Master’s Thesis (Coursework) (1st year)
Software and System Engineering Programme
School of Software Engineering / FCS

Full Name: Alpeev Vladislav Sergeevich
Student ID – M221МСПИН017

Title: Structural Analysis of Systems in the Presence of Uncertainty of Relationships Between Their Elements (Modified Q-Analysis Procedure)

NATIONAL RESEARCH
UNIVERSITY

Supervisor: Degtiarev K.Y., Assoc. Professor, Dept. of Software Engineering FCS

Date of Submission: June 20th, 2024

Moscow, 2024

ABSTRACT

This Thesis focuses on the Structural Analysis of Systems in the presence of uncertainty concerning the relationships between their elements, employing a modified Q-Analysis procedure. The work aims to enhance the classical Q-Analysis by incorporating approaches to handle uncertainties in expert assessments, the weight of connections, and other complexities inherent in modeling systems with ambiguous relational data. Key methods explored include the use of simplicial complexes, the calculation of eccentricities, and adaptations through Fuzzy Set theories to accommodate vague and overlapping data inputs. The core contribution lies in extending the Q-Analysis to include not only binary but also weighted and fuzzy relationships, allowing for a more nuanced representation and understanding of system structures.

Key words: Q-analysis; structural analysis; weight of connections; uncertainty of expert assessments; simplex; simplicial complex; eccentricities;

TABLE OF CONTENTS

LIST OF DEFINITIONS, TERMS AND ABBREVIATIONS

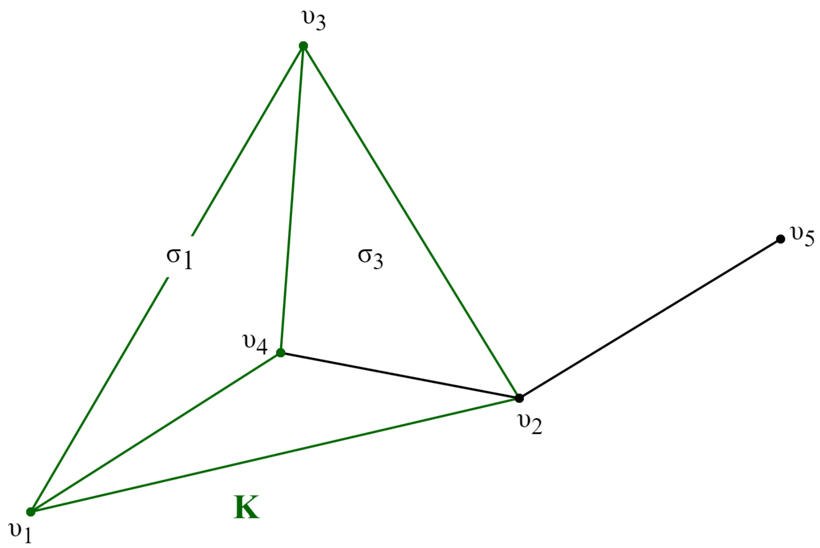

Figure 1. Graphical representation of the elements mentioned in the definitions.

Vertices – υ_i (Fig. 1.) are points where multiple edges or faces converge. In $\mathbb{R}^n$, a vertex is defined as a 0-dimensional element of the polyhedron, which is the intersection of at least n hyperplanes. Thus, considering geometric shapes or object, vertices - element that represents the point or corners.

Simplex (σ) – geometric object formed by vertices and characterized by its dimension. An n-simplex σ can be represented by the convex hull of its $n+1$ vertices $\{v_0, v_1, \ldots, v_n\}$ in an n-dimensional space:

$$\sigma = \left\{ \sum_{i=0}^{n} \lambda_i v_i \;\mid \sum_{i=0}^{n} \lambda_i = 1, \lambda_i \geq 0 \right\}, \qquad (1)$$

where v_i - are the vertices of the n-simplex, λ_i – are the weights (or coefficients) assigned to each vertex v_i.

Vertices (mentioned earlier) need require weights, because λ_i define the precise location of points within the simplex. Geometric representation, depending on the dimension, defined as follows:

- **0-Simplex**: A single point (vertex).
- **1-Simplex**: A line segment bounded by two vertices (edge) like σ_1 (Fig. 1.).
- **2-Simplex**: A triangle bounded by three edges and three vertices like σ_2 (Fig. 1.).

- **n-Simplex**: An n-dimensional polytope bounded by $n + 1$ vertices, where each face is an $(n - 1)$-simplex.

Dimension of simplex (n) – An n-dimensional simplex (or n-simplex) is a set of $n + 1$ vertices $\{v_0, v_1, \dots, v_n\}$ that form an n-dimensional geometric object. The dimension of a simplex is defined as the highest dimension of its constituent vertices.

Simplicial complex (K) – a collection of simplices that satisfies conditions:

- Every face of a simplex in K is also a simplex in K.
- If σ is a simplex in K and F is a face of σ, then F must also be in K.
- The intersection of any two simplices in K is either empty or a common face of both.
- If σ_1 and σ_2 are simplices in K, then $\sigma_1 \cap \sigma_2$ is either empty or a simplex in K.

Simplicial complex is a very important mathematical structure because makes it possible to represent more complex interactions or relationships among elements when one tries to describe the system model consisting of high-dimensional components (simplices).

Simplex face (F) – for an n-dimensional simplex (or n-simplex) is any k-dimensional simplex ($k < n$) that is formed by a subset of the vertices of the n-simplex. Faces of a simplex illustrate the hierarchical structure within a simplicial complex. Higher-dimensional simplices are composed of lower-dimensional faces. This term is very important in Q-analysis because it helps in analyzing how simplices connect and overlap within a simplicial complex.

Facets of simplex – for an n-simplex is an $(n - 1)$-simplex formed by omitting one vertex from the n vertices of the n-simplex.

Connectivity - two simplexes $\sigma^{(\alpha)}$ and $\sigma^{(\beta)}$ are q-connected in complex K, if there is a sequence of intermediate simplexes σ(α) ,σ(τ1) ,...,σ(τn) ,σ(β) such that any pair of simplexes in the sequence share q-face (thus, σ(α) and σ(β) are connected by means of q-chain «links» that correspond to intermediate simplexes mentioned); if two simplexes $\sigma^{(\alpha)}$ and $\sigma^{(\beta)}$ are q-connected, they are also (q-1)-, (q-2)-connected, etc. Besides, any q-simplex q σ is q-connected to itself.

INTRODUCTION

System in its formal definition is “a collection of components organized to accomplish a specific function or set of functions” [1] or in other words it is numerous objects that need to interact with each other. From this definition one got the idea that specifically organized components should be took into consideration during work with the system, where specificity expressed by different relations between components. To identify these two important elements information about the system must be extracted. The most valuable sources of information about the system are stakeholders, analysts, and people involved in its operation. There are a large number of techniques: interviews, workshops, focus groups, questionnaires, etc. It is a good practice to use several techniques at the same time, because it is not so easy to collect information in the right way, and it can be difficult for stakeholders to describe everything in the system in all details [2]. The extracted information, after being received in one way or another, should be formalized and distributed. To do this, it is necessary to have certain frameworks that allow not only to correctly present the existing picture, but also to enrich it. Most often, schemes of various notations like UML, IDEF, DFD, etc. are used for this purpose. A researcher or an analyst of the system must think “in big picture” in order to identify connections between the elements of the system and correctly assess their impact on each other. The mentioned schemes allow one to see the direct connections between the elements, but do not give an explicit idea of the composite connections, are not intended to identify them. In addition, the identification would not allow one to extract any more information about these connections in the future. In this case, there is a need for a tool that gives a mathematical apparatus that allows one to draw additional conclusions based on the available information. One of the possible options is the Q-analysis method presented by R.H. Atkin in the early 1970s [3].

Taking about components it is important to mention level of decomposition. Combining elements into larger components should not be ignored during forming a list of objects under consideration. But the formation of subsystems cannot be realized without describing the connections between smaller components. The emphasis in such case is on relations, not on elements, but mainly on how they are connected. Unfortunately, representation of connections (internal and external) seems to be a sophisticated task without accepting certain boundaries. The system can be described by designing a mathematical model which is related to the simplification of original, stressing those features and characteristics which are of importance for the researcher or available for observation/measurement [4]. The elements of a particular system are elements that are indivisible at a given depth of decomposition, which can be geometrically represented as vertices. In UML notation, these would be different types of entities, including classes. The dependence of one element on another there would be indicated by a line and an arrow. If one looks at this scheme from a geometric point of view, then the appearance of the concept of simplex is obvious. In addition to the connection of elements, the dimensional arrangement of the elements and their associations begin to carry important information and the formed simplicial complex can be considered.

As it was mentioned the simplicial complex forms the mathematical generalization of a planar graph consists of simple relations having only the property of presence or absence. Basically, direction and weight of this connections couldn't be specified inside such a model. Thus, they can be formalized through an incident matrix for two sets (Table 1.).

Table 1. Example of a system's model represented by the matrix.

λ	x1	x2	x3	x4	x5	x6	x7	x8
y1	1	1	1	1	0	0	0	0
y2	1	0	1	0	0	0	0	0
y3	0	0	0	0	1	1	0	0
y4	0	1	0	0	1	0	1	1
y5	0	0	0	0	0	0	1	0
y6	0	1	0	0	1	0	0	0

The table provides some visual representation of the structure and relationships of the components because it allows one to determine the list of vertexes of simplexes line by line. However, this is not enough to talk about how the simplex fits into the total complex. Assessing how well each simplex is "integrated" into the entire structure is important. To quantify this integration, we introduce the concept of "eccentricity" [5]. This value can be calculated in different ways. In this thesis, attention is paid to two approaches, provided by John L. Casti [5] and Lucien Duckstein [6].

John L. Casti Approach

The eccentricity of a simplex σ is given by the formula

$$ecc(\sigma) = \frac{\hat{q} - \check{q}}{\check{q} + 1}, \qquad (2)$$

where $\hat{q}$ is the dimension of σ as a simplex, and $\check{q}$ is the largest q value at which σ is connected to some other simplex in K [5]. Also, it is important to mention that for such formula $ecc(\sigma) = \infty$. This value means that simplex is not connected to any other simplex in K.

Lucien Duckstein Approach

Duckstein suggested another measure of eccentricity called ecc'

$$ecc'(\sigma) = \frac{2\sum_i \frac{q_i}{\sigma_i}}{q_{\max} \quad (q_{max} + 1)}, \qquad (3)$$

where q_i each q-level where σ appears, σ_i the number of elements in σ_i's equivalence class at level q_i and q_{max} the maximum q-level of the complex. Ecc (Equation 2) depends upon only a single simplex other than a while ecc' depends upon all the other simplices. Furthermore, ecc takes on values in $[0, \infty]$ and ecc' in $[0, 1]$ [6].

The mathematical model allows one to evaluate the connections within a well-defined system, which implies the presence of one, indisputable opinion about its structure. It is almost impossible to achieve this, because of two major reasons:

- Binary evaluation of relations is too simplified. Analysts, when assessing the degree of the relation, rather use verbal constructions which should be transform in mathematical form. In this case, other possible sets of values should be considered to fill in the cells of the matrices describing the structure of the system.
- Different experts may have different visions of the system. At the same time, the opinion of each of them should be taken into account, because this is the only way to achieve the most complete understanding of the system.

Often the process of system analysis starts with getting general information which can describe just presence or absence of a relationship that describes the system but it is ideally suited for a standard q-analysis procedure. Obviously, there should be some kind of extension which allows to work with more specific information that can show the weight of a relationship between elements. Relations will be more detailed and better described in this approach. But here we face a problem of transformation of a new matrix to the original binary form. The solution should be discovered, applied and tested. But as was mentioned before not only weight should be taken into account to enrich the model. Due to the human factor and the spread of knowledge about the system, a sophisticated approaches, which will be able to show distribution of opinions, should be found out. The usual experts' lack of confidence and inability to present an agreed unified assessment shows that the factor of vagueness is acceptable. Fuzzy sets are a mathematical tool close to solving such problems. This approach can be applied both in case the uncertainty of one particular expert and in case of the assessments of several people united into a single entity. Here also the problem of converting the matrix to the binary form used in the classical Q-Analysis should be solved and here can be several suitable solutions.

A software solution can combine classical Q-Analysis procedure, improvements mentioned before and also provide “user-friendliness” by using user interface. Ultimately a powerful tool can be obtained if all the above-mentioned difficulties are solved. A deeper study of system structures can be applied in various fields and helps analysts, managers, system architects, etc. to find additional solutions based on previously unknown information about structural complexity. And finally, this approach can be used as a base for an information system having no analogues and capable of becoming the foundation for a valuable expert system.

PROBLEM STATEMENT

The main problem that must be addressed in the work is the development of a software solution that implements the Q-analysis procedure and presents the necessary data for subsequent detailed analysis by stakeholders in a particular subject domain.

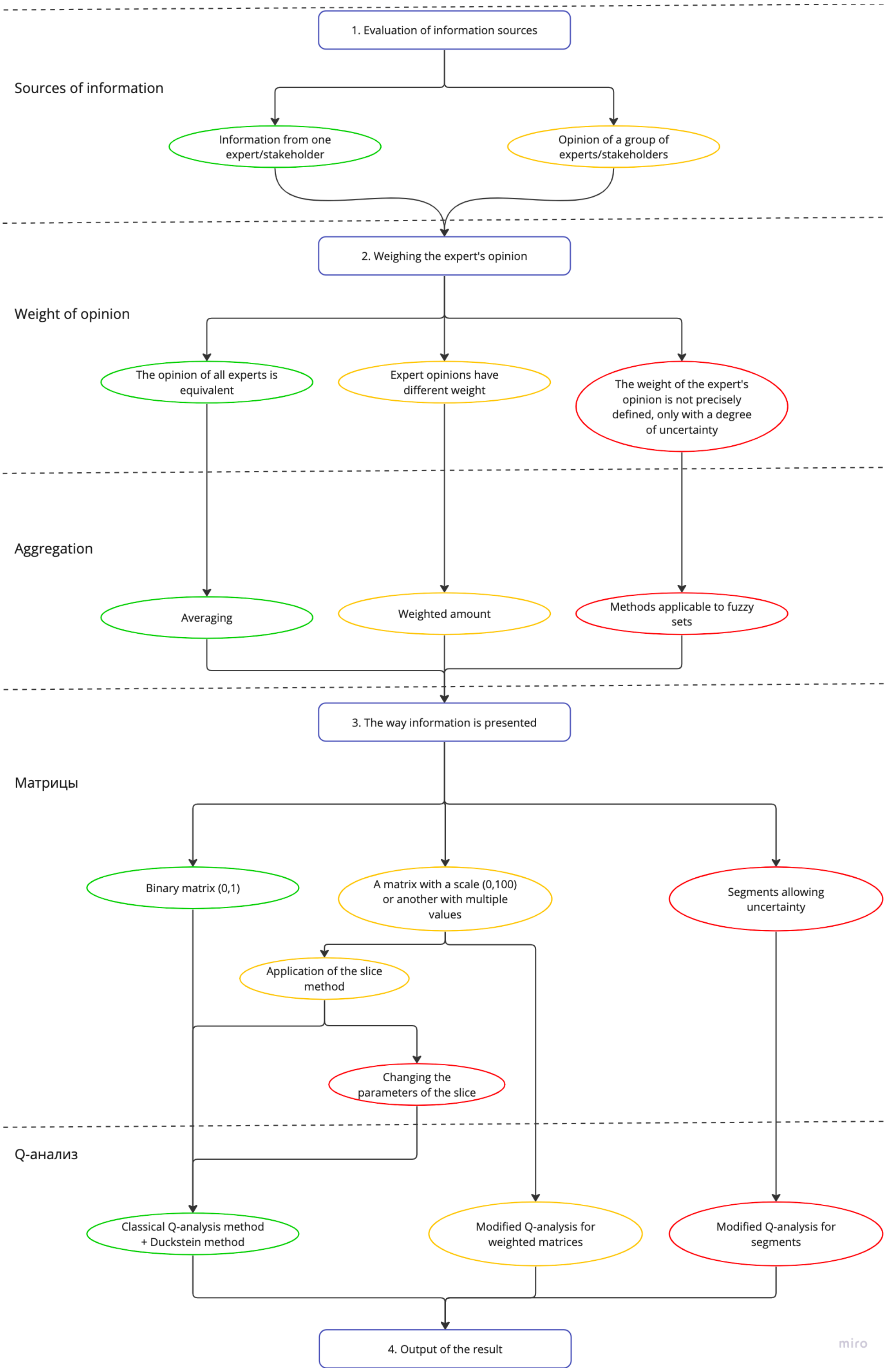

Figure 2. Structure of the task list.

It is also necessary to add the ability to work with more complex representation of relationships between elements. In particular, the relationships can be weighted, so there'll be a need to use approaches that allow us to transform the data (for example: weighted - binary form) before calculations. The basic solution in this case is the use of a slicing procedure. But the expert's assessment of relationships between the elements is not always converted to crisp (exact) values, so corresponding relationship weights can be represented using more sophisticated approaches. Thus, various types of fuzzy sets considered in the project seriously here.

It is proposed to address those tasks that will create the basis for the further development of the Q-analysis method in the field of system structure analysis. At the moment, there is no software implementation that allowed the analyst to use this method when processing information received from stakeholders. The general work plan can be presented in the following sequence (Fig. 2).

Here are three levels of complexity of tasks that allow to divide the work into parts, while getting a certain full-fledged result at each stage. The green level solves the basic problem of transferring a mathematical procedure into the form of software that allows the user to enter data and perform the necessary calculations. The yellow level suggests expanding the basic procedure by adding an additional opportunity to use link weights from the point of view of the Q-analysis procedure. The last red level involves the study of the application of fuzzy sets techniques for the development of the entity of the connection between the elements.

Thus, the main problem that must be addressed in the work is the development of a software solution that implements the Q-analysis procedure and presents the necessary data for subsequent detailed analysis by stakeholders in a particular subject domain. It is also necessary to add the ability to work with more complex representation of relationships between elements. In particular, the relationships can be weighted, so there will be a need to use approaches that allow to transform the data (to binary form) before calculations. Also, various types of fuzzy sets considered in the project seriously here.

THE MAIN PART

Within the framework of the designated set of tasks, it is necessary first of all to consider the basic mathematical procedure of Q-analysis and, based on this example, and determine the set of necessary entities. Also, all the requirements must be defined, without which the design process is impossible.

Example of calculation

Here will be provided an example of calculations from John Casti's book [5]. First of all, structure of the system should be defined (Table 2.). In this example, the domain is predator-prey relations.

Table 2. Predator-Prey Relations.

λ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
4	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
9	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
13	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
15	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0

First of all vector Q should be calculated. It is pretty clear that every row (excluding header) can be considered as a simplex. For every simplex dimension can be calculated using formula n – 1, where n is a number of relations. Here is important to emphasize that should be some rule or condition for relation existence. In case of binary matrix value 1 shows that simplex has a connection, but with the appearance of weight, that rule should be improved. The dimension of this model is 5.

Then, on every dimension (from 5 to 0) connectivity components should be found out. The connectivity level defines the required number of links with another simplex to be considered connected. First such case of such connectivity is presented on the level 1. There simplexes x1, x4, x9, x12, x14, x15 forms a single component. Finally, the composition of the vector Q and its connectivity components look as follows (Table 3).

Table 3. Predator-Prey Relations Q-Vector.

Level (q)	**Q value**	**Connectivity components**
5	1	{x4}
4	1	{x4}
3	2	{x4}, {x15}
2	3	{x4}, {x1}, {x15}
1	2	{x1, x4, x9, x12, x14, x15}, {x5}
0	1	{all except x7, x10}

Based on these table one can say that this system has problems with connectivity on the middle levels (1-3), but well-connected on the first and last levels. Here is important to emphasize that condition of having connection between two different simplices also depends on type of value.

Based on this Q-vector eccentricities can be calculated. Firstly, formula from Casti [5] approach will be used (2). Calculations will be performed for x4 simplex. The largest q value at which σ is connected to some other simplex is 1 (Table 3). Dimension of this simplex is 5 (Table 2). These values will be used inside the formula

$$ecc(\sigma) = \frac{5-1}{1+1} = 2\,. \quad (4)$$

For all other simplices results look as follows (Table 4).

Table 4. Predator-Prey Relations Eccentricity (Casti).

Simplex	**Eccentricity**
1	0.5
2	0

3	0
4	2
5	1
6	0
7	∞
8	0
9	0
10	∞
11	0
12	0
13	0
14	0
15	1

Secondly, formula from Duckstein [6] approach will be used (3). Example calculations will be performed also for x4 simplex. Maximum q level for this model is 5 (Table 3). Values for sum should be taken from all levels (Table 3). These values will be used inside the formula

$$ecc'^{(\sigma)} = \frac{2\,(\frac{5}{1} + \frac{4}{1} + \frac{3}{1} + \frac{2}{1} + \frac{1}{6})}{5 * 6} = \ 0{,}94(4), \qquad (5)$$

For all other simplices results look as follows (Table 5).

Table 5. Predator-Prey Relations Eccentricity (Duckstein).

Simplex	**Eccentricity**
1	0.14(4)
2	0
3	0
4	0.94(4)
5	0.06(6)
6	0
7	0
8	0
9	0.01(1)
10	0

11	0
12	0.01(1)
13	0
14	0.01(1)
15	0.34(4)

The analysis indicates that from the perspective of the predator, the complex exhibits strong connectivity at both the highest and lowest q values, yet it fragments into multiple unconnected components at the middle q levels. With the exception of species 7 and 10, which are not part of the predator complex, we observe that the least uniform member is species 4. This is primarily due to the fact that element 4 has many connections that it does not share with any other large-scale elements. These findings suggest that eccentricity reflects the adaptability of species in their feeding habits—that is, their capacity to withstand changes in the ecosystem without succumbing to starvation. This interpretation is reminiscent of the resilience concept, now applied at the level of individual species.

Proposed methods

As it was mentioned before there is a need to solve some problems related to the adaptation of relations values to the classical Q-analysis procedure.

Weighted (Real-Valued) Case

To fix or evaluate the relation, the analyst is suggested to use the number which can be beyond values 0 and 1. The domain must be defined the same for all respondents, although it is not required for further calculations. Then before performing calculations, it is proposed to transform the collected values using the slice procedure. For that purpose, **slice value** should be set by analyst. Relation with value which is greater than slice value will be considered as the equivalent of a relation with a value 1 from case of binary relations. Same logic will be used for checking connection between simplices. The slice procedure can be repeated several times and a comparison of the results will show how stable is the connectivity, how much it will be weakened when increasing slice value.

Fuzzy Sets Type 1 case

A Type-1 fuzzy set A over a universe of discourse X is characterized by a membership function $\mu_A(x)$, which maps each element in X to a membership value between 0 and 1. Formally, A can be defined as:

$$A = \{ (x, \mu_A(x)) \mid x \in X \},$$

where x is an element of X, $\mu_A(x)$ is the degree of membership of x in A, $\mu_A: X \to [0,1]$ [7]. Type-1 fuzzy sets can handle vagueness and ambiguity effectively by allowing partial membership, which is not possible with standard binary sets. This quality allows one to expand the standard limitations of determining the presence of a relation in a q-analysis. In case of system structure membership function will be used as a value for matrix. Generally, plot bases on this value will look like trapezoid (or special cases of the membership function like triangle) (Fig. 3). To describe the relationship, the analyst must specify the points on the basis of which it will be possible to build a graph. Along the ordinate axis, values are limited by range [0, 1] (possible values of the membership function). Along the abscissa axis, there are no limitations based on Fuzzy Sets theory, but still **domain** should be defined. In such case the relative position of the graphs of several relations can be validated. The presence of a relation is determined by the presence of data in the corresponding cell of the system's structure matrix sufficient for plotting trapezoid. Connection between simplices is evaluated by applying **clipping point**. It is the point on the ordinate axis through which the α line will be drawn. The intersection points of the horizontal α line and the trapezoid form a segment which will be used for comparisons of relation values. Since the comparison does not necessarily have to reveal the complete correspondence of the segments, it is allowed to use **match proportion**. This is a user-defined value that determines the required minimum percentage of line segments that is necessary to consider them equivalent.

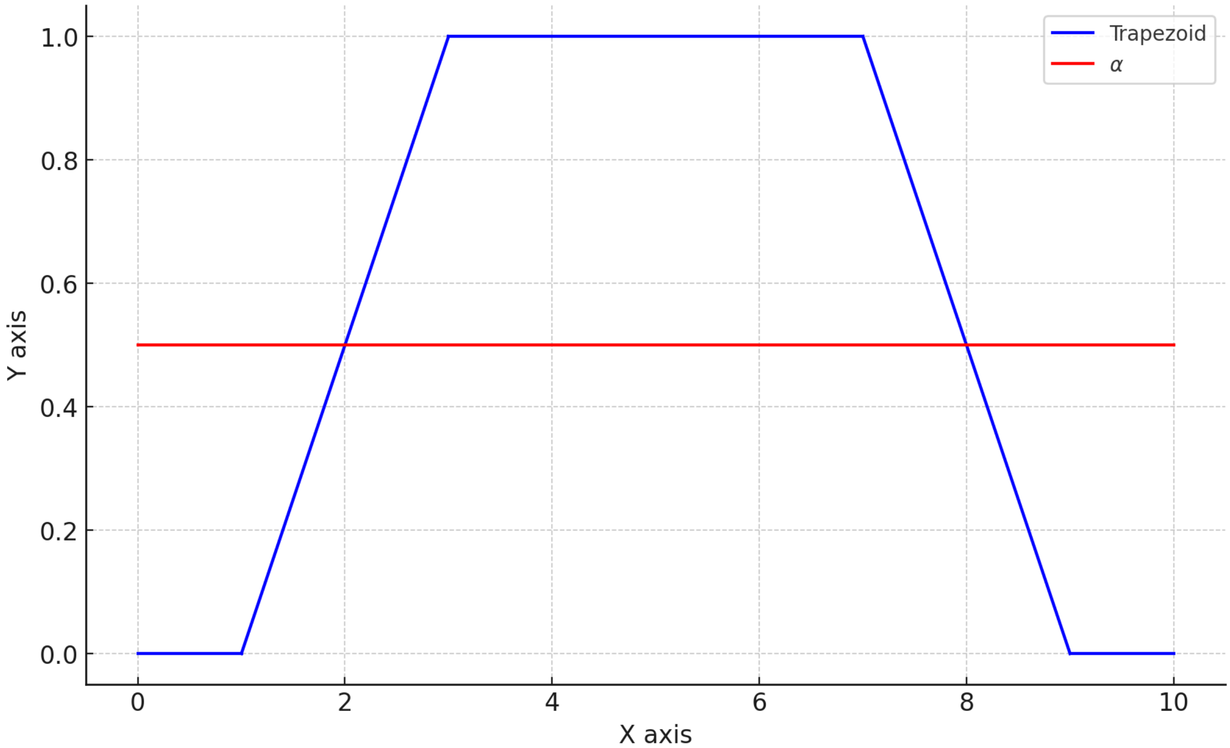

Figure 3. Plot for Fuzzy Sets value.

Only one line does not reflect the correspondence of the trapezoids, so the calculations should be repeated using different values of α. After receiving several results, they should be combined to obtain an overall score. To solve this problem, the aggregation procedure described below will be applied.

Fuzzy Sets Type 2 case

Type-2 fuzzy sets extend the concept of Type-1 fuzzy sets by allowing the degree of membership itself to be fuzzy. A Type-2 fuzzy set A over a universe of discourse X is characterized by a Type-2 membership function $\mu_A(x,u)$, where each element x in X is associated with a fuzzy set on the interval [0,1]. Formally, A can be defined as:

$$A = \{((x,u), \mu_A(x,u) | x \in X, u \in J_x \subseteq [0,1]\},$$

where x is an element of X, u is a secondary membership grade belonging to the fuzzy set J_x, $\mu_A(x,u): X \times [0,1] \to [0,1]$ is the Type-2 membership function that specifies the degree of membership of u for each x. Using of Fuzzy Sets Type 2 is shaped in a very similar way compared to Fuzzy Sets Type 1. The significant difference is that each point of the trapezoid in this case is transformed from a point to a segment. That is why procedure of comparisons of relation values should be improved. There are several ways to do that but the most obvious among them is transformation to Fuzzy Sets Type 1. Consider segment obtained after applying one of the clipping points (Fig. 4). Part [B, C] in any case will be used for further comparison with value of another relation. Parts [A, B] and [C, D] represent the segments taking into account the uncertainty. To transform the values in a form suitable for the case of Fuzzy Sets Type 1 two points should be chosen on segments [A, B] and [C, D]. But when one comes to comparing the segments of two relations, the question arises whether these side segments can be considered completely valuable if their choice is in field of uncertainty.

The segment (for example using segment [A, B]) can be separated on some smaller part and each of them can be assigned a weight before further comparisons. The weight can be in the range from 0 to 1, because point from [B, C] always will always be present in comparison and will be compared with another similar segment with a weight ratio 1:1. Talking about assignment rules, it is clear that the closer a point is to the boundary of the [B, C] segment, the more valuable it is. If one chose point X (Fig. 4) every point between X and B will be included in the result segment. Thus, the point closest to B will almost always be in the resulting segment. Such a model can be represented as a triangle, where one of the sides is a segment [A, B] (or [C, D]) and the other side (it is also the height) will be equal to 1. To calculate the segment, taking into account the weight, it is necessary to calculate the area of the triangle: [A, C] * 1 / 0,5. Since one end up with half the weight, one can simply find the midpoint of the side segments (for [A, B] and for [C, D]) and attach half to the main segment [B ,C]. After calculating the result segment, one can use the procedure previously described for Fuzzy Sets Type 1.

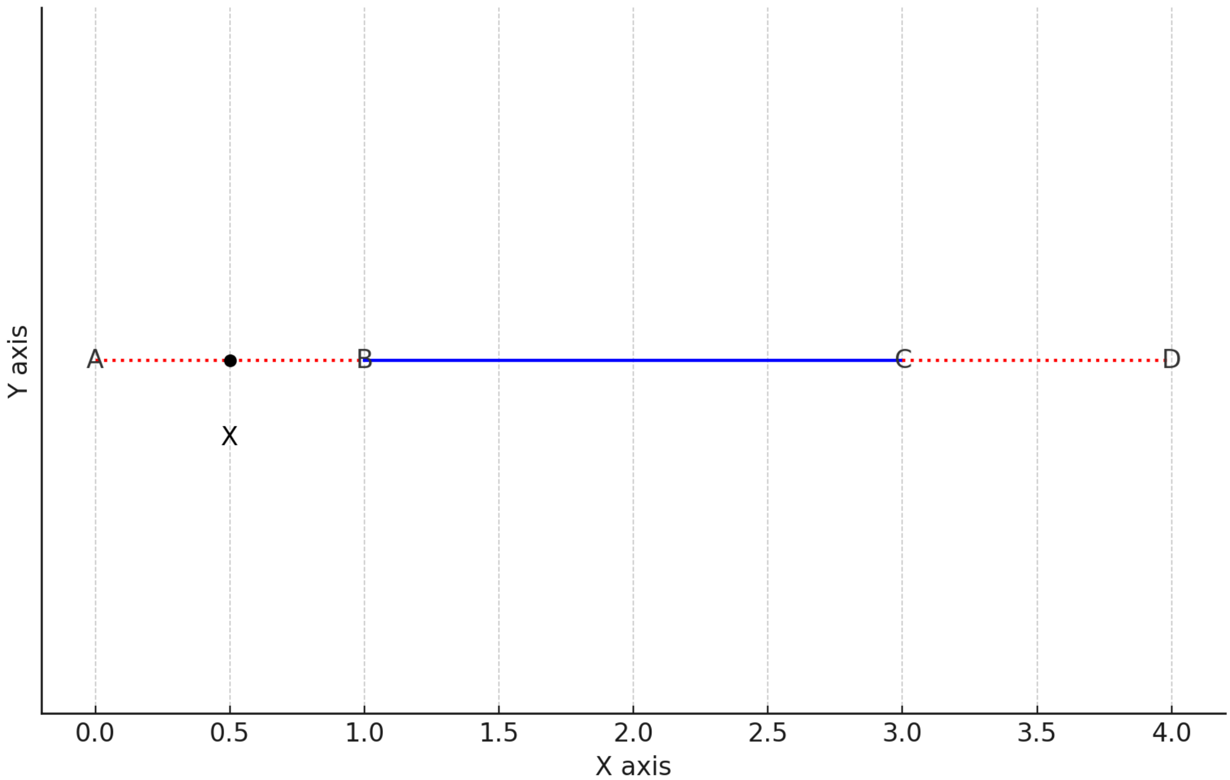

Figure 4. Plot for Fuzzy Sets Type 2 segment after clipping.

Aggregation

As it was mentioned before in cases of Fuzzy Sets user will get multiple results with different clipping points. Both Q-vector and eccentricities should be aggregated in some way. One possibility is to calculate the average values for each component of the vector and each eccentricity. This form will give a general idea of all the results, but will not be able to completely replace them, because in this case it will be problematic to reflect the components of connectivity.

Requirements

These requirements are a reflection of the potential needs of software stakeholders. As in an official document like SRS, they can be divided into two main groups.

Functional requirements

- Input of data
    - The user should be able to enter data using the form
    - The form should provide an opportunity to select/add elements between which it will be built (exists) connection.
    - The user should be able to input value of the relation in form which will be suitable for selected relations type (binary, weight, etc.).
    - The user should be able to enter data using a pre-prepared table (xlsx format for example). It should be possible to transfer a table compiled in a certain format to the application page and then convert the data contained in it to the application model.
    - The user should be able to download data from previous calculations. The implementation should allow loading from the application's memory. The result should be converted into an input form that allows one to make adjustments to the original data.
    - The user should be able to choose the algorithm by which the eccentricities will be calculated.
- Mathematical operations

    - The calculation of eccentricities should be implemented for methods according to J. Casti and by L. Dukstein/Chin.
    - According to the results of the calculations, there should be a representation of the values of the elements of the vector X and their determining number at each dimensional level of the q component of connectivity.
- Saving data
    - Data from previous calculations should be saved during current user session (until reloading/closing page). Data should contain both initial data and results of calculations. This data will be used for comparisons.
- Display of results
    - The output in the form of bar charts should be supported.
    - It should also be possible to review the results by levels presented in tab format.
    - It should be possible to select pairs of input data for comparison.

<u>Non-functional requirements</u>

- Application format
    - The mathematical operations required for the Q-analysis procedure should be implemented within the framework of the library project in order to preserve the possibility of reuse in different types of applications.
    - Access to the library in this implementation should be provided through an API to which various applications could be connected.
    - The front part of the application should be made in the form of a website
- Presentation of results
    - The user must understand what he is working with, therefore, the terms with which the work is performed and their definitions, as well as the formulas used in calculations, must be presented. For a complete understanding, a simple example can be provided, but definitions should clearly show the potential.
    - It should be possible to see the dimension of the vector Q.

    - When all subsequent displays must contain display changes. In the case of a graphical representation, they should be highlighted in an alternative color.
- Performance requirements
    - The program should not have restrictions in the dimension of the relationship tables, because the software is supposed to be used when considering large systems' models.

Architecture and technology stack

According to the requirements application will be implemented in the form of a client-server application. For ease of deployment, all application components are placed in docker containers.

Mathematical algorithms can be implemented as separated library. The calculation processes themselves will not depend in any way on the format of the user's interaction with the software, so the life cycle of the development of this block can be easily separated. There will also be an advantage that allows one to choose a separate technology stack that will meet the needs of this particular component.

Languages from the C family may be suitable for backend API part implementation. These languages give low-level control during development and have a good level of basic optimization, are widely supported by a large company, and are also popular and familiar to many developers. C# seems to be the most convenient from the point of view of development, because one of its key features is portability to a large number of platforms, which will allow one to deploy the application in any environment. In addition, this language has a sufficient number of frameworks that allows you to organize not only a library for calculations, but also an API for providing interaction with other parts of the systems.

To meet the requirements for user interaction, it is necessary to develop the front-end part of the application. Previously, there was a requirement that access to the calculation library should be provided through the API. This is dictated by the fact that calculations can require large resources and places serious demands on the user's device. This approach is unacceptable, therefore, in order to reduce the load on users' devices, it is necessary to resort to a client-server architecture. In this case, the user interface will be built in any convenient way. The form of the browser application was chosen because it gives access to a large number of libraries that simplify working with forms and graphs. Also, the system requirements of browser applications are much smaller and it is easier to support the update process, because the latest version will be loaded by the user on the go with each transition to the application. For a browser application, the JS language is the most widely used and supported solution.

Finally, it is necessary to meet the data storage requirements. There is a requirement that application should store calculation results for further comparison. For that purpose, it is enough to store data in the memory allocated by the browser. Saving and using after a long period of time, one can use the files obtained through the export function.

Data structures and API

Since the work implies the implementation of several approaches to determining the relationship, it is necessary to organize the basic data structures in such a way that the addition of a new method minimally affects the basic algorithms. This principle should be followed in both parts of the client-server application.

Backend

Within the library implementing Q-analysis methods, the algorithm is built in such a way that all calculations can be performed without referring to a specific implementation of the relation.

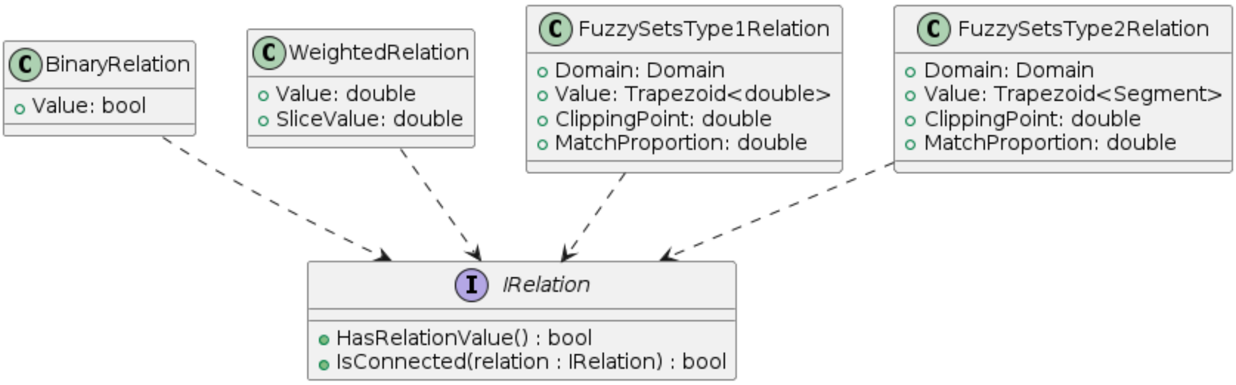

Figure 5. Relations Classes from backend.

Earlier in the "Proposed method" section, two main properties of the relationship were mentioned, which are used in calculations within the Q-analysis procedure:

- Checking for a relation by value in the corresponding cell of the matrix (prepared for system model). This functionality is contained in the function HasRelationValue() (Fig. 5).

- Checking the connectivity of two simplices. To do this, one needs to compare the values of the relations. This functionality is contained in the function IsConnected() (Fig. 5).

In all further algorithms (calculations of q-vector and eccentricities), the interface is already used, so if it is necessary to add a new type of relations, it is enough to create a new class, implement the interface IRelation and add the necessary additional parameters. Another standard is the use of the Value field to store a value, it is not fixed, because the type of value strongly depends on the way the relationship is described.

A similar approach is used for methods of calculating eccentricities. To add a new method, it is enough only to implement the interface IEccentricityCalculator (Fig. 6).

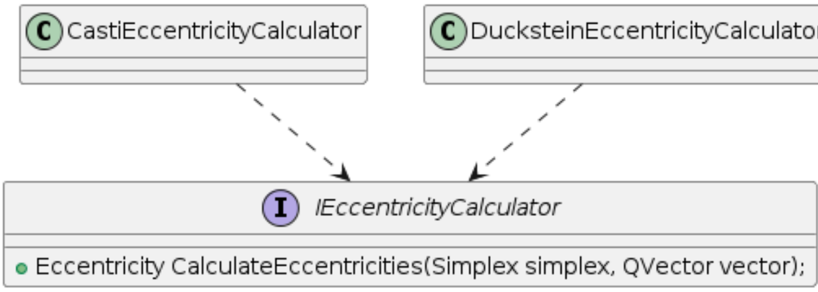

Figure 6. Interface for eccentricity calculator on backend.

But not only the library for calculations is engaged in data processing. The API is also important, where the types are separated. This is due, among other things, to the fact that at the stage of data preprocessing, it is necessary to perform other actions, not like in calculations (Fig. 7).

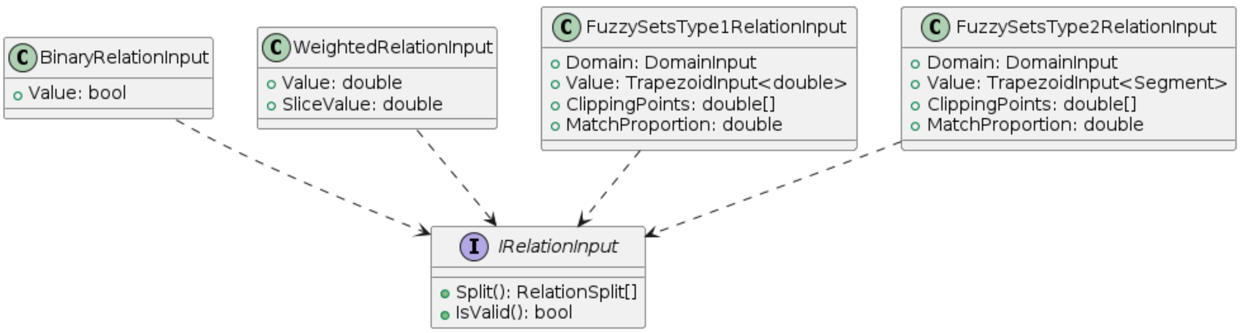

Figure 7. Interface for input data about relations on backend.

Since the API receives user data, validation should be performed according to the type of relationship. To do this, one need to implement the method IsValid(). In the cases of Fuzzy Sets Type 1 and Type 2 appeared the problem with the need for repeated calculation. In order to select the necessary sets of additional parameters from the received data and perform calculations for all their combinations, the function Split() was introduced.

The structure of the input data for the API method is designed in such a way that each relationship can be fully described, including all additional parameters necessary for calculations.

This makes it necessary to duplicate data, but at the same time allows one to reduce the necessary improvements when adding new implementations of relations.

Frontend

During development of the frontend, there was also a task to simplify as much as possible the subsequent addition of new types of relationships. But in this case, this is a more difficult aim, because it is necessary to provide a user-friendly and readable interface for the user, as well as add additional validations. All this requires working directly with the rendered interface, not just with the data type. Nevertheless, a common type is also implemented here (Fig. 8), which allows for a high level of versatility in many functions.

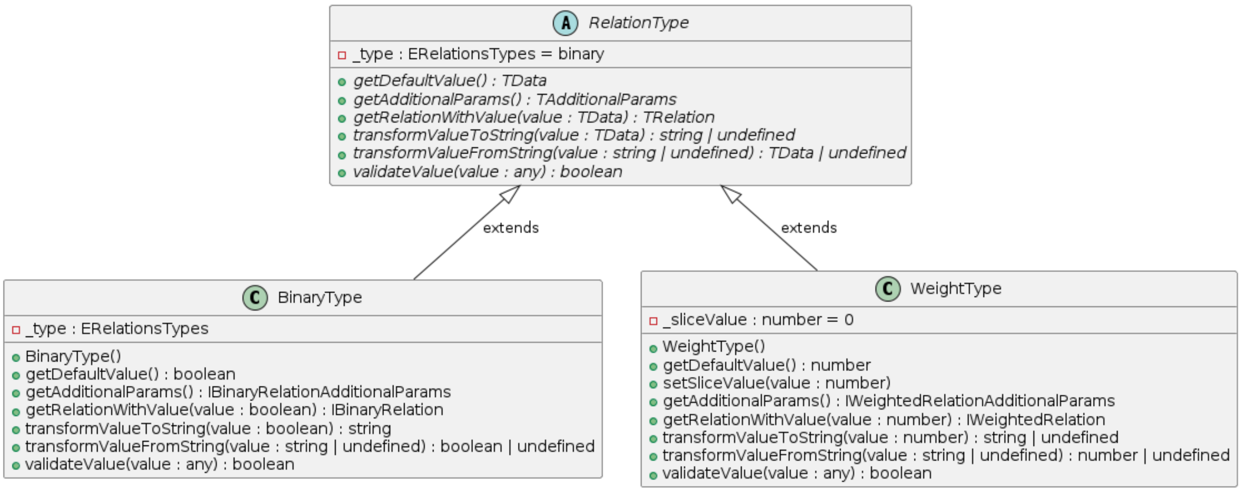

Figure 8.1. Relations classes from frontend Part 1.

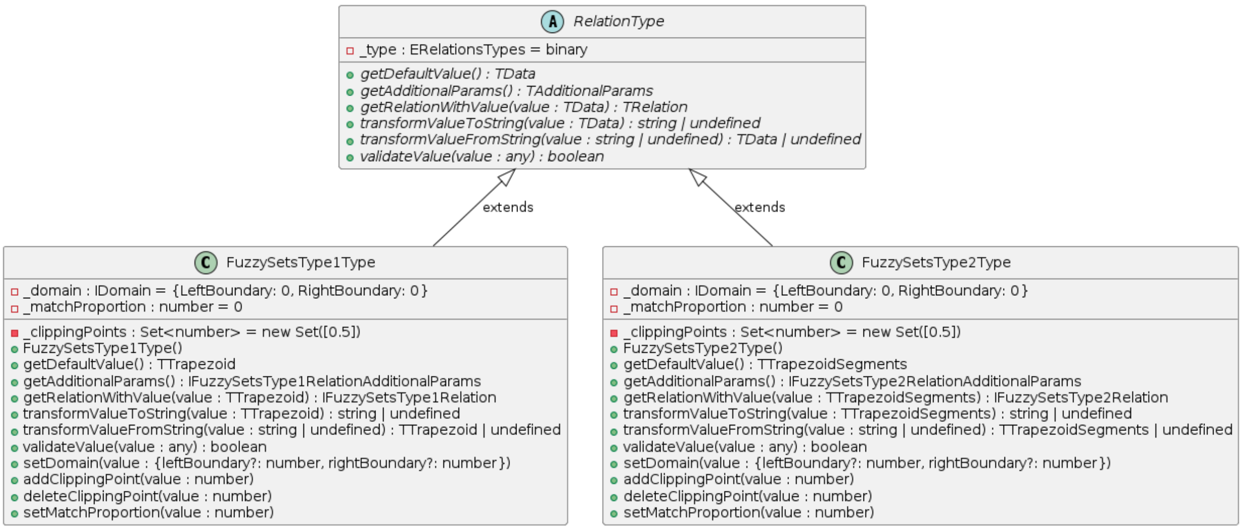

Figure 8.2. Relations classes from frontend Part 2.

The number of functions in this case is greater, since it is necessary to support a large number of functions:

- getDefaultValue: It is used to get the default value of the relationship, which most often corresponded to the absence of a connection.
- getAdditionalParams: It is used to get a complete list of relation-dependent additional parameters.
- getRelationWithValue: It is used to get structured object of the relation which will be sent to API for performing calculations.
- transformValueToString: It is used for transformation value of the relation to string. This function used in import/export functionality and helps to prepare data for excel table.
- transformValueFromString: It is used for parsing string to the value of the relation. This function used in import/export functionality and helps to prepare data for excel table.
- validateValue: It is used for values validation before import.

Program functionality

The ready-made program provides the user with a number of functions that make it comfortable to work with the Q-analysis procedure. The main page is divided into blocks responsible for a separate stage of preparation for calculations.

Eccentricity calculation approach selector

At this stage, the user has the opportunity to choose a method for calculating eccentricities (Fig. 9). It also implements the requirement that the application must contain information about the essence of the calculation method. To view the information, click on the button next to the name of the calculation method. The necessary information will be presented in the modal window (Fig. 10).

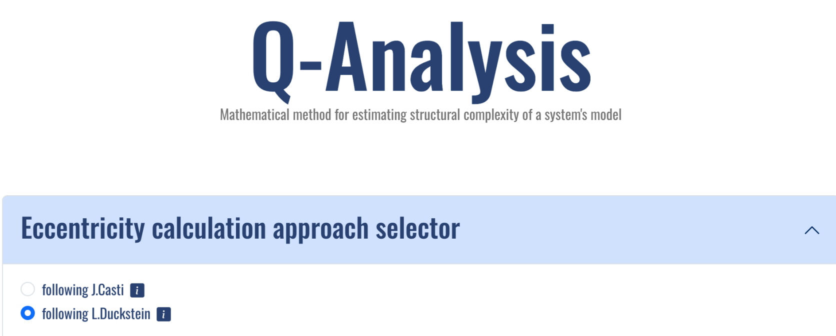

Figure 9. Eccentricity calculation approach selector block.

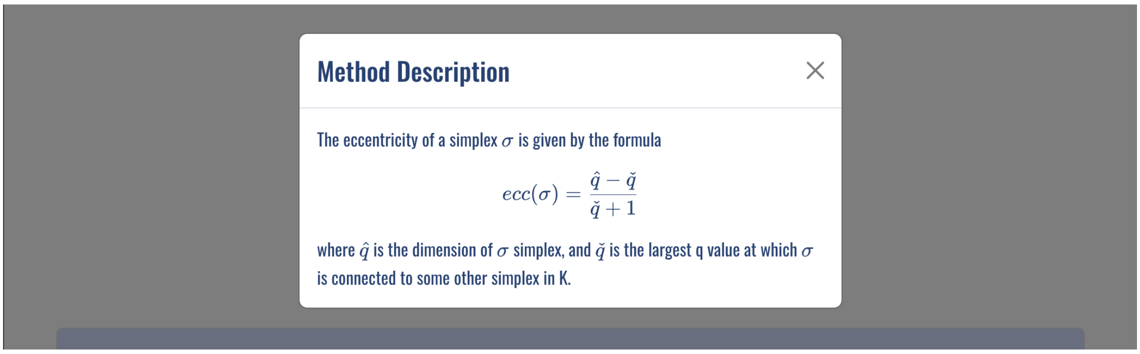

Figure 10. Information about eccentricity calculation method.

<u>Type of system's model selector</u>

At this stage, the user has to choose the type of relations. This step is significant for further logic because validation, data presentation and additional params list determined by this choice (Fig. 11).

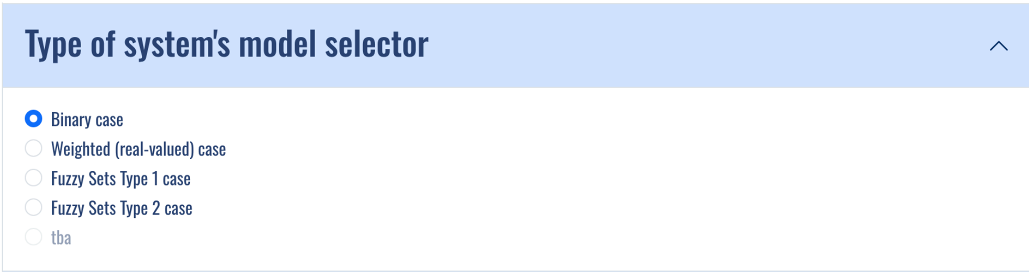

Figure 11. Type of system's model selector.

<u>Additional method params</u>

For some types of relations users should choose additional params. The list of these params is determined by the selection from the previous stage. Earlier the necessary parameters were already highlighted:

- Binary case: no additional params.
- Weighted (real-valued) case: slice value.
- Fuzzy Sets Type 1 case, Fuzzy Sets Type 2 case: domain (universe of discourse), clipping points, match proportion.

Restrictions are also applied to the fields for the input data and validation of the parameters takes place (Fig. 12).

Figure 12. Additional method params.

<u>System's model (matrix form)</u>

At this stage, the user can input data into matrix, import them from xlsx table. All kinds of values of the relations they have their own filling format (Table 6).

Table 6. Relations value.

Type	Value	Empty value
Binary case	number (from 0 to 1)	0
Weighted (real-valued)	number	0
Fuzzy Sets Type 1 case	[double; double; double, double]	null
Fuzzy Sets Type 2 case	[{double, double}; {double, double}; {double, double}; {double, double}]	null

In this block, when selecting Binary and Weighted relations, the user can work with the matrix, edit it (Fig. 13).

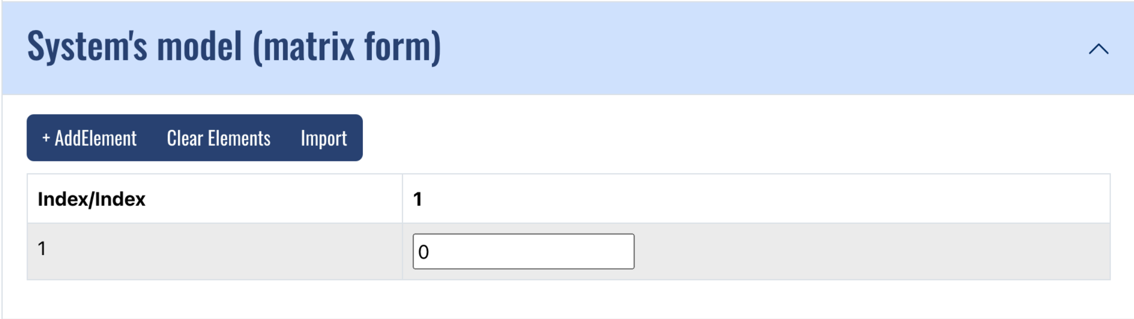

Figure 13. System’s model.

Users can enter the data manually but, also, they can import function (button Import). To open modal button “Import” should be used (Fig. 14.). The file is selected from the memory of the user's local computer. Validation is also provided in this modal before saving matrix.

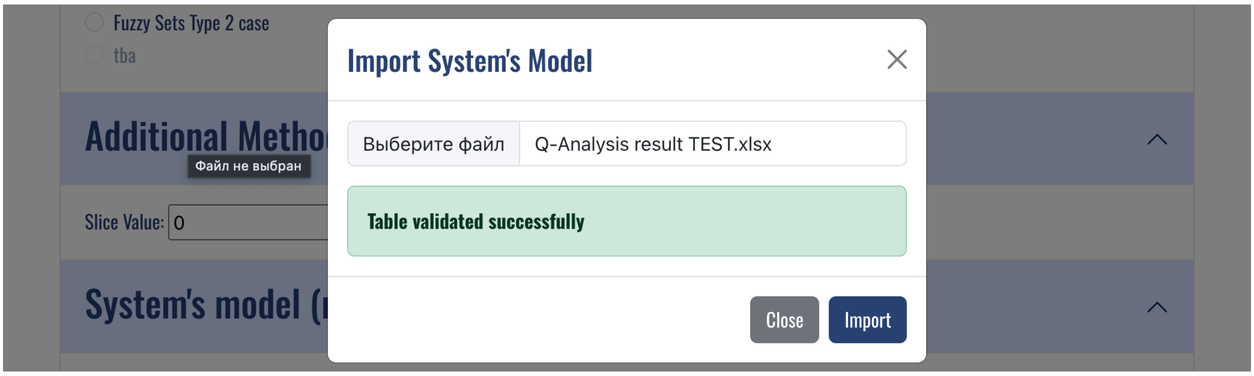

Figure 14. System’s model. Import.

There is an additional view for relations with Fuzzy Sets Type. Here users can look at the imported trapezoids (Fig. 15).

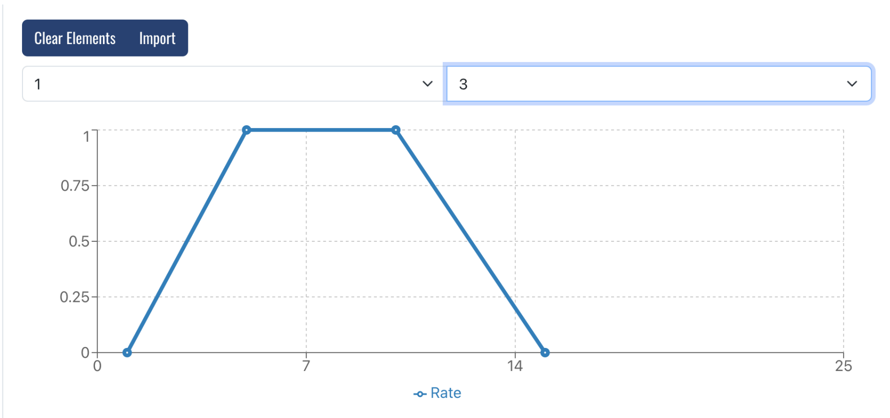

Figure 14. System’s model. Trapezoid.

Results

Finally, stage of the results. Here users can see the last result and the list of previous results (Fig. 15). For sending request button “Perform calculations” can be used. There are 4 additional buttons on each result card. “Show initial data” opens the modal with information about the structure of the system under consideration (Fig 16.). “Rename” used to rename result. By default, every result get it’s name from the time of receiving a message. But it can be inconvinient during further work with the result. Using “Rename” helps to highlight the result among others.

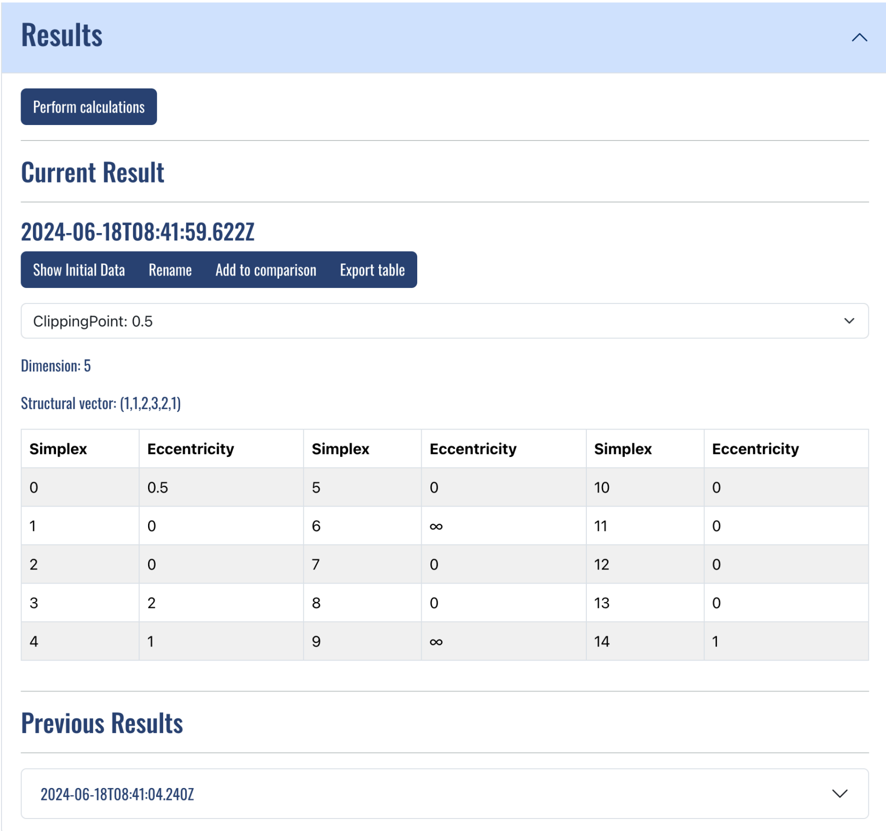

Figure 15. Results.

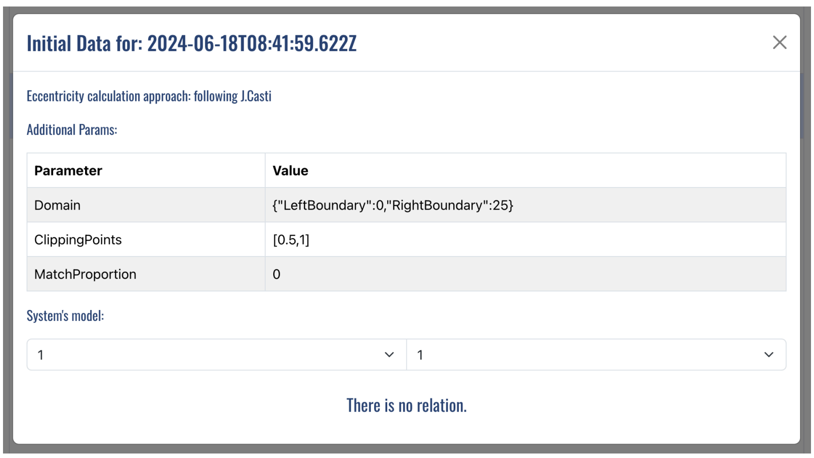

Figure 16. Results. Initial data.

“Export table” starts the process of transformation of the data to xlsx file. User will get a file with initial data (that can be imported once again), selected additional params and the results of the calculations. Finally, “Add to comparison” provide an ability to compare results using bar charts. Any number of results can be selected and displayed on charts at the same time (Fig. 17). There will be presented q-vector components (with connectivity components) and eccentricities values.

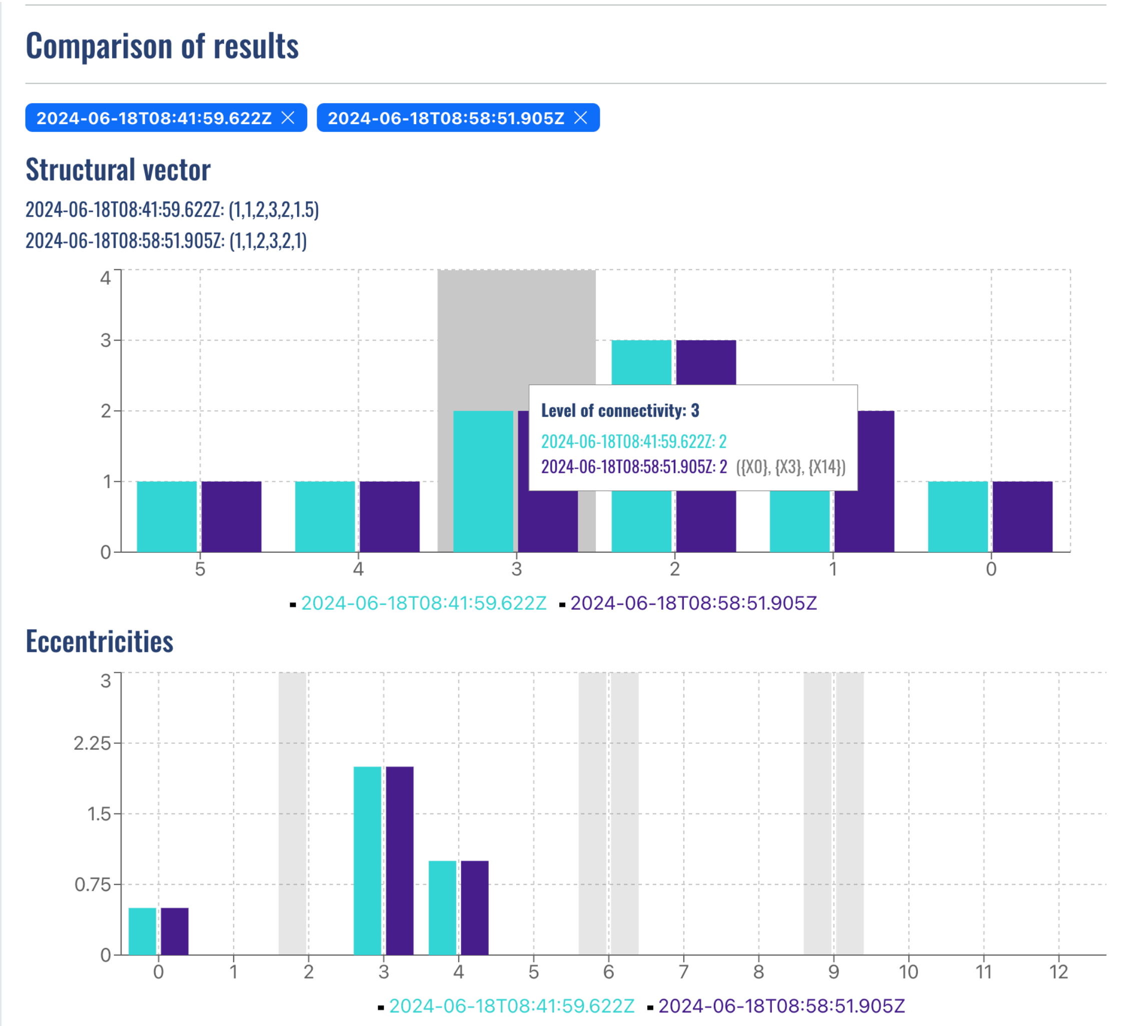

Figure 17. Results. Comparison of results.

CONCLUSION

The thesis successfully extends the Q-Analysis methodology to handle complex and uncertain relational data within system structures, providing a novel approach to model systems that involve fuzzy and weighted relationships. The implementation of the modified Q-Analysis procedure through software demonstrates its potential in providing detailed insights into the connectivity and interaction dynamics of system components. By integrating advanced mathematical tools and algorithms, this work allows analysts and system engineers to derive more profound and actionable insights from their structural analyses. The proposed methods contribute significantly to the field of systems analysis, offering a robust framework for understanding complex interactions in various domains where uncertainty and ambiguity are prevalent.

In future works, new types of fuzzy sets may be added and methods for converting the values of relations may be improved. The program can also be modified so that adding a new method to the user UI becomes even easier.

REFERENCES

[1] 'IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries'. IEEE. doi: 10.1109/IEEESTD.1991.106963.

[2] K. E. Wiegers and J. Beatty, *Software requirements*, Third edition. Redmond, Washington: Microsoft Press, s division of Microsoft Corporation, 2013.

[3] R. H. Atkin, 'From cohomology in physics to q-connectivity in social science', *Int. J. Man-Mach. Stud.*, vol. 4, no. 2, pp. 139–167, Apr. 1972, doi: 10.1016/S0020-7373(72)80029-4.

[4] K. Y. Degtiarev, 'Systems analysis: mathematical modeling and approach to structural complexity measure using polyhedral dynamics approach', *Complex. Int.*, vol. 7.

[5] J. L. Casti, *Connectivity, complexity, and catastrophe in large-scale systems*. in International series on applied systems analysis, no. 7. Chichester [Eng.] ; New York: J. Wiley, 1979.

[6] L. Duckstein and S. A. Nobe, 'Q-analysis for modeling and decision making', *Eur. J. Oper. Res.*, vol. 103, no. 3, pp. 411–425, Dec. 1997, doi: 10.1016/S0377-2217(97)00308-1.

[7] S. Tomasiello, W. Pedrycz, and V. Loia, *Contemporary Fuzzy Logic: A Perspective of Fuzzy Logic with Scilab*, vol. 1. in Big and Integrated Artificial Intelligence, vol. 1. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-98974-3.