AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING, AUTOMATICS, COMPUTER SCIENCE AND BIOMEDICAL ENGINEERING

DEPARTMENT OF APPLIED COMPUTER SCIENCE

Master of Science Thesis

Niespójności niekompletnych macierzy porównań parami
Inconsistency of incomplete pairwise comparisons matrices

Author: *Dawid Talaga*
Fielf od study: *Computer Science*
Thesis supervisor: *Konrad Kułakowski PhD*

Kraków, 2018

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystycznego wykonania albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.", a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.): „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej «sądem koleżeńskim».", oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Serdecznie dziękuję Promotorowi Panu dr. hab. Konradowi Kułakowskiemu za zainteresowanie mnie tematem PC oraz życzliwe wsparcie.
Dziękuję moim Rodzicom, dzięki którym mogłem rozwijać się i studiować oraz bliskim mi osobom, które dodawały mi wiary w moje siły.

Spis treści

1. Introduction

1.1. Pairwise Comparisons method

People have made decisions for ages. Some of them are very simple and come easily but others, more complicated, require deeper analysis. It happens when there are many compared objects, which are complex and the selection criterion is hard to measure precisely. Fortunately, the development of mathematics brought an interesting tool - *The Pairwise Comparisons (PC) Method*. The first case of using the method (in a very simple version) is the election system described by *Ramond Llull* [1] in the thirteenth century. Its rules were based on the fact that the candidates were pairwise compared with each other and the winner was the one who won in the largest number of direct comparisons. The method was reinvented in the eighteenth century by *Condorcet* and *Bord* [2] as they proposed it in their voting system. In the twentieth century, the method found the application in the theory of social choice, the main representatives of which were the Nobel prize winners *Keneth Arrow* [3] and *Amartya Sen* [4]. The current shape of the method was influenced by the changes introduced by *Fechner* and then refined by Thrustone [5]. However, the breakthrough was the introduction to the method The Analytic Hierarchy Process (AHP) by *Saaty* [6], which allowed to compare many more complex objects and create a hierarchical structure.The main aim of this paper is to check which method of calculating the inconsistency is the best in this case. I order to do it, a series of tests was carried out on various known inconsistency indexes, taking into account many different parameters: the matrix size, the amount of missing data and the level of inconsistency. The results of the research are included in this paper.

The PC method is based on the assumption that it is not worth comparing all objects at the same time. It is better to compare them in pairs and then gather the results together. Such pairwise comparisons are much more intuitive and natural for a human being. How can one be sure that these judgments are consistent? Or what to do if some comparisons are missing? In such case, is it worth taking the *PC method* at all?

The answer to the first question is the concept of inconsistency introduced into the method. This paper tries to answer the next two questions - meaning to examine whether available methods for determining inconsistencies give reliable results when a part of the comparisons is missing.

1.2. The aim of the work

The main aim of this paper is to check which method of calculating the inconsistency is the best when some comparisons are missing. In order to do it, a series of tests was carried out on various known inconsistency indexes, taking into account many different parameters: the matrix size, the amount of missing data and the level of inconsistency. The tests were implemented in R language, which properly fit into numerical calculations.

It was not know from the beginning what will be the result of this work. It was considered that one or more existing inconsistency indexes turn out appropriate also for incomplete matrices or the tests show that the inconsistency can not be calculated by these indexes. The results of the research are included below.

1.3. Contents of the work

The work includes the theoretical part and the description of conducted experiments and their results. It consists of six chapters.

The Second section shows the Pairwise Comparisons method and the problem of inconsistency. Understanding the basics is necessary to go into the further chapters.

In the third section sixteen available methods for calculating the inconsistency for the PC matrices are presented.

The fourth chapter shows ideas which allow to perform tests. These are modifications of existing inconsistency indexes which enable to adjust them to incomplete matrices and the algorithm which tests the quality of the modified indexes.

The results of experiments are presented and discussed in the fifth section.

The last chapter contains summary, conclusions and ideas for future researches.

2. Pairwise Comparisons method

2.1. PC matrix

The *PC method* is used to choose the best alternative from a set of concepts. However, this goal is achieved by comparing in pairs. A numerical value is assigned to each pair. It not only determines which alternative is preferred but also informs about the weight of this preference. In this way, the finite set of concepts $C = \{c_1, \ldots, c_n\}$ is transformed into a *PC matrix* $M = (m_{ij})$, where $m_{i,j} \in R$ and $i, j \in \{1, \ldots, n\}$. The *PC matrix* for n concepts is following:

$$M = \begin{pmatrix} 1 & m_{12} & \ldots & m_{1n} \\ m_{21} & 1 & \ldots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \ldots & 1 \end{pmatrix}$$

It is worth noting that the values m_{ij} and m_{ji} represent the same pair. Therefore, one should expect that $m_{ji} = \frac{1}{m_{ij}}$. If

$$\forall i, j \in \{1, \ldots, n\} : m_{ij} = \frac{1}{m_{ji}}, \tag{2.1}$$

the matrix is called a *reciprocal*.

2.2. Weight vector

The PC matrix is the basis for calculating the method. It is used in a function $\mu : C \to R$ that assigns a positive real number to each alternative in the set C. The vector

$$\mu = [\mu(c_1), \ldots, \mu(c_n)]$$

formed in this way is called *the weight vector* or *the priority vector* (see Fig. 2.1.). It informs which alternative has won.

There are many ways to calculate the vector μ. Among the popular ones are the method using the matrix's eigenvalues or the method based on geometric means [7].

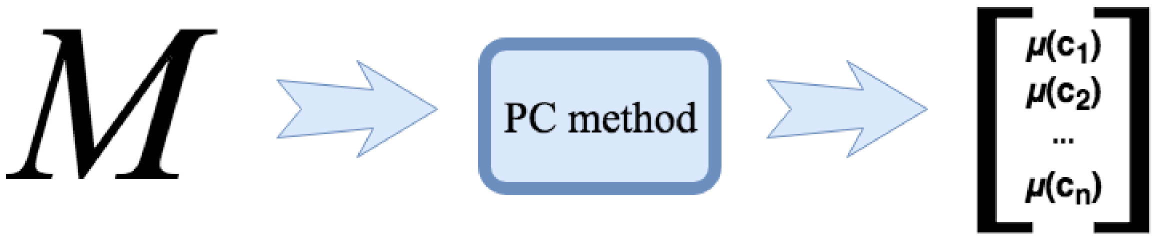


Fig. 2.1. Diagram of calculating *the weight vector*

2.3. Inconsistency

The second important parameter describing the *PC matrix* is *consistency*. The matrix is consistent if

$$\forall i, j, k \in \{1, \ldots, n\} : m_{ik} = m_{ij} m_{jk}. \tag{2.2}$$

Three numbers that should meet this assumption are called a *triad*.

If the matrix is consistent and the weight vector μ is computed, then for each of the variables i, j, where $1 \leq i, j \leq n$ meets the equation:

$$ij = \frac{\mu_i}{\mu_j}. \tag{2.3}$$

In practice, it is very rare for a matrix M to be completely consistent. In the long history of the PC method, many methods have been developed to calculate inconsistencies. Many of them are based directly on the definition of consistency ([eq:consistent]), some methods use the eigenvalues of the matrix, others are based on the assumption that each fully consistent matrix fulfills the condition ([eq:consistent2]).

3. Inconsistency indexes

3.1. Inconsistency indexes for complete matrices

This subsection presents sixteen common inconsistency indexes. Their detailed description, including the formulas, is necessary to modify them in the next step so that they can also work for incomplete matrices. Many of them have been described and tested numerically in article [8]. In all methods, it is assumed that the *PC matrix* is reciprocal.

3.1.1. Saaty index

This is one of the most important and popular indexes and was introduced by *Saaty* [9]. In order to determine inconsistency, the matrix's eigenvalues should be computed. The author used the dependence that the largest eigenvalue of the matrix is equal to its dimension if and only if the given matrix is completely consistent. On this assumption, he based his thoughts and proposed the formula:

$$CI(A) = \frac{\lambda_{max} - n}{n-1}, \tag{3.1}$$

where λ_{max} is the principal eigenvalue of the PC matrix and n is its dimension.

3.1.2. Geometric consistency index

This index on the assumption ([eq:consistent2]) was proposed by *Craford* and *Williams* [10] and then refined by *Aguaròn* and *Moreno-Jimènez* [11]. In this case the priority vector should be calculated using the geometric mean method. Consider ([eq:consistent2]) one can create a matrix:

$$E = \left[e_{ij} \mid e_{ij} = a_{ij}\frac{w_j}{w_i} \right], \quad i,j = 1, ..., n. \tag{3.2}$$

The inconsistency index is calculated as follows:

$$GCI = \frac{2}{(n-1)(n-2)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} ln^2 e_{ij}. \tag{3.3}$$

3.1.3. Koczkodaj index

One of the most popular inconsistency indexes was proposed by *Koczkodaj* [12]. It is based directly on the definition of consistency ([eq:consistent]). The value of the inconsistency index for one triad ([triad]) was defined as:

$$K_{i,j,k} = min\{\frac{1}{a_{ij}} \mid a_{ij} - \frac{a_{ik}}{a_{jk}} \mid, \frac{1}{a_{ij}} \mid a_{ik} - a_{ij}a_{jk} \mid, \frac{1}{a_{jk}} \mid a_{jk} - \frac{a_{ik}}{a_{ij}} \mid\}. \tag{3.4}$$

This formula has been simplified by Duszak and Koczkodaj [13] and is given as:

$$K(\alpha, \beta, \gamma) = min\{\mid 1 - \frac{\beta}{\alpha\gamma} \mid, \mid 1 - \frac{\alpha\gamma}{\beta} \mid\}, \quad where\, \alpha = a_{ij}, \beta = a_{ik}, \gamma = a_{jk} \tag{3.5}$$

Then it was generalized [13] for $n > 2$. Finally, the inconsistency index has the following form:

$$K = max\{K(\alpha, \beta, \gamma) | 1 \leq i < j < k \leq n\}. \tag{3.6}$$

It is worth noting that not only does the coefficient find the greatest inconsistency but also indicates the place in which it occurs.

3.1.4. Kazibudzki indexes

Based on the Koczkodaj inconsistency index and observation that $ln(\frac{\alpha\gamma}{\beta}) = -ln(\frac{\beta}{\alpha\gamma})$, *Kazibudzki* proposed several additional inconsistency indexes [14]. Instead of the formula for inconsistency of the triad [eq:k-abg], he introduced two new formulas:

$$LTI(\alpha, \beta\gamma) = \mid ln(\frac{\alpha\gamma}{\beta}) \mid, \tag{3.7}$$

$$LTI*(\alpha, \beta\gamma) = ln^2(\frac{\alpha\gamma}{\beta}). \tag{3.8}$$

Based on the above equations, *Kazibudzki* proposed new indexes. The simplest ones use the geometric mean of the triads. Thus, new indexes could be written in the form:

$$MLTI(LTI) = \frac{1}{n} \sum_{i=1}^{n} [LTI_i(\alpha, \beta\gamma)], \tag{3.9}$$

$$MLTI(LTI*) = \frac{1}{n} \sum_{i=1}^{n} [LTI*_i(\alpha, \beta\gamma)]. \tag{3.10}$$

After further research *Kazibudzki* introduces another inconsistency index [15], again based on ([eq:lti*]). It was defined as $CM(LTI*) = \frac{MEAN[LTI*(\alpha,\beta,\gamma)]}{1+MAX[LTI*(\alpha,\beta,\gamma)]}$. Hence,

$$CM(LTI*) = \frac{\frac{1}{n}\sum_{i=1}^{n}[LTI*_i(\alpha, \beta, \gamma)]}{1 + max\{LTI*_i(\alpha, \beta, \gamma)\}}. \tag{3.11}$$

3.1.5. Index of determinants

This index was proposed by *Pelaez* and *Lamata* [16] and is also based on the concept of triad. The authors noticed that *PCM matrices* can be constructed on the basis of triads. Their determinant is closely related to the consistency of the matrix.

For every triad (a_{ik}, a_{ij}, a_{jk}) one can build a matrix in the form:

$$T_{ijk} = \begin{pmatrix} 1 & a_{ij} & a_{ik} \\ \frac{1}{a_{ij}} & 1 & a_{jk} \\ \frac{1}{a_{ik}} & \frac{1}{a_{jk}} & 1 \end{pmatrix}, \quad where\, i < j < k. \tag{3.12}$$

The determinant of this matrix is:

$$det(A) = \frac{a_{ik}}{a_{ij}a_{jk}} + \frac{a_{ij}a_{jk}}{a_{ik}} - 2. \tag{3.13}$$

If the matrix is fully consistent, then $det(A) = 0$, else $det(A) > 0$. Based on the above considerations, the authors introduced the new inconsistency index that can be formulated as follows:

$$CI* = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{a_{ik}}{a_{ij}a_{jk}} + \frac{a_{ij}a_{jk}}{a_{ik}} - 2\right). \tag{3.14}$$

3.1.6. Kułakowski and Szybowski indexes

Kułakowski and *Szybowski* proposed two further inconsistency indexes [17], which are also based on triads. They use the fact that the number of triads that can be found in a *PCM matrix* is:

$$\binom{n}{3} = \frac{n!}{(n-3)!3!} = \frac{n(n-1)(n-2)}{6}. \tag{3.15}$$

The index is formulated as follows:

$$I_1 = \frac{6\sum_{t\in T} K(t)}{n(n-1)(n-2)}, \tag{3.16}$$

where $K(t)$ is the Koczkodaj index for triad $t = (\alpha, \beta, \gamma)$ of the set of all triads T.

The second inconsistency index is similar:

$$I_2 = \frac{6\sqrt{\sum_{t\in T} K^2(t)}}{n(n-1)(n-2)}. \tag{3.17}$$

Indexes can be combined with each other to create new coefficients. In this way *Kułakowski* and *Szybowski* proposed two new indexes. The first one is based on ([eq:K]) and ([eq:I1]). This index allows to choose which value should have more impact on the result: the greatest inconsistency found or the average inconsistency of all triads. The new inconsistency index looks as follows:

$$I_\alpha = \alpha K + (1-\alpha) I_1, \tag{3.18}$$

where $0 \leq \alpha \leq 1$.

The second index expands the first one by ([eq:I2]):

$$I_{\alpha,\beta} = \alpha K + \beta I_1 + (1-\alpha-\beta) I_2. \tag{3.19}$$

3.1.7. Harmonic consistency index

This index was introduced by *Stein* and *Mizzi* and it presents a completely new method of inconsistency computing [18]. At the beginning it requires the creation of an auxiliary vector $s = (s_1, ..., s_n)^T$, where n is the dimension of the matrix A, for which the index will be calculated. Each element of the vector s is the sum of values in one column of the matrix A. Hence,

$$s_j = \sum_{i=1}^{n} a_{ji} \quad \forall j. \tag{3.20}$$

The authors proved that if the matrix A is consistent, then $\sum_{j=1}^{n} s_j^{-1} = 1$. The formula for the mean harmonic looks as follows (Brunelli, 2015):

$$HM = \frac{n}{\sum_{j=1}^{n} \frac{1}{s_j}}. \tag{3.21}$$

The final formula for inconsistency index was obtained by normalizing the above equation ([eq:hm']):

$$HCI = \frac{(HM(s) - n)(n+1)}{n(n-1)}. \tag{3.22}$$

3.1.8. Golden and Wang index

This index was introduced by *Golden* and *Wang* [19]. It assumes that the priority vector was calculated using the geometric mean method, then normalized to add up to 1. In this way vector $g* = [g^*_{1,}, ..., g^*_n]$ was obtained, where n is the dimension of the matrix A. The next step is to normalize each column of the matrix A. After this, the sum of the elements of each column in matrix A is 1. The obtained matrix is marked with the symbol A^*. The inconsistency index is defined as follows:

$$GW = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \mid a^*_{ij} - g^*_i \mid . \tag{3.23}$$

3.1.9. Salo and Hamalainen index

The index proposed by *Salo* and *Hamalainen* [20] uses the definition of inconsistency ([eq:consistent]), however it requires the creation of an auxiliary matrix, in which each element contains the smallest and largest discrepancy from consistency based on the formula ([eq:consistent]). The index takes all triads into account:

$$R = (r_{ij})_{nxn} = \begin{pmatrix} [\underline{r}_{11}, \overline{r}_{11}] & \dots & [\underline{r}_{1n}, \overline{r}_{1n}] \\ \vdots & \ddots & \vdots \\ [\underline{r}_{n1}, \overline{r}_{n1}] & \dots & [\underline{r}_{nn}, \overline{r}_{nn}] \end{pmatrix}, \tag{3.24}$$

where $\underline{r_{ij}} = min\left\{a_{ik}a_{kj} \mid k = 1, \dots, n\right\}$, $\overline{r_{ij}} = max\left\{a_{ik}a_{kj} \mid k = 1, \dots, n\right\}$ and n is the dimension of the tested matrix A. A numerical example was presented in [21]. Based on the resulting matrix R, the authors proposed the following inconsistency index:

$$CM = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\overline{r}_{ij} - \underline{r}_{ij}}{\left(1 + \overline{r}_{ij}\right)\left(1 + \underline{r}_{ij}\right)}. \tag{3.25}$$

3.1.10. Cavallo and D'Apuzzo index

The authors *Cavallo* and *D'Apuzzo* based their index on triads but they conducted studies on a new path, generalizing them for linear, ordered abelian groups ([22], [23]). Thanks to this, the index can be used also with other relations [8]. Index for relation max can be presented in the form of a formula:

$$I_{CD} = \prod_{i=1}^{n-2} \prod_{j=i+1}^{n-2} \prod_{k=j+1}^{n} \left(max \left\{ \frac{a_{ik}}{a_{ij}a_{jk}}, \frac{a_{ij}a_{jk}}{a_{ik}} \right\} \right)^{\frac{1}{\binom{n}{3}}}. \tag{3.26}$$

3.1.11. Relative error

This index, proposed by *Barzaili* [24], requires calculation of the weight vector using the arithmetic mean method for each row and creation of two additional matrices. Thus, the weight vector is

$$w_i = \frac{1}{n} \sum_{j=1}^{n} a_{ij},$$

where n is the dimension of the matrix. The two auxiliary matrices are calculated according to the formulas:

$$C = (c_{ij}) = (w_i - w_j)$$

$$E = (e_{ij}) = (a_{ij} - c_{ij})$$

Ultimately, the formula for the *Relative error* is following:

$$RE(A) = \frac{\sum_{ij} e_{ij}^2}{\sum_{ij} a_{ij}^2}. \tag{3.27}$$

3.2. Inconsistency indexes for incomplete matrices

There are no inconsistency indexes for incomplete matrices. However, Those presented in chapter (3) could be used in such cases. It requires usually a slight modification of the index definition or calculation only for selected data. The ways in which the examined indexes have been adjusted to be able to deal with incomplete matrices are presented below.

Saaty index

The input matrix is modified using the method proposed by *Harker* [25]. It means that values $c+1$,where c is the number of non-empty elements in a given row, are placed on the diagonal.

Geometric consistency index

During calculating the weight vector by the geometric mean, empty values are omitted. Additionally, in the formula ([eq:GCI]) only non-empty elements e_{ij} are used. The reason for this exclusion is that the domain of the logarithmic function is R^+.

Koczkodaj index, Kazibudzki indexes, Index of determinants:

Only those triads which do not contain empty values are taken into account.

Kułakowski and Szybowski indexes

Only those triads which do not contain empty values are taken into account. In addition, the number of triads is no longer calculated according to the formula ([eq:KulSzynPo3]) but determined directly by counting the number of triads.

Harmonic consistency index:

No modification.

Golden and Wang index:

During calculating the weight vector by the geometric mean empty values are omitted.

Salo and Hamalainen:

No modification.

Cavallo and D'Appuzo:

During calculating the product ([eq:CavDAp]) empty values are omitted.

Relative index:

No modification.

Proposed methods of adjustments of indexes allow to apply them in incomplete matrices. However, them do not guarantee the best results. It means that further experiments are required. The existing indexes can be modified in many different ways. In this paper one put emphasis on the fact that the adjustments were slight. The aim of this study is to examine existing indexes, not to create new indexes.

4. Studies of inconsistency indexes for incomplete matrices

The presented inconsistency indexes were tested utilizing the Monte Carlo method. Their goal was to select those indexes which will give reliable results for incomplete matrices. Therefore, it was decided that the measure of the indexes' quality would be the *relative error* (expressed as percentage), which took into account the value of the index for a full, inconsistent matrix and the value of the index for the same matrix after partial decomposition. To be sure that the results were fair, all indexes were tested on the same set of matrices. The different sizes of the matrices, the levels of incompleteness and the levels of inconsistency were taken into account. Then, in order to compare the indexes easily and to select the best ones, the results were averaged using the arithmetic mean. The article [15] was used while building the algorithm to solve the problem.

4.1. Algorithm

4.1.1. Steps of the algorithm

Procedure steps:

1. Randomly generate a vector $w = [w_1, ..., w_n]$ and a consistent *PCM matrix* associated with it $PCM = (m_{ij})$, where $m_{ij} = \frac{w_i}{w_j}$.
2. Disrupt the matrix by multiplying its elements (excluding the diagonal) by the value of d, randomly selected from the range $\left(\frac{1}{x}, x\right)$.
3. Replace the values m_{ij}, where $i < j$ by the values m_{ji}.
4. Calculate the values of index with all the methods for the created matrix.
5. Remove some of the values from the matrix. The level of incompleteness should be $g\%$.
6. Calculate the values of inconsistencies by all methods for the incomplete matrix.
7. Calculate the relative error for each index.
8. Repeat steps 1 to 7 X_1 times.
9. Calculate the average relative error for each inconsistency index for the *PCM matrix*.

10. Repeat steps 1 to 9 X_2 times.

11. Calculate the average relative error for each index by averaging the values obtained in step 9.

4.1.2. Details of the algorithm

The above algorithm was carried out for values $X_1 = 100$, $X_2 = 100$. Tests were started for values d the range $(1.1, 1.2, ..., 4)$ and then the results were averaged. It means that the average relative error of one index was calculated on the basis of 4000 matrices, each of which decomposed randomly 100 times. It gave together 400000 tests on how good the index was.
In addition, tests were carried out for various sizes of matrices.

The results are divided into two parts:

1. A constant degree of incompleteness, different size of the matrix.

2. Different degrees of incompleteness, constant size of the matrix.

The aim of a such division is to pay attention to how the inconsistency indexes behave when the size of the matrix and the degree of incompleteness are changing. The results of the research are presented below.

4.2. Implementation

4.2.1. Development environment

Tests of indexes were developed in R language which is appropriate for numerical calculations (see [26]). It contains dozens of functions which support operations on matrices and vectors. Integrated development environment (IDE) called RStudio (see [27]) was used during implementation. This tool allows to create own packages which contain not only code but also documentation and information about licence and author. Package named *indexesForIncomplete* has been created. The most important part of this package is the file *indexes.R* which performs calculations necessary to test the indexes. RStudio supports programmer's work by syntax highlighting, built-in console, easy documentation searching and many others. The program is available on common operating systems. Before using RStudio one has to install R programming language (see [28]).

4.2.2. Implementation of tests of the inconsistency indexes

Implementation of the tests of inconsistency indexes consists of two steps:

1. Implementation of functions which calculate inconsistency indexes for a given matrix (full or incomplete).

2. Implementation of functions which study indexes for different matrices and collect all the results of these tests.

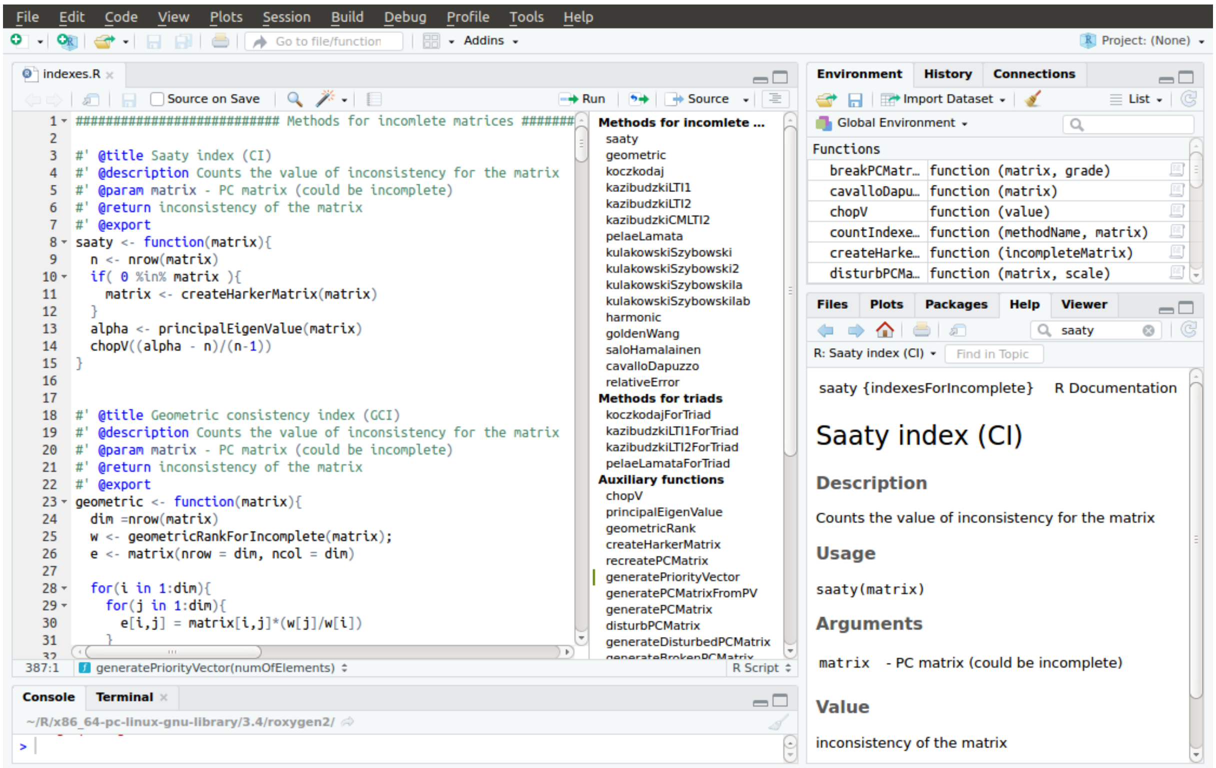


Fig. 4.1. Program *RStudio*

4.2.2.1. Implementation of the inconsistency indexes

Sixteen functions which calculate inconsistency indexes using methods described in chapter 3. The functions were written in such a way that allows handling both full and incomplete matrices. One has not taking account the wrong matrices, it means nonreciprocal or inconsistent PC matrices. Each of these function has only one parameter - PC matrix. Exceptions are the two methods implementing Kulakowski and Szybowski index which additionally takes parameters α, β. The result of each function is value of inconsistency index. The functions were extended by comments which informs about name of index related to given function, parameters and returned value. It allows to easily read and modify the code. Several examples of the functions are presented below.

It is worth drawing attention to function which is called within the functions intended for indexes based on triads. This function generates triads from a matrix and next returns inconsistency for each o them. However, the way to calculate inconsistency for one triad depends on first function parameter. It informs about function name that calculates inconsistency of a triad specified method.

4.2.2.2. Implementation of tests

In the second step, functions calculating the quality of the indexes for incomplete matrices, were created. Functions, which generate specified matrices, play an important role. PC matrices are created depending on the size, the level of inconsistency and the degree of incompleteness.

```r
#' @title Saaty index (CI)
#' @description Counts the value of inconsistency for the matrix
#' @param matrix - PC matrix (could be incomplete)
#' @return inconsistency of the matrix
#' @export
saaty <- function(matrix){
  n <- nrow(matrix)
  if( 0 %in% matrix ){
    matrix <- createHarkerMatrix(matrix)
  }
  alpha <- principalEigenValue(matrix)
  chopV((alpha - n)/(n-1))
}
```

Fig. 4.2. The implementation of *Saaty* index

```r
#' @title Koczkodaj matrix inconsistency
#' @description Counts the value of inconsistency for the matrix
#' @param matrix - PC matrix (could be incomplete)
#' @return inconsistency of the matrix
#' @export
koczkodaj <- function(matrix){
  triadsAndIdxs <- countIndexesForTriads("koczkodajForTriad", matrix)
  chopV(max(triadsAndIdxs))
}
```

Fig. 4.3. The implementation of *Koczkodaj* index

The last part of the functions relates to testing how big relative error occurs for inconsistency indexes after deleting some of the values. To begin with functions, which test one index. They consider matrix size, the level of inconsistency, the degree of incompleteness and the number of attempts which are performed for given matrix.

Then functions, which perform tests for each index, were implemented based on the same set of matrices. Thus, the results are reliable and each index is considered the same way.

4.2.3. Documentation

Comments in code were used to generate the documentation. The package *roxygen2* were made for this purpose.

Exemplary portions of the documentation are presented below.

```r
#' @title Kulakowski and Szybowski consistency index (Ia)
#' @description Counts the value of inconsistency for the matrix
#' @param matrix - PC matrix (could be incomplete)
#' @return inconsistency of the matrix
#' @export
kulakowskiSzybowskiIa <- function(matrix, alfa, beta=0){
  triadsAndIdxs <- countIndexesForTriads("koczkodajForTriad", matrix)
  chopV(alfa*max(triadsAndIdxs) + (1-alfa)*mean(triadsAndIdxs))
}
```

Fig. 4.4. The implementation of *Kulakowski and Szybowski* index

```r
#' @title Salo and Hamalainen consistency index (SH)
#' @description Counts the value of inconsistency for the matrix
#' @param matrix - PC matrix (could be incomplete)
#' @return inconsistency of the matrix
#' @export
saloHamalainen <- function(matrix){
  n <- nrow(matrix)
  sum <- 0

  rMin <- matrix(nrow = n, ncol = n, data = 0)
  rMax <- matrix(nrow = n, ncol = n, data = 0)

  for(i in 1:n){
    for(j in 1:n){
      c <- rep(0,n)
      for(k in 1:n){
        c[k] = matrix[i,k]*matrix[k,j]
      }
      rMin[i,j] = min(c)
      rMax[i,j] = max(c)
    }
  }

  for(i in 1:(n-1)){
    for(j in (i+1):n){
      sum <- sum + (rMax[i,j]-rMin[i,j])/((1+rMax[i,j])*(1+rMin[i,j]))
    }
  }

  cm <- 2/(n*(n-1))*sum
  chopV(cm)
}
```

Fig. 4.5. The implementation of *Salo and Hamalainen* index

```
#' @title Generates inconsistency indexes for each triad from the matrix
#' @param methodName - name of the method which computes inconsistency index for triad
#' @param matrix - PC matrix
#' @return vector of inconsistency indexes for triads
countIndexesForTriads <- function(methodName, matrix){
  triads <- generateTriads(matrix)
  if(is.null(dim(triads)) && length(triads) == 3){
    triads <- matrix(triads, 3, 1)
  }
  triadIdxs <- apply(triads, 2, methodName)
  triadIdxs
}
```

Fig. 4.6. The implementation of method *countIndexesForTriads*, which calculates inconsistency for each triad of a specified matrix

```
#' @title Generates PC Matrix on the basis of the size
#' @param numOfElements - number of elements of the matrix
#' @return reciprocal PC matrix
generatePCMatrix<- function(numOfElements){
  priorityVector <- generatePriorityVector(numOfElements)
  generatePCMatrixFromPV(priorityVector)
}
```

Fig. 4.7. The implementation of function *generatePCMatrix* which generates the PC matrix depending on matrix size

```
#' @title Disturbs the PC matrix
#' @description Disturbs the PC matrix in order to obtain inconsistency
#' @param matrix - PC matrix
#' @param scale - extend of disorders. This parametr is the upper limit
#' of the interval that is used to scale the elements. The lower limit
#' is defined as 1 / scale
#' @return disturbed PC matrix
disturbPCMatrix<- function(matrix, scale){
  dim = nrow(matrix)

  for(r in 1:dim-1)
    for(c in (r+1):dim)
      matrix[r,c] <- runif(1, min = 1/scale, max = scale) * matrix[r,c]

    diag(matrix) <- 1
    recreatePCMatrix(matrix)
}
```

Fig. 4.8. The implementation of function *disturbPCMatrix* which disturbs the PC matrix regarding inconsistency depending on given level of inconsistency

```r
#' @title Breaks PC Matrix
#' @description Breaks PC Matrix to obtain reciprocal, incomplete matrix
#' @param  matrix - PC matrix
#' @param grade - percentage of the value to be removed (not applicable to the diagonal)
#' @return incomplete PC matrix
breakPCMatrix<- function(matrix, grade){

  dim <- nrow(matrix)
  numOfEmptyElements <- grade*0.01*(dim*(dim-1))

  while(numOfEmptyElements > 1){
    rowToClear <- sample(1:dim, 1)
    colToClear <- sample(c(1:dim)[-rowToClear], 1)

    if(matrix[rowToClear, colToClear] != 0) {
      matrix[rowToClear, colToClear] <- matrix[colToClear, rowToClear] <- 0
      numOfEmptyElements <- numOfEmptyElements - 2
    }
  }

  matrix

}
```

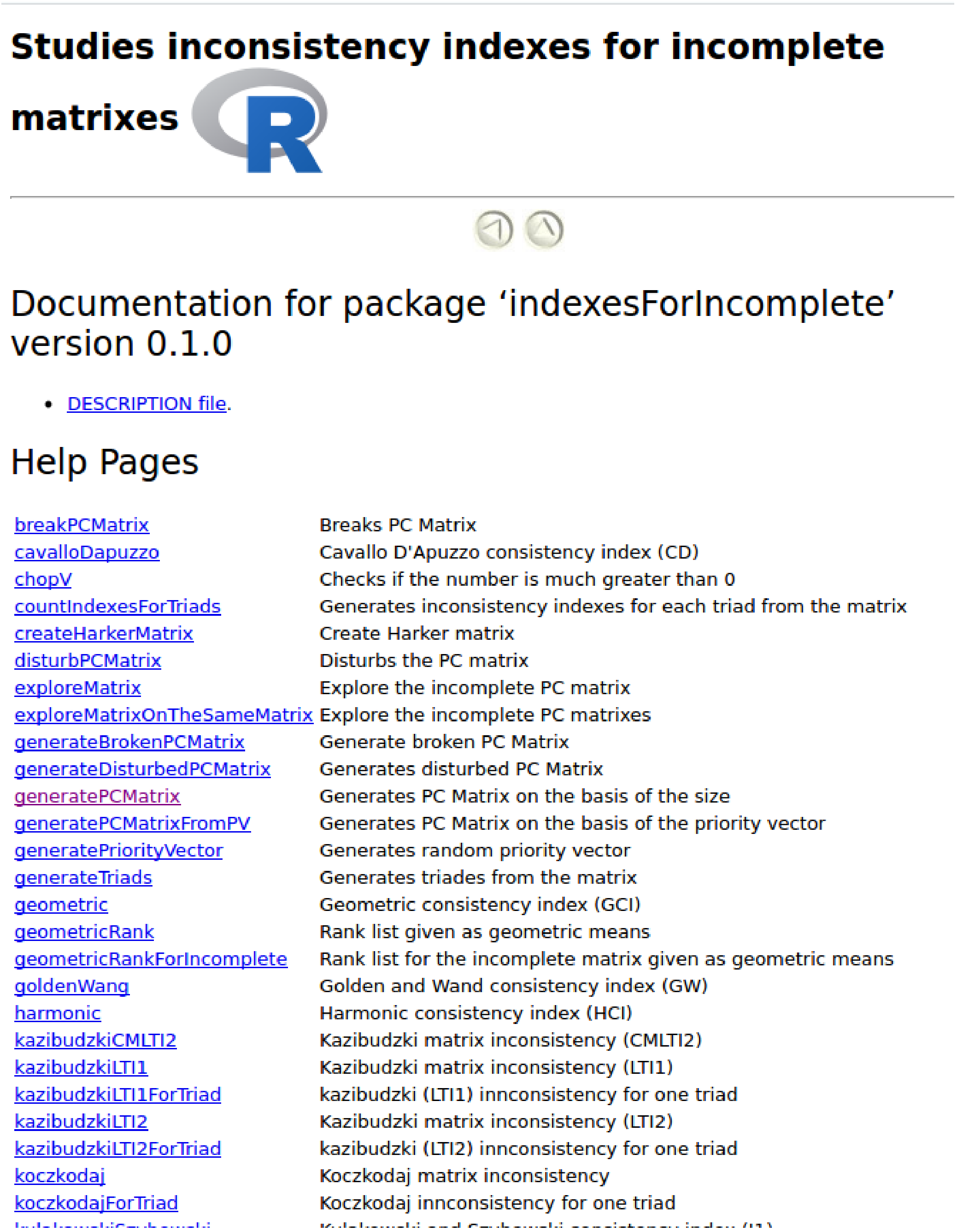
Fig. 4.9. The implementation of function *breakPCMatrix* which breaks up the PC matrix regarding incompleteness depending on given degree of incompleteness

```
#' @title Explore the incomplete PC matrix
#' @description Examines what is the relative error between the full matrix and indomplete matrix
#' @param methodName - a name of the method which is tested
#' @param scale - extend of disorders. This parametr is the upper limit of the interval that is used
#' to scale the elements. The lower limit is defined as 1 / scale
#' @param numOfElements - dimension of tested matrix
#' @param gradeOfIncomplete -  percentage of the value to be removed (not applicable to the diagonal)
#' @param numOfAttempts - number of test cases
#' @param alfa - a parameter for kulakowskiSzybowskiIa and kulakowskiSzybowskiIab method
#' @param beta - a parameter for kulakowskiSzybowskiIab method
#' @return average value of the relative error between the full matrix and indomplete matrix
exploreMatrix <- function(methodName, scale, numOfElements, gradeOfIncomplete, numOfAttempts, alfa=0, beta=0) {

  matrix <- generateDisturbedPCMatrix(numOfElements, scale)
  n <- ncol(matrix)
  if(alfa==0 && beta==0){
    realIdx <- methodName(matrix)
  } else {
    realIdx <- methodName(matrix, alfa, beta)
  }

  vectorOfIdsx <- integer(numOfAttempts)

  for( i in 1:numOfAttempts ) {
    brokenMatrix <- matrix(nrow = n, ncol = n, data = 0)
    numOfTriads <- length(generateTriads(brokenMatrix))/3

    while( (0 %in% (matrix %^% (n-1))) || numOfTriads==0){
      brokenMatrix <- breakPCMatrix(matrix, gradeOfIncomplete)
      numOfTriads <- length(generateTriads(brokenMatrix))/3
    }

    if(alfa==0 && beta==0){
      idx <- methodName(brokenMatrix)
    } else {
      idx <- methodName(brokenMatrix, alfa, beta)
    }

    vectorOfIdsx[i] <- abs(realIdx - idx)
  }

  incompleteIdx <- mean(vectorOfIdsx)

  abs(incompleteIdx/realIdx*100)
}
```

Fig. 4.10. The implementation of function *exploreMatrix* which tests given inconsistency index

```r
#' @title Explore the incomplete PC matrixes for every method and scale <1.1, 1.2, ... , 4.0>
#' @description Examines what is the relative error between the full matrixes and indomplete matrixes.
#' @param numOfElements - dimension of tested matrix
#' @param gradeOfIncomplete -  percentage of the value to be removed (not applicable to the diagonal)
#' @param numOfAttempts - number of tested matrixes
#' @param numOfAttemptsForOneMatrix - number of test cases for each matrix
#' @param alfa - a parameter for kulakowskiSzybowskiIa method
#' @param beta - a parameter for kulakowskiSzybowskiIab method
#' @return average value of the relative error between the full matrix and indomplete matrix
test <- function(numOfElements, gradeOfIncomplete, numOfAttempts, numOfAttemptsForOneMatrix, alfa=0, beta=0){

  results <- matrix(nrow=31, ncol=16, data=0)
  counter <- 1

  for(i in seq(1.1, 4, 0.1)){
    print(counter)
    results[counter,] <- monteCarloOnTheSameMatrix(numOfElements, i, gradeOfIncomplete, numOfAttempts,
                                                   numOfAttemptsForOneMatrix, alfa, beta)
    counter <- counter+1
  }

  for(i in 1:16){
    results[31, i] = sum(results[,i])/30
  }

  results
}
```

Fig. 4.11. The implementation of function *test* which tests all indexes regarding given matrix size and the degree of incompleteness



Studies inconsistency indexes for incomplete matrixes

Documentation for package ‘indexesForIncomplete’ version 0.1.0

- DESCRIPTION file.

Help Pages

breakPCMatrix	Breaks PC Matrix
cavalloDapuzzo	Cavallo D'Apuzzo consistency index (CD)
chopV	Checks if the number is much greater than 0
countIndexesForTriads	Generates inconsistency indexes for each triad from the matrix
createHarkerMatrix	Create Harker matrix
disturbPCMatrix	Disturbs the PC matrix
exploreMatrix	Explore the incomplete PC matrix
exploreMatrixOnTheSameMatrix	Explore the incomplete PC matrixes
generateBrokenPCMatrix	Generate broken PC Matrix
generateDisturbedPCMatrix	Generates disturbed PC Matrix
generatePCMatrix	Generates PC Matrix on the basis of the size
generatePCMatrixFromPV	Generates PC Matrix on the basis of the priority vector
generatePriorityVector	Generates random priority vector
generateTriads	Generates triades from the matrix
geometric	Geometric consistency index (GCI)
geometricRank	Rank list given as geometric means
geometricRankForIncomplete	Rank list for the incomplete matrix given as geometric means
goldenWang	Golden and Wand consistency index (GW)
harmonic	Harmonic consistency index (HCI)
kazibudzkiCMLTI2	Kazibudzki matrix inconsistency (CMLTI2)
kazibudzkiLTI1	Kazibudzki matrix inconsistency (LTI1)
kazibudzkiLTI1ForTriad	kazibudzki (LTI1) innconsistency for one triad
kazibudzkiLTI2	Kazibudzki matrix inconsistency (LTI2)
kazibudzkiLTI2ForTriad	kazibudzki (LTI2) innconsistency for one triad
koczkodaj	Koczkodaj matrix inconsistency
koczkodajForTriad	Koczkodaj innconsistency for one triad

Fig. 4.12. The portion of the documentation: general view

saty {indexesForIncomplete} R Documentation

Saaty index (CI)

Description

Counts the value of inconsistency for the matrix

Usage

```
saaty(matrix)
```

Arguments

`matrix`	- PC matrix (could be incomplete)

Value

inconsistency of the matrix

[Package *indexesForIncomplete* version 0.1.0 Index]

Fig. 4.13. The portion of the documentation: function *saaty*

exploreMatrix {indexesForIncomplete} R Documentation

Explore the incomplete PC matrix

Description

Examines what is the relative error between the full matrix and indomplete matrix

Usage

```
exploreMatrix(methodName, scale, numOfElements, gradeOfIncomplete,
  numOfAttempts, alfa = 0, beta = 0)
```

Arguments

`methodName`	- a name of the method which is tested
`scale`	- extend of disorders. This parametr is the upper limit of the interval that is used to scale the elements. The lower limit is defined as 1 / scale
`numOfElements`	- dimension of tested matrix
`gradeOfIncomplete`	- percentage of the value to be removed (not applicable to the diagonal)
`numOfAttempts`	- number of test cases
`alfa`	- a parameter for kulakowskiSzybowskiIa method
`beta`	- a parameter for kulakowskiSzybowskiIab method

Value

average value of the relative error between the full matrix and indomplete matrix

[Package *indexesForIncomplete* version 0.1.0 Index]

Fig. 4.14. The portion of the documentation: function *exploreMatrix*

5. Results and discussion

5.1. Results

The tests were performed using Operating System *Ubuntu 16.04 LTS* and IDE *RStudio*. The results are presented int he following section.

5.1.1. Tests taking into account different matrix sizes

Table 5.1. Relative error of inconsistency indexes for incomplete matrices with constant degrees of incompleteness $g = 15\%$ and variable matrix size.

Index	n = 4	n = 7	n = 8	n = 10	n = 15	mean	rank
saaty	33,41	19,82	18,78	19,16	17,37	21,71	10
geometric	616,68	124,73	77,94	68,62	39,13	185,42	13
koczkodaj	**13,86**	**3,69**	**2,14**	**1,62**	**0,80**	**4,42**	**1**
kazibudzkiLTI1	24,80	10,21	6,62	4,97	2,73	9,87	6
kazibudzkiLTI2	42,31	17,93	11,88	9,03	5,03	17,24	8
kazibudzkiCMLTI2	35,40	17,07	13,26	11,20	6,81	16,75	7
pelaeLamata	44,65	19,90	13,46	10,36	5,84	18,84	9
kulakSzyb	20,34	7,68	4,88	3,63	1,96	7,70	5
kulakSzyb2	44,61	26,05	27,12	29,64	28,46	31,18	11
kulakSzybIa	16,47	5,18	3,09	2,27	1,16	5,63	3
kulakSzybIab	17,40	4,89	2,81	2,04	1,01	5,63	2
harmonic	9 573,02	1 577,49	1 127,33	1 066,35	866,00	2 842,04	15
goldenWang	115,92	54,37	43,90	43,16	36,26	58,72	12
saloHamalainen	381,57	205,06	176,11	160,06	136,55	211,87	14
cavalloD'Apuzzo	16,94	6,85	4,46	3,36	1,87	6,70	4
relativeError	1 792,64	226 313,60	746,21	100,87	20,42	45 794,75	16

5.1.2. Tests taking into account different degrees of incompleteness

5.1.3. Tests taking into account different levels of inconsistency

Among the tested inconsistency indexes ones, which are given the best results, are presented below. The detailed results are placed in Appendix.

Table 5.2. Relative error of inconsistency indexes for incomplete matrices with varying degrees of incompleteness and constant matrix size $n = 8$.

Index	$g = 4\%$	$g = 7\%$	$g = 14\%$	$g = 25\%$	$g = 50\%$	mean	rank
saaty	4,71	9,40	18,78	32,89	65,56	26,27	10
geometric	23,60	48,44	86,61	135,68	207,99	100,46	13
koczkodaj	**0,48**	**0,99**	**2,17**	**4,52**	16,41	4,92	2
kazibudzkiLTI1	2,90	4,31	6,64	10,05	23,09	9,40	6
kazibudzkiLTI2	5,12	7,71	11,91	18,08	40,77	16,72	7
kazibudzkiCMLTI2	5,16	8,05	13,34	22,03	61,16	21,95	9
pelaeLamata	5,73	8,72	13,52	20,54	45,64	18,83	8
kulakSzyb	2,17	3,18	4,91	7,43	17,30	7,00	5
kulakSzyb2	5,90	12,22	27,12	56,71	202,27	60,84	12
kulakSzybIa	1,20	1,88	3,12	5,13	13,97	5,06	3
kulakSzybIab	1,00	1,65	2,84	4,74	**12,97**	**4,64**	**1**
harmonic	291,74	544,60	1 152,26	1 962,25	3 995,58	1 589,29	16
goldenWang	14,23	25,23	46,18	68,83	98,24	50,54	11
saloHamalainen	88,40	137,70	180,17	182,54	148,74	147,51	14
cavalloD‘Apuzzo	1,95	2,91	4,46	6,81	16,11	6,45	4
relativeError	18,99	20,81	206,74	68,98	1 056,96	274,50	15

5.2. Discussion

Analyzing the above results, one can draw several conclusions.

5.2.1. Tests taking into account different matrix sizes

Along with a growth of the matrix size, examined relative error decreases. Considering the best indexes, the relative error is between 13 and 17 percent for small matrices (size 4) so it is relatively big. However, it decreases rapidly, when the size of PC matrix increases. For matrix of size 7 indexes exist which give the results in range $3 - 5\%$, what is a gratifying score. For PC matrices with a significant size (over 10 concepts), the relative error is just about $1 - 2\%$, what is an acceptable score.

Definitely, the best index found was out *Koczkodaj index*. It wins in each test regardless of the matrix size. It gives really satisfying results for not very small matrices. The good results were obtained also through indexes *kulakowskiSzybowskiIab*, *kulakowskiSzybowskiIa* and *cavalloD‘Apuzzo*. It seems that these indexes are reliable. Absolutely one should reject indexes *relativeError*, *harmonic*, *saloHamalainen*, *geometric*, *goldenWang* for incomplete matrices. Results obtained through these methods shows that score examined inconsistency for incomplete matrices can depart significantly from a right values.

It is worth noticing that the presented results involve matrices with the degree of incompleteness equal 15%. It means that 7 comparisons are missing for matrices with size $n = 10$. The results will be different from presented in table 5.1. for other degrees of incompleteness. Impact the degree of incompleteness on tested relative error is shown consecutive experiments.

5.2.2. Tests taking into account different degrees of incompleteness

Along with growth of the degree of incompleteness, examined relative error grows. Considering the best indexes, the relative error amount to just about $1 - 2\%$ percent for degree of incompleteness (size 4) so it is really satisfying. However, it increases slowly, when the degree of incompleteness decreases. For the degree of incompleteness $g = 14\%$ exist indexes which give the results about $2 - 3\%$, what is a gratifying score. The relative error is equal about a dozen percent for significant matrices ($g = 50\%$).

Definitely, the best indexes came out *Koczkodaj index* and *kulakowskiSzybowskiIab*. First one wins in most cases and gives really satisfy the results. Second one improves along with the growth of the degree of incompleteness. It obtained a significant advantage over other methods in the last test, what caused that the average value turned out to be the lowest. The good results were obtained also through indexes *kulakowskiSzybowskiIa* and *cavalloD'Apuzzo*. It seems that these indexes are reliable. Absolutely one should reject indexes *harmonic*, *relativeError*, *saloHamalainen*, *geometric*. Results obtained through these methods shows that score examined inconsistency for incomplete matrices can differ significantly from the right values.

It is worth noticing that the presented results involve matrices with the size 8%. The results will be different from presented in table 5.2. for others matrix sizes. Impact the degree of incompleteness on tested relative error was shown in previous experiments.

5.2.3. Tests taking into account different levels of inconsistency

All tests for different matrix sizes and degrees of incompleteness was performed taking into account varying levels of inconsistency. The scale of this incompleteness is proportional to the value d included in table 5.3. This table shows detailed results for which average values were presented in table 5.1. It is worth taking a closer look at these scores. They point out that the quality of inconsistency indexes is determined also by the level of inconsistency.

The lowest values of the relative error was obtained by *Koczkodaj index*, what was discussed before. One should notice and emphasize that it gives the best results only from particular moment which the level of inconsistency in matrix begins to grow. For small inconsistencies, the most satisfying index is *Cavallo and D'Apuzzo index*. This in interesting regularity which indicates that the choice of inconsistency index for incomplete matrices should depend on the level of inconsistency.

It is worth noticing that this regularity repeats in each of the performed tests. For small level of inconsistency, regardless of the matrix size and the degree od incompleteness, *Cavallo and D'Apuzzo index* comes out to be the best. The results presented in Appendix confirm it.

5.2.4. General discussion

Certainly, the tests managed to show that the error increases with a growth of the level of incompleteness. At the same time, it decreases when the size of the matrix increases. However, the most important question was about which indexes cope well with incomplete matrices. The Koczkodaj index ([sub:Koczkodaj-index]) won in 9 out of 10 tests and its average error in both cases turns out to be the lowest (below 5%). The next places are occupied by two indexes introduced by Kułakowski and Szybowski ([eq:Ia], [eq:Iab]) and Cavallo and D'Apuzzo index ([sub:Cavallo-and-D'Apuzzo]). It is worth noting that all of these indexes are based on triads.

A question about what makes the Koczkodaj index giving such good results and whether is it worth using, may arise. One should return to the definition of this index ([eq:K]) and notice that it is equal to the value of the most inconsistency triad. Therefore, if the level of incompleteness is low, there is a good chance that after deletion of some values from the matrix and recalculation at the index the value of it will not change at all. It will only change if the element included in the most inconsistent triad is removed. However, in many cases, the examination of the full matrix and th matrix after removing some values give exactly the same results (the error is 0%). This is the only index of this kind among those presented in the paper.

If one uses the Koczkodaj index, one may be worried that the removed comparison belonged to the most inconsistent triad. In such case, it is difficult to predict what error will be contained in the index of the incomplete matrix. It seems that one should pay particular attention to the indexes proposed by Kułakowski and Szybowski. First of them ([eq:I1]) averages the inconsistencies of the triads. Therefore, it is safe and gives good results (in both tests it took the sixth place and achieved an error below 8%). From this perspective, another index suggested by the same authors ([eq:Ia]) turns out to be very interesting. In the tests it took the third place. It has the parameter α allowing to determine the effect of the greatest inconsistency of the triad (α) and the average $(1-\alpha)$. In the tests carried out, the parameter α was 0.4.

Table 5.3. Relative error of inconsistency indexes for incomplete matrices with varying degrees of inconsistency, constant matrix size $n = 8$ and constant level of incompleteness $g = 15\%$.

d	koczkodaj	kazibudzkiLTI1	kulakSzyb	kulakSzybIa	kulakSzybIab	cavalloD'Apuzzo
1.1	4.09	6.38	6.13	4.52	4.35	**0.49**
1.2	3.68	6.45	5.98	4.26	4.06	**0.96**
1.3	3.58	6.55	5.87	4.16	3.96	**1.37**
1.4	3.55	6.62	5.77	4.12	3.91	**1.80**
1.5	3.19	6.35	5.42	3.80	3.59	**2.02**
1.6	3.26	6.57	5.46	3.84	3.64	**2.39**
1.7	3.03	6.59	5.35	3.71	3.48	**2.81**
1.8	**2.47**	6.45	5.13	3.31	3.04	3.07
1.9	**2.47**	6.65	5.26	3.35	3.06	3.34
2.0	**2.20**	6.66	5.13	3.16	2.85	3.64
2.1	**2.38**	6.51	4.93	3.21	2.93	3.83
2.2	**2.16**	6.52	4.81	3.04	2.76	4.09
2.3	**2.14**	6.66	4.89	3.07	2.80	4.25
2.4	**2.00**	6.67	4.85	2.98	2.67	4.49
2.5	**1.96**	6.59	4.73	2.91	2.63	4.71
2.6	**1.73**	6.62	4.72	2.81	2.48	4.87
2.7	**1.94**	6.79	4.73	2.96	2.67	5.14
2.8	**1.75**	6.67	4.64	2.78	2.47	5.24
2.9	**1.56**	6.67	4.54	2.68	2.38	5.60
3.0	**1.82**	6.71	4.68	2.84	2.51	5.56
3.1	**1.47**	6.69	4.48	2.62	2.30	5.74
3.2	**1.46**	6.52	4.40	2.59	2.28	5.75
3.3	**1.48**	6.85	4.57	2.65	2.35	6.14
3.4	**1.38**	6.60	4.30	2.53	2.23	6.15
3.5	**1.17**	6.60	4.25	2.40	2.10	6.50
3.6	**1.25**	6.73	4.39	2.50	2.18	6.65
3.7	**1.18**	6.83	4.41	2.48	2.19	6.77
3.8	**1.24**	6.59	4.17	2.40	2.09	6.68
3.9	**1.28**	6.73	4.25	2.47	2.19	6.79
4.0	**1.11**	6.52	4.17	2.35	2.06	6.79
mean	**2.13**	6.61	4.88	3.08	2.81	4.45

6. Summary

The goal of the work - testing common inconsistency indexes for incomplete PC matrices was achieved. One has examined sixteen different methods for calculating inconsistency. The performed teststook into account many factors which could have impact on results. It considered matrices with different size and the level of inconsistency. They were checked for varying degree of incompleteness (from 4% to 50%). All results was gathered and presented in tables.

All tests was developed in *R* language which is appropriate for numerical calculations and it fulfilled the task. Functions, which are responsible for calculations, was described and documented in details. They can compute inconsistency indexes for both full and incomplete matrices. If applicable, one can implement another indexes easily.

The tests have obtained expressly that some of existing indexes are up to calculating inconsistency for incomplete matrices after slight modifications. The methods of these modifications are presented in this work. Obviously there are many different possibilities to make changes in these indexes. Perhaps they can give even more favorable results. However, the purpose of this work was testing only existing indexes, therefore, one has decided not to make many modifications. The most reliable results was achieved by *Koczkodaj index*, *KulakowskiSzybowskiIa*, *KulakowskiSzybowskiIab* and *Cavallo DAapuzzo*. Selection one from these indexes should be done based on matrix parameters which was described in this work and presented in the results.

Very interesting came out indexes proposed by *Kułakowski* and *Szybowski* containing parameters α and β. Only one assignment of these arguments was checked in the performed tests and it gave promising results. It seems that further experiments of these indexes should be dedicated to choice of values of parameters α and β in a way that will give even better results. How one was mentioned in this work, these parameters allow to choice between a more reliable result and sure that the relative error will be known.

An interesting line of enquiry can be using PC method for incomplete matrices when somehow a part of comparisons is missing. Such works already appear. However, this paper shows that also calculating inconsistency for such matrices is possible. The PC method expands for many years, still discoverS new potential and tests new places where IT can be applied.

Bibliografia

[1] Josep M. Colomer. „Ramon Llull: from 'Ars electionis' to social choice theory". W: *Social Choice and Welfare* 40.2 (2013), s. 317–328. ISSN: 1432-217X. DOI: *10.1007/s00355-011-0598-2*. URL: *https://doi.org/10.1007/s00355-011-0598-2*.

[2] Konrad Kułakowski. „Notes on the existence of a solution in the pairwise comparisons method using the heuristic rating estimation approach". W: *Annals of Mathematics and Artificial Intelligence* 77.1 (2016), s. 105–121. DOI: *10.1007/s10472-015-9474-6*. URL: *https://doi.org/10.1007/s10472-015-9474-6*.

[3] Kenneth Arrow. „A Difficulty in the Concept of Social Welfare". W: *Journal of Political Economy* 58 (1950). URL: *https://EconPapers.repec.org/RePEc:ucp:jpolec:v:58:y:1950:p:328*.

[4] Amartya Sen. „Social Choice Theory: A Re-examination". W: *Econometrica* 45.1 (1977), s. 53–89. URL: *https://EconPapers.repec.org/RePEc:ecm:emetrp:v:45:y:1977:i:1:p:53-89*.

[5] Louis Leon Thurstone. „A law of comparative judgment". W: *Psychological Review* 101(2) (1994), s. 266–270.

[6] Thomas L. Saaty. „Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process". W: *RACSAM - Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas* 102.2 (2008), s. 251–318. ISSN: 1579-1505. DOI: *10.1007/BF03191825*. URL: *https://doi.org/10.1007/BF03191825*.

[7] T.L. Saaty i G. Hu. „Ranking by Eigenvector versus other methods in the Analytic Hierarchy Process". W: *Applied Mathematics Letters* 11.4 (1998), s. 121 –125. ISSN: 0893-9659. DOI: *https://doi.org/10.1016/S0893-9659(98)00068-8*. URL: *http://www.sciencedirect.com/science/article/pii/S0893965998000688*.

[8] Matteo Brunelli, Luisa Canal i Michele Fedrizzi. „Inconsistency indices for pairwise comparison matrices: a numerical study". W: *Annals of Operations Research* 211.1 (2013), s. 493–509. ISSN: 1572-9338. DOI: *10.1007/s10479-013-1329-0*. URL: *https://doi.org/10.1007/s10479-013-1329-0*.

[9] Thomas L Saaty. „A scaling method for priorities in hierarchical structures". W: *Journal of Mathematical Psychology* 15.3 (1977), s. 234 –281. ISSN: 0022-2496. DOI: *https://doi.org/10.1016/0022-2496(77)90033-5*. URL: *http://www.sciencedirect.com/science/article/pii/0022249677900335*.

[10] Gordon Crawford i Cindy Williams. „A note on the analysis of subjective judgment matrices". W: *Journal of Mathematical Psychology* 29.4 (1985), s. 387 –405. ISSN: 0022-2496. DOI: *https://doi.org/10.1016/0022-2496(85)90002-1*. URL: *http://www.sciencedirect.com/science/article/pii/0022249685900021*.

[11] Juan Aguaron i Jose Maria Moreno-Jimenez. „The geometric consistency index: Approximated thresholds". W: *European Journal of Operational Research* 147.1 (2003), s. 137 –145. ISSN: 0377-2217. DOI: *https://doi.org/10.1016/S0377-2217(02)00255-2*. URL: *http://www.sciencedirect.com/science/article/pii/S0377221702002552*.

[12] W.W. Koczkodaj. „A new definition of consistency of pairwise comparisons". W: *Mathematical and Computer Modelling* 18.7 (1993), s. 79 –84. ISSN: 0895-7177. DOI: *https://doi.org/10.1016/0895-7177(93)90059-8*. URL: *http://www.sciencedirect.com/science/article/pii/0895717793900598*.

[13] Zbigniew Duszak i Waldemar W. Koczkodaj. „Generalization of a new definition of consistency for pairwise comparisons". W: *Information Processing Letters* 52.5 (1994), s. 273 –276. ISSN: 0020-0190. DOI: *https://doi.org/10.1016/0020-0190(94)00155-3*. URL: *http://www.sciencedirect.com/science/article/pii/0020019094001553*.

[14] Pawel Kazibudzki. „Redefinition of Triad's Inconsistency and its Impact on the Consistency Measurement of Pairwise Comparison Matrix". W: 2016 (mar. 2016), s. 71–78.

[15] Paul Thaddeus Kazibudzki. „The quality of priority ratios estimation in relation to a selected prioritization procedure and consistency measure for a Pairwise Comparison Matrix". W: *CoRR* abs/1704.01944 (2017). arXiv: *1704.01944*. URL: *http://arxiv.org/abs/1704.01944*.

[16] J.I. Peláez i M.T. Lamata. „A new measure of consistency for positive reciprocal matrices". W: *Computers & Mathematics with Applications* 46.12 (2003), s. 1839 –1845. ISSN: 0898-1221. DOI: *https://doi.org/10.1016/S0898-1221(03)90240-9*. URL: *http://www.sciencedirect.com/science/article/pii/S0898122103902409*.

[17] Konrad Kulakowski i Jacek Szybowski. „The New Triad based Inconsistency Indices for Pairwise Comparisons". W: *Procedia Computer Science* 35 (2014). Knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings, s. 1132 –1137. ISSN: 1877-0509. DOI: *https://doi.org/10.1016/j.procs.2014.08.205*. URL: *http://www.sciencedirect.com/science/article/pii/S1877050914011703*.

[18] William E. Stein i Philip J. Mizzi. „The harmonic consistency index for the analytic hierarchy process". W: *European Journal of Operational Research* 177.1 (2007), s. 488 –497. ISSN: 0377-2217. DOI: *https://doi.org/10.1016/j.ejor.2005.10.057*. URL: *http://www.sciencedirect.com/science/article/pii/S0377221705009288*.

[19] Bruce L. Golden i Qiwen Wang. „An Alternate Measure of Consistency". W: *The Analytic Hierarchy Process: Applications and Studies*. Red. Bruce L. Golden, Edward A. Wasil i Patrick T. Harker. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, s. 68–81. ISBN: 978-3-642-50244-6.

[20] Ahti A. Salo i Raimo P. Hamalainen. „Preference programming through approximate ratio comparisons". W: *European Journal of Operational Research* 82.3 (1995), s. 458 –475. ISSN: 0377-2217. DOI: *https://doi.org/10.1016/0377-2217(93)E0224-L*. URL: *http://www.sciencedirect.com/science/article/pii/0377221793E0224L*.

[21] Matteo Brunelli. *Introduction to the Analytic Hierarchy Process*. Sty. 2015.

[22] B. Cavallo i L. D'Apuzzo. „A general unified framework for pairwise comparison matrices in multicriterial methods". W: *International Journal of Intelligent Systems* 24.4 (2009), s. 377–398. ISSN: 1098-111X. DOI: *10.1002/int.20329*. URL: *http://dx.doi.org/10.1002/int.20329*.

[23] Bice Cavallo i Livia D'Apuzzo. „Characterizations of consistent pairwise comparison matrices over abelian linearly ordered groups". W: *Int. J. Intell. Syst.* 25.10 (2010), s. 1035–1059. DOI: *10.1002/int.20438*. URL: *https://doi.org/10.1002/int.20438*.

[24] Barzilai Jonathan. „Consistency measures for pairwise comparison matrices". W: *Journal of Multi-Criteria Decision Analysis* 7.3 (1998), s. 123–132. DOI: *10.1002/(SICI)1099-1360(199805)7:3<123::AID-MCDA181>3.0.CO;2-8*.

[25] P.T. Harker. „Alternative modes of questioning in the analytic hierarchy process". W: *Mathematical Modelling* 9.3 (1987), s. 353 –360. ISSN: 0270-0255. DOI: *https://doi.org/10.1016/0270-0255(87)90492-1*. URL: *http://www.sciencedirect.com/science/article/pii/0270025587904921*.

[26] 2018 (accessed June 1, 2018). URL: *https://cran.r-project.org*.

[27] 2018 (accessed June 1, 2018). URL: *https://www.rstudio.com/products/RStudio*.

[28] 2010 (accessed June 14, 2018). URL: *http://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/installr.html*.

				BŁĄD WZGLĘDNY												
100 x 100	N = 4	INCOM = 15%														
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	33.19377	12104.95854	17.496633	24.29082	40.79644	40.59259	40.81909	23.73275	47.53033	20.24615	21.48334	176471.1667	924.12427	2591.5607	**1.680931**	97.41760
1.2	33.52463	1567.66910	18.851788	25.92967	45.06119	44.05059	45.17973	24.46757	46.34504	21.30373	22.52788	32823.1770	359.14143	1063.0919	**3.890674**	308.01591
1.3	33.30961	725.43880	17.411693	24.95385	43.15466	41.53533	43.35127	23.14712	46.26999	19.89357	20.99081	12787.4400	213.46967	744.1814	**5.016011**	71.51679
1.4	33.20639	645.41563	16.196975	23.80880	40.21997	37.90405	40.50250	22.10246	45.78984	18.66164	19.60531	16629.0494	204.06217	623.8909	**5.947366**	45.48664
1.5	33.07680	353.32978	15.828859	24.96826	41.91292	38.20009	42.43545	22.53085	46.34582	18.63137	19.59206	5285.3583	126.31377	482.2611	**8.310100**	68.21883
1.6	33.12399	326.59835	15.886603	25.00819	42.01766	37.85251	42.63121	22.18267	46.24526	18.53246	19.49599	4959.7858	115.14350	492.7710	**8.972119**	46.58680
1.7	33.38869	209.27517	16.046038	25.77707	44.03368	38.61411	44.94884	22.35075	45.54104	18.66297	19.60428	2795.6981	101.81936	377.3315	**10.907788**	52.21692
1.8	33.59274	141.17418	14.741311	24.05689	41.89391	35.45785	43.08894	20.39327	44.06566	17.01927	17.88159	1907.4118	77.62078	283.5774	**12.045677**	102.16686
1.9	33.08837	345.42424	15.180474	24.71290	42.12287	36.37760	43.17610	21.25333	45.06910	17.66865	18.76059	3620.0050	103.23373	397.1868	**11.617610**	245.14562
2	33.56351	154.49680	13.946747	23.91799	41.22471	34.32581	42.77828	19.97886	44.30822	16.36896	17.33862	1695.0038	62.10456	256.9582	**13.630334**	20761.78334
2.1	33.53854	151.86598	**14.902911**	26.14765	43.93636	36.29657	45.69949	22.00374	45.22855	17.72234	18.38890	2641.0302	84.37643	292.2531	15.324162	89.12215
2.2	33.53672	128.82471	**14.036823**	24.74335	41.60570	34.28856	43.21714	20.30437	45.28602	16.60362	17.65595	2376.1359	76.89173	266.1821	15.081927	422.96411
2.3	33.29815	95.03890	**14.163792**	24.84662	42.44105	35.36208	44.18066	20.23140	44.78016	16.64251	17.78353	1634.5323	68.91945	229.2499	15.879418	101.09640
2.4	33.57207	211.07610	**13.765527**	24.50627	42.03844	34.03416	43.94484	19.65736	44.54769	16.14138	17.12841	1658.2426	76.01481	279.7526	16.240281	81.57221
2.5	33.84260	90.65566	**14.336666**	26.47804	45.19147	36.24702	47.81429	21.17147	44.31081	16.99458	17.55841	1421.6241	67.19964	208.6712	19.016658	164.14810
2.6	33.42205	81.82519	**13.498720**	25.29041	43.13477	34.45652	45.71955	19.89965	44.36365	16.00933	16.80135	1573.2021	68.03356	214.5739	19.246976	69.39993
2.7	33.04999	109.40815	**13.052938**	24.50541	41.66384	34.11379	44.07505	19.87003	44.35878	15.76175	16.75746	992.4467	58.29010	239.0567	18.818721	1743.72318
2.8	33.27415	166.17184	**12.983248**	24.77826	42.11220	33.68822	44.90787	19.81759	44.03727	15.66560	16.60109	1445.2802	61.24474	244.5711	19.754475	72.94473
2.9	33.20759	160.23100	**12.009758**	24.05097	40.71869	32.46001	43.70787	18.99390	44.08622	14.73379	15.67949	3339.3676	67.32043	208.3890	20.821873	2333.46617
3	33.23799	85.97430	**12.115964**	24.47179	41.06160	32.71360	44.13543	19.23897	44.41326	14.96497	15.89355	1250.5546	57.19278	195.2469	21.982160	332.81789
3.1	33.44827	56.18713	**12.975392**	25.65931	43.98755	34.58271	47.41730	19.47454	43.81940	15.55681	16.51735	798.4669	50.09576	177.5808	23.024785	69.27645
3.2	33.32215	62.35050	**12.876430**	25.18833	43.29570	34.40465	46.43465	19.84006	43.68268	15.58143	16.14785	1035.0932	52.62716	192.2464	21.296776	2688.16110
3.3	33.86956	69.94830	**11.430911**	23.97675	40.86490	32.59108	44.75466	18.19832	43.44761	14.09441	15.02006	898.6684	47.98402	168.6938	23.838817	96.62367
3.4	33.91468	64.68349	**12.068943**	23.78274	41.53405	32.60743	45.21385	18.28564	42.93935	14.44050	15.16581	998.1338	50.90481	182.5599	21.560812	54.36607
3.5	33.54859	65.82020	**12.879343**	26.68684	45.71454	35.43419	50.14129	19.25235	43.76730	15.40381	16.31943	830.1918	48.11440	149.1904	27.249709	99.53210
3.6	33.55067	53.38288	**11.453037**	24.16588	41.34280	32.62269	45.61978	18.08551	43.33441	14.05146	15.06480	765.3219	42.11473	159.2900	25.924704	405.99913
3.7	33.53365	55.87411	**12.126212**	24.93220	42.65348	34.01311	46.70452	18.48565	44.11537	14.59573	15.50311	1116.5471	48.22129	193.1881	24.591261	57.02695
3.8	33.47723	71.37353	**11.524139**	25.62915	43.50901	34.15645	48.49786	20.02655	43.33117	14.84835	15.31330	1303.3484	59.66181	167.6315	27.582501	40.68222
3.9	33.30119	96.37045	**12.260789**	25.19815	42.90155	34.07195	46.90171	19.44575	44.00578	15.03993	15.92218	1453.4005	66.04069	214.0114	24.723968	368.09709
4	33.21129	49.52269	**9.705521**	21.61344	37.21805	28.93320	41.42960	15.81206	42.81709	12.13273	13.51416	684.8222	39.41607	151.8958	24.366487	22689.73282
ŚREDNIA	33.40752	616.67886	**13.858473**	24.80253	42.31213	35.39962	44.64763	20.34115	44.60610	16.46579	17.40056	9573.0169	115.92325	381.5682	16.944836	1792.64359

				BŁĄD WZGLĘDNY												
100 x 100	N = 6	INCOM = 15%														
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	20.09735	1868.92984	6.434853	10.307496	17.88775	17.63533	17.90753	9.938767	26.51083	7.461503	7.294034	24720.2375	470.82282	1378.25055	**0.7847627**	42.04489
1.2	19.80228	626.24056	5.757167	10.172102	17.51832	16.63355	17.59040	9.510405	26.48250	7.037483	6.822257	7130.1332	228.73932	695.06327	**1.4730570**	31.31537
1.3	19.91184	315.86296	5.835830	10.007532	17.23582	15.71163	17.38747	9.102586	26.37577	6.836664	6.638325	3767.4622	143.23739	463.30963	**2.0932104**	345.5845
1.4	19.76527	121.52089	5.381870	10.119758	18.07529	15.77008	18.35624	8.819405	26.37362	6.532083	6.385168	1705.0360	80.18795	327.93064	**2.7658421**	27.43780
1.5	20.22667	123.05595	5.454400	10.100891	17.45626	15.24082	17.82012	8.730824	25.94365	6.481819	6.260828	1329.5146	62.24802	269.95772	**3.2262929**	6787322
1.6	19.96144	69.33523	4.850259	10.292108	17.73903	15.30001	18.21187	8.691002	26.15642	6.149097	5.911386	1011.9394	55.20548	234.20628	**3.8687051**	26.58558
1.7	19.65038	56.02665	4.289249	9.653072	16.81729	14.57848	17.35766	8.007852	26.04008	5.601610	5.352996	907.1446	51.92631	224.46494	**3.9936759**	43.59097
1.8	20.05975	39.81261	4.776926	10.595570	18.59635	15.89642	19.36798	8.546307	26.03685	6.074614	5.787792	735.8898	45.47310	185.13784	**4.7185356**	20.31194
1.9	20.23113	34.55642	**4.176875**	10.346410	18.09488	15.57543	19.01446	8.156017	25.94105	5.562436	5.237522	581.3561	36.38688	159.96625	5.2561184	16.96724
2	19.76385	36.43960	**3.987324**	10.130293	17.84222	15.35191	18.86817	7.905971	26.03965	5.392512	5.111896	526.3546	35.91791	158.19263	5.4411259	17.63695
2.1	19.72543	37.17107	**4.039302**	10.198642	17.99152	15.90108	19.23888	7.862713	26.09941	5.396649	5.118683	408.6556	30.08374	150.79019	5.8414936	31.73146
2.2	19.95502	36.53264	**3.885218**	10.425099	18.45216	15.97074	19.83858	7.859201	25.98000	5.348684	5.035819	398.6650	30.97694	148.00123	6.3079041	34.76873
2.3	19.93441	23.77939	**3.526719**	10.244919	17.86315	16.24825	19.39865	7.701228	25.98021	5.054043	4.728950	332.6475	26.01917	131.13156	6.6572019	23.91516
2.4	19.96963	30.74497	**3.405329**	10.299073	17.84681	16.16004	19.55870	7.686349	25.96123	5.003513	4.680253	302.9812	22.18708	126.65109	7.0113032	559.7964
2.5	19.75359	24.94231	**3.638073**	10.314012	18.14792	17.09276	20.00256	7.533081	26.05494	5.059155	4.760526	286.9388	23.31898	115.85172	7.2576284	26.14573
2.6	19.73395	26.63904	**3.518353**	10.172354	18.08114	17.72092	20.20753	7.412458	25.91082	4.934177	4.617061	307.0523	26.22140	121.68059	7.2778887	16.47764
2.7	19.77336	21.13702	**3.245991**	9.941673	17.34301	17.36256	19.53088	7.203248	26.05366	4.712422	4.406364	250.5867	21.04420	100.35909	7.6445794	39.40290
2.8	19.96610	21.96576	**3.114027**	10.140286	18.01929	16.95724	20.41695	7.179741	25.89782	4.718868	4.400755	252.9760	20.63547	103.64031	8.1733246	21.84916
2.9	19.55083	20.94146	**2.830616**	10.100120	17.69333	17.09006	20.07389	7.147434	26.04278	4.516512	4.190841	261.8878	20.83967	103.83690	8.2152667	21.15079
3	19.88865	19.99464	**3.032424**	10.483554	18.12235	18.06656	20.87414	7.331958	25.86567	4.717755	4.352076	217.3242	20.20669	99.80857	9.0795933	27.13738
3.1	19.74696	19.73121	**2.856096**	10.030063	17.80316	18.22819	20.80996	6.852494	25.86106	4.421881	4.091075	207.2519	19.55855	89.75760	9.0684233	32.38309
3.2	19.70389	18.36457	**2.886541**	10.467474	18.70323	18.83261	22.01207	7.034348	25.97910	4.547684	4.247109	214.9930	19.02013	86.30296	9.4092088	15.73611
3.3	19.75122	20.36396	**2.694493**	10.460970	18.50395	17.92098	21.52279	7.077816	26.00633	4.478651	4.173062	220.4528	19.46583	97.11765	9.4266652	17.05024
3.4	19.54629	19.49335	**2.256849**	10.326000	18.02326	17.36544	21.12168	7.006642	25.97594	4.219476	3.883494	195.4933	18.51250	89.56174	9.6409281	27.51022
3.5	19.73473	19.08160	**2.234151**	10.213723	17.62694	18.11540	20.77861	6.856669	25.99935	4.156783	3.811999	192.4801	19.15069	82.64165	9.9594238	130.7350
3.6	19.83332	16.84716	**2.656405**	10.329227	18.37070	18.29594	22.08177	6.772955	25.93405	4.307550	3.993727	174.2783	16.34948	77.63043	10.0602904	178.7915
3.7	19.58103	19.04770	**2.269374**	9.832769	17.46988	17.61759	21.03715	6.441462	26.01045	3.973501	3.693478	186.3651	17.66742	84.62575	9.9083634	254.3906
3.8	19.76261	18.52163	**2.860100**	10.606471	18.97415	20.55269	23.17829	6.924899	26.04470	4.505600	4.209245	178.7117	18.06507	83.82906	10.4855789	18.01544
3.9	19.54997	17.26431	**2.424964**	10.044828	17.76154	19.13833	21.42948	6.557866	26.02581	4.125075	3.822989	168.9070	17.06280	86.59295	10.0444075	24.03729
4	19.55387	17.59369	**2.370695**	10.020463	17.76862	19.90035	21.99330	6.477028	26.06132	4.038085	3.734062	150.9219	14.62729	75.63339	10.4311382	13.09485
ŚREDNIA	19.81616	124.73127	**3.689682**	10.212565	17.92764	17.07438	19.89959	7.677624	26.05484	5.178863	4.891793	1577.4879	54.37194	205.06414	6.8507312	226313.6

				BŁĄD WZGLĘDNY												
100 x 100	N = 8	INCOM = 15%														
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	18.95832	1225.94665	4.090937	6.382752	11.35475	11.15069	11.37002	6.134483	27.13112	4.522092	4.355400	17996.7899	392.32023	1182.47003	**0.4935174**	13.89572
1.2	18.83182	316.10328	3.680864	6.458161	11.44344	10.86449	11.49805	5.989153	27.12924	4.262745	4.064348	4579.3978	174.74604	580.40194	**0.9619032**	13.93167
1.3	18.97737	180.76883	3.580805	6.558406	11.76087	10.74503	11.88199	5.871278	27.06143	4.165851	3.962433	2345.1261	113.61430	416.95999	**1.3780385**	36.60727
1.4	19.05189	84.51707	3.553722	6.623511	11.77991	10.64539	11.97340	5.772095	27.11282	4.124993	3.917218	1425.0556	75.22200	303.48775	**1.8027343**	250.90099
1.5	19.04023	76.86235	3.193002	6.358902	11.32745	10.44179	11.58609	5.423905	27.02980	3.807940	3.597450	1043.7310	65.40725	263.83915	**2.0291874**	16.09523
1.6	18.90739	45.10083	3.266865	6.571615	11.82338	11.14907	12.20637	5.466522	27.11635	3.846445	3.642482	786.5903	49.23579	218.34754	**2.3957012**	14.15074
1.7	18.81929	31.42474	3.034330	6.590362	11.78382	11.55312	12.31430	5.352045	27.09426	3.711462	3.486838	595.0655	38.11715	182.81874	**2.8173099**	27.70678
1.8	18.92389	25.39390	**2.476960**	6.454247	11.49402	11.29169	12.07339	5.132586	27.08042	3.310848	3.041577	490.2343	30.36175	159.28403	3.0724057	29.65546
1.9	18.81830	24.17022	**2.471195**	6.657841	11.74825	11.64674	12.42523	5.263005	27.12510	3.358879	3.069313	441.4255	30.09642	147.43709	3.3495593	29.22829
2	18.83723	20.33368	**2.204978**	6.666235	11.80731	12.27328	12.59902	5.136900	27.12580	3.161737	2.853897	374.7498	27.40830	130.03984	3.6494440	8708.76246
2.1	18.91067	16.51134	**2.389517**	6.519765	11.63564	12.48972	12.56929	4.939208	27.06767	3.213137	2.939677	306.7203	21.68868	114.90076	3.8359517	12.40769
2.2	18.92453	15.12731	**2.167890**	6.524077	11.87296	13.02101	13.02591	4.811689	27.04064	3.049594	2.768609	279.2756	20.35673	106.13742	4.0996105	27.70767
2.3	18.73863	20.59298	**2.145163**	6.661849	12.03576	13.08124	13.24224	4.899801	27.19102	3.074436	2.802012	273.3625	21.14855	115.29019	4.2551201	1144.59540
2.4	18.91567	17.15720	**2.009510**	6.676261	11.92892	13.19913	13.23274	4.854839	27.03816	2.984005	2.676571	266.6710	21.47425	110.77309	4.4942425	18.08464
2.5	18.58201	15.50414	**1.966027**	6.591953	11.84284	14.01966	13.34160	4.737839	27.15294	2.916170	2.633866	222.5165	17.74982	99.31352	4.7102621	10.69797
2.6	18.74086	15.37388	**1.730726**	6.625968	11.80761	13.35029	13.33444	4.722261	27.06053	2.810515	2.484247	233.5669	20.21476	97.40728	4.8739589	29.43351
2.7	18.83260	14.81335	**1.946950**	6.799545	12.39891	15.18660	14.29759	4.732430	27.10599	2.960137	2.671522	212.8559	18.44541	94.03272	5.1470262	10.79809
2.8	18.63726	15.53603	**1.755744**	6.671352	12.05604	14.49875	14.02773	4.644836	27.16357	2.786333	2.478066	193.1518	16.09198	88.45479	5.2410957	200.86171
2.9	18.64025	14.80973	**1.563749**	6.674614	12.00442	14.43736	14.06811	4.540316	27.24427	2.681703	2.385703	166.1818	15.05333	82.82680	5.6013640	10.75872
3	18.89335	15.89409	**1.829013**	6.718147	11.82940	15.41082	13.82302	4.683159	27.03267	2.842154	2.517478	181.1377	15.86646	82.40629	5.5601805	11.36305
3.1	18.81211	14.19025	**1.477378**	6.691944	12.18102	14.51463	14.51420	4.486856	27.06290	2.624650	2.301464	175.1670	14.96687	78.66690	5.7410075	13.32417
3.2	18.80522	17.15977	**1.467914**	6.528610	11.73706	14.15747	13.98010	4.400275	27.02913	2.599315	2.289956	154.9887	14.71263	78.09228	5.7577832	11651.27523
3.3	18.53992	15.01637	**1.489940**	6.852486	12.33436	15.08011	14.86839	4.573024	27.21592	2.656934	2.351967	158.8784	14.59558	77.46240	6.1420393	13.70406
3.4	18.67270	14.87396	**1.381816**	6.609077	12.06172	14.49803	14.75770	4.309345	27.12383	2.531362	2.230070	142.3836	13.49598	72.61274	6.1595149	10.46514
3.5	18.66517	14.35903	**1.172601**	6.603527	11.89588	14.21881	14.73729	4.257484	27.09490	2.408400	2.101206	128.5171	11.69047	65.40004	6.5010120	15.34537
3.6	18.60717	13.82778	**1.257825**	6.735709	12.04010	15.26965	14.91433	4.393535	27.11211	2.503510	2.182539	136.0581	12.41596	68.00008	6.6500134	16.49683
3.7	18.53702	14.01369	**1.180798**	6.834359	12.28457	14.44461	15.31444	4.410983	27.26400	2.487682	2.195762	138.9636	15.04749	70.50877	6.7721756	10.49891
3.8	18.69654	13.99556	**1.245373**	6.592207	12.03814	14.89828	15.14333	4.179833	27.10588	2.408778	2.097611	125.3483	11.25489	64.67355	6.6882838	11.43341
3.9	18.56170	14.36422	**1.283994**	6.734846	12.41621	15.83813	15.87137	4.253299	27.13734	2.478422	2.198863	120.2718	11.67893	67.99367	6.7921020	12.50160
4	18.49652	14.45691	**1.117155**	6.525911	11.69827	14.56718	14.79856	4.170044	27.24048	2.359060	2.062402	125.5783	12.49405	63.19705	6.7914418	13.69598
ŚREDNIA	18.77919	77.93997	**2.137758**	6.616475	11.88077	13.26476	13.45967	4.884768	27.11634	3.088310	2.812018	1127.3254	43.89907	176.10788	4.4587996	746.21279

				BŁĄD WZGLĘDNY												
100 x 100	N = 10	INCOM = 15%														
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybd	kulakowskiSzybd	kulakowskiSzybd	kulakowskiSzybd	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	19.53075	1155.16420	3.9225786	4.953879	8.899221	8.695295	8.911953	4.754709	29.43042	3.946137	3.875692	16874.4213	410.53291	1099.63227	**0.3743040**	12.985568
1.2	19.39114	253.54042	3.4598777	4.873460	8.862264	8.384827	8.909834	4.495845	29.55969	3.550077	3.448784	4528.2661	170.35563	541.47041	**0.7167935**	14.481201
1.3	19.37825	120.88292	2.9518779	4.818052	8.647425	8.083197	8.739568	4.317802	29.59420	3.186265	3.046841	2174.7630	107.84278	366.14067	**1.0117529**	23.064106
1.4	19.49167	84.28107	2.7770010	4.876652	8.794569	8.368334	8.945711	4.243676	29.51001	3.083045	2.918474	1481.8421	84.23154	300.03417	**1.2943449**	14.447113
1.5	19.33784	42.46339	2.5475202	4.771371	8.558474	8.674135	8.769829	4.038692	29.68737	2.868257	2.706352	957.9391	58.78988	224.77370	**1.5471152**	13.714542
1.6	19.29236	44.20313	2.4941282	4.936930	8.818536	9.415193	9.117621	4.087220	29.59065	2.827132	2.640279	727.9090	53.04274	206.90458	**1.8601608**	11.499559
1.7	19.40545	28.14733	2.3773737	4.902362	8.783318	9.846377	9.163350	3.993623	29.59387	2.768482	2.584836	592.5010	39.56818	176.37009	**2.0339289**	11.844960
1.8	19.22422	22.32667	**1.8840305**	4.889500	8.812583	9.595725	9.308831	3.863038	29.63820	2.429407	2.200361	448.9793	30.53240	148.97144	2.3025021	11.265025
1.9	19.30148	16.86709	**1.9140844**	5.033585	9.050524	10.245691	9.628792	3.905748	29.60534	2.460437	2.213029	395.3586	29.28790	129.76097	2.5646492	19.107794
2	19.27408	18.50430	**1.7498258**	4.848250	8.817335	10.149481	9.487828	3.690328	29.59435	2.302348	2.070307	341.4552	24.36160	118.92148	2.6518077	14.768815
2.1	19.18823	15.14961	**1.4844836**	4.930607	8.998984	10.166150	9.771388	3.671138	29.66671	2.168847	1.909033	303.6707	24.69351	112.29544	2.8500466	17.301647
2.2	19.24594	14.37669	**1.7090594**	4.883862	8.870078	11.274519	9.792397	3.621439	29.64836	2.266971	2.033475	270.6364	20.74047	101.78687	3.0342413	23.241739
2.3	19.17847	13.76267	**1.5679511**	5.006959	8.957620	11.539460	9.918982	3.677494	29.70794	2.211858	1.959609	247.9941	18.54772	97.39158	3.2600643	10.623253
2.4	19.07026	13.75578	**1.4417938**	4.919656	8.997764	11.471419	10.126037	3.521626	29.71088	2.107172	1.860365	229.9678	16.86771	90.29140	3.3546406	9.432669
2.5	19.14573	12.39865	**1.4716844**	5.159139	9.391267	11.895960	10.638143	3.643332	29.68320	2.166328	1.910988	216.7150	16.57998	84.43244	3.6497853	11.088231
2.6	19.19331	13.71996	**1.3049237**	5.059998	9.107785	11.803042	10.414335	3.563594	29.68055	2.062879	1.782631	199.7960	15.47682	81.50642	3.7634760	11.587335
2.7	19.02791	13.45794	**1.2503182**	5.048953	9.247256	12.085053	10.772076	3.484202	29.70186	2.013311	1.750799	188.0345	15.59478	83.83866	3.8865379	9.169348
2.8	19.13952	12.62250	**1.1478723**	4.921742	8.965793	11.459075	10.502786	3.386077	29.63619	1.942884	1.673847	180.3715	14.12685	79.27954	3.9297438	9.212313
2.9	19.15093	12.94850	**1.1386060**	5.066999	9.200025	11.943178	10.786016	3.448583	29.66353	1.933732	1.654667	179.0672	14.91076	77.85483	4.0871876	8.952398
3	19.04499	12.77226	**1.0765951**	5.029597	9.115385	11.975065	10.820288	3.375051	29.66364	1.907751	1.644970	157.4278	13.32368	72.07151	4.2527050	9.186574
3.1	19.11788	13.46875	**1.1622584**	5.055471	9.286233	13.142167	11.209741	3.378380	29.67839	1.953876	1.691230	159.9030	13.94894	70.73339	4.3270537	2667.270385
3.2	19.12337	13.45535	**0.9195216**	5.056728	9.226555	11.955256	11.226868	3.290091	29.66225	1.794280	1.508661	133.4854	10.89720	62.16516	4.5915154	11.418116
3.3	18.89655	13.39585	**0.9271408**	5.087257	9.200267	12.667262	11.272206	3.320845	29.68368	1.821740	1.552473	136.3317	12.08671	62.06331	4.7017560	9.527984
3.4	18.96585	13.98183	**0.9943640**	5.016682	9.231427	12.804399	11.443272	3.261392	29.64364	1.828862	1.554856	136.3229	12.19744	64.44792	4.6421793	8.679033
3.5	19.01799	13.45155	**0.8466721**	5.038250	9.217910	12.296477	11.580343	3.231780	29.64829	1.768576	1.495445	131.7717	12.14054	62.60483	4.8201010	11.259532
3.6	18.88706	13.71035	**0.8922579**	5.004841	9.090914	13.360895	11.577123	3.203560	29.70048	1.767614	1.508003	130.8806	12.03794	60.70485	4.9200924	10.688315
3.7	19.05329	13.98776	**0.8568434**	4.956072	9.178875	12.918004	11.798636	3.101431	29.67868	1.724155	1.475207	118.8133	10.17334	58.10650	4.9643247	8.579437
3.8	18.91835	13.72557	**0.8276112**	4.924357	9.174028	13.374053	11.980610	3.063333	29.70225	1.699069	1.460092	116.5190	10.90365	59.11119	5.0052466	8.567032
3.9	18.94647	13.83107	**0.8351828**	5.155810	9.429906	13.496102	12.237658	3.235766	29.65817	1.773642	1.502971	117.8198	10.76311	55.45508	5.2862896	13.608990
4	18.96601	14.11405	**0.7491045**	4.987828	9.085977	12.955637	11.898488	3.109550	29.65192	1.691943	1.429481	111.5213	10.35499	52.72286	5.2631295	9.527834
ŚREDNIA	19.16351	68.61557	**1.6227514**	4.973828	9.033943	11.201381	10.358357	3.632645	29.64249	2.267569	2.035459	1066.3495	43.16372	160.06145	3.3649159	100.870028

				BŁĄD WZGLĘDNY													
100 x 100	N = 15	INCOM = 15%															
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	17.56956	561.324853	2.0391969	2.627270	4.772600	4.686494	4.779720	2.515973	28.45457	2.0255727	1.9876700	13707.79079	349.068108	973.36565	**0.2018006**	8.356472
1.2	17.55623	157.495118	1.8652410	2.578444	4.708082	4.587810	4.733778	2.376833	28.43773	1.8712074	1.8194207	3743.03311	155.407843	489.03027	**0.3753965**	9.049051
1.3	17.53158	75.967530	1.5259381	2.633607	4.770113	4.766629	4.823543	2.346163	28.47255	1.6394874	1.5492783	1830.62372	96.863766	331.16506	**0.5591217**	8.474644
1.4	17.53675	45.839524	1.6287944	2.611645	4.749901	5.149922	4.839938	2.249913	28.44178	1.6865625	1.6212712	1095.92764	70.111016	251.58251	**0.7140154**	16.675407
1.5	17.57067	28.859660	1.5035638	2.701821	4.926951	5.731641	5.063634	2.269501	28.42677	1.6238126	1.5332299	793.06306	53.864392	202.90219	**0.8760733**	8.323037
1.6	17.54133	16.104878	1.3003329	2.652454	4.844268	5.879040	5.019883	2.166654	28.44149	1.4610043	1.3568135	588.19623	39.717531	165.10212	**0.9946513**	8.319358
1.7	17.51892	14.438701	1.1142820	2.643321	4.804836	5.814332	5.026390	2.115865	28.42739	1.3264658	1.1958318	458.27082	31.099495	142.45901	1.1206277	7.481460
1.8	17.47417	12.649452	**1.1248088**	2.652660	4.843054	6.308693	5.129176	2.071506	28.47436	1.3306175	1.2118605	379.03630	27.805033	123.50304	1.2488585	11.562473
1.9	17.44620	10.245510	**0.9160027**	2.658607	4.835930	6.294194	5.168667	2.039449	28.48762	1.2217269	1.0769572	315.06000	22.467358	108.58824	1.3666117	12.300202
2	17.48754	11.793669	**0.9119701**	2.683446	4.862692	6.426475	5.248273	2.022969	28.41609	1.2090834	1.0522300	283.17300	20.099749	99.76001	1.4813238	7.209268
2.1	17.35157	9.643762	**0.7988279**	2.680674	4.886445	6.579652	5.335908	1.983009	28.48105	1.1265085	0.9707337	248.60545	18.683769	91.90490	1.5746366	8.469397
2.2	17.48374	10.994719	**0.7886020**	2.716942	5.003084	6.686537	5.518463	1.974438	28.42352	1.1371790	0.9745583	228.62177	17.980956	91.60889	1.6578361	7.684215
2.3	17.38174	9.985550	**0.8270707**	2.729655	5.001389	7.397451	5.591114	1.953635	28.48214	1.1394957	0.9820326	205.12552	16.073331	80.29736	1.7803749	7.235851
2.4	17.34786	10.978885	**0.6526492**	2.774271	5.116576	6.832260	5.790109	1.952787	28.47186	1.0694859	0.8927220	189.51567	15.353751	75.25893	1.8919336	8.229062
2.5	17.38455	11.416630	**0.6455287**	2.760649	5.085921	7.185219	5.825702	1.914095	28.46349	1.0534101	0.8778877	167.68595	11.742687	67.58750	1.9917227	9.417241
2.6	17.35505	11.250973	**0.5923524**	2.741617	5.066122	7.174688	5.854339	1.872402	28.47110	1.0051854	0.8316163	157.04409	11.510213	67.60766	2.0482313	6.349716
2.7	17.39768	11.530740	**0.5551547**	2.758795	5.048680	7.140227	5.871358	1.875640	28.43333	1.0071908	0.8240568	150.35027	11.373140	64.87388	2.1384047	11.002171
2.8	17.39555	11.651655	**0.5479667**	2.791256	5.151236	7.410119	6.073372	1.862118	28.42748	1.0042742	0.8219602	139.44073	11.071673	63.63629	2.2371760	6.677101
2.9	17.36423	11.610985	**0.5059128**	2.762752	5.109375	7.263654	6.109188	1.834856	28.44718	0.9825770	0.8020596	137.52428	10.553995	60.07523	2.2636095	6.585978
3	17.27882	11.939180	**0.4445030**	2.769400	5.124577	7.149874	6.183053	1.811215	28.47727	0.9370870	0.7554201	126.44716	9.850103	57.54549	2.3642327	369.888249
3.1	17.21903	12.088469	**0.4594026**	2.778402	5.100282	7.531538	6.195333	1.816854	28.50013	0.9538476	0.7718566	123.41072	9.783112	56.70324	2.4174188	6.854354
3.2	17.21126	12.682974	**0.4176909**	2.789588	5.168410	7.290523	6.401539	1.792452	28.48519	0.9221004	0.7401115	115.03922	8.973862	52.04613	2.5092646	7.635565
3.3	17.21374	11.827315	**0.4359855**	2.771078	5.169086	7.679371	6.500340	1.762155	28.48624	0.9263828	0.7523572	112.28883	10.292019	55.37645	2.5466379	5.802574
3.4	17.22606	12.586082	**0.3579524**	2.778058	5.115051	7.198512	6.414061	1.759804	28.48100	0.9034740	0.7269574	106.63828	9.184148	51.97192	2.6123888	6.491232
3.5	17.24757	12.762806	**0.3973273**	2.794833	5.232974	7.827220	6.698252	1.746825	28.43952	0.9157485	0.7359717	105.74437	9.242133	49.81009	2.6581654	9.756433
3.6	17.16556	12.711824	**0.3567236**	2.805444	5.188109	7.772224	6.664440	1.763140	28.49570	0.9064638	0.7285962	100.16726	8.408228	47.25202	2.7457050	6.935444
3.7	17.25230	13.260068	**0.3493424**	2.796222	5.270708	7.838787	6.934888	1.695636	28.47757	0.8805753	0.7078656	94.80388	7.807079	45.11426	2.8235065	7.288553
3.8	17.15946	13.314994	**0.3594835**	2.833531	5.297039	8.251714	7.052278	1.730305	28.47149	0.8999918	0.7223006	96.17083	8.250863	44.53357	2.8936728	9.026755
3.9	17.19410	13.320021	**0.3413163**	2.815492	5.283644	8.160326	7.045314	1.717998	28.50417	0.8842436	0.7116406	92.30509	7.480085	43.84284	2.9002310	6.533564
4	17.18269	13.577633	**0.3192897**	2.822542	5.281320	8.409701	7.186351	1.696344	28.46948	0.8687308	0.6965595	88.89994	7.796057	41.91926	2.9922853	6.946796
ŚREDNIA	17.36818	39.128472	**0.8027738**	2.730483	5.027282	6.814161	5.835947	1.956351	28.46231	1.1639831	1.0143942	866.00013	36.263850	136.54747	1.8661972	20.418721

				BŁĄD WZGLĘDNY												
100 x 100	N = 8	INCOM = 4%														
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	4.795406	376.415364	1.0082741	2.852310	4.992939	4.913245	4.998768	2.745548	6.070246	1.6300147	1.4527525	4694.29131	124.921095	591.22168	**0.2166438**	216.732681
1.2	4.882833	97.579056	1.1138085	2.861177	5.020670	4.714333	5.043366	2.661920	6.020756	1.6790509	1.5210050	1253.94183	54.787826	301.92748	**0.4203884**	68.513785
1.3	4.777450	35.019926	0.9595231	2.860377	4.998739	4.533723	5.045973	2.581166	6.041379	1.5667816	1.3947507	563.94923	30.017945	191.54899	**0.6057543**	6.430051
1.4	4.711613	28.293165	**0.7573710**	2.890074	5.072433	4.583279	5.147705	2.539961	6.028320	1.4310345	1.2427876	370.61247	22.867306	154.91220	0.7593978	137.955419
1.5	4.741167	15.964511	**0.7752329**	2.983462	5.294846	4.768993	5.414245	2.541754	6.000930	1.4549198	1.2634734	254.74122	17.161562	124.39891	0.9502978	9.006036
1.6	4.675439	14.123711	**0.7214202**	2.850985	5.036444	4.692883	5.191821	2.377617	5.977688	1.3726730	1.1922653	197.42441	14.799875	107.01517	1.0734457	5.572396
1.7	4.701481	11.689979	**0.7769618**	2.947336	5.063651	4.901982	5.238621	2.439381	5.988741	1.4376993	1.2518634	153.17899	12.635932	92.24823	1.2307220	5.997389
1.8	4.694653	9.131828	**0.5615219**	2.856780	5.009176	4.832127	5.246348	2.290345	5.936896	1.2632793	1.0680077	119.44925	10.611606	81.00170	1.3534759	4.746416
1.9	4.705096	8.249270	**0.5400422**	2.941004	5.203838	5.130304	5.488284	2.302050	5.930423	1.2624290	1.0611476	116.59198	10.088939	73.56367	1.4868621	5.958548
2	4.699081	7.215551	**0.5445838**	2.868534	5.054347	5.067104	5.388827	2.218851	5.894635	1.2339893	1.0448315	91.04412	8.779349	69.21975	1.5685283	4.306846
2.1	4.730363	6.329138	**0.5016029**	2.932923	5.157505	5.052926	5.549696	2.240258	5.859758	1.2178831	1.0178468	79.25441	8.168419	59.85022	1.6937280	5.487386
2.2	4.741233	6.550953	**0.3705087**	2.993343	5.268587	5.098261	5.715994	2.234452	5.873044	1.1536520	0.9371741	73.99491	7.426750	54.37719	1.8782314	8.657534
2.3	4.623911	5.650922	**0.3952158**	2.943696	5.257226	5.176077	5.760062	2.170156	5.926945	1.1430881	0.9391127	65.02249	6.901967	55.15956	1.9273775	4.098142
2.4	4.719966	5.744147	**0.4300704**	2.888083	5.019185	5.163593	5.532563	2.137691	5.850835	1.1625855	0.9533787	62.29553	7.057969	51.68420	1.9872425	4.310176
2.5	4.675235	6.007579	**0.4520516**	2.888883	5.113506	5.254277	5.692615	2.113081	5.875332	1.1648383	0.9697754	62.31465	7.190691	53.03144	2.0183045	5.068753
2.6	4.755418	5.633823	**0.4150496**	2.864056	5.058937	5.215594	5.651310	2.069725	5.853569	1.1223203	0.9312580	54.66882	6.556127	49.15058	2.0698498	5.924728
2.7	4.701174	5.324364	**0.3866014**	2.875410	5.116122	5.372652	5.787093	2.017511	5.841202	1.1017728	0.9096190	46.24743	5.892882	44.99666	2.2439364	3.975270
2.8	4.652964	4.970389	**0.3458733**	2.850066	5.070572	5.251777	5.815757	1.997483	5.893060	1.0641115	0.8754495	44.95714	5.661824	42.99197	2.2928113	3.656271
2.9	4.615273	5.064411	**0.3376976**	2.791617	4.922718	5.134301	5.653117	1.963195	5.865530	1.0503169	0.8541676	44.88674	5.596012	42.06830	2.3262247	4.049930
3	4.726685	4.827436	**0.3296210**	2.961941	5.126317	5.273093	5.903843	2.076581	5.835251	1.0982774	0.8845799	42.09139	5.874257	42.69812	2.4900855	9.508435
3.1	4.753531	5.137105	**0.3732540**	2.934766	5.280134	5.508225	6.207245	1.992938	5.833774	1.0867209	0.8947591	44.28094	6.208131	42.68122	2.4941180	4.032267
3.2	4.645642	4.811215	**0.3175031**	2.945837	5.284391	5.531012	6.245251	1.961448	5.885703	1.0462402	0.8573578	39.58669	5.483255	37.55445	2.6188871	7.678748
3.3	4.702373	5.147858	**0.2715509**	3.039590	5.383959	5.463374	6.345116	2.027094	5.837621	1.0517497	0.8428286	41.85610	6.049626	39.32905	2.7069742	5.321808
3.4	4.597561	4.670057	**0.2764797**	2.861914	5.118361	5.371533	6.124834	1.901024	5.859182	1.0103052	0.8128247	35.90442	5.446828	38.53230	2.6798121	7.277101
3.5	4.666226	4.672876	**0.2434911**	2.909842	5.101492	5.237466	6.135105	1.927679	5.846602	1.0053553	0.7991912	34.75780	5.194657	34.87026	2.8224301	3.766544
3.6	4.737565	4.662391	**0.3075332**	2.983118	5.245877	5.769896	6.359515	1.977056	5.834953	1.0599039	0.8557589	34.84559	5.255690	35.56453	2.8749993	3.800782
3.7	4.726691	4.977392	**0.2865611**	2.969628	5.321237	5.735630	6.465364	1.935199	5.829738	1.0331193	0.8284478	37.93142	5.610163	38.99497	2.8721089	3.944733
3.8	4.735247	4.805296	**0.2317841**	2.861728	5.027648	5.328303	6.277554	1.849443	5.804007	0.9790992	0.7773163	30.05421	4.803971	32.01443	3.0182603	7.806664
3.9	4.705977	4.639948	**0.2734816**	2.846286	5.061826	5.494333	6.265986	1.841799	5.826026	0.9956664	0.8019527	30.60004	4.907009	34.05190	2.9467213	3.101696
4	4.679112	4.560662	**0.2152825**	2.873391	5.038081	5.262583	6.245388	1.879683	5.829654	0.9827681	0.7737137	31.45237	4.814470	35.48915	2.9965904	2.995041
ŚREDNIA	4.709212	23.595677	**0.4839984**	2.904272	5.124025	5.161096	5.731246	2.167070	5.898393	1.1953882	1.0003132	291.74093	14.225738	88.40494	1.9541403	18.989386

				BŁĄD WZGLĘDNY												
100 x 100	N = 8	INCOM = 7%														
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	9.669102	811.780796	2.0582587	4.224084	7.468723	7.319700	7.478766	4.056850	12.23471	2.723380	2.544089	8301.56376	231.084424	935.57215	**0.3274788**	8.746293
1.2	9.579291	197.166845	1.9730716	4.290102	7.689627	7.223827	7.726669	3.978765	12.30345	2.653050	2.471618	2497.85609	104.887876	461.83384	**0.6150523**	12.161917
1.3	9.490400	98.924341	1.7960725	4.329007	7.606873	6.912825	7.679724	3.895177	12.26850	2.539399	2.339698	1165.16782	61.264787	321.36176	**0.9068565**	9.172277
1.4	9.418728	54.563922	1.6915485	4.132406	7.384121	6.655588	7.510999	3.609831	12.28334	2.377422	2.201217	688.77448	41.014870	237.88073	**1.1223514**	20.154431
1.5	9.398343	32.138227	1.4845962	4.254185	7.492445	7.031007	7.662952	3.622558	12.22671	2.253365	2.039464	484.39570	28.734274	187.81605	**1.3923284**	8.640615
1.6	9.380470	22.556464	**1.2143088**	4.221019	7.399744	6.912899	7.628521	3.513523	12.27188	2.059403	1.828763	372.01423	24.217726	157.22647	1.6074803	13.693931
1.7	9.515793	18.237108	**1.4628654**	4.234166	7.604358	7.160360	7.929132	3.427354	12.24541	2.199560	2.004105	293.49109	20.129727	138.69255	1.8193579	8.771672
1.8	9.554597	16.805019	**1.3297914**	4.287300	7.705864	7.524017	8.110120	3.408169	12.19454	2.120004	1.909891	262.89872	19.711214	128.91008	2.0089493	298.619204
1.9	9.292938	14.061818	**1.1803868**	4.213057	7.514885	7.375444	7.974808	3.309371	12.28226	1.998951	1.787412	218.65397	17.277817	113.87811	2.1492906	9.879939
2	9.308684	10.371632	**0.9532221**	4.232017	7.563830	7.505233	8.081916	3.244427	12.25008	1.858523	1.616716	179.20012	13.885051	100.44563	2.3309045	8.530573
2.1	9.482386	10.251606	**1.0663069**	4.343776	7.803944	8.109013	8.444227	3.297250	12.17972	1.930188	1.693823	161.86483	13.309739	101.02089	2.5037878	6.732683
2.2	9.498069	11.944297	**1.0387978**	4.466177	7.920480	8.203568	8.628175	3.351817	12.13752	1.959825	1.709636	148.72734	13.740296	93.37520	2.7124597	8.628219
2.3	9.339281	9.507294	**0.9609310**	4.329676	7.739636	7.964852	8.521207	3.188542	12.23778	1.859649	1.624150	135.99146	11.590435	84.24414	2.8288187	6.976810
2.4	9.357524	9.431446	**0.8408512**	4.260205	7.587272	7.995460	8.441489	3.093825	12.20559	1.769094	1.528607	120.78296	10.939179	80.08750	2.9725669	12.869765
2.5	9.557463	9.910220	**0.8694187**	4.303540	7.745660	8.088092	8.655160	3.090218	12.15527	1.775036	1.523843	117.12699	11.052582	76.95583	3.0379564	13.997461
2.6	9.407256	8.447498	**0.7977515**	4.400842	7.823825	8.174591	8.868139	3.108291	12.23103	1.752841	1.503906	106.75615	10.601738	68.73464	3.3326229	8.371676
2.7	9.390641	8.980670	**0.8146945**	4.429451	7.950186	8.396732	9.120513	3.091911	12.21972	1.751017	1.500444	98.98848	9.857244	70.97283	3.4021079	13.240047
2.8	9.501169	8.103783	**0.8070235**	4.342539	7.870678	8.304870	9.092012	2.974754	12.16375	1.712135	1.479683	93.06693	9.724020	63.40738	3.4612367	12.239293
2.9	9.234262	9.106581	**0.6961130**	4.258974	7.668685	8.345876	8.917447	2.900904	12.25843	1.633238	1.396648	90.45803	9.875001	68.90096	3.5378035	6.808457
3	9.345953	8.862775	**0.8192046**	4.280812	7.631489	8.925436	8.927399	2.944867	12.20989	1.710897	1.472335	85.16038	8.968621	67.53116	3.6035984	5.569430
3.1	9.349566	9.015516	**0.7174110**	4.287924	7.751831	8.781579	9.253381	2.905294	12.23830	1.640712	1.394948	81.12074	9.015511	64.45614	3.7360520	6.813968
3.2	9.392647	8.334275	**0.6277343**	4.278895	7.666959	8.358558	9.085198	2.890054	12.15938	1.588464	1.330165	81.59608	9.437004	66.43921	3.7351867	7.964656
3.3	9.180400	7.938747	**0.6370362**	4.350835	7.845878	9.003873	9.527923	2.845905	12.26415	1.589598	1.347875	71.86586	8.226050	54.58276	4.0755738	6.007164
3.4	9.308680	8.013190	**0.6208209**	4.378108	7.860490	8.619323	9.496450	2.847121	12.24485	1.586586	1.343988	74.29435	8.974553	59.46628	4.1199792	28.279982
3.5	9.348322	7.940013	**0.6553633**	4.403825	7.936828	9.031351	9.620607	2.921393	12.24061	1.633200	1.381476	76.91328	8.405980	60.59369	4.0013168	29.781157
3.6	9.268814	7.975444	**0.5525101**	4.421031	7.908171	8.912042	9.654482	2.886273	12.22840	1.570758	1.312959	70.92769	7.881906	53.03548	4.2472138	5.850946
3.7	9.429665	8.595940	**0.6193461**	4.284762	7.795334	8.833710	9.739832	2.757674	12.18386	1.551010	1.320777	68.20947	8.111160	56.08204	4.2189679	6.254166
3.8	9.405135	8.038341	**0.5205660**	4.407564	7.906588	8.407603	9.905359	2.854839	12.21278	1.546962	1.281079	67.06381	8.736570	55.32449	4.3974227	25.991780
3.9	9.247104	8.073315	**0.5827333**	4.268761	7.797569	9.307897	10.066197	2.720022	12.23380	1.520809	1.287614	62.18861	8.220509	51.96996	4.4285055	6.728859
4	9.426998	8.052710	**0.4406813**	4.390464	7.764700	8.221170	9.804845	2.787942	12.16991	1.490132	1.226145	60.82915	8.088015	50.30278	4.6239303	6.841231
ŚREDNIA	9.402656	48.437661	**0.9943139**	4.310183	7.713556	8.053550	8.718455	3.184164	12.22452	1.878507	1.646769	544.59829	25.232129	137.70336	2.9085719	20.810620

				BŁĄD WZGLĘDNY												
100 x 100	N = 8	INCOM = 14%														
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	18.84152	1380.56840	4.276483	6.540891	11.66314	11.45576	11.67802	6.284145	27.15486	4.746101	4.595988	18281.6235	423.39123	1254.40333	**0.4911508**	27.733524
1.2	18.98317	398.44518	3.933444	6.395091	11.33221	10.69441	11.38520	5.941807	27.05923	4.434740	4.253243	5007.5868	197.12184	625.01832	**0.9307917**	19.698494
1.3	19.02154	156.69404	3.834747	6.588464	11.71284	10.74484	11.82972	5.930999	27.02225	4.343820	4.141708	2379.1825	107.55081	412.69935	**1.3693302**	17.839280
1.4	18.88624	108.72052	3.280450	6.560343	11.50421	10.61759	11.68689	5.755410	27.13497	3.926656	3.699113	1471.1698	82.98087	317.21497	**1.7674407**	90.330368
1.5	19.27548	64.64928	3.386560	6.639580	11.75756	10.82868	12.04151	5.658913	26.85011	4.056489	3.815959	983.2775	62.13825	243.64467	**2.1537897**	41.405128
1.6	18.71770	61.81459	3.124545	6.666622	11.82349	11.38801	12.21271	5.552513	27.18080	3.800980	3.571537	724.4830	50.25399	218.59563	**2.4895289**	4628.064119
1.7	18.97216	39.40718	3.046623	6.562829	11.66297	11.37879	12.11377	5.371750	26.91740	3.712912	3.459259	639.9828	44.35330	196.70967	**2.7083752**	14.490401
1.8	18.97488	35.41922	**2.729135**	6.656315	11.81176	11.60041	12.37403	5.345534	27.03301	3.502579	3.216956	516.1431	37.30338	167.58361	3.0303392	15.398438
1.9	18.94712	24.05590	**2.370529**	6.407491	11.33996	11.73840	11.99218	5.048480	27.01854	3.226376	2.920538	424.0963	28.17104	143.05471	3.2260062	13.328645
2	18.77630	20.62999	**2.190705**	6.796029	12.07740	12.08114	12.88030	5.200780	27.22009	3.170826	2.871400	357.2463	25.20113	125.32955	3.7276657	13.504643
2.1	18.62903	19.01073	**2.250584**	6.627158	11.94765	12.93251	12.92781	4.983691	27.24760	3.151492	2.878845	319.5536	23.14272	121.73782	3.8791485	15.007788
2.2	18.89108	19.31895	**2.253798**	6.517660	11.65142	13.24615	12.73689	4.848933	27.13328	3.110293	2.834295	291.5357	23.24106	114.59771	4.0771799	143.936731
2.3	18.67505	16.45404	**1.913008**	6.587528	11.88679	12.73683	13.17484	4.836744	27.19297	2.931988	2.639504	269.5329	21.10796	106.66492	4.3698068	24.187205
2.4	18.80539	16.94159	**2.112260**	6.772549	12.41134	14.12003	13.95072	4.834368	27.06627	3.039301	2.769154	256.2908	20.45627	106.00176	4.6082234	12.054336
2.5	18.94695	16.64769	**1.926879**	6.852281	12.17347	13.57977	13.65754	4.947602	27.06104	2.990105	2.647606	238.8623	19.14742	99.31507	4.8885535	13.024772
2.6	19.02048	14.22338	**2.060830**	6.566687	11.91249	14.16263	13.52963	4.629372	27.06682	2.931882	2.652550	226.6293	18.04547	94.86801	4.8224163	11.969407
2.7	18.65451	15.50560	**1.615286**	6.500659	11.66370	12.78626	13.34068	4.555251	27.19168	2.698066	2.396175	193.8083	14.94575	85.86333	5.0565067	40.880375
2.8	18.66349	14.85702	**1.674530**	6.579373	11.77591	14.34886	13.65577	4.577182	27.16039	2.737943	2.437637	184.8533	17.17717	90.18605	5.2934129	18.259687
2.9	18.61519	14.99938	**1.615197**	6.620554	11.79493	13.88172	13.69251	4.608135	27.19913	2.722537	2.403117	179.8465	15.65538	82.84583	5.4304562	17.614642
3	18.84278	14.01018	**1.758300**	6.645717	12.06907	14.82652	14.25252	4.535552	27.10595	2.765489	2.474733	179.6191	16.86100	87.18549	5.5026602	15.569278
3.1	18.70323	14.06581	**1.644228**	6.914173	12.49351	14.71432	14.89422	4.678986	27.06200	2.783943	2.447092	170.4850	15.22740	82.18971	5.9181030	11.161737
3.2	18.80801	14.62396	**1.540251**	6.805745	12.36970	14.84074	15.00553	4.558679	27.06739	2.688016	2.371954	158.8648	14.96884	80.87418	5.9728467	23.327533
3.3	18.71869	14.68209	**1.428649**	6.684089	12.07655	14.87867	14.63691	4.438534	27.13524	2.580746	2.275282	149.4636	13.83424	73.61663	6.0661192	11.092838
3.4	18.50076	14.07994	**1.286642**	6.739628	11.93900	13.99003	14.39320	4.486522	27.26855	2.538066	2.214691	141.2916	12.21512	68.65441	6.2648157	20.318102
3.5	18.58892	14.42974	**1.416173**	6.868842	12.55732	15.19001	15.47136	4.440673	27.18162	2.606390	2.321747	160.5193	16.49471	73.79900	6.4071552	67.538824
3.6	18.56459	15.01913	**1.384827**	6.517174	11.80034	15.62790	14.90408	4.227826	27.17859	2.503628	2.202075	134.5521	12.59851	67.56976	6.4390836	823.182454
3.7	18.50317	15.28410	**1.174952**	6.736836	12.19708	14.75960	15.32690	4.296281	27.15379	2.443586	2.149894	128.9359	12.86709	66.73240	6.6202435	10.315898
3.8	18.66983	14.62382	**1.432071**	6.652743	12.12449	16.59446	15.51645	4.276941	27.16814	2.541021	2.257156	132.1933	12.68634	63.35141	6.7165740	9.948774
3.9	18.67389	14.82228	**1.232047**	6.551720	11.80468	14.97930	14.81065	4.198480	27.19080	2.401928	2.115308	129.2595	13.37335	69.06716	6.6092682	21.253758
4	18.60592	14.32043	**1.187077**	6.671648	12.06538	15.50699	15.49438	4.212125	27.15213	2.406062	2.106722	136.7782	12.93419	65.82105	7.0283914	9.655311
ŚREDNIA	18.78257	86.61081	**2.169360**	6.640881	11.91335	13.34104	13.51890	4.908740	27.11915	3.116465	2.838041	1152.2556	46.18153	180.17318	4.4621794	206.736416

				BŁĄD WZGLĘDNY											
100 x 100	N = 8	INCOM = 25%													
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	33.27397	2167.61704	8.667965	9.708629	17.41384	17.09729	17.43810	9.318967	56.50863	8.091912	7.896638	30498.7551	604.00505	1247.76110	**0.7556738**	59.03776
1.2	33.26771	649.16782	8.254284	9.709155	17.24137	16.38328	17.32399	9.020478	56.43754	7.717264	7.518478	8539.7593	299.15529	628.96426	**1.4252361**	794.56637
1.3	33.25860	212.87764	7.739146	10.039969	18.23517	16.88742	18.42461	8.973254	56.33939	7.395209	7.119228	4148.4523	172.49305	421.47742	**2.0955663**	28.67632
1.4	33.18833	166.47336	7.549028	10.074234	18.02536	16.74966	18.33468	8.774590	56.54340	7.191381	6.952180	2544.4338	119.86144	308.75658	**2.7188123**	131.94135
1.5	33.12200	130.65807	7.006303	9.882705	17.84185	16.69119	18.28583	8.380008	56.55297	6.730648	6.435188	1823.1573	103.37509	283.57065	**3.2244349**	20.98798
1.6	33.19788	75.84050	5.648614	9.526078	16.80337	15.99293	17.34343	7.919992	56.45605	5.789465	5.373543	1293.7332	78.22087	209.79330	**3.7174740**	28.51895
1.7	33.01180	55.30371	5.644402	9.919544	17.55612	17.60032	18.24625	8.080315	56.73235	5.837300	5.463515	1104.5742	66.51036	197.72175	**4.2366852**	29.85933
1.8	33.19840	44.51685	5.900373	10.200397	18.06381	18.73182	18.99283	8.175194	56.51742	6.015686	5.603883	845.1810	53.71022	165.61927	**4.7461707**	50.15336
1.9	33.14926	42.69200	5.407418	10.205638	18.05340	19.58401	19.10598	8.062238	56.84736	5.731639	5.319260	768.1556	52.11191	156.23817	**5.0785893**	39.15778
2	33.18435	34.27708	**5.028279**	10.296831	18.10393	19.95644	19.34878	8.011683	56.49521	5.514039	5.044545	657.3497	46.28768	148.37807	5.6572024	23.19669
2.1	33.08985	27.50300	**4.675644**	10.011611	17.99358	19.86580	19.44072	7.566916	56.70336	5.165861	4.725523	570.7008	34.90985	124.01684	5.8501187	27.48661
2.2	32.85764	32.52958	**4.858384**	10.123236	18.14617	22.16668	19.82208	7.616698	56.61864	5.314048	4.874044	558.3622	38.91008	128.87692	6.1447653	33.04977
2.3	32.83953	26.90283	**4.385614**	9.932169	17.97866	21.79499	19.88118	7.286102	56.82598	4.935822	4.525513	468.4167	28.83005	113.01159	6.4559833	26.21025
2.4	32.93303	23.92129	**4.223063**	10.036646	18.10615	21.94267	20.17265	7.241155	56.65068	4.878057	4.457805	419.0674	26.48633	99.89864	6.8853661	22.86739
2.5	32.91167	25.93449	**4.112521**	10.165664	18.36433	22.37357	20.60219	7.351573	56.88066	4.821757	4.351489	428.3843	29.51134	105.28410	7.0462214	27.89257
2.6	32.68090	24.02396	**3.663412**	10.133058	18.03429	22.30581	20.42042	7.231746	56.84520	4.589262	4.130574	369.5102	24.26375	92.09207	7.6168625	20.81917
2.7	32.73641	22.22359	**3.880912**	10.149503	18.45823	24.25469	21.27132	7.104578	56.75986	4.657201	4.239247	376.8402	25.41311	89.59679	7.7739591	19.96887
2.8	32.82983	25.80247	**3.350678**	10.019811	18.10659	23.32471	21.03238	6.895607	56.73503	4.339569	3.918475	320.5805	23.43607	84.54474	8.1889744	34.78766
2.9	32.89967	26.06461	**3.355775**	10.172519	18.22184	22.76265	21.26278	7.043527	56.68495	4.413465	3.969507	322.4148	23.64889	82.54323	8.4050766	23.35472
3	32.72796	22.70885	**3.532904**	10.145599	18.37452	25.50166	21.76123	6.947106	56.93435	4.422843	4.002612	316.6389	23.28431	81.53083	8.5650813	51.58822
3.1	32.70094	22.83418	**3.366026**	10.309559	18.79379	26.29278	22.63436	6.966400	56.94169	4.363111	3.955648	277.0681	18.93169	78.66991	8.8314480	21.00936
3.2	32.81334	25.27608	**2.979770**	9.666090	17.60724	23.62627	21.13901	6.397653	56.73439	3.986866	3.595711	289.7300	21.15962	73.84603	8.7730491	81.83718
3.3	32.52185	23.47277	**3.142592**	10.034646	18.26365	26.84532	22.34037	6.633055	56.83668	4.153788	3.758980	275.9330	21.00706	73.26384	9.2708850	18.32622
3.4	32.76044	22.99539	**2.934914**	10.295019	18.35625	23.86474	22.09718	6.852861	56.76935	4.135544	3.685739	281.4038	21.62034	73.42178	9.5498517	45.22912
3.5	32.58742	22.48396	**2.768099**	10.141888	18.31949	24.93995	22.52887	6.584166	56.80836	4.015243	3.635323	229.9443	17.78027	68.70669	9.8466610	45.66032
3.6	32.52885	22.81964	**2.926733**	10.305723	18.58233	26.82863	23.21699	6.764944	56.92022	4.109104	3.694127	240.2062	19.85092	72.17812	9.9707048	245.59437
3.7	32.88165	23.70262	**2.943305**	10.137767	18.32859	26.09952	22.70875	6.612565	56.69127	4.053292	3.607151	259.6233	20.47725	73.83161	9.7610554	21.03131
3.8	32.43938	23.52791	**2.702648**	10.035371	18.28389	27.15501	23.32693	6.486587	56.91548	3.890431	3.510025	225.5716	17.45431	70.25429	10.1419550	24.43111
3.9	32.53215	23.15361	**2.654579**	10.213915	18.75767	27.21726	24.34397	6.399643	56.73282	3.910633	3.570513	207.9116	15.68872	60.67068	10.7915400	53.51946
4	32.71741	23.19368	**2.406588**	9.802187	17.96953	26.18771	23.48182	6.086142	56.93415	3.654093	3.385700	205.7206	16.43708	61.58403	10.6480982	18.78395
ŚREDNIA	32.89474	135.68329	**4.523666**	10.046505	18.07950	22.03416	20.54432	7.426191	56.71178	5.127484	4.744005	1962.2527	68.82773	182.53678	6.8057834	68.98478

				BŁĄD WZGLĘDNY												
100 x 100	N = 8	INCOM = 50%														
	saaty	geometric	koczkodaj	kazibudzkiLTI1	kazibudzkiLTI2	kazibudzkiCMLT	pelaeLamata	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	kulakowskiSzybo	harmonic	goldenWang	saloHamalainen	cavalloDapuzzo	relativeError
1.1	66.40734	3459.49916	28.32739	23.01463	40.38424	40.16536	40.43226	22.16645	198.5462	22.19062	21.04846	63595.4753	875.15110	1211.81034	**1.744506**	2221.06118
1.2	66.64123	715.61436	26.15939	22.74248	39.52716	38.82283	39.70269	21.16023	198.1683	20.54345	19.10415	16658.6106	387.33950	540.93179	**3.376876**	349.52123
1.3	66.19302	484.68624	24.60145	23.36428	40.67100	40.43895	41.02318	21.17333	200.7861	19.46437	18.04342	8496.4316	263.27808	392.04853	**4.816198**	24369.41887
1.4	66.14478	190.89132	24.38358	23.50402	41.06769	41.49169	41.70598	20.63688	199.7127	19.22366	17.79100	4880.9755	182.63774	259.05688	**6.379110**	122.44582
1.5	66.13906	155.90329	21.80958	22.89301	39.69029	40.93284	40.51414	19.67095	200.6584	17.46797	15.93133	3513.2915	153.90942	236.74615	**7.467218**	59.39174
1.6	65.99278	105.84756	21.40091	23.02640	40.27232	43.81693	41.41902	19.37811	201.6641	17.09677	15.70917	2911.8760	126.19905	188.37545	**8.569808**	124.04684
1.7	66.06488	72.20895	20.50430	23.14761	40.28343	45.34432	41.75652	19.15713	201.4251	16.43062	14.91987	2244.1379	93.15431	161.32411	**9.789063**	64.61912
1.8	66.24217	62.37210	19.30994	23.57560	41.01554	46.87900	42.78939	19.13600	200.7294	16.04038	14.67412	1901.9206	86.38996	142.64079	**10.940181**	444.63874
1.9	66.00807	48.99403	18.16783	23.26712	40.67831	49.22700	42.87493	18.37631	202.3648	15.06586	13.72356	1379.5283	59.24932	114.89681	**12.149239**	69.80642
2	65.61347	46.56568	18.32738	23.15622	40.85463	54.80739	43.59015	18.03506	202.2484	14.96471	13.67321	1378.2659	66.18252	111.26896	**12.870979**	64.08887
2.1	65.50982	42.07334	16.62715	22.91071	39.91102	53.98172	42.84409	17.55715	202.1258	13.84961	**12.58981**	1067.2519	44.76605	83.76710	14.102119	541.54031
2.2	65.70964	44.15514	17.46815	23.25739	41.33632	59.64094	44.83296	17.62248	201.5678	14.38213	**12.98010**	1088.7672	55.68172	90.26395	14.479159	300.81596
2.3	65.82751	41.33474	16.30075	23.20291	40.91730	58.01909	44.66997	17.27028	202.5410	13.78803	**12.56555**	933.6882	42.58664	70.52487	15.474174	54.68842
2.4	65.61907	44.33374	16.02005	22.86488	40.61608	61.17014	44.86420	16.69620	201.7641	13.44619	**12.20556**	854.5620	42.40485	75.00634	16.217169	56.43785
2.5	65.57323	47.05665	15.03516	22.77197	40.32240	61.47872	44.87500	16.59336	203.1662	12.92646	**11.76661**	774.5263	37.78611	70.01164	16.789869	1568.23503
2.6	65.55683	41.99469	15.05706	22.71266	40.30732	67.77379	45.35692	16.33430	202.9874	12.66336	**11.53342**	727.9561	37.29048	60.98454	17.637741	188.26397
2.7	65.44990	43.95729	14.39273	23.27154	41.41132	65.17217	46.93231	16.38916	202.0309	12.73100	**11.77859**	690.6799	34.14302	57.02805	18.771641	187.98725
2.8	65.35144	42.68430	13.56313	22.62843	39.94125	65.21246	45.58229	15.97824	202.8432	12.08768	**11.16287**	658.9339	30.21069	56.53863	19.000777	96.28281
2.9	65.31444	44.52658	13.31417	22.93304	40.56287	65.58173	46.51298	16.13015	203.0770	11.99508	**11.21224**	664.7804	31.78285	58.97365	19.259787	61.65942
3	65.74010	45.70196	13.58085	23.00289	40.72377	74.56550	47.57830	15.79486	202.0039	11.93421	**10.94629**	565.1963	28.55905	46.20767	20.737719	115.46222
3.1	64.98157	46.49417	12.84266	23.23358	41.29184	72.81072	48.41054	15.82039	203.4099	11.60843	**10.89054**	525.8839	26.67448	44.17900	21.241790	46.29859
3.2	65.29414	44.01168	12.88117	23.15735	41.11596	73.24102	48.30779	15.73653	202.9554	11.63218	**10.87188**	584.4333	31.00311	48.41265	21.232841	63.50410
3.3	65.14282	46.50541	12.56079	23.27784	41.16330	73.35110	48.91982	15.87062	202.3149	11.61873	**10.84956**	525.8847	27.16285	47.39402	21.831170	76.92566
3.4	64.86465	45.08714	12.19783	23.50305	41.47487	74.82363	49.63962	16.00510	203.5047	11.52666	**10.89846**	509.0470	26.83197	46.14597	22.891705	103.51722
3.5	64.86293	46.54614	11.69320	23.06720	41.15622	76.64867	49.76501	15.37590	204.2826	11.06410	**10.52410**	486.5743	28.31828	43.31545	23.109490	41.02121
3.6	65.01376	46.27499	11.52799	23.42578	41.40562	75.09423	49.85825	15.38946	202.3788	10.98683	**10.55895**	479.8847	24.29683	40.16975	23.883227	76.48649
3.7	64.92148	46.54869	11.45244	22.96158	41.62225	76.39478	50.98558	14.93113	204.1553	10.81640	**10.39345**	444.5415	25.47704	43.28498	23.847627	40.94275
3.8	64.96659	46.89181	11.23273	22.87394	41.11070	80.27604	51.08272	14.84509	204.2389	10.62591	**10.16979**	433.6336	24.81860	41.99130	24.445969	80.16109
3.9	64.88719	45.54133	11.00749	23.15374	41.60436	78.84606	51.83168	14.95058	205.3797	10.55159	**10.27025**	460.0260	28.96584	41.15243	24.920958	59.32123
4	64.83965	45.44513	10.62371	22.92156	40.69130	78.74472	50.58682	14.84886	204.9566	10.43966	**10.27712**	430.6153	24.91579	37.78947	25.274271	60.06963
ŚREDNIA	65.56245	207.99159	16.41237	23.09411	40.77102	61.15845	45.64150	17.30101	202.2663	13.97209	**12.96878**	3995.5793	98.23891	148.74138	16.108413	1056.95534