7.13	Set of vertices with a given in/out degree . . . . .	8
7.14	Inclusion . . . . .	8
7.15	Equality . . . . .	8
7.16	Difference detection . . . . .	8
7.17	Cloning . . . . .	8
<b>8</b>	<b>Structural tests</b>	<b>9</b>
8.1	Connected . . . . .	9
8.2	Complete . . . . .	9
8.3	Null . . . . .	9
8.4	Reflexive . . . . .	9
8.5	Regular . . . . .	9
8.6	Trivial . . . . .	9
8.7	Simple . . . . .	9
8.8	Tree . . . . .	9
8.9	Chordal . . . . .	9
8.10	Hypergraph . . . . .	9
8.11	Multigraph . . . . .	10
8.12	Irreflexive . . . . .	10
8.13	Acyclic . . . . .	10
8.14	Hamiltonian . . . . .	10
	8.14.1 Ore's theorem . . . . .	10
	8.14.2 Dirac's theorem . . . . .	10
8.15	Anti-graph . . . . .	10
8.16	Bipartiteness . . . . .	10
<b>9</b>	<b>Topology generators</b>	<b>10</b>
9.1	GNM . . . . .	10
9.2	GNP . . . . .	10
9.3	Clique . . . . .	10
9.4	Bus . . . . .	11
9.5	Chain . . . . .	11
9.6	Star . . . . .	11
9.7	Grid . . . . .	11
9.8	Inet . . . . .	11
9.9	Wireless backhaul . . . . .	11
9.10	Ad hoc network . . . . .	11
9.11	Random Newman Watts Strogatz . . . . .	11
9.12	Random tree . . . . .	12
9.13	Ring . . . . .	12
9.14	Chordal . . . . .	12
9.15	Brite . . . . .	12
9.16	GLP . . . . .	12
9.17	R-mat . . . . .	12

<b>10 Input/Output</b>	<b>12</b>
10.1 DOT/GraphViz . . . . .	12
10.2 Dimacs . . . . .	12
10.3 LAD . . . . .	12
10.4 GraphML . . . . .	12
10.5 GML . . . . .	13
10.6 DGS (read-only) . . . . .	13
10.7 Grph binary . . . . .	13
10.8 Grph text . . . . .	13
<b>11 Unclassified</b>	<b>13</b>
11.1 Line graph . . . . .	13
11.2 Pruning . . . . .	13
11.3 Minimum vertex cover . . . . .	13
11.4 Maximum independent set . . . . .	13
11.5 Random-walk based centrality . . . . .	13
11.6 Graph isomorphism . . . . .	13
11.7 Subgraph isomorphism . . . . .	13
11.8 Maximum clique . . . . .	14
11.9 Topological sort . . . . .	14
11.10 Complement . . . . .	14
<b>12 Set operations</b>	<b>14</b>
12.1 Independant set . . . . .	14
<b>13 Value distribution</b>	<b>14</b>

# 1 Traversal

Graph traversal algorithms are designed in the form of “abstract traversers”, which means that at each step of the traversal, they invoke dedicated abstract method that the user need to implement in order to meet its needs.

## 1.1 Breadth First Search

Breadth-first search (BFS) is a graph search algorithm that begins at the root node and explores all the neighboring nodes. Then for each of those nearest nodes, it explores their unexplored neighbor nodes, and so on, until it finds the goal, if any.

## 1.2 Depth First Search

Depth-first search (DFS) starts at the root and explores as far as possible along each branch before backtracking.

## 1.3 Random Search

The random search starts from a given root node and select a random neighbor as the next step. It does not guarantee that the entire graph will be visited, even though it is connected. Nodes may be visited several times.

### 1.3.1 Single visit

An adaptation of the previously described scheme, but ensures that any single node will be visited only once.

## 1.4 Parallelization

When searches are requested for multiple source, they are computed in parallel. To do this *Grph* instantiates 4 times more threads than cores on the computer.

# 2 Shortest paths

The shortest path problem is the problem of finding a path between two vertices (or nodes) such that the sum of the weights of its constituent edges is minimized. An example is finding the quickest way to get from one location to another on a road map; in this case, the vertices represent locations and the edges represent segments of road and are weighted by the time needed to travel that segment.

## 2.1 BFS-based

## 2.2 Dijkstra

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959,[1][2] is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree.

## 2.3 Bellman-Ford

The BellmanFord algorithm computes single-source shortest paths in a weighted digraph. For graphs with only non-negative edge weights, the faster Dijkstra's algorithm also solves the problem. Thus, BellmanFord is used primarily for graphs with negative edge weights.

### 2.3.1 David mod

This modification uses two correlated stacks.

## 2.4 Floyd Warshall

# 3 Clustering

## 3.1 Kernighan-Lin

This algorithm runs in  $\Theta(n^2 \log(n))$ .

KernighanLin is an heuristic algorithm for solving the graph partitioning problem. The algorithm has important applications in the layout of digital circuits and components in VLSI.

## 3.2 Number of triangles

Implemented in C. Uses Matthieu Latapy (LIP6) source code.

## 3.3 Chromatic number

When used without any qualification, a coloring of a graph is almost always a proper vertex coloring, namely a labelling of the graphs vertices with colors such that no two vertices sharing the same edge have the same color. The smallest number of colors needed to color a graph G is called its chromatic number.

## 4 Distances

### 4.1 Distance matrix

The graph distance matrix, sometimes also called the all-pairs shortest path matrix, is the square matrix  $d_{i,j}$  consisting of all graph distances from vertex  $v_i$  to vertex  $v_j$ .

### 4.2 Eccentricity

The eccentricity  $\epsilon(v)$  of a graph vertex  $v$  in a connected graph  $G$  is the maximum graph distance between  $v$  and any other vertex  $u$  of  $G$ . For a disconnected graph, all vertices are defined to have infinite eccentricity.

### 4.3 Radius

The radius of a graph is the minimum graph eccentricity of any graph vertex in a graph.

### 4.4 Diameter

The length  $\max_{u,v} d(u,v)$  of the "longest shortest path" (i.e., the longest graph geodesic) between any two graph vertices  $(u,v)$  of a graph, where  $d(u,v)$  is a graph distance. In other words, a graph's diameter is the largest number of vertices which must be traversed in order to travel from one vertex to another when paths which backtrack, detour, or loop are excluded from consideration.

### 4.5 Center vertex

#### 4.5.1 Minimum eccentricity

#### 4.5.2 Two sweeps

Starts from any node. Performs one BFS and choose the farthest node as the new source. Performs another BFS and returns the node in the middle of the path between the source and the farthest node.

#### 4.5.3 Distance matrix-based diameter

This algorithm runs in  $\Theta(n^2)$ .  $n$  being the number of vertices in the graph.

#### 4.5.4 Two sweeps

Also called two-sweeps diameter approximation. Gives a lower bound.

This algorithm runs in  $\Theta(2n)$ .

#### 4.5.5 Four sweeps

Also called two-sweeps diameter approximation. Gives a lower bound.

This is a contribution of Laurent Viennot.

## 5 Connected components

A connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices.

### 5.1 Set of connected components

### 5.2 Connected component containing a given vertex

## 6 Cuts

A cut is a partition of the vertices of a graph into two disjoint subsets. The cut-set of the cut is the set of edges whose end points are in different subsets of the partition. Edges are said to be crossing the cut if they are in its cut-set.

### 6.1 Test if two vertex sets form a cut in the graph

### 6.2 Test if a set of vertex sets form a cut in the graph

### 6.3 Computes the set of edges of a given cut

### 6.4 Computes the size of given cut

## 7 Navigation

### 7.1 Adjacency matrix

The adjacency matrix, sometimes also called the connection matrix, of a simple graph is a matrix with rows and columns labelled by graph vertices, with a 1 or 0 in position  $(v_i, v_j)$  according to whether  $v_i$  and  $v_j$  are adjacent or not. For a simple graph with no self-loops, the adjacency matrix must have 0s on the diagonal. For an undirected graph, the adjacency matrix is symmetric.

### 7.2 Incidence matrix

The incidence matrix of a graph gives the  $(0,1)$ -matrix which has a row for each vertex and column for each edge, and  $(v,e)=1$  iff vertex  $v$  is incident upon edge  $e$  (Skiena 1990, p. 135). However, some authors define the incidence matrix to be the transpose of this, with a column for each vertex and a row for each edge.

### **7.3 Degrees**

Consider in/out edge/vertex degrees.

### **7.4 Clustering coefficient**

A clustering coefficient is a measure of degree to which nodes in a graph tend to cluster together. Evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit groups characterised by a relatively high density of ties.

### **7.5 Set of edges connecting two given vertices**

### **7.6 Set of edges incident to two given vertices**

### **7.7 Headless/tailless edges**

### **7.8 In/out neighbors**

### **7.9 Isolated vertices**

Returns the set of vertices that have no incident edge.

### **7.10 Loops**

Returns the of edges that are self-loops.

### **7.11 Set of vertices at a given distance**

Returns the set of vertices at the given distance.

### **7.12 Set of vertices incident to a given edge**

### **7.13 Set of vertices with a given in/out degree**

### **7.14 Inclusion**

Test if a given graph is included in another graph.

### **7.15 Equality**

Test if a given graph equals another graph.

### **7.16 Difference detection**

Reports the difference with two given graphs.

### 7.17 Cloning

Builds a clone of the given graph. Topology and properties are duplicated.

## 8 Structural tests

### 8.1 Connected

; A graph which is connected in the sense of a topological space, i.e., there is a path from any point to any other point in the graph. A graph that is not connected is said to be disconnected. This definition means that the null graph and singleton graph are considered connected, while empty graphs on  $n \geq 2$  nodes are disconnected.

### 8.2 Complete

A complete graph is a graph in which each pair of graph vertices is connected by an edge.

### 8.3 Null

The term "null graph" is used both to refer to any empty graph and to the empty graph on 0 nodes.

### 8.4 Reflexive

A reflexive graph is a pseudograph such that each vertex has an associated graph loop.

### 8.5 Regular

A graph is said to be regular of degree  $r$  if all local degrees are the same number  $r$ . A 0-regular graph is an empty graph, a 1-regular graph consists of disconnected edges, and a 2-regular graph consists of disconnected cycles. The first interesting case is therefore 3-regular graphs, which are called cubic graphs.

### 8.6 Trivial

### 8.7 Simple

A simple graph, also called a strict graph, is an unweighted, undirected graph containing no graph loops or multiple edges.

## 8.8 Tree

## 8.9 Chordal

## 8.10 Hypergraph

An hypergraph is a graph in which generalized edges (called hyperedges) may connect more than two nodes. A connection between two or more vertices of a hypergraph. A hyperedge connecting just two vertices is simply a usual graph edge.

## 8.11 Multigraph

; The term multigraph refers to a graph in which multiple edges between nodes are either permitted (Harary 1994, p. 10; Gross and Yellen 1999, p. 4) or required.

## 8.12 Irreflexive

## 8.13 Acyclic

## 8.14 Hamiltonian

### 8.14.1 Ore's theorem

### 8.14.2 Dirac's theorem

## 8.15 Anti-graph

An anti-graph is an orientation of a graph where every vertex is a source or a sink.

## 8.16 Bipartiteness

# 9 Topology generators

<http://sndlib.zib.de/home.action>

## 9.1 GNM

Erdos and Renyi GNM model: adds  $m$  edges to a graph of  $n$  vertices, connecting random pairs of vertices.

The maximum number of edges is  $x = n(n - 1)/2$ . If  $m > x$  then the generator starts by constructing a clique of all vertices from which it then randomly removes  $x - m$  edges.

## 9.2 GNP

Erdos and Renyi GNP model: connect each pair of vertices according to a given probability.

## 9.3 Clique

Connect the vertices in the graph as a clique. Vertices are connected to all others. Previously existing edges are ignored.

## 9.4 Bus

Connect the vertices in the graph as a bus: they are all included in a new undirected hyper-edge. Previously existing edges are ignored.

## 9.5 Chain

For instance, the chain topology model can be used to model in-line networks following railways, enabling train passengers to connect to the internet.

## 9.6 Star

## 9.7 Grid

Connect the vertices in the graph as a grid.

## 9.8 Inet

Inet, currently at version 3.0, is an Autonomous System (AS) level Internet topology generator.

## 9.9 Wireless backhaul

Degenerates a grid in the following manner: the farther is an edge from the initial node, the more probability it has to be deleted. An edge is never deleted if deleting it would result in disconnecting the graph.

## 9.10 Ad hoc network

A wireless ad hoc network is a decentralized wireless network.[1] The network is ad hoc because it does not rely on a preexisting infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, and so the determination of which nodes forward data is made dynamically based on the network connectivity.

## 9.11 Random Newman Watts Strogatz

From the SageMath documentation: First create a ring over  $n$  nodes. Then each node in the ring is connected with its  $k$  nearest neighbors. Then shortcuts are created by adding new edges as follows: for each edge  $u-v$  in the underlying ' $n$ -ring with  $k$  nearest neighbors'; with probability  $p$  add a new edge  $u-w$  with " + "randomly-chosen existing node  $w$ .

## 9.12 Random tree

## 9.13 Ring

Connect the vertices in the graph as a ring. Previously existing edges are ignored.

## 9.14 Chordal

## 9.15 Brite

Uses the external Brite topology generator. BRITE supports multiple generation models including models for flat AS, flat Router and hierarchical topologies.

## 9.16 GLP

## 9.17 R-mat

The R-mat topology generator is implemented in C.

# 10 Input/Output

## 10.1 DOT/GraphViz

DOT is a plain text graph description language. It is a simple way of describing graphs that both humans and computer programs can use. DOT graphs are typically files that end with the `.gv` (or `.dot`) extension. Various programs can process DOT files. Some, like `dot`, `neato`, `twopi`, `circo`, `fdp`, and `sfdp`, will read a DOT file and render it in graphical form. Others, like `gvpr`, `gc`, `acyclic`, `cmaps`, `sccmap`, and `tred`, will read a DOT file and perform calculations on the represented graph. Finally, others, like `GVedit`, `KGraphEditor`, `lefty`, `dotty`, and `grappa`, provide an interactive interface.

## 10.2 Dimacs

## 10.3 LAD

## 10.4 GraphML

GraphML is an XML-based file format for graphs. The GraphML file format results from the joint effort of the graph drawing community to define a common format for exchanging graph structure data. It uses an XML-based syntax and supports the entire range of possible graph structure constellations including directed, undirected, mixed graphs, hypergraphs, and application-specific attributes.

## 10.5 GML

Graph Modelling Language (GML) is a hierarchical ASCII-based file format for describing graphs. It has been also named Graph Meta Language.

## 10.6 DGS (read-only)

DGS is the native format used by the GraphStream graph rendering tool. DGS is suited to the description of dynamic graphs.

## 10.7 Grph binary

This format is native to the *Grph* project. It encodes graphs in a compact sequence of bytes. It is not human-readable.

## 10.8 Grph text

This format is native to the *Grph* project. It encodes graphs in a compact sequence of ASCII characters. It is meant to be compact and easily editable by humans.

# 11 Unclassified

## 11.1 Line graph

## 11.2 Pruning

## 11.3 Minimum vertex cover

## 11.4 Maximum independent set

## 11.5 Random-walk based centrality

Based on a random walk.

## 11.6 Graph isomorphism

Integration of Brendan McKay C code (Nauty).

## 11.7 Subgraph isomorphism

LAD is a program for solving subgraph isomorphism problem. It is written by Christine SOLNON, Professor at LIRIS, INSA (Lyon, France).

<http://liris.cnrs.fr/csolnon/LAD.html>

## 11.8 Maximum clique

The computation of the maximum clique is delegated to program AntClique. AntClique is a program for solving maximum clique problems. It is written by Christine SOLNON, Professor at LIRIS, INSA (Lyon, France).

<http://liris.cnrs.fr/csolnon/AntClique.html>

On request of the maximum clique, Grph runs AntClique in a new process and interact with it using standard input/output streams. If the executable file of AntClique is not found (which is very likely at the first call), it will compile the source code on-the-fly.

## 11.9 Topological sort

## 11.10 Complement

The complement (or inverse) of a graph  $G$  is a graph  $H$  on the same vertices such that two vertices of  $H$  are adjacent if and only if they are not adjacent in  $G$ .

## 12 Set operations

Inclusion, suppression, addition, difference, union.

## 12.1 Independant set

## 13 Value distribution

The result of methods returning set of values (one value per vertex) can be further computed so as to generate distribution of these values.

## References