Computer Science Journal of Moldova, vol.**24**, no.3(72), 2016

About Applications of Distances on Monoids of Strings

Mitrofan Choban, Ivan Budanaev

Dedicated to Professor, Corresponding Member of the Academy of Science of Moldova Constantin Gaindric on the occasion of his seventy-fifth anniversary

Abstract

In this article we show that there are invariant distances on the monoid $L(A)$ of all strings closely related to Levenshtein's distance. We will use a distinct definition of the distance on $L(A)$, based on the Markov - Graev method, proposed by him for free groups. As result we will show that for any quasimetric d on alphabet A in union with the empty string there exists a maximal invariant extension d^* on the free monoid $L(A)$. This new approach allows the introduction of parallel and semiparallel decompositions of two strings. In virtue of Theorem 3.1, they offer various applications of distances on monoids of strings in solving problems from distinct scientific fields. The discussion covers topics in fuzzy strings, string pattern search, DNA sequence matching etc.

Keywords: String pattern matching, parallel decomposition, semiparallel decomposition, free monoid, invariant distance, quasimetric, Levenshtein distance, Hamming distance, proper similarity.

1 Introduction

The dynamic transition of our technological civilization to digital processing and data transmission systems created many problems in the design of modern systems in computer science and telecommunications. Providing robustness and noise immunity is one of the most important and difficult tasks in data transmission, recording, playback, and

©2016 by M. Choban, I. Budanaev

storage. The distance between information plays a paramount role in mathematics, computer science, and other interdisciplinary research areas. The first among many scientists in the field, who presented the theoretical solutions to error detection and error correction problems, were C. Shannon, R. Hamming, and V. Levenshtein (see [11], [12], [18]). We begin this section with introductions into the field, focusing mainly on abstract monoid of strings $L(A)$.

A monoid is a semigroup with an identity element. Fix a non-empty set A. The set A is called an *alphabet.* Let $L(A)$ be the set of all finite strings $a_1a_2\dots a_n$ with $a_1, a_2, \dots, a_n \in A$. Let ε be the empty string. Consider the strings $a_1a_2\dots a_n$ such that $a_i = \varepsilon$ for some $i \leq n$. If $a_i \neq \varepsilon$, for any $i \leq n$ or $n = 1$ and $a_1 = \varepsilon$, the string $a_1a_2\dots a_n$ is called a *canonical string.* The set

$$Sup(a_1a_2\dots a_n) = \{a_1, a_2, \dots, a_n\} \cap A$$

is the support of the string $a_1a_2\dots a_n$ and

$$l(a_1\dots a_n) = |Sup(a_1\dots a_n)|$$

is the length of the string $a_1a_2\dots a_n$. For two strings $a_1\dots a_n$ and $b_1\dots b_m$, their product(concatenation) is $a_1\dots a_nb_1\dots b_m$. If $n \geq 2, i < n$ and $a_i = \varepsilon$, then the strings $a_1\dots a_n$ and $a_1\dots a_{i-1}a_{i+1}\dots a_n$ are considered equivalent. In this case any string is equivalent to one unique canonical string. We identify the equivalent strings. In this case $L(A)$ becomes a monoid with identity ε. Let $Sup(a, b) = Sup(a) \cup Sup(b) \cup \{\varepsilon\}$, and $Sup(a, a) = Sup(a) \cup \{\varepsilon\}$.

It is well known that any subset $L \subset L(A)$ is an abstract language over the alphabet A.

2 Distances on spaces

2.1 Definitions

Let A be a non-empty set and $d : X \times X \to \mathbb{R}$ be a mapping such that for all $x, y \in X$ we have:

(i_m) $d(x, y) \geq 0$;

(ii_m) $d(x, x) = 0$.

Then (X, d) is called a *pseudo-distance space* and d is called a *pseudo-distance* on X. In addition,

(iii_m) $d(x, y) + d(y, x) = 0$ if and only if $x = y$,

then (X, d) is called a *distance space* and d is called a *distance* on X. Furthermore,

(iv_m) $d(x, y) = 0$ if and only if $x = y$,

then (X, d) is called a *strong distance space* and d is called a *strong distance* on X.

General problems in distance spaces were studied by different authors (see [1], [3], [4], [8], [15]). The notion of a distance space is more general than the notion of *o*-metric spaces in sense of A. V. Arhangelskii [1] and S. I. Nedev [15]. A distance d is an *o*-metric if from $d(x, y) = 0$ it follows that $x = y$, i.e. d is a strong distance.

Let X be a non-empty set and d be a pseudo-distance on X. Then:

- (X, d) is called a *pseudo-symmetric space* and d is called a *pseudo-symmetric* on X if for all $x, y \in X$

$$(v_m) d(x, y) = d(y, x);$$

- (X, d) is called a *symmetric space* and d is called a *symmetric* on X if d is a distance and a pseudo-symmetric simultaneously;
- (X, d) is called a *pseudo-quasimetric space* and d is called a *pseudo-quasimetric* on X if for all $x, y, z \in X$

$$(vi_m) d(x, z) \leq d(x, y) + d(y, z);$$

- (X, d) is called a *quasimetric space* and d is called a *quasimetric* on X if d is a distance and a pseudo-quasimetric simultaneously;
- (X, d) is called a *pseudo-metric space* and d is called a *pseudo-metric* if d is a pseudo-symmetric and a pseudo-quasimetric simultaneously;
- (X, d) is called a *metric space* and d is called a *metric* if d is both symmetric and quasimetric;

- a distance d is called discrete if $d(x,y) \in \omega = \{0,1,2,\dots\}$ for all $x,y \in X$.

Let G be a semigroup and d be a pseudo-distance on G. The pseudo-distance d is called:

- Left (respectively, right) invariant if $d(xa,xb) \leq d(a,b)$ (respectively, $d(ax,bx) \leq d(a,b)$) for all $x,a,b \in G$;
- Invariant if it is both left and right invariant.

A distance d on a semigroup G is called *stable* if $d(xy,uv) \leq d(x,u) + d(y,v)$ for all $x,y,u,v \in G$.

Proposition 1. *Let d be a pseudo-quasimetric on a semigroup G. The next assertions are equivalent:*

1. *d is invariant,*
2. *d is stable.*

2.2 Extension of pseudo-quasimetrics on free monoids

Fix an alphabet A and let $\bar{A} = A \cup \{\varepsilon\}$. We assume that $\varepsilon \in \bar{A} \subseteq L(A)$ and ε is the identity of the monoid $L(A)$. Let ρ be a pseudo-quasimetric on the set $\bar{A}$ and $Q(\rho)$ be the set of all stable pseudo-quasimetrics d on $L(A)$ for which $d(x,y) \leq \rho(x,y)$ for all $x,y \in \bar{A}$. The set $Q(\rho)$ is nonempty since it contains the trivial pseudo-quasimetric $d(x,y) = 0$ for all $x,y \in L(A)$. For all $a,b \in L(A)$ let $\hat{\rho}(a,b) = sup\{d(a,b) : d \in Q(\rho)\}$. We say that $\hat{\rho}$ is the maximal stable extension of ρ on $L(A)$.

The following properties are proved in [5].

Property 2.1. $\hat{\rho} \in Q(\rho)$.

For any $r > 0$ let $d_r(a,a) = 0$ and $d_r(a,b) = r$ for all distinct points $a,b \in L(A)$. Then d_r is an invariant metric on $L(A)$.

Property 2.2. *Let $r > 0$ and $\rho(x,y) \geq r$ for all distinct points $x,y \in A$. Then $\hat{\rho}$ is a quasimetric on $L(A)$, $d_r \in Q(\rho)$, and $\hat{\rho}(a,b) = r$ for all distinct points $a,b \in L(A)$.*

For any $a, b \in L(A)$ let

$$\bar{\rho}(a,b) = inf\{\Sigma\{\rho(x_i, y_i) : i \leq n\}\},$$

where $n \in \mathbb{N} = \{1, 2, \ldots\}$, $x_1, y_1, x_2, y_2, \ldots, x_n, y_n \in \bar{A}$, $a = x_1 x_2 \ldots x_n, b = y_1 y_2 \ldots y_n$. Let

$$\rho^*(a,b) = inf\{\bar{\rho}(a, z_1) + \cdots + \bar{\rho}(z_i, z_{i+1}) + \cdots + \bar{\rho}(z_n, b)\},$$

where $n \in \mathbb{N}, z_1, z_2, \ldots, z_n \in L(A)$.

Property 2.3. *$\bar{\rho}$ is a pseudo-distance on $L(A)$ and $\bar{\rho}(x,y) \leq \rho(x,y)$ for all $x, y \in \bar{A}$.*

Property 2.4. *$\bar{\rho}(x,y) = \rho(x,y)$ for all $x, y \in X$.*

Property 2.5. *The pseudo-distance $\bar{\rho}$ is invariant on $L(A)$.*

Property 2.6. *The pseudo-distance ρ^* is a stable pseudo-quasimetric on $L(A)$ and $\rho^* \in Q(\rho)$.*

Property 2.7. *If ρ is a quasimetric on X, then $\bar{\rho}$ is a distance on $L(A)$.*

Property 2.8. *Let $a, b \in L(A)$ be two distinct points in $L(A)$ and $r(a,b) = min\{\rho(x,y) : x \in Sup(a,a), y \in Sup(b,b), x \neq y\}$. Then*

$$\hat{\rho}(a,b) = \rho^*(a,b) \geq r(a,b).$$

The following properties follow from Property 2.8.

Property 2.9. *If ρ is a quasimetric on $\bar{A}$, then ρ^* and $\hat{\rho}$ are quasimetrics on $L(A)$.*

Property 2.10. *If ρ is a strong quasimetric on $\bar{A}$, then ρ^* and $\hat{\rho}$ are strong quasimetrics on $L(A)$.*

Property 2.11. *Let ρ be a pseudo-quasimetric on $\bar{A}$, Y be a subspace of $\bar{A}$, and $\varepsilon \in \bar{Y}$. Let $M(Y) = L(Y)$ be the submonoid of the monoid $L(A)$ generated by the set Y, and by d_Y be the extension $\hat{\rho}|Y$ on $M(Y)$ of the pseudo-quasimetric ρ_Y on Y, where $\rho_Y(y,z) = \rho(y,z)$ for all $y, z \in \bar{Y}$. Then*

1. *$d_Y(a,b) = \hat{\rho}(a,b)$ for all $a,b \in M(Y)$,*
2. *If ρ is a (strong) quasimetric on Y, then $\hat{\rho}$ is a (strong) quasimetric on $M(Y)$,*
3. *If ρ is a metric on Y, then $\hat{\rho}$ is a metric on $M(Y)$,*
4. *If $a,b \in L(A)$ are distinct points and ρ is a quasimetric on $Sup(a,b)$, then $\hat{\rho}(a,b) + \hat{\rho}(b,a) > 0$,*
5. *If $a,b \in L(A)$ are distinct points and ρ is a strong quasimetric on $Sup(a,b)$, then $\hat{\rho}(a,b) > 0$ and $\hat{\rho}(b,a) > 0$,*
6. *For any $a,b \in L(A)$ there are $n \in \mathbb{N}$, $x_1, x_2, \ldots, x_n \in Sup(a,a)$ and $y_1, y_2, \ldots, y_n \in Sup(b,b)$ such that $a = x_1 x_2 \cdots x_n$, $b = y_1 y_2 \cdots y_n$ ρ, $n \le l(a) + l(b)$ and $\bar{\rho}(a,b) = \Sigma\{\rho(x_i, y_i) : i \le n\}$,*
7. *$\hat{\rho} = \bar{\rho} = \rho^*$.*

Property 2.12. *For any $a{=}a_1 a_2 \ldots a_n$ we put $a^{-1}{=}a_n \ldots a_2 a_1$. Then $\rho^*(a,b){=}\rho^*(a^{-1}, b^{-1})$ and $(ab)^{-1}{=}b^{-1}a^{-1}$ for all $a,b \in L(A)$.*

Remark 2.1. *The method of extensions of distances for free groups, used by us, was proposed by A. A. Markov [13] and M. I. Graev [9]. For free universal algebras it was extended in [3], for free groups and varieties of groups it was examined in [6], [17].*

2.3 Discrete distances on $L(A)$

Fix an alphabet A and $\bar{A}{=}A \cup \{\varepsilon\}$. Consider on A some linear ordering for which $\varepsilon < x$ for any $x \in A$. On $\bar{A}$ consider the following distances ρ_l, ρ_r, ρ_s, where $\rho_l(x,x) = \rho_r(x,x) = 0$ for any $x \in \bar{A}$; if $x,y \in \bar{A}$ and $x < y$, then $\rho_l(x,y) = 1, \rho_l(y,x) = 0$, $\rho_r(x,y) = 0, \rho_r(y,x) = 1$, $\rho_s(x,y) = \rho_l(x,y) + \rho_r(x,y)$. By construction, ρ_l and ρ_r are quasimetrics and ρ_s is a metric on $\bar{A}$. Then ${\rho_l}^*(x,y)$ and ${\rho_r}^*(x,y)$ are invariant discrete quasimetrics on $L(A)$ and ${\rho_s}^*$ is a discrete invariant metric on $L(A)$.

Theorem 2.1. *Let ρ be a quasimetric on $\bar{A}$, and $\rho(a,\varepsilon) = \rho(b,\varepsilon)$ for all $a,b \in A$. Then $\rho^*(ac,bc) = \rho^*(ca,cb) = \rho^*(a,b)$ for all $a,b,c \in L(A)$.*

Corollary 2.1. *If $\rho^* = \rho_s^*$, then $\rho^*(ac,bc) = \rho^*(ca,cb) = \rho^*(a,b)$ for all $a,b,c \in L(A)$.*

3 Parallel decompositions of two strings

The longest common substring and pattern matching in two or more strings is a well known class of problems. For any two strings $a, b \in L(A)$ we find the decompositions of the form $a = v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}$, which can be represented as $a = a_1a_2 \cdots a_n$, $b = b_1b_2 \cdots b_n$ with the following properties:

- some a_i and b_j may be empty strings, i.e. $a_i = \varepsilon$, $b_j = \varepsilon$;
- if $a_i = \varepsilon$, then $b_i \neq \varepsilon$ and if $b_j = \varepsilon$, then $a_j \neq \varepsilon$;
- if $u_1 = \varepsilon$, then $a = v_1$ and $b = w_1$;
- if $u_1 \neq \varepsilon$, then there is a sequence $1 \leq i_1 \leq j_1 < i_2 \leq j_2 < \cdots < i_k \leq j_k \leq n$ such that:
    - $u_1 = a_{i_1} \cdots a_{j_1} = b_{i_1} \cdots b_{j_1}$, $u_2 = a_{i_2} \cdots a_{j_2} = b_{i_2} \cdots b_{j_2}$, $u_k = a_{i_k} \cdots a_{j_k} = b_{i_k} \cdots b_{j_k}$;
    - if $v_1 = w_1 = \varepsilon$, then $i_1 = 1$;
    - if $v_{k+1} = w_{k+1} = \varepsilon$, then $j_k = n$;
    - if $k \geq 2$, then for any $i \in \{2, \cdots, k\}$ we have $v_i \neq \varepsilon$ or $w_i \neq \varepsilon$.

In this case

$$l(u_1) + l(u_2) + \cdots + l(u_k) = |\{i : a_i = b_i\}|.$$

The above decomposition forms are called *parallel decompositions* of strings a and b. For any parallel decompositions $a = v_1u_1 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1 \cdots w_ku_kw_{k+1}$ the number

$$E(v_1u_1 \cdots v_ku_kv_{k+1}, w_1u_1 \cdots w_ku_kw_{k+1}) = \sum_{i \leq k+1} \{\max\{l(v_i), l(w_i)\}\}$$

is called the efficiency of the given parallel decompositions. The number $E(a, b)$ is equal to the minimum of the efficiencies of all parallel decompositions of the strings a, b and is called the *common efficiency of the strings a,b*. It is obvious that $E(a, b)$ is well determined. We say that the parallel decompositions $a = v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}$ are optimal if

$$E(v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}, w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}) = E(a, b).$$

These types of parallel decompositions are associated with the problem of approximate string matching [14]. If the decompositions $a = v_1u_1 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1 \cdots w_ku_kw_{k+1}$ are optimal and $k \geq 2$, then we may consider that $u_i \neq \varepsilon$ for any $i \leq k$.

Any parallel decompositions $a = a_1a_2 \cdots a_n = v_1u_1 \cdots v_ku_kv_{k+1}$ and $b = b_1b_2 \cdots b_n = w_1u_1 \cdots w_ku_kw_{k+1}$ generate a common subsequence $u_1u_2 \cdots u_k$. The number

$$m(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n) = l(u_1) + l(u_2) + \cdots + l(u_k)$$

is the *measure of similarity* of the decompositions [2], [16]. There are parallel decompositions $a = v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}$ and $b = w_1u_1w_2u_2 \cdots w_ku_kw_{k+1}$ for which the measure of similarity is maximal. The maximum value of the measure of similarity of all decompositions is denoted by $m^*(a, b)$. The maximum value of the measure of similarity of all optimal decompositions is denoted by $m^\omega(a, b)$. We can note that $m^\omega(a, b) \leq m^*(a, b)$. For any two parallel decompositions $a = a_1a_2 \cdots a_n$ and $b = b_1b_2 \cdots b_n$ as in [16], we define the *penalty factor* as

$$p(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n) = |\{i \leq n : a_i = \varepsilon\}| + |\{j \leq n : b_j = \varepsilon\}|$$

and

$$\begin{aligned} &M(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n) \\ &\quad = m(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n) - p(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n) \end{aligned}$$

as the *measure of proper similarity.* The number

$$d_H(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n) = |\{i \leq n : a_i \neq b_i\}|$$

is the Hamming distance between decompositions and it is another type of penalty. We have that

$$p(a_1 \cdots a_n, b_1 \cdots b_n) \leq d_H(a_1 \cdots a_n, b_1 \cdots b_n).$$

Theorem 3.1. *Let a and b be two non-empty strings, $a = a_1a_2 \cdots a_n$ and $b = b_1b_2 \cdots b_n$ be the initial optimal decompositions, and $a = a'_1a'_2 \cdots a'_q$ and $b = b'_1b'_2 \cdots b'_q$ be the second decompositions, which are arbitrary. Denote by*

$$\begin{aligned} m_0 &= m(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n), & m_1 &= m(a'_1a'_2 \cdots a'_n, b'_1b'_2 \cdots b'_q), \\ p_0 &= p(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n), & p_1 &= p(a'_1a'_2 \cdots a'_n, b'_1b'_2 \cdots b'_q), \\ r_0 &= d_H(a_1a_2 \cdots a_n, b_1b_2 \cdots b_n), & r_1 &= d_H(a'_1a'_2 \cdots a'_n, b'_1b'_2 \cdots b'_q), \\ M_0 &= m_0 - p_0, & M_1 &= m_1 - p_1. \end{aligned}$$

The following assertions are true

1. *If $m_1 \geq m_0$, then $M_0 \geq M_1$ and $p_1 - p_2 = 2(m_1 - m_0) + 2(r_1 - r_0)$,*
2. *If $m_1 \geq m_0$ and the second decompositions are non-optimal, then $M_0 > M_1$,*
3. *If $m_1 = m_0$ and the second decompositions are optimal, then $p_0 = p_1$ and $M_0 = M_1$,*
4. *If $m_1 \leq m_0$ and the second decompositions are non-optimal, then $m_1 - r_1 < m_0 - r_0$.*

Proof. Firstly, we prove the following claims:

Claim 1. *If $m_1 > m_0$, then $M_0 > M_1$ and $p_1 - p_2 = 2(m_1 - m_0) + 2(r_1 - r_0)$.*

Assume that $M_0 \leq M_1$. Hence,

$$m_0 - p_0 \leq m_1 - p_1, p_0 \leq r_0, p_1 \leq r_1, n = m_0 + r_0, q = m_1 + r_1.$$

Moreover, $l(a) + l(b) = 2n - p_0 = 2q - p_1$. Since $m_0 < m_1$, $r_0 \leq r_1$ and $m_0 = n - r_0 < q - r_1 = m_1$, we obtain that $n < q$. From $l(a) + l(b) = 2n - p_0 = 2q - p_1$ it follows that $p_0 < p_1$.

Let $m_1 = m_0 + \delta_0$ and $p_1 = p_0 + \delta_1$, with $\delta_0 > 0$ and $\delta_1 > 0$. Then, from assumptions, we have that $m_0 - p_0 \leq m_1 - p_1 = m_0 + \delta_0 - p_0 - \delta_1 = (m_0 - p_0) + (\delta_0 - \delta_1)$. Hence

$$\delta_1 \leq \delta_0. \tag{1}$$

On the other hand, $q = m_1 + r_1 = m_0 + \delta_0 + r_1 = n - r_0 + \delta_0 + r_1$ and $q = (n + \delta_0) + (r_1 - r_0)$. Since $p_1 = 2q - l(a) - l(b)$ and $p_0 = 2n - l(a) - l(b)$, after substitutions, we obtain that $p_1 + l(a) + l(b) = p_0 + l(a) + l(b) + 2\delta_0 + 2(r_1 - r_0)$, or $p_0 + \delta_1 = p_0 + 2\delta_0 + 2(r_1 - r_0)$, or

$$\delta_1 = 2\delta_0 + 2(r_1 - r_0). \tag{2}$$

From (2), $\delta_1 > \delta_0$, a contradiction with inequality (1). Hence $M_0 > M_1$ provided that $m_1 > m_0$. From (2) it follows that $p_1 - p_0 = 2(m_1 - m_0) + 2(r_1 - r_0)$, provided that $m_1 > m_0$. The claim is proved.

Claim 2. *If $m_1 = m_0$, then $M_0 \geq M_1$ and $p_1 - p_2 = 2(r_1 - r_0)$.*

We have that $n = m_0 + r_0$ and $q = m_0 + r_1$. Since $r_0 \leq r_1$, we have that $n \leq q$. Assume that $M_0 < M_1$. Then $m_0 - p_0 < m_0 - p_1$, $p_1 = 2q - l(a) - l(b)$ and $p_0 = 2n - l(a) - l(b)$. Hence $m_0 - 2n + l(a) + l(b) < m_0 - 2q + l(a) + l(b)$, or $-2n < -2q$ and $n > q$, a contradiction.

From Claims 1 and 2, Assertions 1-3 of the Theorem 3.1 follow immediately. Since $r_1 > r_0$, from $m_1 \leq m_0$ it follows that $m_1 - r_1 < m_0 - r_0$. Assertion 4 and Theorem 3.1 are proved. □

Remark 3.1. *From Assertions 1 and 3 of Theorem 3.1 it follows that on the class of all optimal decompositions of two strings:*

- *The maximal measure of proper similarity is attained on the optimal parallel decomposition with minimal penalties (minimal measure of similarity),*

- *The minimal measure of proper similarity is attained on the optimal parallel decomposition with maximal penalties (maximal measure of similarity).*

For any two non-empty strings there are parallel decompositions with maximal measure of similarity and optimal decompositions on which the measure of similarity is minimal.

The following example shows that there are some exotic non-optimal parallel decompositions $a = a'_1 a'_2 \cdots a'_q$ and $b = b'_1 b'_2 \cdots b'_q$, such that for optimal decompositions $a = a_1 a_2 \cdots a_n$ and $b = b_1 b_2 \cdots b_n$ we have $m_1 < m_0$, $p_1 < p_0$, and $M_1 > M_0$.

Example 3.1. *Let*

$$\begin{array}{ccccccc} A & A & A & A & C & C & C \\ C & C & C & B & B & B & B \end{array}$$

be trivial optimal decompositions of strings a, b, and

$$\begin{array}{cccc} A & A & A & A \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{array} \begin{pmatrix} C & C & C \\ C & C & C \end{pmatrix} \begin{array}{cccc} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ B & B & B & B \end{array}$$

be their non-optimal decompositions. Then

$$m_1 = 3, r_1 = 8, p_1 = 8,$$

$$m_0 = 0, r_0 = 7, p_0 = 0.$$

In this example we have that $-5 = m_1 - r_1 > m_0 - r_0 = -7$ and $-5 = m_1 - p_1 = M_1 < M_0 = m_0 - p_0 = 0$.

Example 3.2. *Let*

$$\begin{array}{cccc} A & B & C & D \\ C & D & E & F \end{array} \begin{pmatrix} E \\ E \end{pmatrix} \begin{array}{c} F \\ D \end{array}$$

be trivial non-optimal decompositions of strings a, b and

$$\begin{array}{cc} A & B \\ \varepsilon & \varepsilon \end{array} \begin{pmatrix} C & D & E & F \\ C & D & E & F \end{pmatrix} \begin{array}{cc} \varepsilon & \varepsilon \\ E & D \end{array}$$

be their optimal decompositions. Then

$$m_1 = 1, r_1 = 5, p_1 = 0,$$

$$m_0 = 4, r_0 = 4, p_0 = 4.$$

We have that $m_1 - p_1 = M_1 > M_0 = m_0 - p_0$, and $m_1 - r_1 < m_0 - r_0$.

The above examples show that Theorem 3.1 cannot be improved in the case of $m_1 < m_0$.

Decompositions with minimal penalty and maximal proper similarity are of significant interest. Moreover, if we solve the problem of text editing and correction, the optimal decompositions are more favorable. Therefore, the optimal decompositions are the best parallel decompositions and we may solve the string match problems only on class of optimal decompositions.

Remark 3.2. *The optimal decompositions:*

- *describe the proper similarity of two strings,*
- *permit to obtain long common sub-sequences,*
- *permit to calculate the distance between strings,*
- *permit to appreciate changeability of information over time.*

4 Relations to Hamming and Levenshtein Distances

If $a, b \in L(a, b)$ and $a = a_1 a_2 \cdots a_n, b = b_1 b_2 \cdots b_m$ are the canonical decompositions, then for $m \leq n$ the number

$$d_H(a,b) = d_H(b,a) = |\{i \leq m : a_i \neq b_i\}| + n - m$$

is called the *Hamming distance* [11] between strings a and b.

The *Levenshtein distance* [12] between two strings $a = a_1 a_2 \cdots a_n$ and $b = b_1 b_2 \cdots b_m$ is defined as the minimum number of insertions, deletions, and substitutions required to transform one string to the other. A formal definition of Levenshtein's distance $d_L(a,b)$ is given by the following formula:

$$d_L(a_1 \cdots a_i, b_1 \cdots b_j) = \begin{cases} i, & \text{if j=0,} \\ j, & \text{if i=0,} \\ \min \begin{cases} d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_j) + 1 \\ d_L(a_1 \cdots a_i, b_1 \cdots b_{j-1}) + 1 \\ d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_{j-1}) + 1_{(a_i \neq b_j)}, \end{cases} \end{cases}$$

where $1_{(a_i \neq b_j)}$ equals to 0 if $a_i = b_j$ and to 1 otherwise.

Theorem 4.1. $d_L(a,b) = \rho^*(a,b) \leq d_H(a,b)$ *for any* $a, b \in L(A)$.

Proof. To prove the equality $d_L(a,b) = \rho^*(a,b)$, we will first prove that $d_L(a,b) \leq \rho^*(a,b)$, and then that $d_L(a,b) \geq \rho^*(a,b)$.

We begin with the observation that the parallel decompositions of two strings a, b allow more transparent evaluation of the Levenshtein distance $d_L(a,b)$. If $a = v_1 u_1 v_2 u_2 \cdots v_n$ and $b = w_1 u_1 w_2 u_2 \cdots w_n$ are optimal parallel decompostions, then for transformation of b to a it is sufficient to transform any w_i to v_i. The cost of transformation of w_i to v_i is $\leq \max\{l(w_i), l(v_i)\}$. Hence $d_L(a,b) \leq \rho^*(a,b)$.

The proof of the inequality $d_L(a,b) \geq \rho^*(a,b)$ is based on the Levenshtein distance formula, as well as the construction of the transformation of string a to string b. We observe that the Levenshtein distance is calculated recursively using the *memoization* matrix and *dynamic programming* technique [7, pp. 359–378]. A small snapshot of the memoization matrix calculation is presented below.

Table 1. Construction of memoization matrix for Levenshtein distance

Diag	Above
Left	min(Above + delete, Left + insert, Diag + $1_{a_i \neq b_j}$)

Distance d_L calculated on subtrings $a_1 \cdots a_i$ of string a and substring $b_1 \cdots b_j$ of string b is equal to the minimum of the following values:

- $d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_j) + 1,$ (1)

- $d_L(a_1 \cdots a_i, b_1 \cdots b_{j-1}) + 1,$ (2)

- $d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_{j-1}) + 1_{a_i \neq b_j}.$ (3)

Remark : the operation (1) is the delete operation, (2) is the insert operation, and (3) is the substitution operation.

Once all of the above values are calculated and the memoization matrix is filled, the distance is given by the value in the cell on the n^{th} row and m^{th} column.

The construction of the transformation of string a into string b is based on the values of the memoization matrix. At each point of the construction process, we will execute operations on both strings a and b, and obtain another pair of strings a' and b' equivalent to the initial pair a and b. We use the top-down analysis approach to describe the transformation process step by step. The process below starts with $i = n, j = m$, $p = 0$, $q = 0$ and both a', b' as empty strings:

- if when calculating $d_L(a_1 \cdots a_i, b_1 \cdots b_j)$ we used operation (1), then we deleted a character from string a at position i, which is equivalent to inserting the ε character in string b at the corresponding position. In this case, in the building process of a' and b', we put $p := p+1$, $v'_p = \{a_i\}$,$w'_p = \{\varepsilon\}$, $a' := v'_p \cup a'$, $b' := w'_p \cup b'$. Next, we proceed to calculate $d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_j)$.

- if when calculating $d_L(a_1 \cdots a_i, b_1 \cdots b_j)$ we used operation (2), then we inserted the ε character in string a at position i. In this case, in the building process of a' and b', we put $p := p + 1$, $v'_p = \{\varepsilon\}$,$w'_p = \{b_j\}$, $a' := v'_p \cup a'$, $b' := w'_p \cup b'$. Next, we proceed to calculate $d_L(a_1 \cdots a_i, b_1 \cdots b_{j-1})$.

- if when calculating $d_L(a_1 \cdots a_i, b_1 \cdots b_j)$ we used operation (3), then we either substituted the character at position i of string a with the character at position j of string b, or we did not make any change in case if $a_i = b_j$. If $a_i = b_j$, we put $q =: q + 1$,$u'_q = \{a_i\}$, $a' := u'_q \cup a'$, $b' := u'_q \cup b'$. If $a_i \neq b_j$, we put $p =: p+1$, $v'_p = \{a_i\}$, $w'_p = \{b_j\}$, $a' := v'_p \cup a'$, $b' := w'_p \cup b'$. Next, we proceed to calculate $d_L(a_1 \cdots a_{i-1}, b_1 \cdots b_{j-1})$.

According to the above steps, we observe that string a' is equivalent to string a, and string b' is equivalent to b by construction. But, we also have that the decomposition $a' = v'_p u'_q v'_{p-1} u'_{q-1} \cdots u'_1 v'_1$ and $a' = w'_p u'_q w'_{p-1} u'_{q-1} \cdots u'_1 w'_1$ obtained from the above construction process, represent a parallel decomposition of strings a and b. Thus, we have that $d_L(a,b) = E(a,b) \geq \rho^*(a,b)$. This completes the proof of the equality $d_L(a,b) = \rho^*(a,b)$.

We will now prove the second part of the theorem, namely that $\rho^*(a,b) \leq d_H(a,b)$. Let $d_H(a,b) < max\{l(a), l(b)\} = n$, where $n = l(a) \geq l(b) = m$. Then $a = a_1 a_2 \cdots a_n, b = b_1 b_2 \cdots b_m$, $a_i \neq \varepsilon$ for any $i \leq n$, and or $m = 1$ and $b_1 = \varepsilon$, or $b_j \neq \varepsilon$ for any $j \leq m$. In this case $d_H(a,b) = n - |\{i \leq m : a_i = b_i\}|$ and we have the representations $a = (a_1)(a_2) \cdots (a_m)(a_{m+1} \cdots a_n)$ and $b = (b_1)(b_2) \cdots (b_m)(\varepsilon)$ which generate two parallel decompositions α, β with $E(\alpha, \beta) = d_H(a,b)$. Therefore $\rho^*(a,b) \leq E(\alpha,\beta) = d_H(a,b)$. The proof is complete.

□

Corollary 4.1. *Distance d_L is strictly invariant, i.e. $d_L(ac, bc) = d_L(ca, cb) = d_L(a,b)$ for any $a, b, c \in L(A)$.*

Remark 4.1. *The Hamming distance d_H is not invariant.*

Example 4.1. *Let $n = m + p$ and strings $a = (01)^n, b = (10)^m$, $c = (01)^p$. We obtain the following distance values for the above strings:*

$$d_L(a,b) = 2p, \rho^*(a,b) = 2p, d_H(a,b) = 2n,$$
$$d_L(ac,bc) = 2p, \rho^*(ac,bc) = 2p, d_H(ac,bc) = 2n.$$

Remark 4.2. *If $l(a) = l(b)$, then $d_H(ac, bc) = d_H(a,b)$ for any $a, b, c \in L(A)$. Additionally, the following equality always holds:*

$$d_H(ca, cb) = d_H(a,b).$$

5 Applications

First and foremost let us look at how we can apply the results of this article in information distance problems such as string search, text correction, and pattern matching. We have presented one such example in the previous section – the edit distance.

We also mentioned the problem of DNA/RNA sequence alignment, which goes back as early as 1970 [16]. Other bioinformatic applications of the distance ρ^* include phylogenetic analysis, whole genome phylogeny, and detection of acceptable mutations.

We begin this section with the pseudo-codes of two algorithms: distance calculation and decompositions alignment.

The first algorithm describes how to calculate the distance between two strings a and b. The approach is based on dynamic programming and it has a complexity of $O(mn)$, where m and n are the lengths of a and b.

Algorithm 1.

```
Description: Computes the metric $\rho^*$ on strings a and b.
Input: Strings $a, b \in L(A)$
Output: Value of $\rho^*(a, b)$
Initialisation: $m := l(a)$, $n := l(b)$, $D[m, n] := 0$
Pseudocode:
for i := 0 to m D[i,0] := i;
for j := 0 to n D[0,j] := j;
for j := 1 to n do
        for i := 1 to m do
              if a[i]= b[j] then
                        D[i,j] := D[i-1,j-1]
              else
                        D[i,j] := min(D[i-1,j] + 1,
                        min(D[i,j-1] + 1, D[i-1,j-1] + 1));
return D[m,n];
```

The algorithm that follows constructs the optimal parallel decompositions of strings a and b that give the value of distance ρ^*. This algorithm uses the memoization matrix $D[m, n]$ calculated in the previous algorithm. The idea is to traverse from the bottom right cell $D[m, n]$ to the top left cell $D[0, 0]$ and at each step to evaluate whether the minimal distance was obtained by replacement, deletion or insertion. The algorithm uses recursive *backtracking* to reconstruct all decompositions of strings a and b. We modified the classical version of the pseudo-code to print only the most optimal decomposition, instead of printing all possible paths.

Algorithm 2.

```
Description: Constructs optimal parallel decompositions
of strings a and b.
Input: $n, m$ - current indexes in matrix D
        $a_r, b_r$ - recontructed decompositions
Output: Optimal parallel decompositions of strings a, b
Initialisation: Read $D[m, n]$ from Algorithm 1
Pseudocode:
if (n=0) and (m=0) then return $a_r, b_r$
if ((n>0)and(m>0)) and((D[n,m]=D[n-1,m-1]+$c_{dist}$)
      or ((D[n,m]=D[n-1,m-1]) and ($c_{dist}$=0)))
            then recOPD(n-1, m-1, $a_r + a[n]$, $b_r + b[m]$)
    else
    if (n>0) and (D[n,m]=D[n-1,m] +$cost_r$)
            then recOPD(n-1, m, $a_r$+a[n], $b_r+\varepsilon$)
    else
    if (m>0) and (D[n,m]=D[n,m-1] +$cost_i$)
            then recOPD(n, m-1, $a_r+\varepsilon$, $b_r$+b[m])
```

In the worst case scenario its complexity is $O(m+n)$ (this happens when we separately traverse the matrix horizontally and vertically). This result is achieved with the help of prioritizing the direction of analysis when traversing the matrix. We first look to the north-west and only afterwards to the northern and western cell values. We stop the reconstruction process once the algorithm reaches the cell at $D[0, 0]$. The reasoning behind this decision is to find the most optimal decomposition among all possible decompositions of strings a and b. The example that follows is a good illustration of this approach.

Example 5.1. *Let's investigate the example where $a =$ industry and $b =$ interest. In this case we have $\rho^*(a, b) = 6$. The possible decompositions of strings a and b are as follows:*

industry	*in$\varepsilon\varepsilon$dustry*	*inεdεustry*	*ind$\varepsilon\varepsilon$ustry*	*inεduεstry*
interest	*interest$\varepsilon\varepsilon$*	*interest$\varepsilon\varepsilon$*	*interest$\varepsilon\varepsilon$*	*interest$\varepsilon\varepsilon$*

The first pair of parallel string decompositions is the optimal one as it has minimal string length. Another good example of two strings decomposition into their building blocks u_i, v_j, and w_j is illustrated below.

Example 5.2. *Consider the alphabet* $\bar{A} = \{\varepsilon, X, Y, Z, W\}$ *and two strings* $a = XXYYWZYX$ *and* $b = YXXWZWXY$. *For this example we obtain that* $\rho^*(a, b) = 5$ *as well as the following optimal decomposition:*

$$\begin{matrix}\varepsilon\\ Y\end{matrix}\begin{pmatrix}X & X\\ X & X\end{pmatrix}\begin{matrix}Y & Y\\ W & Z\end{matrix}\begin{pmatrix}W\\ W\end{pmatrix}\begin{matrix}Z\\ X\end{matrix}\begin{pmatrix}Y\\ Y\end{pmatrix}\begin{matrix}X\\ \varepsilon\end{matrix}$$

Lets look at results in detection of the mutational events. We extend the parallel decompositions and present the construction of the *semiparallel decompositions.* We take into consideration the ordering $\preceq$ and the corresponding distance ρ^*_l. From this point of view, for any two strings $a, b \in L(A)$ we find the decompositions of the form $a = v_1u_1v_2u_2 \cdots v_ku_kv_{k+1}$ and $b = w_1u'_1w_2u'_2 \cdots w_ku'_kw_{k+1}$, where

- u_i, u'_i are canonical substrings of the strings a and b and u_i, u'_i may be empty strings;
- v_j is a substring of a and v_j may be an empty string;
- w_j is a substring of b and w_j may be an empty string;
- $\rho^*_l(u_i, u'_i) = 0$ for all $i \leq k$.

Like in the case with parallel decompositions, the semiparallel decompositions are optimal if

$$\rho^*_l(a, b) = \Sigma\{\rho(v_i, w_i) : i \leq k + 1\}.$$

This given interpretation of the metric and string decompositions can be used in the study of the minimum number of acceptable and unacceptable (when metric ρ^*_r is used) *mutational events* required to convert one sequence to another.

To illustrate the application of the semiparallel decomposition let us partition the strings from the previous example.

Example 5.3. *Let $a = XXYYWZYX$ and $b = YXXWZWXY$, with the alphabet $\bar{A} = \{\varepsilon, X, Y, Z, W\}$, on which we consider the classic ordering $\preceq$, meaning that $\rho_l^*(z_i, z_j) = 0$ for all $z_i, z_j \in \bar{A}$, where $z_i \preceq z_j$. This time we obtain that $\rho_l^*(a, b) = 3$, as well as the following optimal decomposition:*

$$\begin{pmatrix} X & X \\ Y & X \end{pmatrix} \begin{matrix} Y \\ X \end{matrix} \begin{pmatrix} Y \\ W \end{pmatrix} \begin{matrix} W \\ Z \end{matrix} \begin{pmatrix} Z \\ W \end{pmatrix} \begin{matrix} Y \\ X \end{matrix} \begin{pmatrix} X \\ Y \end{pmatrix}$$

For semiparallel decompositions we can define measure of similarity, penalty, and proper similarity.

Remark 5.1. *Our algorithms are effective for any quasimetric on $\bar{A}$. Some authors consider the possibility to define the generalized Levenshtein metric with distinct values $\rho(a, b)$ and $\rho(b, a)$. It is necessary to require that $\rho(a, b)$ is a quasimetric. In other cases we may obtain some confusions as will be seen from the next example.*

Example 5.4. *Let $A = \{a, b\}, \bar{A} = \{\varepsilon, a, b\}$. The following table defines the distance ρ on $\bar{A}$:*

0	0	1	ε
1	0	0	a
0	1	0	b
ε	a	b	y \ x

In this example we have $0 = \rho(a, b) + \rho(b, \varepsilon) < \rho(a, \varepsilon) = 1$ and:

1. *for $u = aba, v = ba$ we get $\bar{\rho}(u, v) = \bar{\rho}(v, u) = 0$,*
2. *for $u = a, v = b$ we get $\bar{\rho}(u, v) = \bar{\rho}(v, u) = 0$, when $\rho(v, u) = 1$.*

Example 5.5. *Let us examine the example from [16] in the context of the results achieved. We have strings $a = AJCJNRCKCRBP$ and $b = ABCNJROCLCRPM$ for which there are eight pairs of optimal decompositions. We present two of them, the shortest and the longest:*

$$\begin{pmatrix} A \\ A \end{pmatrix} \begin{matrix} J \\ B \end{matrix} \begin{pmatrix} C \\ C \end{pmatrix} \begin{matrix} \varepsilon \\ N \end{matrix} \begin{pmatrix} J \\ J \end{pmatrix} \begin{matrix} N & R \\ R & O \end{matrix} \begin{pmatrix} C \\ C \end{pmatrix} \begin{matrix} K \\ L \end{matrix} \begin{pmatrix} C & R \\ C & R \end{pmatrix} \begin{matrix} B & P \\ P & M \end{matrix}$$

$$\begin{pmatrix} A \\ A \end{pmatrix} \begin{matrix} J \\ B \end{matrix} \begin{pmatrix} C \\ C \end{pmatrix} \begin{matrix} J \\ \varepsilon \end{matrix} \begin{pmatrix} N \\ N \end{pmatrix} \begin{matrix} \varepsilon \\ J \end{matrix} \begin{pmatrix} R \\ R \end{pmatrix} \begin{matrix} \varepsilon \\ O \end{matrix} \begin{pmatrix} C \\ C \end{pmatrix} \begin{matrix} K \\ L \end{matrix} \begin{pmatrix} C & R \\ C & R \end{pmatrix} \begin{matrix} B \\ \varepsilon \end{matrix} \begin{pmatrix} P \\ P \end{pmatrix} \begin{matrix} \varepsilon \\ M \end{matrix}$$

For the first pair we have $\rho^* = 7$, $m = 6$, $p = 1$, and $M = 5$. For the second pair we have $\rho^* = 7$, $m = 8$, $p = 5$, and $M = 3$. Our algorithms allow us to calculate all optimal decompositions with distinct measure of similarity. Authors from [16] prefer the second pair of decomposition since it has maximal possible measure of similarity. We consider more preferable the first pair, which has the maximal proper similarity.

6 Conlusion

We showed that there are invariant distances on $L(A)$ closely related to Levenshtein's distance, which help us solve various problems in mathematics, computer science, and bioinformatics. The results can be applied in different areas such as data correction of signals transmitted over channels with noise, finding matching DNA sequence after mutations, text searching with possible typing errors, and estimation of dialect pronunciations proximity [8], [14]. For construction of the matching sequence we propose the method of optimal decompositions of strings, priority of which is confirmed by Theorem 3.1. Our distances of ρ^* type can be defined for distinct values $\rho(a, b)$ of strings a,b, in general, and for $\rho(a, b) \neq \rho(b, a)$. In such a case, the metric can be used in solving the stable marriage problem [10].

References

[1] A. V. Arhangel'skii, "Mappings and spaces," *Uspekhi Mat. Nauk,* vol. 21, no. 4, pp. 133–184, 1966. [in Russian] (English translation: *Russian Math. Surveys,* vol. 21, no. 4, PP. 115–162, 1966).

[2] V. B. Barahnin, V. A. Nehaeva, and A. M. Fedotov, "Prescription of the similarity measure for clustering text documents," *Vestnik Novosib. Gos. Univ., Ser.: Informacionnye tehnologii*, vol. 1, pp. 3–9, 2008. [in Russian]

[3] M. M. Choban, "The theory of stable metrics," *Math. Balkanica,* vol. 2, pp. 357–373, 1988.

[4] M. M. Choban, "Some topics in topological algebra," *Topol. Appl.,* vol. 54, pp. 183–202, 1993.

[5] M. M. Choban and I. A. Budanaev, "Distances on Monoids of Strings and Their Applications," In *Conference on Mathematical Foundations of Informatics: Proceedings MFOI2016, July 25-29, 2016, Chisinau, Republic of Moldova*, Chişinău, Institute of Mathematics and Computer Science, pp. 144–159, 2016. ISBN: 978–9975–4237–4–8

[6] M. M. Choban and L. L. Chiriac, "On free groups in classes of groups with topologies," *Bul. Acad. Ştiinţe Repub. Moldova, Matematica,* no. 2-3, pp. 61–79, 2013.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms.* (3rd ed.), MIT Press and McGraw-Hill, 2009. ISBN: 0–262–03384–4.

[8] M. M. Deza and E. Deza, *Encyclopedia of Distances*, Berlin: Springer, 2009. ISBN: 978-3-642-00233-5; e-ISBN: 978-3-642-00234-2; DOI 10.1007/978-3-642-00234-2.

[9] M. I. Graev, "Free topological groups," *Izv. Akad. Nauk SSSR Ser. Mat.,* vol. 12, no. 3, pp. 279–324, 1948. [in Russian] (English translation: *Amer. Math. Soc. Transl.* (1), vol. 8, pp. 305–364, 1962).

[10] D. Gusfield and R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, Cambridge, MIT Press, 1989. ISBN: 9780262515528.

[11] R. W. Hamming, "Error Detecting and Error Correcting Codes," *The Bell System Technical Journal*, vol. 29, no 2, pp. 147–160, 1952.

[12] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *DAN SSSR,* vol. 163, no 4, pp. 845–848, 1965. [in Russian] (English translation: *Soviet Physics – Doklady*, vol. 10, no. 8, pp. 707–710, 1966).

[13] A. A. Markov, "On free topological groups," *Izv. Akad. Nauk. SSSP, Ser. Matem.*, vol. 9, no. 1, pp. 3–64, 1945. [in Russian]

(English translation: *Amer. Math. Soc. Transl.* (1), vol 8, no. 1, pp. 195–272, 1962).

[14] G. Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys,* vol. 33, no. 1, pp. 31–88, 2001.

[15] S. I. Nedev, "o-metrizable spaces," *Trudy Moskov. Mat.Ob-va,* vol. 24, pp. 213–247, 1974. [in Russian] (English translation: *Trans. Moscow Math. Soc.*, vol. 24, pp. 213–247, 1974).

[16] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology,* vol. 48, no 3, pp. 443–453, 1970.

[17] S. Romaguera, M. Sanchis and M. Tkachenko, "Free paratopological groups," *Topology Proceed.*, vol. 27, no 2, pp. 613–640, 2003.

[18] C. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal,* vol. 27, pp. 379–423, pp. 623–656, 1948.

Mitrofan Choban, Ivan Budanaev,

Received September 22, 2016

Mitrofan Choban
Professor, Doctor of Science,
Academician of the Academy of Science of Moldova
Tiraspol State University, Republic of Moldova
str. Iablochkin 5, Chisinau, Moldova
Phone: +373 22 754906
E–mail: mmchoban@gmail.com

Ivan Budanaev
Doctoral School of Mathematics and Information Science
Institute of Mathematics and Computer Sciences of ASM
Tiraspol State University, Republic of Moldova
str. Academiei, 3/2, MD-2028, Chisinau, Moldova
Phone:+373 60926999
E–mail: ivan.budanaev@gmail.com