Paper

Data analysis and flow graphs

Zdzisław Pawlak

Abstract—In this paper we present a new approach to data analysis based on flow distribution study in a flow network. Branches of the flow graph are interpreted as decision rules, whereas the flow graph is supposed to describe a decision algorithm. We propose to model decision processes as flow graphs and analyze decisions in terms of flow spreading in the graph.

Keywords—data mining, data independence, flow graph, Bayes' rule.

1. Introduction

In this paper we present a new approach to data analysis based on flow distribution study in a flow network, different to that proposed by Ford and Fulkerson [2], called here a flow graph. Branches of the flow graph are interpreted as decision rules, whereas the flow graph is supposed to describe a decision algorithm. Thus we propose to model decision processes as flow graphs and analyze decisions in terms of flow spreading in the graph.

With every decision rule three coefficients are associated, called strength, certainty and the coverage factors. These coefficients have a probabilistic flavor, but it will be shown in the paper that they can be also interpreted in a deterministic way, describing flow distribution in the flow graph. Moreover, it is shown that these coefficients satisfy Bayes' rule. Thus, in the presented approach Bayes' rule has entirely deterministic interpretation, without reference to its probabilistic nature, inherently associated with classical Bayesian philosophy. This leads to new philosophical and practical consequences. A simple example, of a telecom customer, which is a slight modification of example given in [4], will be used to illustrate ideas presented in the paper.

This paper is a continuation of ideas given in [4–7] and refers to some thoughts presented in [3].

2. Flow graphs

A flow graph is a *directed, acyclic, finite* graph $G = (N, \mathscr{B}, \varphi)$, where N is a set of *nodes*, $\mathscr{B} \subseteq N \times N$ is a set of *directed branches*, $\varphi : \mathscr{B} \to R^+$ is a *flow function* and R^+ is the set of non-negative reals.

If $(x, y) \in \mathscr{B}$ then x is an *input* of y and y is an *output* of x.

If $x \in N$ then $I(x)$ is the set of all inputs of x and $O(x)$ is the set of all outputs of x.

Input and *output* of a graph G *are* defined
$I(G) = \{x \in N : I(x) = \varnothing\}$, $O(G) = \{x \in N : O(x) = \varnothing\}$.

Inputs and outputs of G are *external nodes* of G; other nodes are *internal nodes* of G.

If $(x, y) \in \mathscr{B}$ then $\varphi(x, y)$ is a *troughflow* from x to y. We will assume in what follows that $\varphi(x, y) \neq 0$ for every $(x, y) \in \mathscr{B}$.

With every node x of a flow graph G we associate its *inflow*:

$$\varphi_+(x) = \sum_{y \in I(x)} \varphi(y, x) \tag{1}$$

and *outflow*

$$\varphi_-(x) = \sum_{y \in O(x)} \varphi(x, y). \tag{2}$$

Similarly, we define an inflow and an outflow for the whole flow graph G, which are defined as

$$\varphi_+(G) = \sum_{x \in I(G)} \varphi_-(x), \tag{3}$$

$$\varphi_-(G) = \sum_{x \in O(G)} \varphi_+(x). \tag{4}$$

We assume that for any internal node x, $\varphi_+(x) = \varphi_-(x) = \varphi(x)$, where $\varphi(x)$is a *troughflow* of node x.

Obviously, $\varphi_+(G) = \varphi_-(G) = \varphi(G)$, where $\varphi(G)$ is a *troughflow* of graph G.

The above formulas can be considered as *flow conservation equations* [2]. We will define now a *normalized flow graph*.

A normalized flow graph is a *directed, acyclic, finite* graph $G = (N, \mathscr{B}, \sigma)$, where N is a set of *nodes*, $\mathscr{B} \subseteq N \times N$ is a set of *directed branches* and $\sigma : \mathscr{B} \to <0, 1>$ is a *normalized flow* of (x, y) and

$$\sigma(x, y) = \frac{\varphi(x, y)}{\varphi(G)} \tag{5}$$

is *strength* of (x, y). Obviously, $0 \leq \sigma(x, y) \leq 1$. The strength of the branch expresses simply the percentage of a total flow through the branch.

In what follows we will use normalized flow graphs only, therefore by a flow graphs we will understand normalized flow graphs, unless stated otherwise.

With every node x of a flow graph G we associate its normalized *inflow* and *outflow* defined as

$$\sigma_+(x) = \frac{\varphi_+(x)}{\varphi(G)} = \sum_{y \in I(x)} \sigma(y, x), \tag{6}$$

$$\sigma_-(x) = \frac{\varphi_-(x)}{\varphi(G)} = \sum_{y \in O(x)} \sigma(x, y). \tag{7}$$

Obviously for any internal node x, we have $\sigma_+(x) = \sigma_-(x) = \sigma(x)$, where $\sigma(x)$ is a *normalized troughflow* of x.
Moreover, let

$$\sigma_+(G) = \frac{\varphi_+(G)}{\varphi(G)} = \sum_{x \in I(G)} \sigma_-(x), \tag{8}$$

$$\sigma_-(G) = \frac{\varphi_-(G)}{\varphi(G)} = \sum_{x \in O(G)} \sigma_+(x). \tag{9}$$

Obviously, $\sigma_+(G) = \sigma_-(G) = \sigma(G) = 1$.

3. Certainty and coverage factors

With every branch (x, y) of a flow graph G we associate the *certainty* and the *coverage factors*.
The *certainty* and the *coverage* of (x, y) are defined as

$$cer(x, y) = \frac{\sigma(x, y)}{\sigma(x)} \tag{10}$$

and

$$cov(x, y) = \frac{\sigma(x, y)}{\sigma(y)}, \tag{11}$$

respectively, where $\sigma(x) \neq 0$ and $\sigma(y) \neq 0$.
Below some properties, which are immediate consequences of definitions given above are presented:

$$\sum_{y \in O(x)} cer(x, y) = 1, \tag{12}$$

$$\sum_{x \in I(y)} cov(x, y) = 1, \tag{13}$$

$$\sigma(x) = \sum_{y \in O(x)} cer(x, y)\,\sigma(x) = \sum_{y \in O(x)} \sigma(x, y), \tag{14}$$

$$\sigma(y) = \sum_{x \in I(y)} cov(x, y)\,\sigma(y) = \sum_{x \in I(y)} \sigma(x, y), \tag{15}$$

$$cer(x, y) = \frac{cov(x, y)\,\sigma(y)}{\sigma(x)}, \tag{16}$$

$$cov(x, y) = \frac{cer(x, y)\,\sigma(x)}{\sigma(y)}. \tag{17}$$

Obviously the above properties have a probabilistic flavor, e.g., Eqs. (14) and (15) have a form of total probability theorem, whereas formulas (16) and (17) are Bayes' rules. However, these properties in our approach are interpreted in a deterministic way and they describe flow distribution among branches in the network.
A (*directed*) *path* from x to y, $x \neq y$ in G is a sequence of nodes $x_1, \ldots, x_n$ such that $x_1 = x$, $x_n = y$ and $(x_i, x_{i+1}) \in \mathscr{B}$ for every i, $1 \leq i \leq n-1$. A path from x to y is denoted by $[x \ldots y]$.

The *certainty*, the *coverage* and the *strength* of the path $[x_1 \ldots x_n]$ are defined as

$$cer[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cer(x_i, x_{i+1}), \tag{18}$$

$$cov[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cov(x_i, x_{i+1}), \tag{19}$$

$$\sigma[x \ldots y] = \sigma(x)\,cer[x \ldots y] = \sigma(y)\,cov[x \ldots y], \tag{20}$$

respectively.
The set of all paths from x to $y (x \neq y)$ in G denoted $< x, y >$, will be called a *connection* from x to y in G. In other words, connection $< x, y >$ is a sub-graph of G determined by nodes x and y.
For every connection $< x, y >$ we define its certainty, coverage and strength as shown below:

$$cer < x, y > = \sum_{[x \ldots y] \in < x, y >} cer[x \ldots y], \tag{21}$$

the *coverage* of the connection $< x, y >$ is

$$cov < x, y > = \sum_{[x \ldots y] \in < x, y >} cov[x \ldots y], \tag{22}$$

and the *strength* of the connection $< x, y >$ is

$$\sigma < x, y > = \sum_{[x \ldots y] \in < x, y >} \sigma[x \ldots y] = \sigma(x)\,cer < x, y > = \sigma(y)\,cov < x, y > . \tag{23}$$

Let $[x \ldots y]$ be a path such that x and y are input and output of the graph G, respectively. Such a *path* will be referred to as *complete*.
The set of all complete paths from x to y will be called a *complete connection* from x to y in G. In what follows we will consider complete paths and connections only, unless stated otherwise.
Let x and y be an input and output of a graph G respectively. If we substitute for every complete connection $< x, y >$ in G a single branch (x, y) such $\sigma(x, y) = \sigma < x, y >$, $cer(x, y) = cer < x, y >$, $cov(x, y) = cov < x, y >$ then we obtain a new flow graph G' such that $\sigma(G) = \sigma(G')$. The new flow graph will be called a *combined* flow graph. The combined flow graph for a given flow graph represents a relationship between its inputs and outputs.

4. Flow graph and decision algorithms

Flow graphs can be interpreted as decision algorithm.
Let us assume that the set of node of a graph is interpreted as a set of formulas, denoted Φ, Ψ, etc. The formulas are understood as propositional functions.
Then every branch (Φ, Ψ) can be understood as a decision rule $\Phi \to \Psi$, read if Φ *then* Ψ; Φ will be referred to as a *condition*, whereas Ψ—*decision* of the rule. Such a rule is characterized by three numbers, $\sigma(\Phi, \Psi)$, $cer(\Phi, \Psi)$ and $cov(\Phi, \Psi)$.

Thus every path $[\Phi_1 \ldots \Phi_n]$ determines a sequence of decision rules $\Phi_1 \to \Phi_2$, $\Phi_2 \to \Phi_3, \ldots, \Phi_{n-1} \to \Phi_n$.
From previous considerations it follows that this sequence of decision rules can be interpreted as a single decision rule $\Phi_1 \Phi_2 \ldots \Phi_{n-1} \to \Phi_n$, in short $\Phi^* \to \Phi_n$, where $\Phi^* = \Phi_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_{n-1}$, characterized by

$$cer(\Phi^*, \Phi_n) = cer[\Phi_1 \ldots \Phi_n], \tag{24}$$

$$cov(\Phi^*, \Phi_n) = cov[\Phi_1 \ldots \Phi_n], \tag{25}$$

and

$$\sigma(\Phi^*, \Phi_n) = \sigma(\Phi_1)\, cer[\Phi_1 \ldots \Phi_n] = \\ = \sigma(\Phi_n)\, cov[\Phi_1 \ldots \Phi_n], \tag{26}$$

where $\sigma(\Phi)$ is truth value of the formula Φ and $\sigma(\Phi, \Psi)$ in the strength of the decision rule $\Phi \to \Psi$.
Similarly, every connection $< \Phi, \Psi >$ can be interpreted as a single *decision rule* $\Phi \to \Psi$ such that:

$$cer(\Phi, \Psi) = cer < \Phi, \Psi >, \tag{27}$$

$$cov(\Phi, \Psi) = cov < \Phi, \Psi >, \tag{28}$$

and

$$\sigma(\Phi, \Psi) = \sigma(\Phi)\, cer < \Phi, \Psi > = \\ = \sigma(\Psi)\, cov < \Phi, \Psi >, \tag{29}$$

Let $[\Phi_1 \ldots \Phi_n]$ be a path such that Φ_1 is an input and Φ_n an output of the flow graph G, respectively. Such a path and the corresponding connection $< \Phi_1, \Phi_n >$ will be called *complete*.
The set of all decision rules $\Phi_{i_1} \Phi_{i_1} \ldots \Phi_{i_{n-1}} \to \Phi_{i_n}$ associated with all complete paths $\Phi_{i_1} \ldots \Phi_{i_n}$ will be called a *decision algorithm* induced by the flow graph.

5. Dependencies in flow graphs

Let $(x, y) \in \mathscr{B}$. Nodes x and y are independent on each other if

$$\sigma(x, y) = \sigma(x)\, \sigma(y). \tag{30}$$

Consequently

$$\frac{\sigma(x, y)}{\sigma(x)} = cer(x, y) = \sigma(y) \tag{31}$$

and

$$\frac{\sigma(x, y)}{\sigma(y)} = cov(x, y) = \sigma(x). \tag{32}$$

If

$$cer(x, y) > \sigma(y) \tag{33}$$

or

$$cov(x, y) > \sigma(x), \tag{34}$$

then x and y *depend positively* on each other.

Similarly, if

$$cer(x, y) < \sigma(y) \tag{35}$$

or

$$cov(x, y) < \sigma(x) \tag{36}$$

then x and y *depend negatively* on each other.
Let us observe that relations of independency and dependences are symmetric ones, and are analogous to that used in statistics.
For every $(x, y) \in \mathscr{B}$ we define a *dependency factor* $\eta(x, y)$ defined as

$$\eta(x, y) = \frac{cer(x, y) - \sigma(y)}{cer(x, y) + \sigma(y)} = \frac{cov(x, y) - \sigma(x)}{cov(x, y) + \sigma(x)}. \tag{37}$$

It is easy to check that if $\eta(x, y) = 0$, then x and y are independent on each other, if $-1 < \eta(x, y) < 0$, then x and y are negatively dependent and if $0 < \eta(x, y) < 1$ then x and y are positively dependent on each other.

6. Illustrative example

We will illustrate the above ideas by means of a simple tutorial example concerning a telecom provider. This example is a modification of the example given in [4].
Suppose we have three groups of telecom customers classified with respect to age: *young* (*students*), *middle aged* (*workers*) and *old* (*pensioners*). Moreover, suppose we have data concerning place of residence of customers: *town, village* and *country*.
Let us assume that the customers are buying a telecom service in vacation promotion and some of the customers are leaving the telecom provider in 3 months.
The initial data are presented in Fig. 1.

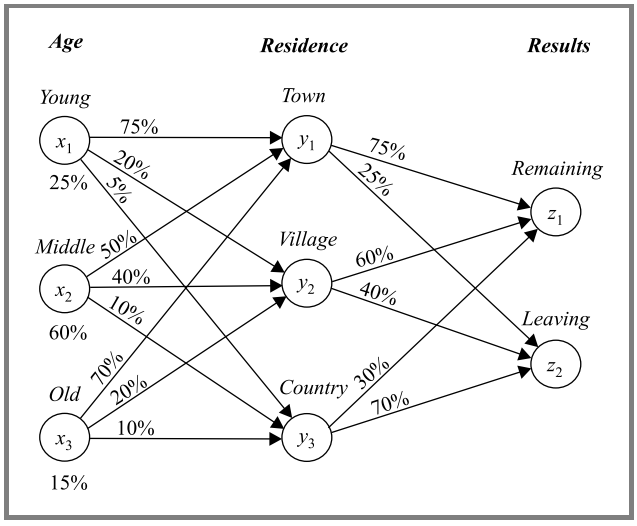

Fig. 1. Initial data.

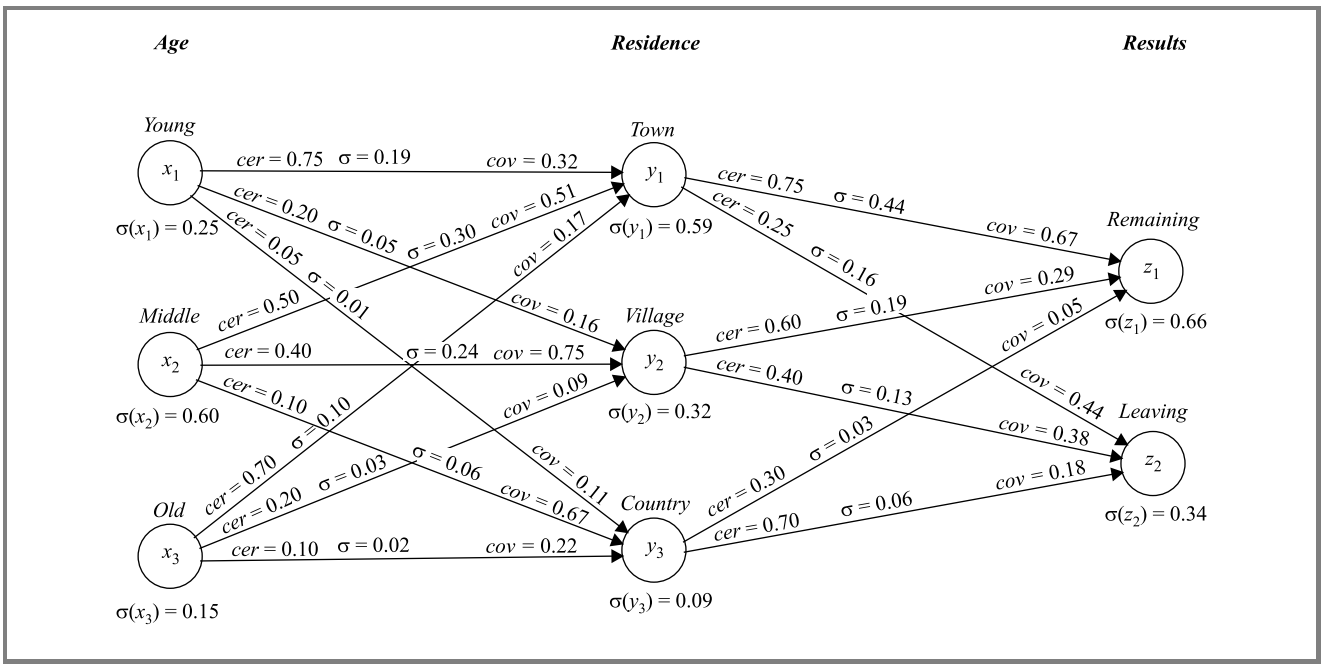

Fig. 2. Final results.

That means that these are 25% young customers, 60%—middle aged and 15% old—in the data base. Moreover, we know that 75% of young customers are living in towns, 20%—in villages and 5%—in the country, etc. We also have from the database that 75% town customers are not leaving the provider, whereas 25% are leaving the provider after 3 month, etc.

We want to find a relationship between the customer's group and the final result of the promotion after three month.

Applying the ideas presented in previous sections we get the results presented in Fig. 2.

Figure 2 shows general structure of patterns between customers and promotion results. Many interesting conclusions can be drawn from the picture, but we leave them for the interested reader.

We might be also interested in finding the relationship between age group and final result of the promotion. To this end we have to eliminate from the flow graph *residence*. In other words we have to compute all connections between

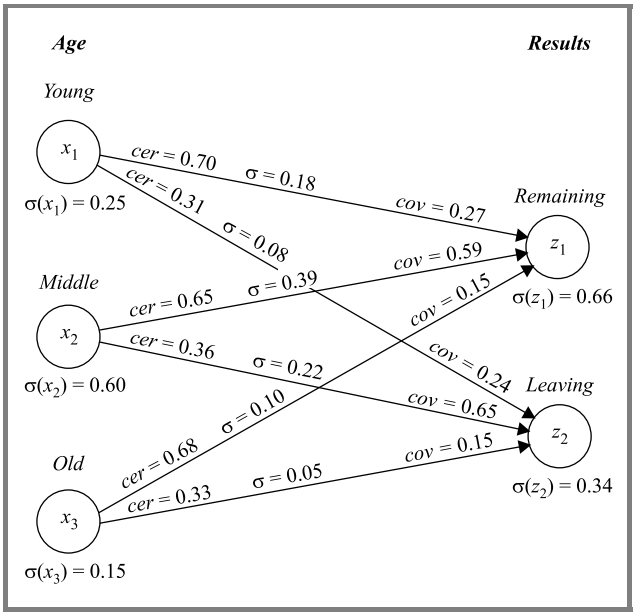

Fig. 3. Simplified flow graph.

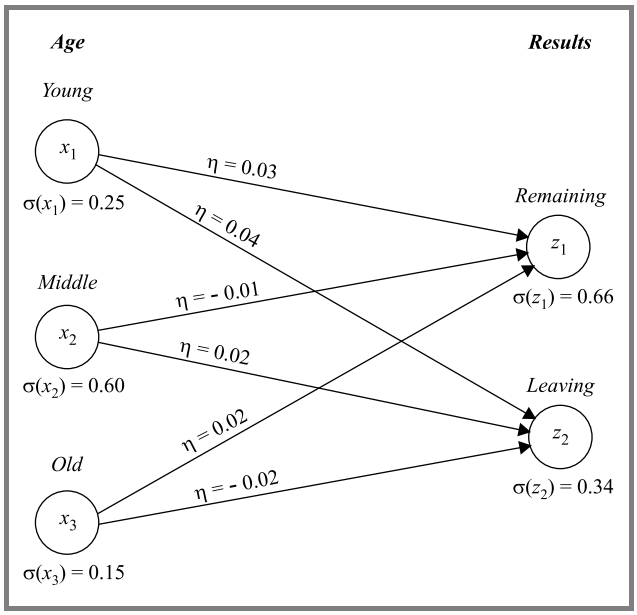

Fig. 4. Dependency coefficient.

age groups and the results, or—the relationship between input and output of the flow graph. The result is shown in Fig. 3.

Many interesting decision rules can be obtained from Figs. 2 and 3. Again we leave the task for the interested reader.

Dependences in flow graph presented in Fig. 3 are shown in Fig. 4.

It can be seen from the flow graph that all the dependency factors are very low and almost close to zero. That means, that in view of the data, practically, there is no relationship between groups of customers and the final result.

7. Conclusions

The paper presents a new approach to decision algorithm analysis. It is revealed that certain classes of decision algorithms can be represented as flow networks, and basic properties of such algorithms can be expressed in terms of flow distribution in a corresponding flow network. A method of simplification of such algorithms is presented.

References

[1] M. Berthold and D. J. Hand, *Intelligent Data Analysis—an Introduction*. Berlin [etc.]: Springer-Verlag, 1999.

[2] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton [etc.]: Princeton University Press, 1962.

[3] J. Łukasiewicz, "Die logishen Grundlagen der Wahrscheinilchkeitsrechnung" (Kraków, 1913), in *Jan Łukasiewicz—Selected Works*, L. Borkowski, Ed. Amsterdam [etc.], North Holland Publ., Warsaw: Polish Scientific Publ., 1970.

[4] M. Kryszkiewicz, H. Rybiński, and M. Muraszkiewicz, "Data mining methods for telecom providers", Institute of Computer Sciences, Warsaw University of Technology, Research Report 28/02 and "MOST—mobile open society thorough wirless telecommunications technology for mobile society", M. Muraszkiewicz, Ed., Warsaw University of Technology, 2003, pp. 210–220.

[5] Z. Pawlak, "Probability, truth and flow graphs", in *RSKD—Rough Sets in Knowledge Discovery, Proc.*, A. Skowron and M. Szczuka, Eds., Warsaw, Poland, 2003, pp. 1–9.

[6] Z. Pawlak, "Flow graphs and decision algorithms", in *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, G. Wang, Y. Yao, and A. Skowron, Eds., *Lecture Notes in Artificial Intelligence*. Berlin: Springer, 2003, vol. 2639, pp. 1–10.

[7] *Bayes's Theorem*, Proceedings of the British Academy, R. Swinburne, Ed. Oxford University Press, 2002, vol. 113.

Zdzisław Pawlak was born in Łódź (Poland), in 1926. He obtained his M.Sc. in 1951 in electronics from Warsaw University of Technology, Ph.D. in 1958 and D.Sc. in 1963 in the theory of computation from the Polish Academy of Sciences. He is a Professor of the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences and the University of Information Technology and Management and Member of the Polish Academy of Sciences. His current research interests include intelligent systems and cognitive sciences, in particular, decision support systems, knowledge representation, reasoning about knowledge, machine learning, inductive reasoning, vagueness, uncertainty and decision support. He is an author of a new mathematical tool, called rough set theory, intended to deal with vagueness and uncertainty. About two thousand papers have been published by now on rough sets and their applications world wide. Several international workshops and conferences on rough sets have been heldin recent years. He is a recipient of many awards among others the State Award in Computer Science in 1978, the Hugo Steinhaus Award for achievements in applied mathematics in 1989. Doctor honoris causa of Poznań University of Technology in 2002. Member of editorial boards of several dozens international journals. Deputy Editor-in-Chief of the *Bulletin of the Polish Academy of Sciences*. Program committee member of many international conferences on computer science. Over forty visiting university appointments in Europe, USA and Canada, about fifty invited international conference talks, and over one hundred seminar talks given in about fifty universities in Europe, USA, Canada, China, India, Japan, Korea, Taiwan, Australia and Israel. About two hundred articles in international journals and several books on various aspects on computer science and application of mathematics. Supervisor of thirty Ph.D. theses in computer science and applied mathematics.

e-mail: zpw@ii.pw.edu.pl
Institute for Theoretical and Applied Informatics
Polish Academy of Sciences
Bałtycka st 5
44-100 Gliwice, Poland

University of Information Technology and Management
Newelska st 6
01-447 Warsaw, Poland