NAME

edgepaint - edge coloring to disambiguate crossing edges

SYNOPSIS

[*options*] [**-o** *outfile*] [*files*]

DESCRIPTION

edgepaint takes as input a graph in DOT format with node position information (the *pos* attribute) and colors the edges in a manner making it easier to tell them apart.

OPTIONS

The following options are supported:

--accuracy=*e*
Accuracy with which to find the maximally different coloring for each node with regard to its neighbors. Default *e* = 0.01.

--angle=*a*
Color two edges differently if their incidence angle is less than *a* degrees. Default *a* = 15.

--random_seed=*s*
Random seed to use. *s* must be an integer. If *s* is negative, we do |*s*| iterations with different seeds and pick the best.

--lightness=*l1*,*l2*
Only applies for the "lab" color scheme: *l1* and *l2* must integers, with 0 <= *l1* <= *l2* <=100. By default, we use "0,70".

--share_endpoint
If this option is specified, edges that share a node are not considered in conflict if they are close to parallel but are on the opposite sides of the node (around 180 degree).

-o *f* Write output to file *f* (default: stdout).

--color_scheme=*c*
Specifies the color scheme. This can be "rgb", "gray", "lab" (default); or a comma-separated list of RGB colors in hex (e.g., "#ff0000,#aabbed,#eeffaa") representing a palette; or a string specifying a Brewer color scheme (e.g., "accent7"; see https://graphviz.org/doc/info/colors.html#brewer).

-v Turns on verbose mode.

-? Print usage and exit.

BUGS

At present, **edgepaint** does not handle graphs with loops or directed multiedges. So, a graph with edges *a* -> *b* and *b* -> *a* is acceptable, but not if it has edges *a* -> *b* and *a* -> *b* or *a* -- *b* and *a* -- *b*. Ports are ignored in this analysis, so having *a.x* -> *b* and *a.y* -> *b* is also not supported.

AUTHOR

Yifan Hu <yifanhu@yahoo.com>

SEE ALSO

gvmap(1), sfdp(1), neato (1), dot(1)