Tatra Mt. Math. Publ. **16** (1999), 225–234

TATRA MOUNTAINS
Mathematical Publications

SOLUTION OF FUZZY RELATION EQUATIONS BY GENETIC ALGORITHMS

Štefan Boor — Vladimír Kvasnička — Jiří Pospíchal

ABSTRACT. General fuzzy relation equations are solved by genetic algorithms. Genetic algorithms represent powerful stochastic optimization methods that are able to look for global minima of complex multimodal objective functions. A modification of genetic algorithms for purposes of finding solution of general fuzzy relations is proposed. Basic concept of genetic algorithms is a population of chromosomes, which is for the considered fuzzy relation equations composed of matrices that correspond to solutions of problem. An objective function assigns to each chromosome a value that is vanishing if the chromosome corresponds to a correct solution. In the opposite case, if the chromosome is not a solution, then the function value is positive. This means that the solution of fuzzy relation can be transformed onto an equivalent minimization problem with the above specified objective function. A crossover operation between two chromosomes–matrices is determined, such that there is selected randomly (vertical) crossover line, and then matrix parts that are behind (below) the crossover line are mutually exchanged. A mutation operation goes through all chromosome elements, to which a small random number $r(0,\sigma)$ is added, where $r(0,\sigma)$ is a random number with a normal distribution, zero mean and a dispersion σ. Simple illustrative examples are discussed.

1. Introduction

Fuzzy relation equations belong to very important concepts and notions of the fuzzy set theory [1]. There exist many problems and particular tasks that can be solved by applying fuzzy relation equations. The main problem preventing their broad applications consists in an unpleasant fact that fuzzy relation equations can be solved by direct methods only in very special cases; there is not available a general method for their direct solutions. Moreover, we don't know necessary and sufficient conditions for existence of their solution, therefore in many cases

2000 Mathematics Subject Classification: 03B52.

Keywords: fuzzy relation, t-norm, genetic algorithms, crossover, mutation, population.

The research summarized in this paper was supported by the Grant Agency of the Slovak Republic, grant No. 1/5229/98, 1/4209/97.

of practical interest we don't know whether a considered fuzzy relation equation has a solution or not. The aim of this paper is to apply genetic algorithms (GAs) to solution of fuzzy set relations (where the achieved solution may be close to an exact one, if it exists, on the other hand, the achieved solution is approximate, if an exact one does not exist). GAs were invented by Holland [2] on the basis of Darwin evolution theory, and immediately after their formulation it was demonstrated that GAs belong to very effective, robust, and universal optimization methods.

2. Fuzzy relation equations and general matrix equations

Let us consider three matrices $\mathbf{P} = (p_{ij})$, $\mathbf{Q} = (q_{jk})$, and $\mathbf{R} = (r_{ik})$ that are of types $t(\mathbf{P}) = m \times n$, $t(\mathbf{Q}) = n \times r$, and $t(\mathbf{R}) = m \times r$, where $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$ are matrix representations of fuzzy relations on $X \times Y$, $Y \times Z$, and $X \times Z$, respectively, and X, Y, Z are fuzzy sets $X = \{x_i \colon i \in [1,m]\}$, $Y = \{y_j \colon j \in [1,n]\}$, and $Z = \{z_k \colon k \in [1,r]\}$. Members of sets X, Y, and Z are from the closed interval $[0,1]$. These matrices are related by the following matrix equation

$$\mathbf{P} \circ \mathbf{Q} = \mathbf{R}, \tag{1}$$

where the symbol $\circ$ represents a binary matrix operation (in general noncomutative, that assigns to a couple of matrices another matrix, with types specified above) and we postulate that $\mathbf{Q}$ and $\mathbf{R}$ are known matrices whereas $\mathbf{P}$ is an unknown matrix. In matrix theory, if the binary operation (1) is determined as follows

$$r_{ik} = \sum_{j=1}^{n} p_{ij} q_{jk} \,. \tag{2}$$

Then matrix equation (1) may be simply solved (at least, approximately) by making use a concept of pseudoinverse matrix, we get $\mathbf{P} = \mathbf{R}\mathbf{Q}^T(\mathbf{Q}\mathbf{Q}^T)^{-1}$, where the matrix $\mathbf{Q}^T(\mathbf{Q}\mathbf{Q}^T)^{-1}$ is called the pseudoinverse matrix. Its construction belongs to standard numerical tasks of contemporary numerical mathematics [3]. In fuzzy theory, the following three types of binary matrix operations are frequently used [1]:

(1) Max-min composition $\mathbf{P} \circ \mathbf{Q} = \mathbf{R}$, where matrix entries of $\mathbf{R}$ are

$$r_{ik} = \max_{j \in [1,n]} \min(p_{ij}, q_{jk}) \,. \tag{3}$$

(2) Sup-i composition based upon continuous t-norm i, $\mathbf{P} \overset{i}{\circ} \mathbf{Q} = \mathbf{R}$, which is a generalization of the previous matrix binary operation

$$r_{ik} = \max_{j \in [1,n]} i(p_{ij}, q_{jk}) \,. \tag{4}$$

(3) Inf-ω_i composition based upon a continuous t-norm i, $\mathbf{P} \overset{\omega_i}{\circ} \mathbf{Q} = \mathbf{R}$

$$r_{ik} = \min_{j\in[1,n]} \omega_i(p_{ij}, q_{jk}), \tag{5}$$

where $\omega_i(a,b) = \sup\{x \in [0,1]\colon i(a,x) \le b\}$.

For max-min composition (3) a solution (if any) of matrix equation (1) can be constructed by a simple direct method [1]. For other compositions direct methods for construction of a solution of (1) are not available, this is the main reason why we turn our attention to GAs. GAs are algorithmic methods that can construct a solution of (1) for an arbitrary binary matrix operation by applying an iterative evolutionary process. If the matrix equation (1) has a solution, then GAs are able to construct it automatically, otherwise, if (1) does not have any solution, then GAs produce an approximate solution that minimizes an objective function.

Let us consider an objective function

$$F(\mathbf{P}) = |\mathbf{P} \circ \mathbf{Q} - \mathbf{R}|, \tag{6}$$

where $|\cdot|$ is a matrix norm (e.g., L_1 norm determined as a sum of absolute values of all matrix elements). This objective function assigns to each matrix $\mathbf{P} = (p_{i,j})$ a nonnegative number; if this number is vanishing, then the matrix $\mathbf{P}$ approaches an exact solution of (1), otherwise, if this number is positive, then $\mathbf{P}$ is an approximate solution that minimizes the right-hand side of (6). Matrix $\mathbf{P}$ is determined by

$$\mathbf{P} = \arg\min_{\tilde{\mathbf{P}}\in\mathbb{R}^{m\times n}} F(\tilde{\mathbf{P}}), \tag{7}$$

where the minimization is performed over all matrices of the type $m \times n$. The main goal of this paper is to use GAs for finding solutions of the above minimization problem (7).

3. Specification of genetic algorithm

In this chapter we provide basic concepts of GA [4] that are specified for solving of the optimization problem (7). A chromosome $\boldsymbol{x}$ is a matrix of the type $m \times n$, $\boldsymbol{x} = (x_{ij})$. Let X be a set of all possible chromosomes that may be constructed, in other words, the set X is composed of all possible matrices of the type $m \times n$ with their entries inside the interval [0,1],

$$X = \{\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}'' \dots\}. \tag{8}$$

The objective function (see Eq. (6)) $F\colon X \to \bar{\mathbb{R}}_+ = [0,\infty)$ assigns to each chromosome $\boldsymbol{x} \in X$ a nonnegative real number $F(\boldsymbol{x})$. We look for its global minimum over the set X (see Eq. (7))

$$\boldsymbol{x}_{\mathrm{opt}} = \arg\min_{\boldsymbol{x}\in X} F(\boldsymbol{x}). \tag{9}$$

In GA the notion of objective function serves as a representation of *environment* in which chromosomes live and are evolved by Darwinian evolution. The goal of the evolution is to "construct" such chromosomes that are closely related (or even equal) to the optimal chromosome $\mathbf{x}_{\mathrm{opt}}$ specified by (9).

A *mutation* is an operator $O_{\mathrm{mut}} : X \to X$ that assigns stochastically to each chromosome $\mathbf{x} \in X$ a new chromosome $\mathbf{x}' = O_{\mathrm{mut}}(\mathbf{x}) \in X$. This modification of the chromosome–matrix $\mathbf{x}$ is done in such a way that entries of the original chromosome $\mathbf{x}$ are slightly modified by random numbers

$$x'_{ij} = x_{ij} + r(0, \sigma) , \tag{10}$$

$r(0, \sigma)$ is a random number with normal distribution, zero mean, and dispersion σ. The dispersion σ will be considered in our forthcoming discussion as a parameter of the used GA, it will be set to a small positive value, say $\sigma = 0.1$. Since entries $\mathbf{x}_{ij}$ are kept inside the interval $[0, 1]$, mutated entries x'_{ij} should be repaired as follows

$$\begin{aligned} x'_{ij} < 0 \Rightarrow x'_{ij} = \delta , \\ x'_{ij} > 1 \Rightarrow x'_{ij} = 1 - \delta , \end{aligned} \tag{11}$$

where δ is a small positive number (e.g., $\delta = 10^{-5}$).

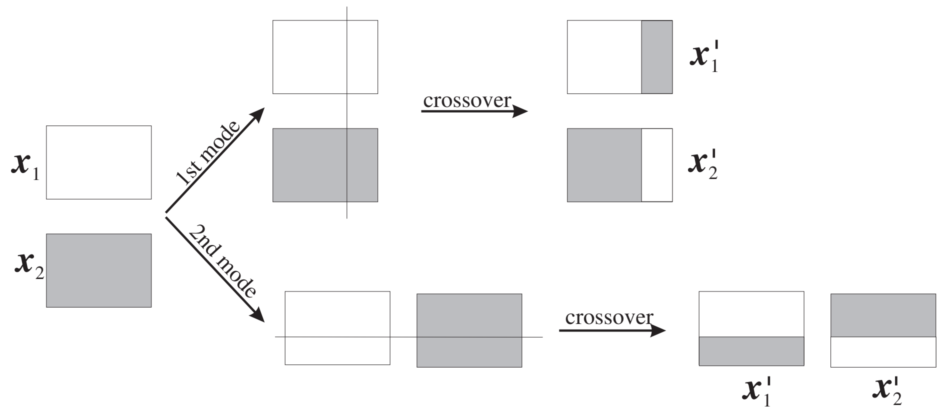

FIGURE 1. Diagrammatic visualization of two different ways of crossover performance. In the first mode a vertical crossover line is randomly generated and then the matrix parts behind the line are exchanged. Similarly, in the second mode, a vertical line is randomly generated and the matrix parts below the line are exchanged.

A *crossover* is an operator $O_{\mathrm{cross}} : X^2 \to X^2$. This operator assigns stochastically to a pair $\mathbf{x}_1, \mathbf{x}_2 \in X$ a new pair $(\mathbf{x}'_1, \mathbf{x}'_2) = O_{\mathrm{cross}}(\mathbf{x}_1, \mathbf{x}_2)$. This operation is performed in two different ways, we select randomly an index of a column (row), and then matrix columns (rows) with higher indices are mutually exchanged, see Fig. 1.

Mutation and crossover operators are unified in an operator called the *reproduction* $O_{\mathrm{repro}}\colon X^2 \to X^2$ determined as follows:

$$(\mathbf{x}_1', \mathbf{x}_2') = O_{\mathrm{repro}}(\mathbf{x}_1, \mathbf{x}_2), \quad (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 = O_{\mathrm{cross}}(\mathbf{x}_1, \mathbf{x}_2), \tag{12}$$

$$\mathbf{x}_1' = O_{\mathrm{mut}}(\tilde{\mathbf{x}}_1), \quad \mathbf{x}_2' = O_{\mathrm{mut}}(\tilde{\mathbf{x}}_2). \tag{13}$$

This means that in the first stage of reproduction the crossover is performed and then the resulting chromosomes are mutated. A stochastic character of operators O_{mut} and O_{cross} is controlled by a probability P_{cross} and by the dispersion σ. The crossover operator O_{cross} is stochastically applied in the framework of the reproduction operator O_{repro}. For each pair of chromosomes selected for a reproduction, the crossover process will be used with a probability P_{cross}. If the crossover process is not used, then chromosomes in the second relation of (12) are simply copied.

A *population* P is a multiset with fixed finite cardinality of chromosomes from the set X, $P = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\} \subseteq X$. A *fitness* of a chromosome from the population P is a positive real number determined by a mapping $f\colon P \to \mathbb{R}_+$ restricted by the following condition

$$F(\mathbf{x}_1) \leq F(\mathbf{x}_2) \Rightarrow f(\mathbf{x}_1) \geq f(\mathbf{x}_2). \tag{14}$$

Each chromosome from the population P is evaluated by a fitness. It reflects how close an objective function value $F(\mathbf{x})$ is to zero, which is the value corresponding to an exact solution. A chromosome evaluated by a smaller function value $F(\mathbf{x})$ is evaluated by a greater fitness $f(\mathbf{x})$ so that the condition (14) is satisfied. In our GA calculations the fitness is determined by

$$f(\mathbf{x}) = \frac{1}{\varepsilon + F(\mathbf{x})}, \tag{15}$$

where ε is a small positive number and the condition (14) is automatically fulfilled.

The fitness values are used for the calculation of probabilities of selection of chromosomes from the population P. Let $O_{\mathrm{select}}(P)$ be a stochastic operator — *proportional selector* — which realizes a selection of chromosomes from the population P in such a way that the probability of this selection is equal to a normalized fitness. A chromosome with a greater normalized fitness has a greater probability to be selected by the selector $O_{\mathrm{select}}(P)$. Formally, the term $O_{\mathrm{select}}(P)$ is understood as a chromosome $\mathbf{x} \in P$ selected from the population P with the probability determined as follows

$$\mathrm{prob}(x) = \frac{f(\mathbf{x})}{\sum\limits_{y \in P} f(\mathbf{y})}. \tag{16}$$

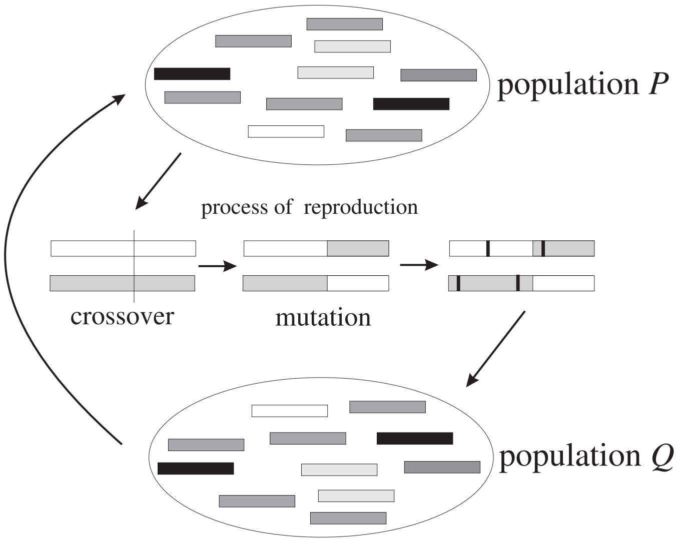

FIGURE 2. Diagrammatic outline of Algorithm 1 with simple specification of crossover and mutation operations

procedure GA;
begin $P :=$ *a* set composed of p randomly
  generated chromosomes;
  $t := 0$;
  while $t < t_{\max}$ **do**
  begin $t := t + 1$;
    evaluate all chromosomes by fitness;
    $Q := \emptyset$;
    while $|Q| < p$ **do**
    begin $\mathbf{x}_1 := O_{\text{select}}(P)$; $\mathbf{x}_2 := O_{\text{select}}(P)$;
      $(\mathbf{x}'_1, \mathbf{x}'_2) := O_{\text{repro}}(\mathbf{x}_1, \mathbf{x}_2)$;
      $Q := Q \bigcup \{\mathbf{x}'_1, \mathbf{x}'_2\}$;
    end;
    $P := Q$;
  end;
  $\mathbf{x}_{\text{opt}} :=$ the best chromosome of P;
end;

ALGORITHM 1. An implementation of simple GA. Algorithm starts with a random generation of population P. Integer variable t is a counter of epochs in GA, outer while-do cycle is repeated a prescribed number of times determined by $t_{\max}$. Inner while-do cycle forms a new population Q, which is initialized as an empty set, it is sequentially occupied by offspring created by the reproduction operator applied on a pair of quasirandomly selected chromosomes $\mathbf{x}_1$ and $\mathbf{x}_2$ from the population P. Input parameters are n (length of binary vectors determining the precision of real number entries of chromosomes), p (number of chromosomes in population), $t_{\max}$ (maximal number of generations), the probability P_{cross}, and the dispersion σ. As an output of GA is a chromosome $\mathbf{x}_{\mathrm{opt}}$, determined as that one with the smallest function value $F(\mathbf{x})$ from the population P.

GA imitates the Darwinian evolution over a population of chromosomes that are realized by matrices of real entries and "live" in an environment specified by the objective function (6). A simple realization of GA is outlined in Algorithm 1, see Fig. 2.

4. Illustrative example

In order to illustrate the described genetic algorithm procedure, let us consider fuzzy relation of the type (4), where the t-norm is determined as a product, and matrices $\mathbf{Q}$ and $\mathbf{R}$ are specified by

$$\mathbf{Q} = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix} \qquad \text{and} \qquad \mathbf{R} = \begin{pmatrix} 0.12 \\ 0.18 \\ 0.27 \end{pmatrix}.$$

The unknown matrix $\mathbf{P}$ of the type 3×3 is determined by the genetic algorithm minimization of the problem (7). The initial population is composed of 200 randomly generated chromosomes–matrices $\mathbf{P}$ with entries from the closed interval $[0,1]$. The best solution in the first iteration is $\mathbf{P} = \begin{pmatrix} 0.0933 & 0.1368 & 0.4837 \\ 0.9750 & 0.8234 & 0.5679 \\ 0.9237 & 0.5951 & 0.8839 \end{pmatrix}$,

$\mathbf{P} \overset{i}{\circ} \mathbf{Q} = \begin{pmatrix} 0.1451 \\ 0.1704 \\ 0.2652 \end{pmatrix}$, and $F(\mathbf{P}) = 0.03954009$.

After 64th iteration we get a final solution

$$\mathbf{P} = \begin{pmatrix} 0.7529 & 0.4259 & 0.4 \\ 0.6725 & 0.8907 & 0.6 \\ 0.8846 & 0.1278 & 0.9 \end{pmatrix}, \ \mathbf{P} \overset{i}{\circ} \mathbf{Q} = \begin{pmatrix} 0.12 \\ 0.18 \\ 0.27 \end{pmatrix}, \quad \text{and } F(\mathbf{P}) = 0.00000009,$$

where the value of objective function is smaller than a prescribed precision 10^{-7}; this fact implies that the achieved solution is correct, i.e., the calculated matrix $\mathbf{P}$ fulfills the matrix equation (1) with a binary operation specified by (4). Plots of average fitness and best fitness are displayed in Fig. 3.

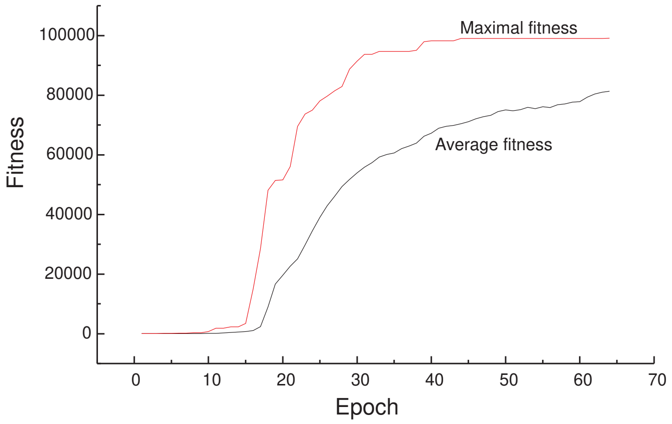

FIGURE 3. Plots of average fitness and best fitness vs. epochs of GA. The population of chromosomes evolves during the run of GA so that the average fitness is gradually increasing while the best fitness is increasing stepwise.

The second illustrative example is a matrix equation (1) with a binary operation (4) (where the t-norm is determined as a product), where this matrix equation does not have a solution, i.e., the proposed GA offers only an approximative solution. The known matrices $\mathbf{Q}$ and $\mathbf{R}$ are specified as follows

$$\mathbf{Q} = \begin{pmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{pmatrix} \qquad \text{and} \qquad \mathbf{R} = (\,0.5 \;\; 0.6\,)\,.$$

The best solution in the first iteration is

$$\mathbf{P} = (\,0.1828 \;\; 0.9967\,)$$

then $\mathbf{P} \overset{i}{\circ} \mathbf{Q} = (\,0.1993 \;\; 0.3987\,) = \mathbf{R}_{\text{calcul}}$, this means that a value of the objective function F is determined by an $F(\mathbf{P}) = |\,\mathbf{R} - \mathbf{R}_{\text{calcul}}| = 0.50195929$. After 100th iteration we get a final solution (that is stable, it is not changed in the course of further iterations)

$$\mathbf{P} = (\,0.4588 \;\; 1.0000\,)\,,$$

then $\mathbf{P} \overset{i}{\circ} \mathbf{Q} = (\,0.2000 \;\; 0.4000\,)$ and $F(\mathbf{P}) = 0.50000002$. Since the final optimal value of objective function is positive, the obtained solution for the matrix $\mathbf{P}$ is only an approximative one, it does not exactly fulfill the matrix equation but only minimizes the objective function.

In both cases the number of generations and the time of computation on PC 586 with 120 MHz required to achieve a solution with the prescribed precision was estimated from 30 runs of the program. In the first example the largest number of generations was 67 with computational time 11 s, the average was 55.8 generations with computational time 8.33 s. In the second case the largest number of generations was 148 with computational time 8 s, the average was

64.5 generations with computational time 4.45 s. The convergence properties of the method follow from the properties of genetic algorithm.

5. Discussion

The proposed version of genetic algorithm offers very simple numerical constructions of solutions of the matrix equation (1), where the binary matrix operator can be very vaguely specified; especially, this operator may be specified by (3) to (5) so that the matrix equation (4) corresponds to a fuzzy relation. The proposed method can be simply modified so that it will provide either maximal or minimal solutions, such that a sum of absolute values of entries of the unknown matrix $\mathbf{P}$ is maximized or minimized. This goal may be simply achieved for a minimal matrix $\mathbf{P}$ in such a way that the objective function (6) is modified as follows

$$F_{\min}(\mathbf{P}) = |\mathbf{P} \circ \mathbf{Q} - \mathbf{R}| + \omega \sum_{i=1}^{m} \sum_{j=1}^{n} |p_{ij}|, \tag{17}$$

where ω is a small positive coefficient (called the penalization coefficient). Both terms on its r.h.s. are simultaneously minimized, if the obtained final solution fulfills the matrix equation (1) (i.e., the unmodified objective function $F(\mathbf{P})$ is vanishing), then the obtained solution $\mathbf{P}$ is a correct solution with minimal sum of its entries (the so-called minimal solution). In a similar as in (17), we may define an objective function that produces maximal solutions

$$F_{\max}(\mathbf{P}) = |\mathbf{P} \circ \mathbf{Q} - \mathbf{R}| + \omega \sum_{i=1}^{m} \sum_{j=1}^{n} (1 - p_{ij}), \tag{18}$$

where we have assumed that entries of $\mathbf{P}$ are taken from the closed interval [0,1]. Minimizing this objective function we get a solution of the matrix equation (1) restricted so that a sum of its entries is maximal. Both the above objective functions (17) and (18) will be studied in our forthcoming paper.

REFERENCES

[1] KLIR, G. J.—BO YUAN : *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Upper Saddle River, NJ, 1995.

[2] HOLLAND, J. H. : *Adaptation in natural and artificial systems*, The University of Michigan Press, Ann Arbor, 1975.

[3] *The Numerical Recipes*, The Cambridge Scientific Press, (adjusted continually) available at `http://cfatab.harward.edu/nr`.

[4] GOLDBERG, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

Received March 31, 1998

Department of Mathematics
Faculty of Chemical Technology
Slovak Technical University
Radlinského 9
SK–81237 Bratislava
SLOVAKIA
E-mail: boor@cvt.stuba.sk
kvasnicka@cvt.stuba.sk
pospichal@cvt.stuba.sk