DIMACS Technical Report 94-36
June 1994

A Comprehensive Manual for NETPAD Users with Exercises on Discrete Mathematics

by

Yanxi Liu[1]

[1]Supported in part by DIMACS

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

This document is intended to serve as a comprhensive reference for NETPAD starters, especially for those whose only interest is to use NETPAD as a tool instead of customizing NETPAD as a programmer user. Several examples and exercises on discrete math using NETPAD are also included at the end of this manual.

1 Getting Started

Get to a x-window shell (assume that the NETPAD system is installed on your computer system). At the command line type **netpad** then hit the return key. After several seconds, you should see two display windows popping up: the bigger one is for displaying a graph; the smaller one is for showing the numerical result of an algorithm applied to a graph.

Once you get NETPAD started you have two basic ways to make things happen. One is to use the mouse to create and modify a graph (mouse-driven), the other is to use the NETPAD menu (menu-driven). You can also mix the two methods to generate the graph you desire.

In the bigger window of NETPAD you should see a menu bar with boxes containing the following keywords on them:

Win, I/O, Make, Graph, Node, Link, ... etc.

This is the list of the top-level choices you can make using the mouse. If you click on a box with *any* button of the mouse, you should see a menu of the next level of choices. If you click on the same box again this table will disappear. If you click on a different box this menu will disappear as well, but another menu will pop up.

To run algorithms, you first set up a graph on the screen, and then click on the "Algo" box. Select an algorithm by moving the mouse to the appropriate place and then click. Move to the **EXEC** box and click to execute the algorithm. For more information on the function of each algorithm, see Section 5.8. For more information on the content of the menu bar, see Section 5.

2 Using the Mouse to Create a Graph

There are three buttons on your mouse, let us call them **L**eft, **M**iddle, and **R**ight.

- The right button is used for **creating** or **deleting** things: you can make vertices by clicking on the right button, or make edges by holding down the right button and moving from one vertex to another, then letting it go. To get rid of a vertex you move the arrow onto the vertex then click the right button while holding down the shift key. You do the same to get rid of an edge except hold down the control key **Ctrl**. To get rid of a part or the whole graph, you can hold down the shift key and the right button while you move the arrow from upper left to the lower right to make a rectangle, everything contained in the box will be deleted.

- The left button is used for **selecting/ unselecting** things. In order to select a vertex or an edge, you simply move the arrow to the item and click on the left button. You can also select a part or the whole graph by holding down the left button and moving from upper left to the lower right to make a rectangle, everything *completely* contained in the box is selected/unselected.

- The middle button is used for moving things in a graph. You can move the arrow over to a vertex and relocate the vertex while holding down the middle button, when you get to the desired position release the middle button.

3 Using NETPAD to Explore — a few suggested problems

3.1 Constructing Graphs

Recall that a graph G is said to have degree sequence (a-m,b-n, c-o,d-p,...) if it has m vertices of degree a, n vertices of degree b, o vertices of degree c, p vertices of degree d, etc. Try to construct on the NETPAD window connected graphs with the following degree sequences:

1) (4-1,3-4,2-2)

2) (5-4,3-2,2-1)

3) (4-1,3-12)

4) (5-3,3-7)

5) (5-4,4-1,3-8)

6) (5-5,4-3,3-6)

3.2 Isomorphic Graphs

Draw the first of each pair of graphs on the screen. Then try to create an isomorphism between the two graphs in each example (Figure 1, Figure 2, Figure 3), i.e. move the first graph around until you get the second graph.
Question: what rules on the forms of a graph have you discovered?

3.3 Additional Exercises using NETPAD

Please see the appendix of this manual for some more interesting exercises which were put together by our teachers and Nate Dean from Bellcore in a previous session with NETPAD.

4 Samples of Using NETPAD

4.1 Getting out of NETPAD

To get out of NETPAD, Click on **Win**, then on **Kill Netpad** of the table popped up, and finally on the small box just below **Win** with **exec** in it. All the NETPAD windows should disappear right in front of your eyes.

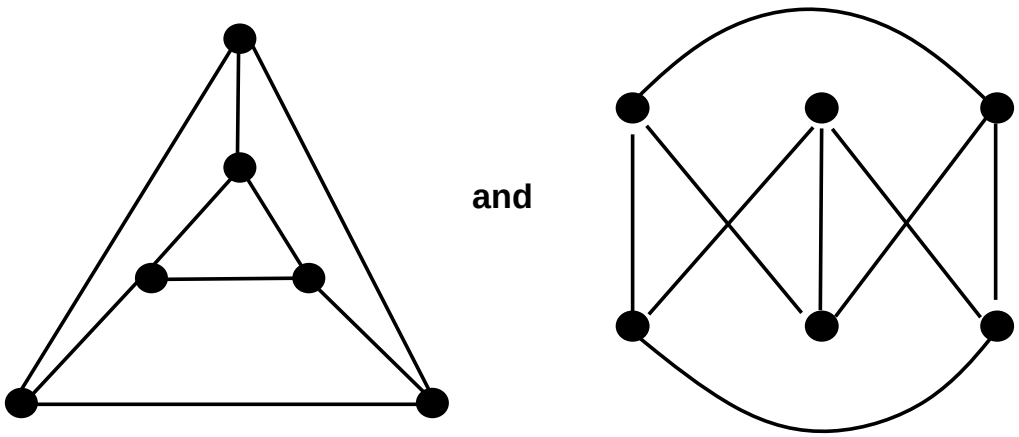

Figure 1: Example one

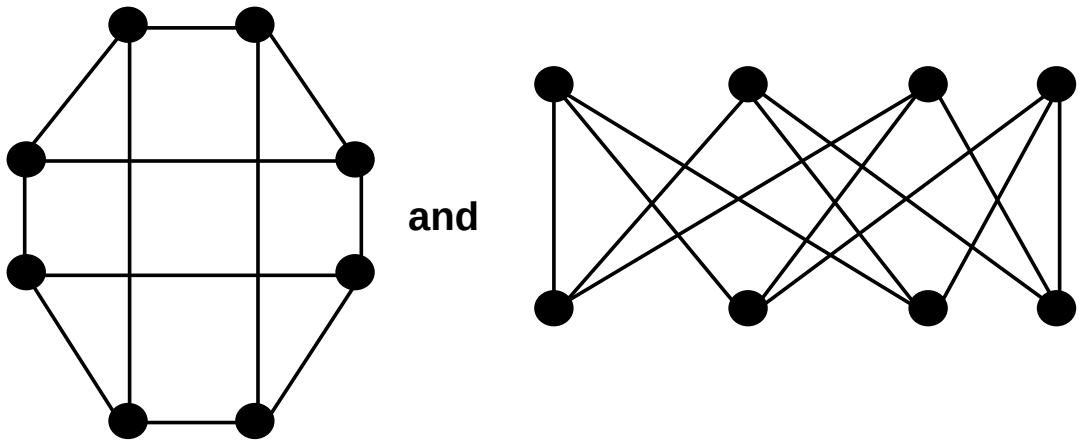

Figure 2: Example two

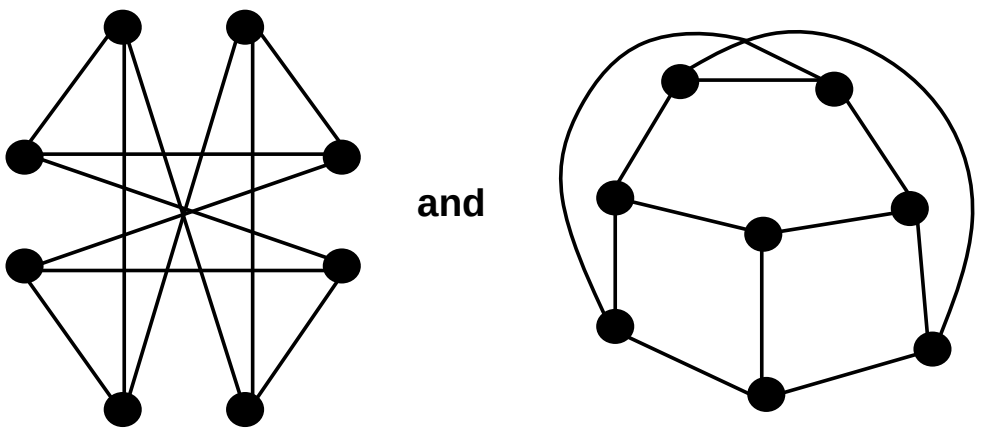

Figure 3: Example three

4.2 Saving the graph you created

Suppose you have created a graph and you want to keep a copy: click **I/O** on the menu bar, then click **Save**. Now you should see a box with **directory** and **filename** in it. You can fill in the directory and the filename you choose and then click **exec** box, or simply click **exec** in which case you will find a file called default.grf in your current direcotry (the directory where you opened NETPAD). This file contains the graph on the NETPAD window and can be loaded by using **Load** command in **I/O**.

4.3 Applying an algorithm to a graph

Suppose you use NETPAD to create a cycle: click on **Make** then on **Cycle**, if you click **exec** now you should see a graph of eight nodes (eight is a default value, you can change it by typing in the NodeCount box) forming a cycle. Now click on **Algo** box on the menu bar, and let us see whether the graph is bipartite by clicking on **bipartite ?** and then on the **exec** box. You should see the answer on the line directly above the menu bar. Try a different node number for the cycle.

5 Making a Choice with the Menu

There are many existing algorithms in NETPAD for creating graphs and computing certain properties of a graph. These algorithms can be called through the menu bar. Now let us examine each of these boxes on the menu bar one by one. Note, it is not necessary to read the following in a sequential order.

5.1 Win

Grid Settings A graph window may or may not display a grid of points in its backgrounds. Depending on the "Show Grid" and "Grid Type" parameters, NETPAD shows no background in a graph window, or a background of dots, horizontal or vertical lines, or a grid of horizontal and vertical lines. The horizontal distance between the dots or lines is given by the "Grid Width" parameter in normalized coordinates, that is, as a value between 0.0 and 1.0. The vertical distance is given by the "Grid Height" parameter.
Refresh redraws the graph.
Open opens a new window containing an empty graph.
Duplicate opens a new window containing a copy of the graph displayed in the current window.
Kill Window closes the current window.
Kill Netpad closes all windows and exits.

5.2 I/O

Load reads the graph described in the given file from the given directory and replaces the window's graph by the graph read. The file may be one that was procuded by "graf.save" or an animation file produced bby an external algorithm.
Add reads (like **Load**) the graph described in the given file from the given directory. The whole graph is added if "all?" parameter is true. Otherwise the selected subgraph is added, that is all selected nodes and thosed selected links that connect two selected nodes.

The ids of added nodes and links are changed, so that they do not overwrite an attribute defined in the wondow's graph which is added.
Save opens the given file in the given directory and saves the graph to it. The format used is rather berbose and human readable.
Print saves the graph in Postscript format to the given file in the given directory. "Width" and "Height" given the size of the Postscript picture. "Rotate" indicates whether the picture is rotated by 90 degrees. If "Frame" is true, the graph picture is surrounded by a frame. If the "Header" ("Trailer") parameter is not empty it is printed above (below) the graph's postscript picture.
Cut saves the graph to a global buffer file.
Paste is equivalent to **Add** called with a "directory" and "filename" parameter which together give the name of the global buffer file.

5.3 Make

Complete removes all nodes from the input graph and creates N nodes and positions them on a circle. Each pair of nodes is connected by a link.
Cycle removes all nodes from the input graph and creates N nodes and positions them on a circle. Nodes that are neighbors on the circle are connected by a link.
Wheel removes all nodes from the input graph and creates NumSpokes nodes and positions them on a circle. Nodes that are consecutive on the circle are connected by an edge. One more node is created in the center and joined to all the rest. It creates a wheel with N spokes; i.e. N+1 nodes.
Grid constructs a grid of (width x height) nodes.
Bipartite creates a complete bipartite graph with nodes on straight verticle lines.
N-Bipartite constructs a n-partite graph.
Rec. Triangles creates nested triangles.
Rand graph constructs random graph with given node count and edge probability. The nodes are positioned on the vertices of a regular polygon.
Rand n,k graph constructs random graph with given node and edge count. The nodes are positioned on the vertices of a regular polygon.
Random planar graph
2-server net constructs a digrid corresponding to 2-server queue.

5.4 Graph

Control Panel When executing **Control Panel** all it's parameter values are saved as the corresponding default settings of the graph. These defaults are used by NETPAD to determine which color, shape, style, and type a new node or link assumes.
Delete Sel deletes all selected nodes and all selected links from the graph. The deletion of a node includes the deletion of all links adjacent to that node.
Delete deletes all nodes and links from the graph. If the "with attr ?" parameter is true, it deletes all attribute definitions as well.
Color sets depending on the "all ?" parameter the color of all or all selected nodes and links in the graph to the "color" parameter value.
Sel Change toggles the selection of all nodes and links in the graph. **Select** selects and **UnSelect Unselect** unselects all nodes and links in the current graph.
X Select Color selects, unselects or toggles (depending on the "mode" parameter) the selection of all nodes and links with the given color.
Complement first adds a link between each pair of nodes that is not yet connected by a link, then it deletes all links that were contained in the original graph.

Added links lead from the node with the higher id to the one with the lower id.

Sel complement applies to the selected subgraph only. It first adds a link between each pair of unconnected selected nodes and then removes all links that connected a pair of selected nodes in the original graph.
Join connects each selected node with all unselected nodes by a new link. The new links lead from the selected nodes to the unselected ones.
Complete adds a link between each pair of nodes that is not yet connected by a link.

Added links lead from the node with the higher id to the one with the lower id.

Sel complete applies to the selected subgraph only; that is, it adds a link between each pair of selected nodes that is not yet connected.
Sel isolate deletes all links that connect a selected node with an unselected node.
Line Graph replaces the given graph by it's line graph, that is each link is replaced by a new node. The new nodes are connected iff the old links were adjacen
k-Power finds the k-power graph of graph in window A k-power graph is a graph with the edges connecting all vertices with a distance ¡= k.
Edit Attr has a variable number of parameters, that is it has a parameter for each currently defined graph attribute. The parameter's name and type is the same as the attribute's name and type. It also sets the value of each attribute to the value given as parameter value.

5.5 Node

Label Control has a variable number of parameters, that is besides the "node id ?" parameter it has a parameter for each currently defined node attribute in the graph. Each parameter has two values: a flag value and a choice value. Whenever the flag value is set to true NETPAD displays depending on the choice value a label to the north, east south, or west of each node.

In the case of the "node id ?" parameter NETPAD sets the label to the node's id. Otherwise, NETPAD sets the label to the value the node carries for the given attribute, or if the node does not carry a value to the empty string which is invisible to the user.

If there are several labels to be displayed in the same direction, NETPAD stacks them up in the same order as they appear in the parameter list.

Delete Sel deletes all selected nodes.

Sel Change toggles the selection of all nodes in the graph.

Select selects all node.

UnSelect unselects all node.

X Select Shape , depending on the "mode" parameter selects, unselects or toggles the selection of all nodes with the given shape.

X Select Color , depending on the "mode" parameter, selects, unselects or toggles the selection of all nodes and links with the given color.

Color sets the color of all or all selected nodes.

Shape sets the shape of all or all selected nodes to the given "shape" parameter value. A node's shape is the symbol that NETPAD uses to display it in the graph window.

Renumber gives new ids to all nodes, so that each node gets a unique id and all ids from 1 to m are assigned to a node, where m is the number of nodes in the graph.

Identify Sel replaces all selected nodes by a single selected node. All links that connected a selected node with an unselected node are changed to connect to the new selected node.

Edit Attr Val has a parameter for each currently defined node attribute. It sets the value refering to the different attributes of all selected nodes to the values given as parameter values.

Edit Label Attr Val has a parameter for each node attribute whose values are currently displayed as node labels. The parameter's name and type is the same as the attribute's name and type. It applies to all selected nodes. It sets the value of each attribute corresponding to a parameter to the value given as parameter value.

Set Attr Val sets depending on the "all ?" and "scope" parameters (with the "scope" parameter set to node), the value referring to the node or link attribute definition with the given name of all (selected) nodes or links in the graph. If the "scope" parameter is set to "graph", attr.set sets the graph's attribute value referring to the graph attribute with the given name. The "all ?" parameter value is in this case irrelevant.

The attribute values are set to the value defined by the "value" parameter string, which must be convertible to the type of the node, link or graph attribute definition.

If the value referring to an attribute definition is set for a node, link or graph we say, that it carries an explicit value for that attribute.

Unset Attr Val It unsets, that is removes attribute values referring to the given node, link, or graph attribute from all (selected) nodes or links of the graph or from the graph itself.

Now all nodes, links, or the graph itself, who's attribute values were unset, do no more carry an explicit value for the given attribute. External algorithms are free to use the attribute definition's default value as an attribute value for such a node, link or graph or to consider the attribute value as undefined. The NETPAD interface does the latter. When

labeling nodes and links with attribute values, it shows only those values as labels, that are explicitly set. Nodes and links without explicitly set values are displayed without labels.
Unset All Attr Val removes all attribute values from the graph itself. Otherwise it removes depending on the "scope" (with the "scope" parameter set to node) and "all ?" parameters all attribute values from all or all selected nodes or links.
Make Rand nodes generates nodes at random positions.
Select Rand nodes randomly selects nodes.

5.6 Link

Label Control has a "link id ?" parameter and a parameter for each currently defined link attribute. If the "link id ?" parameter is true NETPAD labels all links with their ids. For each link-attr-name parameter that is set to true NETPAD displays a label on each link that shows the value the link carries for the given attribute or an empty string. If there are several labels to be displayed, NETPAD stacks them up in the same order as they appear in the parameter list.
Delete Sel deletes all selected links from the graph.
Delete deletes all links from the graph.
Sel Change toggles the selection of all links in the graph.
Select selects all links.
UnSelect unselects all links.
X Select Style , depending on the "mode" parameter, selects, unselects or toggles the selection of all solid, dotted, dashed, dot_dashed, or dot_dot_dashed links.
X Select Color , depending on the "mode" parameter, selects, unselects or toggles the selection of all nodes and links with the given color. It applies only to nodes only.
X Select Type , depending on the "mode" parameter, selects, unselects or toggles the selection of all directed or all undirected links.
Color sets the color of all or all selected links.
Style sets the style of all or all selected links to the given "style" parameter value. A link's style corresponds to the line style which NETPAD uses to draw a link.
Type makes, depending on the "type" parameter, all or all selected links directed or undirected.
Reverse reverses the direction of all directed or all selected and directed links.
Bisect , for each link in the graph, first adds a new node, which it positions onto the midpoint of the link, then removes the link and adds two new links. If the removed link lead from node n to node m, the new links lead from n to the new node and from the new node to m. All new links have the same attribute values, color, style, type and selection state as the links they replace.
Bisect Sel , similar to **Bisect**, applies to selected links only.
Renumber gives new ids to all links, so that each link gets a unique id and all ids from 1 to m are assigned to a link, where m is the number of links in the graph.

Edit Attr Val has a parameter for each currently defined link attribute. It sets the attribute values of all selected links.
Edit Label Attr Val does the same as **Edit Label Attr Val** in **Node** for links and their attributes.
Set Attr Val sets depending on the "all ?" and "scope" parameters (with the "scope" parameter set to link), the value referring to the node or link attribute definition with the given name of all (selected) nodes or links in the graph.
Unset Attr Val unsets, that is removes attribute values referring to the given node, link, or graph attribute from all (selected) nodes or links of the graph or from the graph itself.
Unset All Attr Val removes all attribute values from the graph itself. Otherwise it removes depending on the "scope" and "all ?" parameters all attribute values from all or all selected links.
Make Rand links randomly adds links between pairs of unconnected nodes.
Select Rand links randomly selects links.

5.7 Attr

Def Attr defines, depending on the scope parameter, a new node, link or graph attribute for the graph. The attribute's name, type and default value are set to the respective parameters. It fails, if the graph already contains a node, link or graph attribute with the given name or if the "def value" string parameter can not be converted to the attribute's type.
Redef Attr changes depending on the "scope" parameter the definition of the node, link, or graph attribute with the name given by the "old name" parameter. If the "new type" parameter is not equal to "AS OLD" and all values referring to the given attribute can be converted to the new type, they are converted and the attribute's type is changed to the new type. If the "new name" parameter value is not empty, the attribute's name is changed to the new name, unless the graph contains another node, link, or graph attribute with that name. The attribute's default value is set to the new default value.

Whenever it fails to convert the new default value or some of the values referring to the given attribute to the attribute's (new) type, or when it cannot rename the attribute with the new name, or the graph contains no attribute with the given old name, it does not change any of the graph's attributes and returns with an appropriate error message.
Undef Attr deletes depending on the "scope" parameter the node, link, or graph attribute definition with the given name from the graph. This includes the deletion of all attribute values that refer to the deleted attribute definition.
Undef All Attr undefines depending on the "scope" parameter all node, link, or graph attribute definitions from the graph. This includes the deletion of all attribute values referring to deleted attribute definitions.
Set Attr Val sets depending on the "all ?" and "scope" parameters, the value referring to the node or link attribute definition with the given name of all (selected) nodes or links in the graph.

If the "scope" parameter is set to "graph", attr.set sets the graph's attribute value referring to the graph attribute with the given name. The "all ?" parameter value is in this case irrelevant.

The attribute values are set to the value defined by the "value" parameter string, which must be convertible to the type of the node, link or graph attribute definition.

If the value referring to an attribute definition is set for a node, link or graph we say, that it carries an explicit value for that attribute.

Unset Attr Val It unsets, that is removes attribute values referring to the given node, link, or graph attribute from all (selected) nodes or links of the graph or from the graph itself.

Now all nodes, links, or the graph itself, who's attribute values were unset, do no more carry an explicit value for the given attribute. External algorithms are free to use the attribute definition's default value as an attribute value for such a node, link or graph or to consider the attribute value as undefined. The NETPAD interface does the latter. When labeling nodes and links with attribute values, it shows only those values as labels, that are explicitly set. Nodes and links without explicitly set values are displayed without labels.

Unset All Attr Val removes all attribute values from the graph itself. Otherwise it removes depending on the "scope" and "all ?" parameters all attribute values from all or all selected nodes or links.

X Select Attr Val , depending on the "mode" and "scope" parameters, selects, unselects or toggles the selection of all nodes or links, that carry the given "value" for the given attribute "name".

X Select Attr No Val , depending on the "mode" and "scope" parameters, selects, unselects or toggles the selection of all nodes or links, that carry no value for the given attribute "name".

List Attr Def lists the definitions of all node, link or graph attributes.

List Attr Val lists the attribute values carried by all or all selected nodes, links or the graph itself. The "all ?" parameter value is irrelevant, when listing the graph's attribute value.

List Named Attr Val behaves like **List Attr Val** except that it restricts its list to attribute values refering to the node, link, or graph attribute given by the "name" parameter.

5.8 Algo

connected ? finds whether graph is connected.

 # components ? counts number of components.

mark component selects all links in component of the selected node.

component list prints the edge list for each component of the graph. The node ids of the input graph should be consecutive; otherwise the program may fail.

bicomponent list prints the edge list for each bicomponent of the graph. A bicomponent is a maximal nonseparable subgraph containing at least one edge.

bipartite ? tests whether graph is bipartite. If graph is bipartite, one partition is selected and the other unselected. Otherwise the graph is left unchanged.

claw-free ? checks if a given graph is claw-free.

bridges finds all bridges of the given graph and selects them.
cutedges finds all cut edges.
select minmaxdeg finds a node with the minimum (maximum) degree and selects them.
shortest a,b-path finds path between two selected nodes.
neighbor set , given set S of selected vertices, select S and neighbors of S.
mst prim seraches minimum spanning tree; selects the tree found, if graph is connected. MST is a tree that goes to every node once and only traverses and enge once. It also ahs the property that the sum of the edges that make up this traversal is the smallest of the possible spanning trees. We use unit weights for all the links in the graph.
all spanning trees produces animation file, which enumerates all the spanning trees in a graph.
count spanning trees counts the total number of spanning trees at selected nodes.
Rwalk spanning trees simulates and analyses random traversals of large state spaces.
select match finds and selects a maximum cardinality matching. Matched edges are unselected, unmatched edges are selected.
select match (anim) unit two-matching
random walk does a random walk in the graph.
token walk n token walk randomly and collide.
Hamilton cycle/path produces animation file, which enumerates all hamiltonian cycles or path.
Max Independent Set selects a maximum stable set of nodes
MIS-critical links selects the alpha increasing edges.
Set X,Y, Length Attr
Nearest Neighbor put in edges to the nearest neighbours.
Held Karp
Read X,Y Data reads node coor file and makes sgma nodes with x,y attributes.

5.9 Geom

On Grid repositions nodes on the closest point on a grid. All routines take care that no nodes will be moved out of the unit square.
Scale scales node positions and takes care that no nodes will be moved out of the unit square.
Move moves the positions of all or all selected nodes by the vector defined by the "delta x" and "delta y" parameters.
Rotate rotates the positions of all or all selected nodes around the point given by the "center x" and "center y" parameters by the angle given by the "alpha" parameter. The "center x" and "center y" parameter values must lie in the interval $[0.0, 1.0]$. The angle is given in degree.
On cycle pepositions nodes on a cycle, starting with (selected) node of highest id.
Dilate dilates the graph's drawing by a given factor up to the limit of the bounds of the window.

Tutte embedding (iterative) draws the Tutte embedding of the graph, using an iterative method.
Tutte embedding (matrix) draws the Tutte embedding of the graph, by solving the linear system.
Schnyder's Fary embedding draws a Fary embedding of the graph onto a n by n grid, using Schnyder's algorithm.
Is graph planar ? tests to see if the graph is planar (can be drawn in the plane without lines crossing).
Triangulate graph adds edges to a planar graph to make every face a triangle.
Radial Tree Draw draws spanning tree in circles of increasing radius around root
Randomize randomly distributes the nodes of the graph.
Num Crossings counts the number of crossings in drawing selected edges.
Delaunay removes all links, finds delaunay traingulation and adds links to show it.
Delaunay MST finds mst on delaunay triangulation
Kruskal MST (Kruskal's Minimum Spanning Tree) uses euclidian norm, produces an animation file, if desired
Approx. TSP (Travelling Salesman Tour) finds tsp tour from the mst tree of the delaunay triangulation.
Euclid distances prints matrix of Euclidean distances (i.e., every edge length).

5.10 Nets

strongly connected ? finds whether graph is strongly connected.
dicomponent list prints the vertex set for each dicmponent of the digraph.
tree of paths selects all links in a tree of paths rooted at the selected node. It finds the tree of shortest paths rooted at the single selected node. The length of a path is defined as it's number of links.
shortest a-cycle finds cycle around a selected node.
shortest a,b-path, L2/max finds path between two selected nodes.
shortest a,b-path, weighted finds path between two selected nodes.
shortest a-cycle, L2/max
set dist attr, L2/max
prt dist matrix, L2/max
f-matching finds and selects an f matching. Output vector: multiplicity.
f-factor finds and selects an f-factor.
select unit flow finds the maxflow for undirected network with unit capacities between the two selected nodes.

Two nodes must be selected to indicate source and sink. The Edges used to carry unit flow are selected, edges that carry no flow are unselected.
directed local edge conn

5.11 Anim

Init opens the given file in the given directory as the animation file of the current window.
Run reads, interprets, and executes actions from the animation file until it arrives at the end of the file or until the user selects a menu or types a key on the graph window.
delay
Next reads, interprets, and executes actions from the current animation file until it encounters a NEXT action with a level at least as high as the given "level" parameter, the end of the file or the user selects a menu or types a key on the graph window.
Step reads, interprets, and executes the next action from the current animation file.
StepStep reads, interprets, and executes actions from the current animation file until it encounters a NEXT action, the end of the file or the user selects a menu or types a key on the graph window.
Messages sets a flag in the current window, whether the NETPAD interface should print the string value of MESSAGE actions.
Trace sets the animation trace flag of the current window to the given value. Whenever an action is read by **anim.step**, **anim.step.step**, **anim.next**, or **anim.run** and the trace flag is true, the action is printed in a human readable format.
Stop closes the current animation file. You may call it to stop the NETPAD.

6 Appendix

6.1 Three exercises on discrete math using NETPAD

These exercises have been put together by Dr. Nate Dean from Bellcore and the teachers in the Leadership Program in Discrete Mathematics of DIMACS organized by Profesor Joseph G. Rosenstein [2] of Rutgers University, in a previous session with NETPAD. The teachers are: Carol Avelsgaard, Pat Cline, Susan Simon and Don Cook.

6.1.1 Edge Crossings

6.1.2 Traveling Salesman Problem

6.1.3 Triangulation

6.2 A list of questions for NETPAD Users

- Is this simple manual informative enough?
- Is there any unnecessary information in the manual for doing explorations with the NETPAD?
- Did you find something useful missing from the manual?
- Do you feel that it is better to use NETPAD than pencil and paper to do those simple graph problems? why and why not?
- How do you like the minimizing crossing number exercise?
- Are these problems and exercises using NETPAD interesting?
- What kind of problems/exercises would you like to see?
- What kind of problems/exercises would you like to develop?
- What kind of computers (hardware) you are using now? in the near future? five years from now?
- What new functions and features would you like to see in NETPAD or other discrete math software package?
- Besides graphs, what other math objects would you be interested in learning, such as sets, sequences, etc?
- What can be done to make a discrete math software more fun? more useful and easier to use?
- What do you wish discrete math software could do?
- NETPAD is not a tutorial program but have you learned anything from playing with it?
- Do you think a non-tutorial math software is useful? Why?
- Is exploring math via computer software *helpful* for you? *Necessary*? *Useless*?
- What do you wish your computers could do?
- What would have you learned from playing with NETPAD as you just did?
- Do you think NETPAD is friendly enough? what to be desired?

References

[1] Monika Mevenkamp, Nathaniel Dean, Clyde Monma. *NETPAD User's Guide*. October 18, 1990.

[2] Joseph G. Rosenstein. *An introduction to SIGMA*. October 8, 1990.