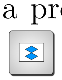
Getting started with AutoGraphiX-III (version 3.1.X)

Gilles Caporossi
GERAD & HEC Montréal
3000, chemin de la côte-ste-Catherine
H3T 2A7 Montreal, Canada

Abstract

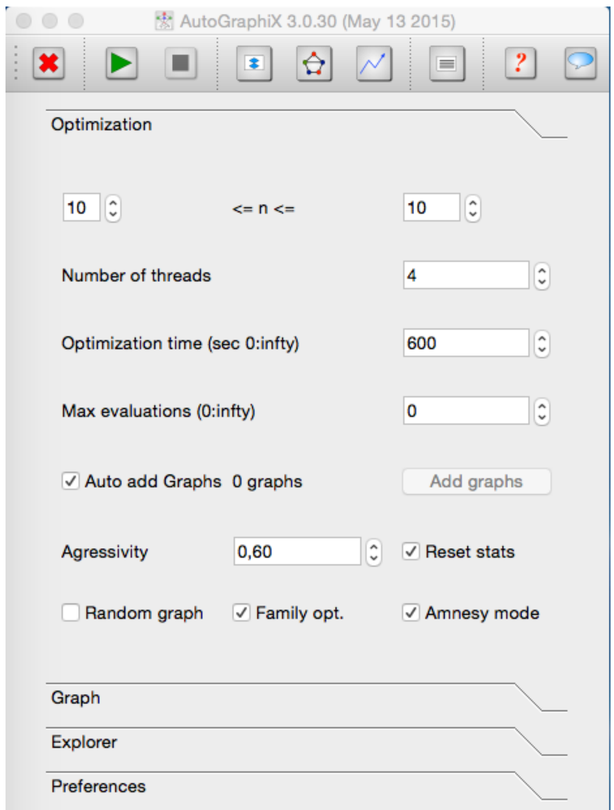
AutoGraphiX-III is a computer aided graph theory system. This document proposes a set of progressive examples using AGX with step by step explanations. It is writen for users of the version 3.1 of the software.

1 First steps

When launching AGX-III for the first time, a window like that on figure 1 should appear. The first step to use AGX is to define a problem to study. For this, one needs to open the *White Board* by clicking the  icon from the top bar of the window. The *Widte Board* is the window from which the problem under study is defined. The definition of a problem is achieved by items that may be combined as functions. There are 4 main types of items :

- The invariants (this is a slight abuse of language as the items considered as invariants are not necessarily invariants in the technical sense) are computed directly from the current graph. There are 4 main kinds of invariants :
  - Values, which are real invariants, functions that associate to each graph a numerical value regardless the labeling of the vertices (for example, the number of vertices ORDER, or the number of edges SIZE).
  - The vectors which provide a numerical value associated to each vertex (for example the degree of each vertex, DEGREE).

Figure 1: AutoGraphiX Control Center



  - Matrices (for example, the adjacency matrix ADJACENCY, or the distance matrix DISTANCE).
  - Family indicators are simply boolean invariants that may be used to determine whether the graph under study belong to the considered family (for example whether the graph is a tree, TREE). Note that this class of invariants is not intended to be used for the optimization, but just for description.

- Operators are used altogether with invariants as they are computed from items that are used as arguments. Examples of operators varies from sum, difference, product or ratio to more sophisticated ones like eigenvalues.

- Constraints or Objectives are items that cannot be used as arguments by other items and are used by the system to define the constraints and objective functions to be used for the optimization. Note that AGX from the version AGX-III, the optimizer is designed to handle multiple objectives, and each objective may be minimized, maximized or both. In case an objective is minimized and maximized, the two problems are treated one after the other or simultaneously, depending on the number of threads available on the computer. As more than one objective may be minimized or maximized, or both, the number of problems to solve is 2^k where k is the number of objectives that needs to be minimized and maximized.

To add one item on the white board, one just needs to select it by the mean of the context menu (right click).

2 Use of the optimizer : Minimizing the Randić index

Let's start with a first simple example. Suppose one is interested in finding graphs that maximize or minimize the Randić index. The Randić index is defined as follows :

$$Ra = \sum_{(i,j)\in E} \frac{1}{\sqrt{d_i d_j}} \tag{1}$$

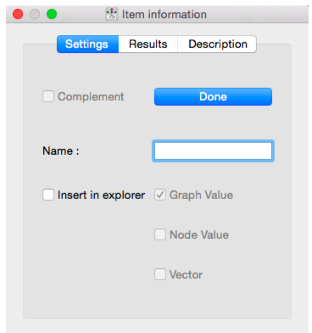
where d_i indicates the degree of the vertex i. The first step is to open the white board, then insert the RANDIC invariant that could be accessed through the context menu in Invariants→Value→RANDIC. The value of the Randić index for the current graph is then displayed in a rectangle with name RANDIC. To make it the objective function to me minimized, one must add the minimization objective by choosing MINIMIZE from the context menu of the white board.

At that step, we have the RANDIC item and an arrow pointing downward on the white board (as well as 4 blue lozenge that will be described later). To express that the objective to minimize is the Randic index, one need to connect RANDIC and MINIMIZE. This is achieved by selecting both items (clicking on then, or by selecting an area that contains them on the white board). To unselect an item, one just needs to click it. Unselecting all items may either be done through the context menu, or by pressing "Ctrl+U". It is also possible to select all items by pressing "Ctrl+A", or by the context menu. As the RANDIC item is the first (and only) argument of MINIMIZE, we just need to press the "1" key, or click the button. A red link joining the bottom of RANDIC to the top of MINIMIZE should appear with the number "1" in its middle. This indicates that RANDIC is the argument of MINIMIZE. To find a tentative solution of the problem, one then just needs to call the optimizer by clicking the button either from the top of the control center, or the top of the white board. A window with the optimizer's progress bar appears.

Note During the optimization process, the white board is tentatively updated, but its display may not be up to date. It is important to update it after optimization by clicking the button. It is also possible to keep a track of the optimization by opening the Console Log (by clicking the button from the top of the Control Center).

Clicking cancel on that window, or the button will cause the optimization to stop. To see the graph obtained by agx, click on the graph button . The *Graph Window* appears then. It is likely that the solution provided by agx is then the empty graph, a graph with no edge. This solution may not be the desired one as one usually searches for properties of connected graph. The addition of a constraint stating that the graph is connected is required. This is simply achieved by inserting the CONNECTED constraint through the context menu of the white board (Constraints*rightarrow*CONNECTED). A red octagon named CONNECTED should appear on the white board, with a number (probably 9 in this case). The red color indicates that the corresponding constraint is not respected. The number "9" is the value of the violation. Values of violations in constraints are sometimes complicated formulas that aims at indicating the optimizer a direction toward better

Figure 2: Item information for EIGENVALUES

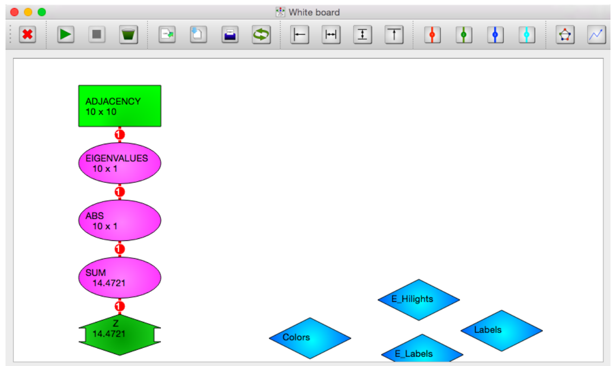


solutions. In the case of connexity, the value is simply the number of connected components - 1, *i.e.,* the number of components that should be removed to get a connected graph. In any case, when a constraint is respected, the octagon appears in green, and its value is 0.

3 Problem definition using the white board : Maximizing and/or minimizing the Energy of a graph

This example is a way to get used to more sophisticated formulas. The energy is the sum of the absolute values of the eigenvalues of the adjacency matrix. This invariant is already available in AGX, but we will explain how it could be possible to build it. The first step is to add the adjacency matrix to the white board (Invariants→Matrix→ADJACENCY). Then, we add the EIGENVALUES item (Operators→EIGENVA and connect ADJACENCY as first argument of EIGENVALUES in the same manner as the previous example, by selecting both items and pressing the key "1" (or clicking the  button). Double clicking on the EIGENVALUES ellipse will open the window on figure 2. This window has three tabs. On the "Settings" tab, the user may select some parameters that will be discussed later, when discussing the *Properties Explorer*. The "Results" tab provides the results of the calculation of the item. In this case, it provides the list of eigenvalues of the adjacency matrix, and the "Description" tab provides a short description of the item, a definition and the function. It is therefore possible to explore the eigenvalues of the adjacency matrix through this window. However, to find a graph maximizing the energy, one must

Figure 3: White board for the optimization of the energy



first automate its computation and provide it to AGX as objective function. Another item must be used to compute the absolute values of the adjacency matrix, and this item is ABS. Again, the insertion of the item is achieved through the context menu (Operators→ABS), and the argument is set by selecting EIGENVALUES and ABS before clicking the  button. In a similar way, we compute the sum of the absolute values of the eigenvalues of the adjacency matrix by linking the operator SUM to ABS, ABS being the argument. Finally, one needs to add the MAXIMIZE item and link it to SUM, by setting SUM as first argument of MAXIMIZE.

Clicking the  button will then start the optimization.

Double clicking on the objective item will open its information window as previously explained for the EIGENVALUES item. It is possible then to check the "Minimize" button (and it is then possible to uncheck the option "Maximize"). Depending on the optimization sense, an upward arrow represents the objective (Maximization), or a downward arrow (Minimization), or a double arrow in the case both maximization and minimization are required. If both maximization and minimization are required, when starting the optimization process, the progress window of the optimizer will yield 2 progress bars (one for maximization and the other for minimization).

At that stage, the white board should look like figure 3.

Depending on the number of available threads, bots could be done at the same time. After the optimization, AGX sends the optimal solutions found to the *Properties Explorer*. It is possible to pass through the various graphs found by pressing the space bar when the graph representation window is active. In order to avoid the accumulation of graphs that the user may not want to keep, it is important to click the  button. Otherwise, the new solutions will simply be added to the list of previously found solutions

4 Problem modification : Trying to find graphs with an integral spectra

In this section, we will explain how the syntax of the white board is used and how to modify a problem definition without the need to write it again from scratch.

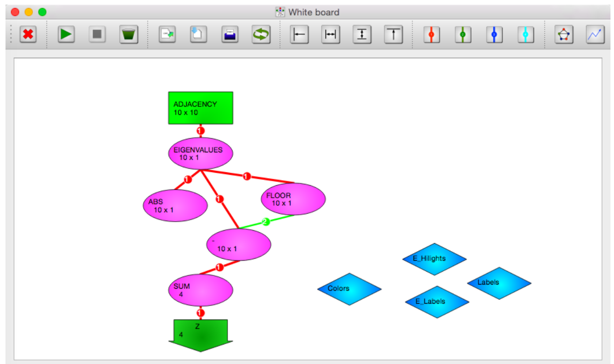
The items in the white board are stored in a stack, and the computations are achieved from the first to the last. As Invariants do not have any argument, they are always placed at the beginning of the stack. On the opposite, objective function items and constraints are never used as arguments and are always placed at the end of the stack. The Operators are in the middle, and their relative position is important. Indeed, if the item ITEM1 is before ITEM2, linking them with an argument relation will make ITEM1 the argument of ITEM2.

By default, when adding a new Operation item, it is placed on the stack at the end of the operators items. If one needs to replace the item SUM by another operator, there is no problem, as we just need to add that new operator item, and link EIGENVALUES to it as first argument.

Suppose we want AGX to search for graphs minimizing the sum of the fractional parts of its eigenvalues. The first thing to do would be to add the FLOOR operator between EIGENVALUES and SUM. For this, we must select SUM, press the "Alt" key and add the operator FLOOR to the white board. By pressing the "Alt" key, AGX inserts the next operator before the first selected operator. The next step consists in linking EIGENVALUES and FLOOR. To find the fractional part of the eigenvalues, one may substract the floor from the eigenvalues, which is done as follows :

1. select SUM (only)
2. insert "-" (Operators→"-")
3. link EIGENVALUES as first argument of "-"
4. link FLOOR as second argument of "-".

Figure 4: White board to search for integral graphs



Finally, link "-" as the first argument of SUM and change the objective to MINIMIZATION.

The white board should be as displayed on figure 4. The ABS item is no more needed and may be removed by pressing "Ctrl+X" after being selected.

5 Use of the Properties Explorer : Maximizing and Minimizing the Balaban index in the graph and its complement

In this section, we will explain some basics for the use of the Properties Explorer to study invariants, a single value for each graph.

Before any further consideration, it is important to notice that AGX-III was build to deal with multiple objectives. Instead of the parametrization that was used in the previous versions of the software, we suggest to use more than one objective function, and to consider their maximization, minimization, or both.

As an example, we will consider here the problem to maximize and minimize the Balaban index in the graph and its complement.

Then, we will use AGX capabilities to display the results.

The Balaban index is defined as follows :

$$J = \frac{m}{m-n+2} \sum_{(i,j)\in E} \frac{1}{\sqrt{t_i \times t_j}} \tag{2}$$

where t_i indicates the transmission of the vertex i. The transmission of a vertex is defined as follows :

$$t_i = \sum_{j=1}^{n} d_{ij}, \tag{3}$$

where d_{ij} is the geodesic distance between vertices i and j.

The Balaban index is available in AGX. Studying the bounds on the possible values of the Balaban index in the graph and its complement could be done as follows :

- Insert the BALABAN invariant in the white board
- add the objective function to be maximized and minimized (MAXMIN), and connect the BALABAN to MAXMIN.
- add another occurence of BALABAN
- double click on the last added BALABAN
- on the "Settings" tab of the window, check "Complement". This will cause the Balaban index to be computed on the complement of the graph instead of the graph itself. The rectangle should have a white circle with the letter "c" inside, indicating that.
- connect the second BALABAN to another MAXMIN.

When running the optimizer, we notice that 4 problems will be solved. After completion of the optimization, we have a wide number of solutions, first because 4 problems were solved, but also because each of then yields 2 objectives, which means that the solution is a set of Pareto optimal solutions, not just a single solution for each problem. These solutions are sent to the *Properties Explorer* and may now be studied. In order to see the results in the properties explorer, we must configure the corresponding items.

This is achieved adding a "REPORT" operator which will have the desired item as argument. In the current case, these values are invariants, a single value for the whole graph, the "Graph value" should be checked. For invariants, it is defined by the system and cannot be changed by the user, but it is an option to check when operation items are considered. If the values are associated to the vertices, the "Vertex value" must be checked (for example, the degrees), while multiple values that are not associated to vertices must be identified as "Vector" (for example, the eigenvalues of a matrix). As the name of the variables used by the explorer is that of the Invariant or Operator, it is recommended to define a custom name. Indeed, specially in this case, the name BALABAN will be used for the graph as well as its complement, which is not very convenient. In the settings of the item, a line may be filled to define the custom name of the item. We will chose "J" for the Balaban index of the graph and "JB" (J bar) for the value computed in the complement.

To start the exploration of the results, we must now open the Properties Explorer by clicking the  button from any of the main windows (control center, white board or graph display). Once the properties explorer is launched, it is recommended to click on the  button to update the database according to the items and their properties as defined on the white board. As the study refers to invariants (a single value for the whole graph), the *Graph-values* tab must be active (in this case, the Vertex-values tab will contain as many copies of each observation as there are vertices in each graph).

Four subtabs are available then :

- Filter
- Table view
- X-Y plot
- Conjectures

Filter may be used to narrow the selection of observations to be displayed, for example by only considering values above or below a given threshold.

Table view displays the values, an obserbation per line, and a measure per column. Figure 5 represents the table view for the current problem.

X-Y Plot is a facility by which a 2D-plot may be displayed representing the values of a measure as a function of another. The X-scale as well as the Y-scale may be changed to improve the view. The radius of the dots may also be modified if needed.

Figure 6 represents the XY-plot for the current problem with both X and Y scales adjusted to 100.

Finally, the Conjecture subtab provides a facility to find conjectures. A detailed illustration of this feature will be given in the next example.

Figure 5: Table view for the Balaban index in the graph and its complement

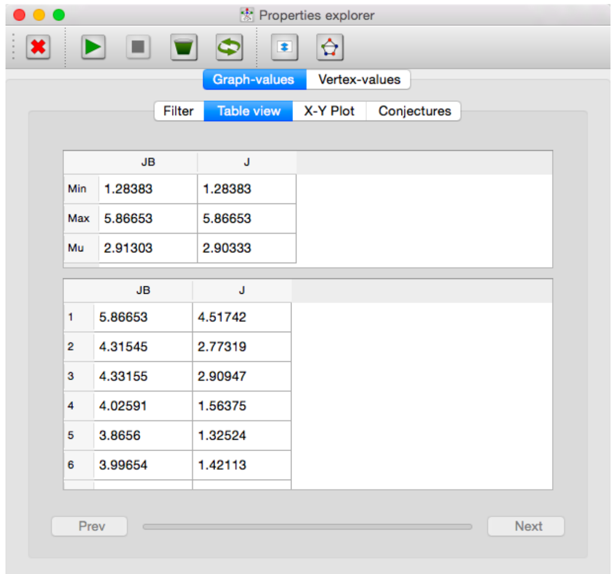
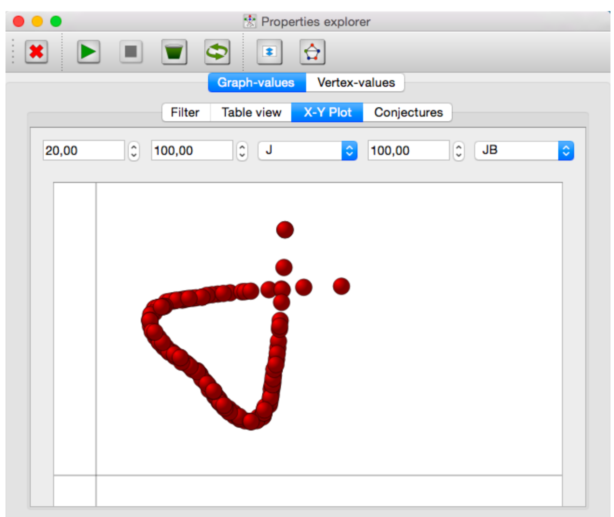


Figure 6: XY-plot view for the Balaban index in the graph and its complement



6 Available invariants and operators

6.1 Invariants

AutoGraphiX may use a wide range of invariants, but it may also use values defined as vectors (one value per node, or a vector for the whole graph). Node related values are cannot invariants (they change with the labeling of the graph), but they may are considered as such here as long as they are computed directly from the graph.

6.1.1 Invariant Values

- ABC : Atom bond connectivity index, defined as $ABC = \sum_{(i,j)\in E} \sqrt{\frac{d_i+d_j-2}{d_i \times d_j}}$ where d_i is the degree of the vertex i.
- AVDIST : Average geodesic distance between pairs of vertices.
- BALABAN : Balaban index, defined as $\frac{m}{m-n+2} \sum_{(i,j)\in E} \frac{1}{\sqrt{t_i t_j}}$ where t_i is the transmission of the vertex i, m the number of edges and n the number of vertices.
- CLAWS : Number of claws in the graph, number of times the star S_4 appears as induced subgraph.
- CLIQMAX : Cliq number, the cardinality of the largest set of vertices that are pairwise adjacent.
- CHROMATIC : Chromatic number, the minimum number of colors that are needed to color each vertex such that two adjacent vertices do not share the same color.
- DEGMAX : Maximum degree among vertices of the graph.
- DEGMIN : Minimum degree among vertices of the graph.
- DIAMETER : Maximum distance between pairs of vertices.
- DOMINATION : Cardinality of the minimum dominating set S such that $\forall v \in V$ either $v \in S$, or $(v, w) \in E$ and $w \in S$.
- EE : Estrada index, sum of the exponents of the eigenvalues of the adjacency matrix.

- ENERGY : Energy, sum of the absolute values of the eigenvalues of the adjacency matrix.

- GA : Geometric-algebraic index $GA = \frac{\sqrt{d_i \times d_j}}{(d_i + d_j)/2} \quad \forall (i,j) \in E$, $= 0$ otherwise. Where d_i is the degree of the vertex i.

- GIRTH : Girth, the size of the smallest cycle in the graph.

- HARARY : Harary index, $\sum_{v,w \in V} \frac{1}{d(v,w)}$, where $d(v,w)$ is the geodesic distance between vertices v and w.

- ISUURBALLE : Suurballe index, the sum of the shortest cycle involving pairs of vertices.

- IGSUURBALLE : Generalized Suurballe index, sum of the sizes of the shortest cycle involving each pair of vertices using distinct vertices.

- K4 : The number of K4 in the graph.

- KIRCHHOFF : KIRCHHOFF index defined as $n \sum_{k=1}^{n-1} \frac{1}{\mu_k}$, where μ_k is the k^{th} eigenvalue of the laplacian matrix of G.

- NCONN : Node connectivity index, the minimum number of vertices that must be removed from the graph to disconnect it.

- ORDER : Order of the graph, $n = |V|$ the number of vertices of G.

- PENDING : Number of pending vertices, vertices of degree 1.

- PLATT : Platt index, defined as $P = \sum_{(i,j) \in E} d_i + d_j - 2$ where d_i is the degree of the vertex i.

- RADIUS : Minimum eccentricity : $r = \min_v \max_w d(v,w)$.

- RANDIC : Randić index, defined as $\chi = \sum_{(i,j) \in E} \frac{1}{\sqrt{\delta_i \delta_j}}$ where δ_i is the degree of the vertex i.

- SIZE : Size of the graph, $m = |E|$, the number of edges of the graph.

- STABLEMAX : Stability number, the cardinality of the largest set of vertices that are pairwise non adjacent.

- SZEGED : Szeged index, $\sum_{(u,v)\in E} n_u \times n_v$, where n_u is the number ov vetrices of G that are closer to u than v and vice versa.
- TRIANGLES : The number of triangles in the graph.
- WIENER : Wiener index, $\sum_{v,w\in V} d(v,w)$, where $d(v,w)$ is the geodesic distance between vertices v and w.
- ZAGREBM1 : Zagreb M1 index, defined as $M1 = \sum_{v\in V} d(v)^2$ where $d(v)$ is the degree of the vertex v.
- ZAGREBM2 : Zagreb M2 index, defined as $M2 = \sum_{(i,j)\in E} d(i)d(j)$ where $d(i)$ is the degree of the vertex i.

6.1.2 Invariant Vectors

- CENTRALITY : Betweenness adjusted centrality : Sum of betweenness of edges adjacent to v.
- CLOSENESS : 1/(Sum of distances to other vertices).
- CUTCENTRALITY : Cut centrality : Sum of normalized cut values of edges adjacent to v.
- CHROMATICVAL : label of the color that is assigned to each vertex in a way that minimizes the total number of colors and respect the coloration condition (2 adjacent vertices do not share the same color).
- DCENTRALITY : Sum of DBETWEENNESS of adjacent edges.
- DCENTRALITY2 : Sum of DBETWEENNESS2 of adjacent edges.
- DEGREES : Degree vector, number of vertices that are adjacent to each individual vertex.
- DISTCLOSESTLEAF : Distance of the vertex to the closest leaf.
- ECCENTRICITY : Maximum distance from a vertex to other vertices: $e_v = \max_w d(v,w)$ where $d(v,w)$ denotes the geodesic distance between the vertex v and the vertex w.
- HARMONIC : Sum of inverse distances to other vertices.

- IINDEX : I-Index for identifying actors between communities.
- INCLIQMAX : Vertices in a maximum cliq. To be used for display purpose.
- INDOMINATION : Vertices in a minimum domination set. To be used for display purpose.
- INSTABLEMAX : Vertices in a maximum stable set. To be used for display purpose.
- KCHERRIES : KC_i is the number of pending vertices that are attached to the vertex i.
- NUMERO : Vector of size n with numbers from 1 to n.
- NWIENERIMPACT : Impact on the Wiener index, of the removal of each node.The value is corrected by the addition of the transmission of the vertex v. NWIENERIMPACT = W(G)-W(G-v)+T(v).If the graph gets disconnected, an error is reported.
- ONEVEC : Column vector of 1's.
- SUMDISTP : Sum of distances to pending vertices.
- TRANSMISSION : Sum of distances to other vertices.

6.1.3 Invariant Matrices

- ADJACENCY : Adjacency matrix
- BETWEENNESS : Betweenness b_{ij} as proposed by Freeman (1977), the number of shortest paths among all pairs of vertices that are using the edge (i,j).
- COMNEIGH : Number of common neighbors
- DBETW : Distance-Betweenness db_{ij} modified version of Freeman's betweenness (1977), the number of shortest paths among all pairs of vertices that are using the edge (i,j) weighted by 1/path length.
- DBETW2 : Distance-Betweenness-2 db^2_{ij} modified version of Freeman's betweenness (1977), the number of shortest paths among all pairs of vertices that are using the edge (i,j) weighted by $1/(pathlength)^2$.

- DISTANCES : Distance matrix, geodesic distance between pairs of vertices.
- EINCLIQMAX : Edges in a maximum cliq. To be used for display purpose.
- EWIENERIMPACT : Impact on the Wiener index, of the removal of each edge. If the graph gets disconnected, an error is reported.
- GSUURBALLE : Generalized Suurballe matrix, Suv is the size of the shortest cycle involving vertices u and v using distinct vertices.
- IDENTITY : A matrix of order n with 1 on the main diagonal and 0 on all other entries.
- INGIRTH : Is the edge in the smallest cycle (only one such cycle is considered). Function for hilight purpose only.
- INDIAMETER : Edges belonging to the diameter of the graph.
- INSHORTESTPATH : Is the edge in the shortest path tree from the last selected node (only one such tree is considered). Function for hilight purpose only.
- LAPLACIAN : Laplacian matrix, $L = D - A$, where D is a matrix with degrees on the main diagonal and A is the adjacency matrix.
- MODULARITY : Modularity matrix
- MBETWEENNESS : Mandatory Betweenness mb_{ij}, the number of shortest paths among all pairs of vertices that are must use the edge (i,j).
- NLAPLACIAN : Normalized Laplacian matrix,$I - A'$, where $A' = \{a'_{ij} = fraca_{ij}\sqrt{d_i d_j}\}$.
- ONEMAT : Matrix of 1's.
- SLLAPLACIAN : Signless Laplacian matrix, $SLL = D + A$, where D is a matrix with degrees on the main diagonal and A is the adjacency matrix.
- SUURBALLE : Suurballe matrix, Suv is the size of the shortest cycle involving vertices u and v.

6.1.4 Invariant Families

- CATERPILLAR : Boolean function that indicates wether the graph is a caterpillar.
- COMET : Boolean function that indicates wether the graph is a comet.
- LOBSTER : Boolean function that indicates wether the graph is a lobster. A lobster is a tree that is made in such a way that if you remove all its leaves, then you have a caterpillar. The distance between any vertex and the root path is at most 2.
- PATH : Boolean function that indicates wether the graph is a path.
- SPIDER : Boolean function that indicates wether the graph is a spider.
- STAR : Boolean function that indicates wether the graph is a star.
- TREE : Boolean function that indicates wether the graph is a tree.

6.2 Operators

- CONSTANT : Constant value, vector or matrix defined by the user.
- + : Term by term addition of values, vectors or matrices (may handle up to 4 arguments). A value may also be added to a vector or matrix.
- - : Term by term substraction of values, vectors or matrices. A value may also be substracted from a matrix or vector.
- * : Matrix or vector multiplication. A matrix or vector may also be multiplied by a value.
- .* : Term by term multiplication of values, vectors or matrices.
- / : Division of a value, vector or matrix by a value, or the reverse.
- ADDCOLS : Concatenation of matrices by columns (adding the columns of the second after the columns of the first).
- ANGLE : Angle (Rad) between 2 vectors.
- ABS : Absolute values of the entries of the value, vector or matrix.

- CEIL : ceil values of the entries of the value, vector or matrix.
- CENTER : Centering of a vector or matrix by substracting the average value. In a matrix, the diagonal elements are considered in the computation of the average value.
- CHOLESKY : Cholesky decomposition (of a square matrix with non negative entries).
- COLUMN : Selection of a column in a matrix. The second argument indicates the number of the column to select. The output is a column vector.
- CORR : Correlation coefficient between 2 vectors.
- DETERMINANT : Determinant of the square symmetric matrix.
- DIAG : Construction of a diagonal matrix from a vector.
- DPALINDROMIC : Distance to palindromicity of a vector. Sum of the absolute values of the differences between the first and last element, second and last second, etc.
- ECORR : Edge correlation coefficient.
- EIGENVALUES : Eigenvalues of the square symmetric matrix.
- EIGENVECTORS : Eigenvectors (normalized) of the square matrix. Vectors are aranged as column vectors in the decreasing order of the corresponding eigenvalues.
- EPROD : Construction of a matrix from a vector v by computing $v_i \times v_j \quad \forall (i,j) \in E, = 0$ otherwise.
- ESUM : Construction of a matrix from a vector v by computing $v_i + v_j \quad \forall (i,j) \in E, = 0$ otherwise.
- EUCLIDIANDIST : Euclidian distance between 2 vectors or matrices
- EXP : Exponent of the entries of the value, vector or matrix.
- FIRST : First value of the argument. It it is a matrix, the first column is used.
- FLOOR : Floor values of the entries of the value, vector or matrix.

- GAMATRIX : Construction of the geometric-algebraic matrix from a vector v by computing $\frac{\sqrt{v_i \times v_j}}{(v_i+v_j)/2} \quad \forall (i,j) \in E$, $= 0$ otherwise.

- IFGE : If the first argument is larger or equal to the second (term by term, of same dimension). Returns the third value, vector or matrix. Otherwise the fourth value, vector or matrix is returned.

- INERTIA : Inertia of a square symmetric matrix, [number of negative eigenvalues, number of zero eigenvalues, number of positive eigenvalues].

- LAPLACIANOF : Laplacian matrix from a symmetric matrix M. $L = D - M$ where D is a matrix whose diagonal term $d_i = \sum_{j=1}^{n} m_{ij}$.

- LAST : Last value of the argument. It it is a matrix, the last column is used.

- MANHATTANDIST : MANHATTAN distance between 2 vectors or matrices

- MAX : Maximum value over a vector or a matrix. Not to be confused with MMAX.

- MAXCOL : Maximum over each column, if the argument is a matrix, a row vector is obtained.

- MAXROW : Maximum over each row. If a matrix is given as argument, a column vector is obtained.

- MIN : Minimum value of a vector or matrix. Not to be confused with MMIN.

- MMAX : Term by term maximum valuea over different vectors or matrices, or between a value and a vector or a matrix.

- MMIN : Term by term minimum valuea over different vectors or matrices, or between a value and a vector or a matrix.

- MOMENT : Inertia moment for each vertex, $\sum_{v \in V} d(u,v)w(v)$. Where d(u,v) is the distance between vertex u and v and $w(v)$ is the weight of the vertex v given as vector argument (default=degree).

- NBEQ : Frequency of each term of the second argument (value, vector or matrix) in the vector or matrix provided as first argument. Args=[matrix or vector M1,value (default, a value=0) M2,epsilon (default=1e-6)].

- NBVALUES : Number of different values in a vector/matrix.
- NORMALIZEMAT : Construction of a matrix from a vector v by computing $\frac{1}{\sqrt{v_i v_j}} \quad \forall (i,j) \in E$, $= 0$ otherwise.
- ROW : Selection of a row in a matrix. The second argument indicates the number of the row to select. The output is a column vector.
- SECOND : Second value of the argument. It it is a matrix, the second column is used.
- SLLAPLACIANOF : Signless Laplacian matrix from a symmetric matrix M. $L = D + M$ where D is a matrix whose diagonal term $d_i = \sum_{j=1}^{n} m_{ij}$.
- SORT : Sort a vector in decreasing order.
- SQRT : Term by term square root.
- SUM : Sum of the values in a vector or matrix. Not to be confused with + which adds corresponding elements of different matrices.
- SUMCOL : Sum of the terms of each column of a matrix, providing a row vector.
- SUMGE : Summation of the terms of a matrix that are greater than or equal to the second argument. By default, the value of the second argument is 0.
- SUMPRODUCT : Sum of the products of matching terms of 2 vectors or matrices.
- SUMROW : Sum of the terms of each row of a matrix, providing a column vector.
- TRANSPOSE : Transposition of a vector or matrix.
- VALUE AT : Value at the position i from the vector provided as first argument, i being provided as second argument.
- VAR : Variance of a vector or matrix.

6.3 Constraints

- < : The first argument should be less than the second.
- <= : The first argument should be less than or equal to the second.
- == : The first and second arguments should be equal.
- >= : The first argument should be greater than or equal to the second.
- > : The first argument should be greater than the second.
- CONNECTED : The graph should be connected.
- DEGMAXLEQ : The maximum degree should not exceed the argument.
- MAXDIAMETER : The diameter should not exceed the argument.

Aknowledgments

This work was supported by NSERC (Canada).

References

[1] M. Aouchiche, J.-M. Bonnefoy, A. Fidahoussen, G. Caporossi, P. Hansen, J. Lacheré, and A. Monhait. Variable neighborhood search for extremal graphs. 14. The AutoGraphiX 2 system. In L. Liberti and N. Maculan, editors, *Global Optimization. From Theory to Implementation*. Springer Science, New-York, 2006.

[2] M. Aouchiche, G. Caporossi, and P. Hansen. Variable neighborhood search for extremal graphs. 20. automated comparison of graph invariants. *MATCH Commun. Math. Comput. Chem.*, 58:365–384, 2007.

[3] M. Aouchiche and P. Hansen. Proximity and remoteness in graphs: Results and conjectures. *Networks*, 58:95–102, 2011.

[4] M. Aouchiche and P. Hansen. The normalized revised szeged index. *MATCH Commun. Math. Comput. Chem.*, 67:369–381, 2012.

[5] M. Aouchiche, P. Hansen, and M. Zheng. Variable neighborhood search for extremal graphs. 19. further conjectures and results about the randic index. *MATCH Commun. Math. Comput. Chem.*, 58:83–102, 2007.

[6] A. T. Balaban. Distance connectivity index. *Chem. Phys. Lett.*, 89:399–404, 1982.

[7] G. Caporossi and P. Hansen. Variable neighborhood search for extremal graphs. 1. The AutoGraphiX system. *Discrete Math.*, 212:29–44, 2000.

[8] G. Caporossi and P. Hansen. Variable neighborhood search for extremal graphs. v. Three ways to automate finding conjectures. *Discrete Math.*, 276:81–94, 2004.

[9] G. Caporossi and P. Hansen. A learning optimization algorithm in graph theory. versatile search for extremal graphs using a learning algorithm. *Lecture Notes in Computer Science*, 7219:16–30, 2012.

[10] G. Caporossi, M. Paiva, D. Vukicević, and M. Segatto. Centrality and betweenness: Vertex and edge decomposition of the wiener index. *MATCH Commun. Math. Comput. Chem.*, 68:293–302, 2012.

[11] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.

[12] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European J. Oper. Res.*, 130:449–467, 2001.

[13] S. Majstorović and G. Caporossi. Bounds and relations involving adjusted centrality of the vertices of a tree. *Graphs and combinatorics*, 2015 (to appear).

[14] H. Melot. Facet defining inequalities among graph invariants: The system graphedron. *Discrete Applied Mathematics*, 156:1875–1891, 2008.

[15] N. Mladenović and P. Hansen. Variable neighborhood search. *Comput. Oper. Res.*, 24:1097–1100, 1997.

[16] D. Vukicević and G. Caporossi. Network descriptors based on betweenness centrality and transmission and their extremal values. *Discrete Applied Mathematics*, 161:2678–2686, 2013.