Computer Science Journal of Moldova, vol.**27**, no.3(81), 2019

On Operations over Language Families

Alexander Meduna, Radim Krčmář,
Adam Kövári, Zuzana Beníčková

Abstract

Let O and $\mathbf{F}$ be an operation and a language family, respectively. So far, in terms of closure properties, the classical language theory has only investigated whether $O(\mathbf{F}) \subseteq \mathbf{F}$, where $O(\mathbf{F})$ is the family resulting from O applied to all members of $\mathbf{F}$. If $O(\mathbf{F}) \subseteq \mathbf{F}$, $\mathbf{F}$ is closed under O; otherwise, it is not.

This paper proposes a finer and wider approach to this investigation. Indeed, it studies almost all possible set-based relations between $\mathbf{F}$ and $O(\mathbf{F})$, including $O(\mathbf{F}) = \emptyset$; $F \not\subset O(\mathbf{F})$, $O(\mathbf{F}) \not\subset \mathbf{F}$, $\mathbf{F} \cap O(\mathbf{F}) \neq \emptyset$; $\mathbf{F} \cap O(\mathbf{F}) = \emptyset$, $O(\mathbf{F}) \neq \emptyset$; $O(\mathbf{F}) = \mathbf{F}$; and $\mathbf{F} \subset O(\mathbf{F})$. Many operations are studied in this way. A sketch of application perspectives and open problems closes the paper.

Keywords: language operations; language families; closure properties; finer approach; new trend; set theory.

MSC 2010: 68Q45, 03D05.

1 Introduction

Over its entire history, formal language theory has primarily studied operations over language families in terms of closure properties by analogy with the investigation of these properties in discrete mathematics as a whole. To give an insight into this study, consider a languare family $\mathbf{F}$, a language operation O, and $O(\mathbf{F})$ as the language family resulting from the application of O to all languages in $\mathbf{F}$. In essence, so far, the language theory has restricted its attention only to the study whether or not $O(\mathbf{F}) \subseteq \mathbf{F}$. If so, $\mathbf{F}$ is closed with respect to O; otherwise, it is not.

The authors believe that formal language theory can approach the investigation of operations over language families in a much finer way than it has been done so far. In fact, there already exist many results that can be combined together to express some properties related to closure or non-closure results more precisely; unfortunately, formal language theory lacks a proper universal terminology or framework, which would allow it to express results of this kind in a uniform and general way. To illustrate this finer approach by a simple example, recall that the family of context-sensitive languages is not closed under homomorphism (see the Corollary on p. 279 in [8]). Apart from this non-closure result, it is well known that every recursively enumerable language L coincides with h(K), where h is a homomorphism and K is a context-sensitive language. Of course, if a language is not recursively enumerable, it cannot be expressed in this way. Putting these results together, we can naturally say that homomorphism expands the family of context-sensitive languages onto that of recursively enumerable languages. Therefore, the present paper proposes a new terminology for results like this and illustrates it by many examples, observations and results.

More specifically, the present paper introduces these notions— (1) if $\mathbf{F} - O(\mathbf{F}) \neq \emptyset$, O reduces $\mathbf{F}$; (2) if $O(\mathbf{F}) = \emptyset$, O eliminates $\mathbf{F}$; (3) if $O(\mathbf{F}) \subset \mathbf{F}$, O properly reduces $\mathbf{F}$; (4) if $\mathbf{F} \not\subset O(\mathbf{F}), O(\mathbf{F}) \not\subset \mathbf{F}$, and $\mathbf{F} \cap O(\mathbf{F}) \neq \emptyset$, then O incomparably reduces $\mathbf{F}$; (5) if $\mathbf{F} - O(\mathbf{F}) = \mathbf{F}$ and $O(\mathbf{F}) \neq \emptyset$, O expels $\mathbf{F}$; (6) if $O(\mathbf{F}) = \mathbf{F}$, O unchanges $\mathbf{F}$; and (7) if $\mathbf{F} \subset O(\mathbf{F})$, O expands $\mathbf{F}$. In terms of these notions, the paper discusses a broad variety of operations, ranging from classical operations, such as complement, up to newly introduced operations. It starts from utterly straightforward observations about simple operations and gradually proceeds towards more complicated operations and results concerning them. Sometimes, it applies these operations to well-known language families, such as the family of linear languages. Most often, however, as the main direction of this newly proposed investigation trend, the paper establishes general results concerning language families satisfying some prescribed properties. In its conclusion, in a greater detail, the paper suggests several special branches of study within this

newly suggested trend as well as application perspectives.

2 Preliminaries

This paper assumes that the reader is familiar with discrete mathematics (see [4]). Most importantly, it assumes an in-depth knowledge of the language theory (see [2], [3], [5]). Let X and Y be two sets. X and Y are comparable if $X \subseteq Y$ or $Y \subseteq X$; otherwise, X and Y are incomparable. In other words, X and Y are incomparable if and only if $X \not\subseteq Y$ and $Y \not\subseteq X$ (notice that if X and Y are disjoint, then they are necessarily incomparable). For a set Q, $power(Q)$ denotes the power set of Q, and $card(Q)$ denotes its cardinality. For a total function f over Q, f^i denotes its ith power, for $i \geq 0$. $\mathbb{N}$ denotes the set of natural numbers.

For an alphabet V, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. Any member $w \in V^*$ is called a *word*. For any $w \in V^*, |w|$ and $reversal(w)$ denote the length of w and the reversal of w, respectively. For any $N \subseteq V, occur(w, N)$ denotes the number of symbols from N occurring in w. For every $i \in \{0, 1, \ldots, |w|\}$, $suffix(w, i)$ denotes the suffix of w of length i; analogously, $prefix(w, i)$ denotes the prefix of w of which length is i. A language L is any subset of V^*, $L \subseteq V^*$. Let **ALL** denote the set of all languages; in other words, **ALL** represents the *universal set* of languages throughout this paper. A *language family* **F** is any subset of **ALL**, $\mathbf{F} \subseteq \mathbf{ALL}$; notice that $\mathbf{F} \subseteq \mathbf{ALL}$ is synonymous with $\mathbf{F} \in power(\mathbf{ALL})$. Observe that both $\emptyset$ and $\{\emptyset\}$ are language families, but $\emptyset \neq \{\emptyset\}$; indeed, $card(\emptyset) = 0$ while $\text{card}(\{\emptyset\}) = 1$. Set $alph(L) = \{a \mid a \text{ occurs in a word in } L\}$, and $alph(\mathbf{F}) = \{a \mid a \in alph(L), L \in \mathbf{F}\}$.

A *grammar* is a quadruple, $G = (N, T, P, S)$, where N and T are nonterminal and terminal alphabets, respectively; $N \cap T = \emptyset$. N contains S—the start symbol of G. P is a finite non-empty set of productions of the form $x \to y$, where $x, y \in (N \cup T)^*$ so $N \cap alph(x) \neq \emptyset$. For every $p \in P$ of the form $x \to y$, x is the *left-hand side* of p, $lhs(p)$,

and y is the *right-hand side* of p, $rhs(p)$. To express that $card(N) = n$, where $n \in \mathbb{N}$, we write ${}_nG$. If $x \to y \in P$, $v = uxz$, $w = uyz$ with $u, z \in (N \cup T)^*$, then v *directly derives* w in G, symbolically written as $v \Rightarrow w$ in G. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The *language of* G, $L(G)$, is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$, and any derivation of the form $S \Rightarrow^* w$ in G with $w \in T^*$ is called a *successful derivation*.

Let $G = (N, T, P, S)$ be a grammar. G is referred to as *context-sensitive* if every production in P is of the form $uAv \to uyv$ with $A \in N$, $u, v \in (N \cup T)^*$, $y \in (N \cup T)^+$. G is referred to as *context-free* if every $x \to y \in P$ satisfies $x \in N, y \in (N \cup T)^*$. A context-free grammar is in the *Chomsky normal form* if every $x \to y \in P$ satisfies $y \in NN \cup T$. Let $j \in \mathbb{N}$; a context-free grammar G is referred to as j-linear if for any $x \to y \in P$, $S \notin alph(y)$ and either $x = S$ and $occur(y, N) \leq j$ or $x \neq S$ and $occur(y, N) \leq 1$. Instead of a 1-linear grammar, we often simply say a *linear grammar*. For $j \in \mathbb{N}$; ${}_j\mathbf{LIN}$ denotes the language family generated by j-linear grammars; recall that for every $k \in \mathbb{N}$; ${}_k\mathbf{LIN} \subset {}_{k+1}\mathbf{LIN}$ (see Theorem 8.3 in [5]).

REG, **LIN**, **CF**, **CS**, **DEC**, and **RE** denote the families of regular, linear, context-free, context-sensitive, decidable, and recursively enumerable languages, respectively. Recall (see [2])

$$\mathbf{REG} \subset \mathbf{LIN} \subset \mathbf{CF} \subset \mathbf{CS} \subset \mathbf{DEC} \subset \mathbf{RE} \subset \mathbf{ALL}$$

3 Definitions

Let $n \in \mathbb{N}$. An n-ary *language operation* on **ALL** is a total function O from the n-ary Cartesian product $\mathbf{ALL}^n$ into **ALL**. An n-ary language-family operation over $power(\mathbf{ALL})$ is a total function O from the n-ary Cartesian product $(power(\mathbf{ALL}))^n$ into $power(\mathbf{ALL})$.

Throughout this paper, we discuss only unary and binary language operations, and we only consider unary language-family operations. For any $\mathbf{F} \subseteq \mathbf{ALL}$, $O(\mathbf{F})$ denotes the image of $\mathbf{F}$ over O, so $O(\mathbf{F}) = \{O(L) \mid L \in \mathbf{F}\}$. In what follows, we automatically assume that $\mathbf{F} \neq \emptyset$

(of course, $O(\mathbf{F}) = \emptyset$ is not ruled out). Notice that in the next main definition of this paper, 2 through 5 are, in effect, special cases of 1.

Definition 3.1. Let $\mathbf{F} \subseteq \mathbf{ALL}$.

1. If $\mathbf{F} - O(\mathbf{F}) \neq \emptyset$, O reduces $\mathbf{F}$.
2. If $O(\mathbf{F}) = \emptyset$, O eliminates $\mathbf{F}$.
3. If $O(\mathbf{F}) \subset \mathbf{F}$, O properly reduces $\mathbf{F}$.
4. If $\mathbf{F} \not\subset O(\mathbf{F}), O(\mathbf{F}) \not\subset \mathbf{F}$, and $\mathbf{F} \cap O(\mathbf{F}) \neq \emptyset$, then O incomparably reduces $\mathbf{F}$.
5. If $\mathbf{F} - O(\mathbf{F}) = \mathbf{F}$ and $O(\mathbf{F}) \neq \emptyset$, O expels $\mathbf{F}$.
6. If $O(\mathbf{F}) = \mathbf{F}$, O unchanges $\mathbf{F}$.
7. If $\mathbf{F} \subset O(\mathbf{F})$, O expands $\mathbf{F}$.

Suppose that O expands $\mathbf{F}$ and $O(\mathbf{F}) = \mathbf{W}$, where $\mathbf{W}$ is a well-known language family, such as any of the families listed in the conclusion of Section 2. Under these circumstances, we sometimes explicitly point out that O expands $\mathbf{F}$ onto $\mathbf{W}$ (see, for instance, Observation 4.13 and Theorem 4.28). We often make analogical statements in terms of the other parts of Definition 3.1 unless a confusion arises (see, for instance, Observation 4.6).

4 Results

Simply put, the present section illustrates Definition 3.1 by a large variety of language-family operations. Starting from part 1, it proceeds, in essence, towards part 7 of the definition. Consequently, it begins with a variety of reducing operations and ends with expanding operations. As far as the mathematical level is concerned, the section opens its discussion with utterly trivial operations and closes it with more complicated operations and results about them.

4.1 Reducing Operations

For all $\mathbf{F} \in power(\mathbf{ALL})$, define operations
$EmptySetConcatenation(\mathbf{F}) = \{L\emptyset \mid L \in \mathbf{F}\}$, and
$EmptyStringConcatenation(\mathbf{F}) = \{L\{\varepsilon\} \mid L \in \mathbf{F}\}$. The next observation is obvious.

Observation 4.1. *For all* $\mathbf{F} \in power(\mathbf{ALL})$,
$EmptySetConcatenation(\mathbf{F}) = \emptyset$ *and* $EmptyStringConcatenation(\mathbf{F}) = \mathbf{F}$. *In words, the former eliminates* $\mathbf{F}$ *while the latter unchanges* $\mathbf{F}$.

Let Complement denote the well-known unary language operation of complement. For all $\mathbf{F} \in power(\mathbf{ALL})$, define $Complement(\mathbf{F}) = \{Complement(L) \mid L \in \mathbf{F}\}$.

Observation 4.2. $Complement(\mathbf{RE}) = \mathbf{DEC} \cup Complement(\mathbf{RE} - \mathbf{DEC})$, *where*
$Complement(\mathbf{RE} - \mathbf{DEC}) \subset \mathbf{ALL} - \mathbf{RE}$. *Therefore, Complement incomparably reduces* $\mathbf{RE}$.

Proof. Let $L \in \mathbf{RE}$. If $L \in \mathbf{DEC}$, then $Complement(L) \in \mathbf{DEC}$ (see Theorem 18.3 in [6]). If $L \in \mathbf{RE} - \mathbf{DEC}$, $Complement(L) \notin \mathbf{RE}$ because $L \in \mathbf{DEC}$ if and only if $L \in \mathbf{RE}$ and $Complement(L) \in \mathbf{RE}$ (see Theorem 4.22 in [7]). Thus, $Complement(\mathbf{RE}) = \mathbf{DEC} \cup Complement(\mathbf{RE} - \mathbf{DEC})$ with $Complement(\mathbf{RE} - \mathbf{DEC})$ out of $\mathbf{RE}$, so $Complement(\mathbf{RE} - \mathbf{DEC}) \subset \mathbf{ALL} - \mathbf{RE}$. The rest of this observation follows from part 4 of Definition 3.1. □

Let $a \in alph(\mathbf{ALL})$. For all $L \in \mathbf{ALL}$, define operation $a\text{-}End(L) = L\{a\}$, and for all $\mathbf{F} \subseteq \mathbf{ALL}$, $a\text{-}End(\mathbf{F}) = \{a\text{-}End(L) \mid L \in \mathbf{F}\}$. As usual, $a\text{-}End^i$ denotes the ith power of $a\text{-}End$, $i \geq 1$. Consider these two disjoint language families

$$\mathbf{ODD}_a = \{L \mid L \subseteq \{a\}^*, |x| \text{ is odd for all } x \in L\}, \text{ and}$$

$$\mathbf{EVEN}_a = \{L \mid L \subseteq \{a\}^*, |x| \text{ is even for all } x \in L\}.$$

Observation 4.3. *Let* $i \geq 1$. *Then:*

$a\text{-}End^i(\mathbf{ODD}_a) = \mathbf{EVEN}_a$ *if* i *is odd, and*

$a\text{-}End^i(\mathbf{EVEN}_a) = \mathbf{ODD}_a$ *if* i *is even.*

Proof. Clear. □

Corollary 4.4. $a\text{-}End(\mathbf{ODD}_a) = \mathbf{EVEN}_a$ *and* $a\text{-}End(\mathbf{EVEN}_a) = \mathbf{ODD}_a$. *In words, a-End expels* $\mathbf{ODD}_a$ *onto* $\mathbf{EVEN}_a$, *and it expels* $\mathbf{EVEN}_a$ *onto* $\mathbf{ODD}_a$.

Proof. Consider Observation 4.3 for $i = 1$ to see that this corollary holds true. □

Define the homomorphism $a\text{-}Coding$ as $a\text{-}Coding(b) = a$ for all $b \in alph(\mathbf{ALL})$. For all $\mathbf{F} \in power(\mathbf{ALL})$, let $a\text{-}Coding(\mathbf{F}) = \{a\text{-}Coding(L) \mid L \in \mathbf{F}\}$, where $a\text{-}Coding(L) = \{a\text{-}Coding(x) \mid x \in L\}$.

In the following lemma and observation, we narrow our attention to **CF** and **REG**.

Lemma 4.5. *Let* $L \in \mathbf{CF}$; *then,* $a\text{-}Coding(L) \in \mathbf{REG} \cap power(\{a\}^*)$.

Proof (sketch). Let $L \in \mathbf{CF}$. Let G be a context-free grammar in the Chomsky normal form such that $L(G) = L$. In G, change every production $A \to b$ to $A \to a\text{-}Coding(b)$, where b is a terminal; otherwise, keep G unchanged. Let H be the context-free grammar resulting from this simple change. Clearly, $a\text{-}Coding(L) = a\text{-}Coding(L(G))$ and $a\text{-}Coding(L(G)) \in power(\{a\}^*)$. Hence, $a\text{-}Coding(L(G)) \in \mathbf{REG}$ because every context-free language over $\{a\}$ is regular (see Theorem 6.3.1 on page 194 in [8]). Thus, $a\text{-}Coding(L) \in \mathbf{REG} \cap power(\{a\}^*)$, so this lemma, whose fully rigorous proof is left to the reader, holds true. □

Observation 4.6. $a\text{-}Coding(\mathbf{CF}) = \mathbf{REG} \cap power(\{a\})^*$, *so* $a\text{-}Coding$ *properly reduces* **CF** *onto* $\mathbf{REG} \cap power(\{a\}^*)$.

Proof. Take any $L \in \mathbf{REG} \cap power(\{a\}^*)$. Clearly, $L \in a\text{-}Coding(\mathbf{REG})$, so $L \in a\text{-}Coding(\mathbf{CF})$. Hence, $\mathbf{REG} \cap power(\{a\}^*) \subseteq a\text{-}Coding(\mathbf{CF})$. From Lemma 4.5, $a\text{-}Coding(\mathbf{CF}) \subseteq \mathbf{REG} \cap power(\{a\}^*)$. Thus, Observation 4.6 holds. □

Let $n \in \mathbb{N}$. Recall that ${}_nG$ means that G has n nonterminals (see Section 2). For all $L \in \mathbf{ALL}$, define ${}_n\,\mathrm{GrammaticalDefinition}(L) = L$ if there exists a grammar ${}_nG$ such that $L({}_nG) = L$, and ${}_n\,\mathrm{GrammaticalDefinition}(L) = \emptyset$ otherwise.

For all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define operation

$${}_n\,\mathrm{GrammaticalDefinition}(\mathbf{F}) = \{{}_n\,\mathrm{GrammaticalDefinition}(L) \mid L \in \mathbf{F}\}.$$

Observation 4.7. ${}_2\,\mathrm{GrammaticalDefinition}(\mathbf{ALL}) = \mathbf{RE}$*, so* ${}_2\,\mathrm{GrammaticalDefinition}$ *properly reduces* **ALL** *onto* **RE**, *for* $i = 2, 3, \ldots$.

Proof. To prove $\mathbf{RE} \subseteq {}_2\,\mathrm{GrammaticalDefinition}(\mathbf{ALL})$, take any $L \in \mathbf{RE}$. If $L = \emptyset$, then ${}_2\,\mathrm{GrammaticalDefinition}(L) = \emptyset$, and there obviously exists a grammar ${}_2G$ so $L({}_2G) = \emptyset$. Let $L \in \mathbf{RE}$ and $L \neq \emptyset$. By Church's thesis, there is a grammar ${}_nG$, $L({}_nG) = L$, for some $n \geq 2$. Let ${}_nG = (N, T, P, S), \{0, 1\} \subseteq N$, and $a \in T$. Introduce a homomorphism h from $N \cup T$ into $\{1\}\{a\}^+\{1\}$. Next, we construct a grammar ${}_2H$ so $L({}_nG) = L({}_2H)$. Set ${}_2H = (\{0, 1\}, T, R, 0)$ with

$$\begin{aligned} R = &\{0 \to 111h(S)1111, 1111111 \to \varepsilon\} \\ &\cup \{h(x) \to h(y) \mid x \to y \in P\} \\ &\cup \{111h(b) \to b111 \mid b \in T\}. \end{aligned}$$

A rigorous proof that $L({}_nG) = L({}_2H)$ is simple and left to the reader.

Thus, $\mathbf{RE} \subseteq {}_2\,\mathrm{GrammaticalDefinition}(\mathbf{ALL})$. By Church's thesis, a language is generated by a grammar if and only if it belongs to **RE**, so we definitely have ${}_2\,\mathrm{GrammaticalDefinition}(ALL) \in \mathbf{RE}$. Thus, the observation holds true. □

Can Obseration 4.7 be established for $i = 1$? The answer is no as proved next.

Observation 4.8. ${}_1\,\mathrm{GrammaticalDefinition}(\mathbf{ALL}) \subset \mathbf{RE}$*, so* ${}_1\,\mathrm{GrammaticalDefinition}$ *properly reduces* **ALL** *into* **RE**.

Proof (by Contradiction). Set $L_{prime} = \{a^j \mid j \text{ is a prime}\}$. Of course, $L_{prime} \in \mathbf{RE}$. For the sake of contradiction, assume that a grammar ${}_1G = (\{S\}, \{a\}, P, S)$ satisfies $L_{prime} = L({}_1G)$. Having a single nonterminal, ${}_1G$ would have derivations in the form

$$S \Rightarrow^* uSw \Rightarrow^* uvSw \Rightarrow^* uvvSw \Rightarrow^* \cdots \Rightarrow^* uv^iSw \Rightarrow^* uv^iw,$$

where $u, w \in T^*, |uw| \geq 2, v \in T^+$, so $uv^iw \in L_{prime}$ for all $i \geq 0$. Of course, $uv^iw \in L_{prime}$. Set $|uw| = m$. Take uv^mw. Observe that $|uv^mw| = |uw| + m|v| = m + m|v| = m(1 + |v|)$, so $uv^mw \notin L_{prime}$—a contradiction. Thus, this observation holds. □

For all $\mathbf{F} \in power(\mathbf{ALL})$, define operations

$$Union(\mathbf{F}) = \{L \cup K \mid K, L \in \mathbf{F}\}, \text{ and}$$
$$DifferentUnion(\mathbf{F}) = \{L \cup K \mid K, L \in \mathbf{F}, K \notin L\}.$$

As it is shown next, while the latter represents a reducing operation (see Observation 4.10), the former does not (see Observation 4.11).

Lemma 4.9. *Let* $\mathbf{F} \in power(\mathbf{ALL})$*, and let* $L \in \mathbf{F}$ *satisfy* $card(L) \leq card(K)$*, for all* $K \in \mathbf{F}$*. Then,* $L \notin DifferentUnion(\mathbf{F})$*.*

Proof. Let $L \in \mathbf{F}$ with $card(L) \leq card(K)$, for all $K \in \mathbf{F}$. If $L = \emptyset, \emptyset \cup \emptyset \notin DifferentUnion(\mathbf{F})$, so $\emptyset \notin DifferentUnion(\mathbf{F})$. Let $L \neq \emptyset$, $card(L) \leq card(K)$. By the definition of *DifferentUnion*, $L - \emptyset \notin DifferentUnion(\mathbf{F})$ and $L \cup L \notin DifferentUnion(\mathbf{F})$. Therefore, every $J \in DifferentUnion(\mathbf{F})$ satisfies $card(L) < card(J)$. Thus, $L \notin DifferentUnion(\mathbf{F})$. □

Observation 4.10. *For all* $\mathbf{F} \subseteq \mathbf{ALL}$*, DifferentUnion reduces* $\mathbf{F}$*.*

Proof. Recall that we always assume $\mathbf{F} \neq \emptyset$ (see Section 3) to see that Lemma 4.9 implies this observation. □

4.2 Expanding Operation

Throughout the rest of this section, we discuss mostly operations that unchange or, more often, expand language families (see parts 6 and 7 in Definition 3.1).

Observation 4.11. *Let* $\mathbf{F} \in power(\mathbf{ALL})$. *If there are* $K, L \in \mathbf{F}$ *such that* $L \cup K \notin \mathbf{F}$, *then Union expands* $\mathbf{F}$; *otherwise, Union unchanges* $\mathbf{F}$.

Proof. Let $\mathbf{F} \in power(\mathbf{ALL})$. Notice that $\mathbf{F} = \{J \cup J \mid J \in \mathbf{F}\}$. If there are $K, L \in \mathbf{F}$ such that $L \cup K \notin \mathbf{F}$, then $\mathbf{F} \subset Union(\mathbf{F})$; otherwise, $\mathbf{F} = Union(\mathbf{F})$. □

For all $\mathbf{F} \in power(\mathbf{ALL})$, define $Intersection(\mathbf{F}) = \{L \cap K \mid K, L \in \mathbf{F}\}$.

Observation 4.12. *Let* $\mathbf{F} \in power(\mathbf{ALL})$. *If there are* $K, L \in \mathbf{F}$ *such that* $L \cap K \notin \mathbf{F}$, *then Intersection expands* $\mathbf{F}$; *otherwise, Intersection unchanges* $\mathbf{F}$.

Proof. By analogy with the proof of Observation 4.11. □

Let *Homomorphism* denote the common language operation of homomorphism (*a-Coding*, discussed in Lemma 4.5 and Observation 4.6, represents its special case). For all $\mathbf{F} \in power(\mathbf{ALL})$, define:

$$Homomorphism(\mathbf{F}) = \{Homomorphism(L) \mid L \in \mathbf{F}\}.$$

Observation 4.13. *Homomorphism expands* **CS** *onto* **RE**.

Proof. By Theorem 9.10 in [5], $\mathbf{RE} \subseteq Homomorphism(\mathbf{CS})$. By Church's thesis, $Homomorphism(\mathbf{CS}) \subseteq \mathbf{RE}$. Thus, Observation 4.13 holds true. □

Throughout the rest of this section, we narrow our attention to operations applied only to **CF** or its subfamilies.

Observation 4.14. *Homomorphism unchanges* **CF**.

Proof. $Homomorphism(\mathbf{CF}) \subseteq \mathbf{CF}$ (see Theorem 8.12 in [2]). To prove that $\mathbf{CF} \subseteq Homomorphism(\mathbf{CF})$, take any $L \in CF$. Consider the homomorphism h over $alph(L)^*$ as the identity $h(a) = a$, for all $a \in alph(L)$. Clearly, $h(L) = L$, so $\mathbf{CF} \subseteq Homomorphism(\mathbf{CF})$. Thus, Observation 4.14 holds true. □

Next, we state some specific results concerning *Union* applied to **CF** and its proper subfamily of inherently ambiguous context-free languages, denoted by ${}_{amb}\mathbf{CF}$. Set ${}_{unamb}\mathbf{CF} = \mathbf{CF} - {}_{amb}\mathbf{CF}$.

Observation 4.15. *Union unchanges* **CF**.

Proof. For any $K, L \in \mathbf{CF}, L \cup K \in \mathbf{CF}$, so this observation follows from Obsevation 4.11. □

Observation 4.16. *Union expands* ${}_{unamb}\mathbf{CF}$ *into* **CF**.

Proof. Of course, ${}_{unamb}\mathbf{CF} \subseteq Union({}_{unamb}CF) \subseteq \mathbf{CF}$. Take $L = \{a^n b^n c^m : n, m \geq 1\}$ and $K = \{a^m b^n c^n \mid n, m \geq 1\}$, both of which are in ${}_{unamb}\mathbf{CF}$. Recall that $L \cup K = \{a^i b^j c^k \mid i, j, k \geq 1, i = j \text{ or } j = k\}$ belongs to ${}_{amb}\mathbf{CF}$ (see Example 2.47 on page 205 in [9]). Thus, ${}_{unamb}CF \subset Union({}_{unamb}\mathbf{CF}) \subseteq \mathbf{CF}$, so this observation holds true. □

For every context-free grammar G, set

$$ {}_{CFG}Amb(L(G)) = \{x \in L(G) \mid x \text{ is the frontier of two or more distinct derivation trees for } G\}. $$

Lemma 4.17. *Let* $L \in \mathbf{RE}$. *Then,* $L = {}_{CFG}Amb(L(G))$, *where* G *is a context-free grammar.*

Proof. Let $L \in RE$. Express L as $L = h(L(I) \cap L(J))$, where $L(I)$ and $L(J)$ are deterministic context-free languages, and I, J are unambiguous context-free grammars (see Theorem 10.3.1 on page 310 in [8] and Theorem 6.21 on page 250 in [10]). Let $I = (N_I, T, P_I, S_I)$ and $J = (N_J, T, P_J, S_J)$, $N_I \cap N_J = \emptyset$. Define the homomorphism

g over $(N_I \cup N_J \cup T)^*$ as $g(A) = A$ for every $A \in N_I \cup N_J$ and $g(a) = h(a)$ for every $a \in T$. Construct the context-free grammar $G = (N_I \cup N_J \cup \{X\}, T, P_G, S_G)$, where X is a new nonterminal and

$$P_G = \{A \to g(x) \mid A \to x \in P_I \cup P_J\} \cup \{S_G \to XS_I, S_G \to S_J X, X \to \varepsilon\}.$$

Observe that $L(G) = h(L(I) \cup L(J))$. Recall that I and J are unambiguous. Thus, $(h(L(J)) - h(L(I))) \cap {}_{CFG}Amb(L(G)) = \emptyset$ and $(h(L(I)) - h(L(J))) \cap {}_{CFG}Amb(L(G)) = \emptyset$, so ${}_{CFG}Amb\ (L(G)) \subseteq h(L(I) \cap L(J))$. Notice that $\{S_G \to XS_I, S_G \to S_J X, X \to \varepsilon\} \subseteq P_G$. Thus, $h(L(I) \cap L(J))$ is necessarily contained in ${}_{CFG}Amb(L(G))$, so $h(L(I) \cap L(J)) \subseteq {}_{CFG}Amb(L(G))$. Hence, ${}_{CFG}Amb(L(G)) = h(L(I) \cap L(J)) = L$. Therefore, Lemma 4.17 holds. □

Define language-family operation ${}_{CF}Amb$ as follows. For every $\mathbf{F} \subseteq \mathbf{CF}$:

$${}_{CF}Amb(\mathbf{F}) = \{{}_{CFG}Amb(L(G)) \mid G \text{ is a context-free grammar}\},$$

and for every $\mathbf{F} \subseteq \mathbf{ALL} - \mathbf{CF}$, ${}_{CF}Amb(\mathbf{F}) = \emptyset$.

Observation 4.18. $\mathbf{RE} = {}_{CF}Amb(\mathbf{CF})$*, so* ${}_{CF}Amb$ *expands* **CF** *onto* **RE**.

Proof. By Lemma 4.17, $\mathbf{RE} \subseteq {}_{CF}Amb(\mathbf{CF})$. From Church's thesis, ${}_{CF}Amb(\mathbf{CF}) \subseteq \mathbf{RE}$. Thus, Observation 4.18 holds. □

Next, we introduce *i-Power* as a language-family operation and demonstrate that its application to **LIN** gives rise to an infinite hierarchy of language families.

Let $i \in \mathbb{N}$. For all $\mathbf{F} \subseteq \mathbf{ALL}$, recursively define operation *i-Power*$(\mathbf{F})$ as follows: (i) *1-Power*$(\mathbf{F}) = \mathbf{F}$, (ii) for all $i \geq 1$, $(i+1)$*-Power*$(\mathbf{F}) = \{LK \mid L \in$ *i-Power*$(\mathbf{F}), K \in \mathbf{F}\}$.

Theorem 4.19. *For all* $i \geq 1$, *i-Power*$(\mathbf{LIN}) \subset i+1$*-Power*$(\mathbf{LIN})$*; in words,* $i+1$*-Power properly expands i-Power*$(\mathbf{LIN})$.

Proof. Let $j \in \mathbb{N}$. Observe that j-*Power*(**LIN**) $= {}_j\mathbf{LIN}$ (see Section 2 for ${}_j\mathbf{LIN}$); a proof of this observation is simple and left to the reader. Recall that for all $i \geq 1, {}_i\mathbf{LIN} \subset {}_{i+1}\mathbf{LIN}$ (see Theorem 8.3 in [5]). Thus, Theorem 4.19 holds true. □

In what follows, without any loss of generality, we assume that # represents a special delimited marker exclusively used as described in the following definitions of operations *Middle* and *SymmetricMiddle*.

For all $L \in \mathbf{ALL}$, define

$$Middle(L) = \{w \mid x\#w\#y \in L, x, w, y \in (alph(L) - \#)^*\}, \text{ and}$$

$$SymmetricMiddle(L) =$$

$$= \{w \mid x\#w\#y \in L, x, w, y \in (alph(L) - \#)^*, x = reversal(y)\}.$$

For all $\mathbf{F} \subseteq \mathbf{ALL}$, define

$$Middle(\mathbf{F}) = \{Middle(L) \mid L \in F\}, \text{ and}$$

$$SymmetricMiddle(\mathbf{F}) = \{SymmetricMiddle(L) \mid L \in \mathbf{F}\}.$$

At a glance, *Middle* and *SymmetricMiddle* resemble each other very much. However, while *Middle* unchanges ${}_j\mathbf{LIN}$ for any $j \geq 1$, *SymmetricMiddle* expands **LIN** onto **RE**.

Theorem 4.20. *Let $j \in \mathbb{N}$. $Middle({}_j\mathbf{LIN}) = {}_j\mathbf{LIN}$, so Middle unchanges ${}_j\mathbf{LIN}$.*

Proof. Let $j \in \mathbb{N}$. To prove $Middle({}_j\mathbf{LIN}) \subseteq {}_j\mathbf{LIN}$, take any $L \in Middle({}_j\mathbf{LIN})$. That is, $L = Middle(L(G))$, where $G = (N, T, P, S)$ is a j-linear grammar. Assume that in every rule of the form $S \to x \in P, x \in N^j$, and every nonterminal is terminating—that is, there is a derivation of a terminal word starting from it. A simple proof that any j-linear grammar can be turned to a j-linear grammar satisfying this assumption is simple and left to the reader.

Next, we construct a j-linear grammar H so $Middle(L(G)) = L(H)$. Set $H = (M, T, R, \langle S \rangle)$, whose components are constructed as follows. Set

$$M = \{\langle aAb\rangle \mid A \in N, a, b \in \{[,], \$, \varepsilon\}, ab \in \{\varepsilon, \$\$, \$], [\$, []\}\},$$

where [,], and \$ are new symbols not contained in $N \cup T$. Construct R by performing 1 through 8, given next. In this construction, we automatically assume that $u, v, w, x, y, z \in T^*$, and $A, B \in N - \{S\}$. Initially, set $R = \emptyset$. Perform

1. for all $S \to A_1 \dots A_h \dots A_j \in P$, where $h \in \{1, \dots, j\}$, add $\langle S\rangle \to \langle [A_h]\rangle$ to R;

2. for all $S \to A_1 \dots A_h A_{h+1} \dots A_{i-1} A_i \dots A_j \in P$, where $h \in \{1, \dots, j-1\}, i \in \{h+1, \dots, j\}$, add $\langle S\rangle \to$ $\langle [A_h\$\rangle\langle \$A_{h+1}\$\rangle \dots\dots \langle \$A_{i-1}\$\rangle\langle \$A_i]\rangle$ to R;

3. for all $A \to uBv \in P$, add $\langle A\rangle \to \langle B\rangle$ and $\langle \$A\$\rangle \to u\langle \$B\$\rangle v$ to R;

4. for all $A \to uBv\#y \in P$, add $\langle \$A]\rangle \to u\langle \$B\$v\rangle$ to R;

5. for all $A \to u\#vBy \in P$, add $\langle [A\$\rangle \to v\langle \$B\$\rangle y$ to R;

6. for all $A \to u\#vBx\#y \in P$, add $\langle [A]\rangle \to v\langle \$B\$\rangle x$ to R;

7. for all rules of the form $A \to vBu\#x\#y, A \to u\#x\#vBy$, and $A \to \#x\#P$, add $\langle [A]\rangle \to x$ to R;

8. for all rules of the form $A \to w \in P$, add $\langle \$A\$\rangle \to w$ to R.

Gist. *In essence, H uses [and] as boundary markers that delimit the corresponding subword w occurring in between the two #s generated in G. In this way, H determines every $w \in Middle(L(G))$ and generates it, so $L(H) = Middle(L(G))$. For instance, suppose that G makes*

$$
\begin{aligned}
S \Rightarrow & A_1 \dots A_h \dots A_j \\
\Rightarrow^* & u_1 A_h u_2 \\
\Rightarrow & u_1 u_2 \# u_3 A_2 u_4 u_2 \\
\Rightarrow^* & u_1 u_2 \# u_3 u_4 A_3 u_5 u_4 u_2 \\
\Rightarrow & u_1 u_2 \# u_3 u_4 u_6 A_4 u_7 \# u_8 u_5 u_4 u_2 \\
\Rightarrow^* & u_1 u_2 \# u_3 u_4 u_6 u_9 A_5 u_{10} u_7 \# u_8 u_5 u_4 u_2 \\
\Rightarrow & u_1 u_2 \# u_3 u_4 u_6 u_9 u_{11} u_{10} u_7 \# u_8 u_5 u_4 u_2 .
\end{aligned}
$$

As a result, $u_3 u_4 u_6 u_9 u_{11} u_{10} u_7 \in Middle(\mathbf{LIN})$. *Then, H simulates the generation of the string* $u_3 u_4 u_6 u_9 u_{11} u_{10} u_7$ *in this way*

$$
\begin{aligned}
\langle S \rangle \Rightarrow^* & \langle [A_h] \rangle \\
\Rightarrow & u_3 \langle \$ A_2] \rangle \\
\Rightarrow^* & u_3 u_4 \langle \$ A_3] \rangle \\
\Rightarrow & u_3 u_4 u_6 \langle \$ A_4 \$ \rangle u_7 \\
\Rightarrow^* & u_3 u_4 u_6 u_9 \langle \$ A_4 \$ \rangle u_{10} u_7 \\
\Rightarrow & u_3 u_4 u_6 u_9 u_{11} u_{10} u_7 .
\end{aligned}
$$

Consider all other possible forms of generating $x\#w\#y \in L(G)$ *such that* $w \in Middle(L(G))$. *H simulates them by analogy with the simulation sketched above, so* $Middle(L(G)) \subseteq L(H)$. *Similarly, we can establish* $L(H) \subseteq Middle(L(G))$, *so* $Middle(L(G)) = L(H)$. *A fully rigorous proof of this identity is simple, but lengthy and tedious, so we omit it; the reader can easily fill in all the details.*

Thus, this theorem holds true. □

Considering Theorem 4.20, we find it surprising that *SymmetricMiddle* properly expands **LIN** onto **RE** (see Theorem 4.28). Since proving this result is more complicated than the previous proof, we provide it in a greater detail. To start with, we need the notion of a queue grammar.

A *queue grammar* (see [1]) is a six tuple, $Q = (V, T, W, F, s, P)$, where V and W are alphabets satisfying $V \cap W = \emptyset, T \subseteq V, F \subseteq W, s \in (V - T)(W - F)$, and $P \subseteq (V \times (W - F)) \times (V^* \times W)$ is a finite relation such that for every $a \in V$, there exists an element $(a, b, x, c) \in P$. If $u, v \in V^*W$ such that $u = arb; v = rzc; a \in V; r, z \in V^*; b, c \in W$; and $(a, b, z, c) \in P$, then $u \Rightarrow v[(a, b, z, c)]$ in G or, simply, $u \Rightarrow v$. In the usual manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The language of Q, $L(Q)$, is defined as $L(Q) = \{w \in T^* : s \Rightarrow^* wf, \text{ where } f \in F\}$. Now, we slightly modify the notion of a queue grammar. A *left-extended queue grammar* is a sixtuple , $Q = (V, T, W, F, s, P)$, where V, T, W, F, and s have the same meaning as in a queue grammar. $P \subseteq (V \times (W - F)) \times (V \times W)$ is a finite relation (as opposed to an ordinary queue grammar, this definition does not require that for every $a \in V$, there exists an element $(a, b, x, c) \in P$). Furthermore, assume that $\# \notin V \cup W$. If $u, v \in V^*\{\#\}V^*W$ so that $u = w\#arb; v = wa\#rzc$; $a \in V; r, z, w \in V^*$; $b, c \in W$; and $(a, b, x, c) \in P$, then $u \to v[(a, b, z, c)]$ in G or, simply, $u \Rightarrow v$. In the usual manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The language of $Q, L(Q)$, is defined as $L(Q) = \{v \in T^* : \#s \Rightarrow^* w\#vf \text{ for some } w \in V^* \text{ and } f \in F\}$. Less formally, during every step of a derivation, a left-extended queue grammar shifts the rewritten symbol over #; in this way, it records the derivation history, which plays a crucial role in the proof of Lemma 4.22.

Lemma 4.21. *For every recursively enumerable language, L, there exists a left-extended queue grammar, Q, satisfying $L(Q) = L$.*

Proof. Recall that every recursively enumerable language is generated by a queue grammar (see [1], [3]). Clearly, for every queue grammar, there exists an equivalent left-extended queue grammar. Thus, this lemma holds. □

Lemma 4.22. *Let H be a left-extended queue grammar. Then, there exists a left-extended queue grammar, $Q = (V, T, W, F, s, R)$, such that $L(H) = L(Q)$ and every $(a, b, x, c) \in R$ satisfies $a \in V - T, b \in W - F$, and $x \in ((V - T)^* \cup T^*)$.*

Proof. Let $H = (\varsigma, T, \Omega, \phi, \sigma, \Pi)$ be any left-extended queue grammar. Set $\Omega' = \{q' : q' \in \Omega\}$, $\Omega'' = \{q'' : q'' \in \Omega\}$, and $\varsigma' = \{a' : a \in \varsigma\}$. Define the bijection α from Ω to Ω' as $\alpha(q) = q'$ for every $q \in \Omega$. Analogously, define the bijection β from Ω to Ω'' as $\beta(q) = q''$ for every $q \in \Omega$. Finally, define the bijection δ from ς to ς' as $\delta(a) = a'$ for every $a \in \varsigma$. In the standard manner, extend δ so it is defined from ς^* to $(\varsigma')^*$. Set

$$U = \{\langle y, p\rangle : y \in T^*, p \in \Omega, \text{ and } (a, q, xy, p) \in \Pi$$

$$\text{for some a } \in \varsigma, q \in \Omega, x \in \varsigma^*\}.$$

Without any loss of generality, assume that $(\delta(\varsigma) \cup T \cup \alpha(\Omega) \cup \beta(\Omega) \cup U) \cap \{1, f\} = \emptyset$. Set $V = \delta(\varsigma) \cup \{1\} \cup T, W = \alpha(\Omega) \cup \beta(\Omega) \cup \{f\} \cup U, F = \{f\}$, and $s = \delta(a)\alpha(q)$. Define the left-extended queue grammar

$$Q = (V, T, W, F, s, R)$$

with R constructed in the following way:

I if $(a, q, xy, p) \in \Pi$, where $a \in \varsigma$; $q \in \Omega - \Phi; x, y \in \varsigma^*$; and $p \in \Omega$, then add $(\delta(a), \alpha(q), \delta(x)\delta(y), \alpha(p))$ and $(\delta(a), \alpha(q), \delta(x)1\delta(y), \alpha(p))$ to R;

II if $(a, q, xy, p) \in \Pi$, where $a \in \varsigma, q \in \Omega - \Phi, x \in \varsigma^*, y \in T^*, p \in \Omega(\langle y, p\rangle \in U)$, then add $(\delta(a), \alpha(q), \delta(x), \langle y, p\rangle)$ and $(1, \langle y, p\rangle, y, \beta(p))$ to R;

III if $(a, q, x, p) \in \Pi$, where $a \in \varsigma, q \in \Omega - \Phi, x \in T^*$, and $p \in \Omega$, then add $(\delta(a), \beta(q), \delta(x), \beta(p))$ to R;

IV if $(a, q, x, p) \in \Pi$, where $a \in \varsigma$, $q \in \Omega - \Phi, x \in T^*$, and $p \in \Phi$, then add $(\delta(a), \beta(q), x, f)$ to R (recall that $F = \{f\}$).

Clearly, for every $(a, b, x, c) \in R$, $a \in V - T, b \in W - F$, and $x \in ((V - T)^* \cup T^*)$. Leaving a rigorous proof that $L(H) = L(Q)$ to the reader, we next give its sketch.

To see that $L(H) \subseteq L(Q)$, consider any $v \in L(H)$. As $v \in L(H)$,

$$\#\sigma \Rightarrow^* w\#vt$$

in $H, w \in \varsigma^*$, $v \in T^*$, and $t \in \Phi$. Express $\#\sigma \Rightarrow^* w\#vt$ in H as

$$\#\sigma \Rightarrow^* u\#zq \Rightarrow ua\#xyp \Rightarrow^* w\#vt,$$

where $a \in \varsigma, u, x \in \varsigma^*$, $y = \mathit{prefix}(v, |y|)$, $z = ax, w = uax$, and during $ua\#xyp \Rightarrow^* w\#vt$, only terminals are generated so that the resulting terminal string equals v. Q simulates $\#\sigma \Rightarrow^* u\#zq \Rightarrow ua\#xyp \rightarrow *w\#vt$ as follows. First, Q uses productions introduced in I to simulate $\#\sigma \Rightarrow^* u\#zq$. During this initial simulation, it once uses a production that generates 1 so that it can then simulate $u\#zq \Rightarrow ua\#xyp$ by making two derivation steps according to productions $(\Delta(a), \alpha(q), \Delta(x), \langle y, p\rangle)$ and $(1, \langle y, p\rangle, y, \beta(p))$ (see II). Notice that by using $(1, \langle y, p\rangle, y, \beta(p))$, Q produces y, which is a prefix of v. After the application of $(1, \langle y, p\rangle, y, \beta(p))$, Q simulates $ua\#xyp \rightarrow *w\#vt$ by using productions introduced in III followed by one application of a production constructed in IV, during which Q enters f and, thereby, completes the generation of v. Thus, $L(H) \subseteq L(Q)$.

To establish $L(Q) \subseteq L(H)$, consider any $v \in L(Q)$. Since $v \in L(Q)$,

$$\#s \Rightarrow^* w\#vf$$

in Q, where $w \in V^*$ and $v \in T^*$. Examine I through IV. Observe that Q passes through states of $\alpha(W), U, \beta(W)$, and $\{f\}$ in this order so that it occurs several times in states of $\alpha(W)$, once in a state of U, several times in $\beta(W)$, and once in f. As a result, Q uses productions introduced in I, and during this initial part of derivation it precisely once uses a production that generates 1 so that it can subsequently make two consecutive derivation steps according to $(\delta(a), \alpha(q), \delta(x), \langle y, p\rangle)$ and $(1, \langle y, p\rangle, y, \beta(p))$ (see II). By using the latter, Q produces y, which is a prefix of v. After the application of $(1, \langle y, p\rangle, y, \beta(p))$, Q applies productions introduced in III, which always use states of $\beta(\Omega)$. Finally, it once applies a production constructed in IV to enter f and, thereby, complete the generation of v. To summarize these observations, we can express $\#s \Rightarrow^* w\#vf$ in Q as

$$\#s \Rightarrow^* u\#zq \Rightarrow ua\#xyp \Rightarrow^* w\#vf,$$

where $a \in V, x \in V^*, y \in T^*, w = uax$ so that during $\#s \Rightarrow^* u\#zq$, Q uses productions introduced in I, then it applies $(1, \langle y, p\rangle, y, \beta(p))$ from II to make $u\#zq \Rightarrow ua\#xyp$, and finally, it performs $ua\#xyp \Rightarrow^* w\#vf$ by several applications of productions introduced in III and one application of a production constructed in IV. At this point, based on an examination of I through IV, we see that H makes

$$\#i\sigma \Rightarrow^* u\#zq \Rightarrow ua\#xyp \Rightarrow^* w\#vt$$

with $t \in \Phi$, so $v \in L(H)$. Therefore, $L(H) \subseteq L(Q)$.

As $L(H) \subseteq L(Q)$ and $L(Q) \subseteq L(H)$, $L(H) = L(Q)$. □

Lemma 4.23. *Let Q be a left-extended queue grammar. Then, there exists a linear grammar, $G = (N, T, P, S)$, such that $L(Q) = SymmetricMiddle(L(G))$.*

Proof. Let $Q = (V, T, W, F, s, R)$ be a left-extended queue grammar. Without any loss of generality, assume that Q satisfies the properties described in Lemma 4.22 and that $\{0, 1, \} \cap (V \cup W) = \emptyset$. For some positive integer n, define an injection ι from VW to $(\{0,1\}^n - 1^n)$ so that ι is an injective homomorphism when its domain is extended to $(VW)^*$; after this extension, ι thus represents an injective homomorphism from $(VW)^*$ to $(\{0,1\}^n - 1^n)^*$ (a proof that such an injection necessarily exists is simple and left to the reader). Based on ι, define the substitution ν from V to $(\{0,1\}^n - 1^n)$ as $\nu(a) = \{\iota(aq) : q \in W\}$ for every $a \in V$. Extend the domain of ν to V^*. Furthermore, define the substitution μ from W to $(\{0,1\}^n - 1^n)$ as $\mu(q) = \{reversal(\iota(aq)) : a \in V\}$ for every $q \in W$. Extend the domain of μ to W^*. Set $U = \{\langle p, i\rangle : p \in W - F \text{ and } i \in \{1, 2\}\} \cup \{S\}$.

Construction. *Introduce the linear grammar $G = (U, T \cup \{0, 1, \#\}, P, S)$ with P constructed in the following way. Initially, set $P = \emptyset$. To construct P, perform the following steps 1 through 5.*

1 *if* $a_0q_0 = s$, *where* $a \in V - T$ *and* $q \in W - F$, *then add* $S \to u\langle q, 1\rangle v$ *to* P, *for all* $u \in \nu(a_0)$ *and* $v \in \mu(q_0)$;

2 *if* $(a, q, y, p) \in R$, *where* $a \in V - T, p, q \in W - F$, *and* $y \in (V - T)^*$, *then add* $\langle q, 1\rangle \to u\langle p, 1\rangle v$ *to* P, *for all* $u \in \nu(y)$ *and* $v \in \mu(p)$;

3 *for every* $q \in W - F$, *add* $\langle q, 1\rangle \to \#\langle q, 2\rangle$ *to* P;

4 *if* $(a, q, y, p) \in R$, *where* $a \in V - T, p, q \in W - F, y \in T^*$, *then add* $\langle q, 2\rangle \to y\langle p, 2\rangle v$ *to* P, *for all* $v \in \mu(p)$;

5 *if* $(a, q, y, p) \in R$, *where* $a \in V - T, q \in W - F, y \in T^*$, *and* $p \in F$, *then add* $\langle q, 2\rangle \to y\#$.

Basic Idea. *G can generate every* $y \in L(G)$ *as* $S \Rightarrow^* u_0\#y\#v_0$, *where* $u_0 \in \nu(a_0a_m)$ *with* $a_0, \ldots, a_m \in T, u_i \in$ *suffix*$(n(a_0 \ldots a_m), |\nu(a_0 \ldots a_m)| - i)$ *for* $i = 1, \ldots, m - 1$, $v_0 \in \nu(q_mq_0)$ *with* $q_0, \ldots, q_m \in Q, v_j \in$ *prefix*$(\mu(q_m \ldots q_0), |\mu(q_m \ldots q_0))| - j)$ *for* $j = 1, \ldots, m - 1$, $u_0 = reversal(v_0)$. *Examine the construction of* P *to see that* $S \Rightarrow^* u_0\#y\#v_0$ *in* G *if and only if* Q *makes* $a_0q_0 \Rightarrow^* a_0 \ldots a_myf$ *according to* (a_0, q_0, z_0, q_1) *through* (a_m, q_m, z_m, q_{m+1}), *where* $q_{m+1} \in F$. *From this equivalence,* $L(Q) = SymmetricMiddle(L(G))$.

Formal Proof. *For brevity, the following rigorous proof omits some obvious details, which the reader can easily fill in. Claim 4.24, proved next, establishes a derivation form by which G can generate each member of* $L(G)$. *This claim fulfills a crucial role in the demonstration that* $SymmetricMiddle(L(G)) \subseteq L(Q)$, *given later in this proof (see Claim 4.26).*

Claim 4.24. *G can generate every $h \in L(G)$ in this way*

$$
\begin{aligned}
& S \\
\Rightarrow & X\langle q_0, 1\rangle t_0 \Rightarrow g_0\langle q_1, 1\rangle t_1 \Rightarrow \cdots \Rightarrow g_{k-1}\langle q_k, 1\rangle t_k \\
\Rightarrow & g_k\langle q_{k+1}, 1\rangle t_{k+1} \Rightarrow g_k\#\langle q_{k+1}, 2\rangle t_{k+1} \\
\Rightarrow & g_k\#y_1\langle q_{k+2}, 2\rangle t_{k+2} \Rightarrow g_k\#y_1y_2\langle q_{k+3}, 2\rangle t_{k+3} \Rightarrow \ldots \\
\Rightarrow^* & g_k\#y_1y_2\ldots y_{m-1}\langle q_{k+m}, 2\rangle t_{k+m} \\
\Rightarrow & g_k\#y_1y_2\ldots y_{m-1}y_m\#t_{k+m}
\end{aligned}
$$

in G, where $k, m \geq 1; q_0, q_1, \ldots, q_{k+m}W - F; y_1, \ldots, y_m \in T^*; X \in \nu(a_0)$, where $a_0 \in (V - T)$ and $s = a_0q_0$; $t_i \in \mu(q_i \ldots q_1q_0)$ for $i = 0, 1, \ldots, k+m; g_j \in \nu(d_0d_1 \ldots d_j)$ with $d_0 = a_0$ and $d_1, \ldots, d_j \in (V-T)^*$ for $j = 0, 1, \ldots, k$; $d_0d_1 \ldots d_k = a_0a_1 \ldots a_{k+m}$ with $a_1, \ldots, a_{k+m} \in V - T$ (that is, $g_k \in \nu(a_0a_1 \ldots a_{k+m})$); $g_k = reversal(t_{k+m})$; $h = y_1y_2 \ldots y_{m-1}y_m$.

Proof of Claim 4.24. Examine the construction of P. Observe that every derivation begins with an application of a production having S on its left-hand side. Set 1-$U = \{\langle p, 1\rangle : p \in W\}$, 2-$U = \{\langle p, 2\rangle : p \in W\}$, 1-$P = \{p : p \in P$ and $lhs(p) \in$ 1-$U\}$, and 2-$P = \{p : p \in P$ and $lhs(p) \in$ 2-$U\}$. Observe that in every successful derivation, all applications of productions from 1-P precede the applications of productions from 2-P. Furthermore, notice that

$$
\begin{aligned}
F(G) - \{S\} \subseteq & \{\#, \varepsilon\}\{0, 1\}^*(1\text{-}U)\{0, 1\}^*\{\#, \varepsilon\} \\
& \cup \{\#, \varepsilon\}\{0, 1\}^*T^*(2\text{-}U)\{0, 1\}^*\{\#, \varepsilon\}.
\end{aligned}
$$

Thus, we can always express the derivation so that the generation of

$h \in L(G)$ can be expressed as

$$
\begin{aligned}
&S \\
&\Rightarrow X\langle q_0, 1\rangle t_0 \Rightarrow g_0\langle q_1, 1\rangle t_1 \Rightarrow \cdots \Rightarrow g_{k-1}\langle q_k, 1\rangle t_k \\
&\Rightarrow g_k\langle q_{k+1}, 1\rangle t_{k+1} \Rightarrow g_k\#\langle q_{k+1}, 2\rangle t_{k+1} \\
&\Rightarrow g_k\#y_1\langle q_{k+2}, 2\rangle t_{k+2} \Rightarrow g_k\#y_1y_2\langle q_{k+3}, 2\rangle t_{k+3} \\
&\Rightarrow \cdots \Rightarrow g_k\#y_1y_2\ldots y_{m-1}\langle q_{k+m}, 2\rangle t_{k+m} \\
&\Rightarrow g_k\#y_1y_2y_{m-1}y_m\#t_{k+m},
\end{aligned}
$$

where all the involved symbols have the meaning described in Claim 4.24. During the first $|g_k|$ steps, every sentential form has the form $\gamma\#y_1y_2y_{m-1}y_m\$m\#\tau$ with $\gamma, \tau \in \{0,1\}^*, 0 \leq |\gamma| = |\tau| \geq |g_k|$, and $\gamma = reversal(\tau)$. Thus, $g_k = reversal(t_{k+m})$; $h = y_1y_2y_{m-1}y_m$. As a result, Claim 4.24 holds. □

Claim 4.25. *Q generates every $h \in L(Q)$ in this way*

$$
\begin{aligned}
&\#a_0q_0 && \\
&\Rightarrow a_0\#x_0q_1 && [(a0, q0, z0, q1)] \\
&\Rightarrow a_0a_1\#x_1q_2 && [(a_1, q_1, z_1, q_2)] \\
&\ldots && \\
&\Rightarrow a_0a_1\ldots a_k\#x_kq_{k+1} && [(a_k, q_k, z_k, q_{k+1})] \\
&\Rightarrow a_0a_1\ldots a_ka_{k+1}\#x_{k+1}y_1q_{k+2} && [(a_{k+1}, q_{k+1}, y_1, q_{k+2})] \\
&\ldots && \\
&\Rightarrow a_0a_1\ldots a_ka_{k+1}\ldots a_{k+m-1}\#x_{k+m-1}y_1\cdots y_{m-1}q_{k+m} && [(a_{k+m-1}, q_{k+m-1}, y_{m-1}, q_{k+m})] \\
&\Rightarrow a_0a_1\ldots a_ka_{k+1}\ldots a_{k+m}\#y_1\ldots y_mq_{k+m+1} && [(a_{k+m}, q_{k+m}, y_m, q_{k+m+1})],
\end{aligned}
$$

where $k, m \geq 1; a_i \in V - T$ for $i = 0, \ldots, k+m; x_j \in (V-T)^*$ for $j = 1, \ldots, k+m; s = a_0q_0$; $a_jx_j = x_{j-1}z_j$ for $j = 1, \ldots, k; a_1\ldots a_kx_{k+1} = z_0\ldots z_k$; $a_{k+1}\ldots a_{k+m} = x_k; q_0, q_1, \ldots, q_{k+m} \in W - F$; $q_{k+m+1} \in F, z_1, \ldots, z_k \in (V-T)^*; y_1, \ldots, y_m \in T^*$; $h = y_1y_2\ldots y_{m-1}y_m$.

Proof of claim 4.25. Recall that Q satisfies the properties given in Lemma 4.22. These properties imply that Claim 4.25 holds. □

Claim 4.26. *Let G generate $h \in L(G)$ in the way described in Claim 4.24; then, $h \in L(Q)$.*

Proof of Claim 4.26. Let $h \in L(G)$. Take the generation of h as described in Claim 4.24. Taking this into consideration, examine the construction of P to see that, R contains $(a_0, q_0, z_0, q_1), \ldots, (a_k, q_k, z_k, q_{k+1})$, $(a_{k+1}, q_{k+1}, y_1, q_{k+2}), \ldots, (a_{k+m-1}, q_{k+m-1}, y_{m-1}, q_{k+m})$, $(a_{k+m}, q_{k+m}, y_m, q_{k+m+1})$, where $z_1, \ldots, z_k \in (V - T)^*$, and $y_1, \ldots, y_m \in T^*$. Then, Q makes the generation of h in the way described in Claim 4.25. Thus, $h \in L(Q)$. □

Claim 4.27. *Let Q generates $h \in L(Q)$ in the way described in Claim 4.25; then, $h \in L(G)$.*

Proof of Claim 4.27. This is left to the reader. □

Claims 4.24 through 4.27 imply that $L(Q) = L(G)$, so this lemma holds.

□

Theorem 4.28. **RE** = *SymmetricMiddle*(**LIN**)*, so SymmetricMiddle properly expands* **LIN** *onto* **RE**.

Proof. From Lemmas 4.21,4.22, and 4.23,
RE $\subseteq$ *SymmetricMiddle*(**LIN**). From Church's thesis,
SymmetricMiddle(**LIN**) $\subseteq$ **RE**. Thus, Theorem 4.28 holds. □

Let $L \in$ **ALL** with $card(alph(L)) \geq 2$. For all $L \in$ **ALL** and $F \subseteq$ **ALL**, define $BinarySymmetricMiddle(L) = \{w \mid x\#w\#y \in L, \# \notin alph(w), x, w, y \in (alph(L) - \{\#\})^*, x = reversal(y), card(alph(\{x, y\})) = 2\}$. For all $F \subseteq$ **ALL**, define $BinarySymmetricMiddle(\mathbf{F}) = \{BinarySymmetricMiddle(L) \mid L \in \mathbf{F}\}$.

Corollary 4.29. **RE** = *BinarySymmetricMiddle*(**LIN**)*, so BinarySymmetricMiddle expands* **LIN** *onto* **RE**.

Proof. This corollary follows from the demonstration of Theorem 4.28. The details are left to the reader. □

5 Concluding Remarks

This section closes the study by suggesting and illustrating five new investigation trends concerning the subject of this paper.

1 We can simplify some proofs above if we restrict our attention only to special cases of the results that are demonstrated. To illustrate, reconsider Theorem 4.20 and its proof. Next, we rephrase the result just for **LIN** and prove it in a simpler way than the proof of Theorem 4.20. Indeed, observe that in the following proof, the construction of H is shorter. The resulting H is also more succinct and economical with respect to the number of nonterminals.

Theorem 5.1. *Middle*(**LIN**) = **LIN**, *so Middle unchanges* **LIN**.

Proof. Let $j \in \mathbb{N}$. To prove $Middle(\mathbf{LIN}) \subseteq \mathbf{LIN}$, take any $L \in Middle(\mathbf{LIN})$. That is, $L = Middle(L(G))$, where $G = (N, T, P, S)$ is a linear grammar. Next, we construct a linear grammar H so $Middle(L(G)) = L(H)$. Set $H = (M, T, R, \langle S \rangle)$, whose components are constructed as follows. Set

$$M = \{\langle aAb \rangle : A \in N, a, b \in \{\#, \varepsilon\}\}$$

Initially, set $R = \emptyset$. Construct R by performing (1) through (5), given next, where $u, v, w, x, y, z \in T^*$, and $A, B \in N$.

1. for all $A \to uBv \in P$, add $\langle A \rangle \to \langle B \rangle$, $\langle \#A\# \rangle \to u\langle \#B\# \rangle v$, $\langle A\# \rangle \to \langle B\# \rangle v$, $\langle \#A \rangle \to u\langle \#B \rangle$, $\langle \#\#A \rangle \to \langle \#\#B \rangle$, and $\langle A\#\# \rangle \to \langle B\#\# \rangle$ to R;

2. for all $A \to uBv\#y \in P$, add $\langle A \rangle \to \langle B\# \rangle v$, $\langle \#A \rangle \to u\langle \#B\# \rangle v$, and $\langle A\# \rangle \to \langle B\#\# \rangle y$ to R;

3. for all $A \to u\#vBy \in P$, add $\langle A \rangle \to v\langle \#B \rangle$, $\langle A\# \rangle \to v\langle \#B\# \rangle y$, and $\langle \#A \rangle \to \langle \#\#B \rangle u$ to R;

4. for all $A \to u\#vBx\#y \in P$, add $\langle A \rangle \to v\langle \#B\# \rangle x$ to R;

5. for all $A \to uBv\#x\#y \in P$, add $\langle A \rangle \to \langle B\#\# \rangle x$ to R;

6. for all $A \to u\#v\#xBy \in P$, add $\langle A\rangle \to v\langle\#\#B\rangle x$ to R;
7. for all $A \to w \in P$, add $\langle\#A\#\rangle \to w$, $\langle\#\#A\rangle \to \epsilon$, and $\langle\#\#A\rangle \to \epsilon$ to R.

Basic Idea. *Suppose that G makes*

$$\begin{aligned} S &\Rightarrow^* u_1A_1u_2 \\ &\Rightarrow u_1u_2\#u_3A_2u_4u_2 \\ &\Rightarrow_* u_1u_2\#u_3u_4A_3u_5u_4u_2 \\ &\Rightarrow u_1u_2\#u_3u_4u_6A_4u_7\#u_8u_5u_4u_2 \\ &\Rightarrow^* u_1u_2\#u_3u_4u_6u_9A_5u_{10}u_7\#u_8u_5u_4u_2 \\ &\Rightarrow u_1u_2\#u_3u_4u_6u_9u_{11}u_{10}u_7\#u_8u_5u_4u_2, \end{aligned}$$

where As and us are nonterminals and terminal strings, respectively. As a result, $u_3u_4u_6u_9u_{11}u_{10}u_7 \in Middle(\mathbf{LIN})$. Then, H simulates the generation of $u_3u_4u_6u_9u_{11}u_{10}u_7$ in this way

$$\begin{aligned} \langle S\rangle &\Rightarrow^* \langle A_1\rangle \\ &\Rightarrow u_3\langle\#A_2\rangle \\ &\Rightarrow^* u_3u_4\langle\#A_3\rangle \\ &\Rightarrow u_3u_4u_6\langle\#A_4\#\rangle u_7 \\ &\Rightarrow^* u_3u_4u_6u_9\langle\#A_4\#\rangle u_{10}u_7 \\ &\Rightarrow u_3u_4u_6u_9u_{11}u_{10}u_7. \end{aligned}$$

Consider all the other possible generations of $x\#w\#y \in L(G)$ such that $w \in Middle(L(G))$. H simulates these generations by analogy with the simulation sketched above, so $Middle(L(G)) \subseteq L(H)$. Similarly, we can establish $L(H) \subseteq Middle(L(G))$. Thus, $Middle(L(G)) = L(H)$. A fully rigorous proof of this identity is simple, but lengthy and tedious, so we omit it because the reader can easily fill in all the details.

Thus, $Middle(\mathbf{LIN}) = \mathbf{LIN}$. □

2 Most classical books about formal languages contain many results concerning closure properties. Reconsider and reformulate them in terms of the notions introduced in the present paper. For instance, for all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define operation

$$\mathit{Iteration}(\mathbf{F}) = \{L^* \mid L \in \mathbf{F}\}.$$

Take $\mathit{Iteration}(\mathbf{REG})$. Clearly, $\{\varepsilon\}^* = \emptyset^* = \{\varepsilon\}$. For any $L \in \mathbf{REG} - \{\emptyset, \{\varepsilon\}\}, L^* \in {}_{inf}\mathbf{REG}$, where ${}_{inf}\mathbf{REG}$ denotes the family of infinite regular languages. As obvious, ${}_{inf}\mathbf{REG} - \mathit{Iteration}(\mathbf{REG}) \neq \emptyset$; in words, there exist (infinitely many) regular languages K satisfying $K \neq L^*$, for all $L \in \mathbf{REG}$. For instance, $K = \{a\} \cup \{b\}^*$ is a regular language that does not represent the iteration of any regular language. Thus, in terms of Definition 3.1, *Iteration* properly reduces **REG** into ${}_{inf}\mathbf{REG} \cup \{\varepsilon\}$.

3 In this paper, we restricted our attention to unary and binary language operations, and we only considered unary language-family operations. Drop this restriction. Study n-ary language operation as well as n-ary language-family operation in general, for any $n \geq 1$. To illustrate, define binary language-family operation *Intersection* from $(\mathit{power}(\mathbf{ALL}))^2$ into $\mathit{power}(\mathbf{ALL})$ so for all $\mathbf{E}, \mathbf{F} \in \mathit{power}(\mathbf{ALL})$,

$$\mathit{Intersection}(\mathbf{E}, \mathbf{F}) = \{K \cap L \mid K \in \mathbf{E}, L \in \mathbf{F}\}.$$

Set $\mathbf{UNARY} = \{L \mid L \in \mathbf{ALL}, \mathit{card}(\mathit{alph}(L))\} = 1\}$. Recall that $\mathbf{UNARY} \cap \mathbf{CF} \subseteq \mathbf{REG}$ (see Theorem 6.3.1 on page 194 in [8]). Thus, $\mathit{Intersection}(\mathbf{UNARY}, \mathbf{CF}) = \mathit{Intersection}(\mathbf{UNARY}, \mathbf{REG})$, so $\mathit{Intersection}(\mathbf{UNARY}, \mathbf{CF}) \subseteq \mathbf{REG}$ and $\mathit{Intersection}(\mathbf{UNARY}, \mathbf{CF} - \mathbf{REG}) = \emptyset$. Of course, further investigation in this direction would necessitate a proper generalization of Definition 3.1, restricted to unary language-family operations in this paper.

4 Apart from operations over language families, we can study operations over other mathematical notions, including notions used in formal language theory. To illustrate a study of this kind by an example closely related to the subject of the present paper, consider families of grammars. Let **GRAMMARS** denote the family of all grammars,

and let ${}_n\mathbf{GRAMMARS}$ denote the family of all n-nonterminal grammars, where $n \in \mathbb{N}$ (see Section 2). Is there a total function f from **GRAMMARS** into ${}_n\mathbf{GRAMMARS}$ so $L(f(G)) = L(G)$ for every $G \in \mathbf{GRAMMARS}$? If so, what is the smallest $n \in \mathbb{N}$ for which such a function exists? Reconsider the proofs of Theorems 4.7 and 4.8 to see that we can always find such a function f from **GRAMMARS** into ${}_n\mathbf{GRAMMARS}$, where $n = 2$ is the smallest number. That is, for $n = 1$, no function like this exists.

5 Apart from a theoretical viewpoint, results concerning operations over language families are important from a practical standpoint, too. For instance, take multilingual translators that contain parsers, whose techniques are restricted to a language family **F**. If prior to parsing, the translators modify some languages in **F** so they are expelled from this family, these techniques cannot parse them; consequently, any possible expulsion like this has to be ruled out. On the other hand, assume $\mathbf{F} \subset \mathbf{E}$. If the translators can reduce **E** into **F** so this reduction makes the parsing techniques applicable, then the reduction is obviously highly desirable in practice. To illustrate this practical standpoint even more specifically in terms of multi-natural-language translation, take **F** as the languages offically used in the EU states, and consider **E** as the same family extended by other languages used in these states together with their major dialects. For instance, apart from French as the official language, many French people speak other languages, such as a broad variety of Gallo-Romance languages, including several Oïl and Occitan languages. As obvious, from this viewpoint, the reduction sketched above together with its application-related advantages have its practical importance.

Acknowledgements

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602; the TAČR grant TE01020415; and the BUT grant FIT-S-17-3964. I also thank Jakub Martiško for his help concerning this paper.

References

[1] H. C. M. Kleijn and G. Rozenberg, "On the Generative Power of Regular Pattern Grammars," *Acta Informatica*, vol. 20, no. 4, pp. 391–411, 1983.

[2] A. Meduna, *Formal Languages and Computation*, London: CRC Press, 2014.

[3] *Handbook of Formal Languages, Volumes 1 through 3*, G. Rozenberg and A. Salomaa, Eds. Berlin: Springer, 1997.

[4] J. G. Pace, *Mathematics of Discrete Structures for Computer Science*, London: Springer, 2012.

[5] A. Salomaa, *Formal Languages*, London: Academic Press, 1973.

[6] J. Martin, *Intro to Lan and the The of Com, 3rd Edition*, Boston: McGraw-Hill, 2003.

[7] M. Sipster, *Introduction to the Theory of Computation,* 3rd ed., Boston: Cengage Learning, 2012.

[8] M.A. Harrison, *Introduction to Formal Language Theory*, Berkeley: Addison-Wesley, 1978.

[9] A.V. Aho and J.D. Ullman, *The theory of parsing, translation, and compiling, Volume I*, Upper Saddle River, NJ: Prentice-Hall, Inc., 1972.

[10] J.E. Hopcroft, R. Motwani, and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed., Boston: Pearson, 2006.

A. Meduna[1], R. Krčmář[2], A. Kövári[3], Z. Beníčková[4] Received October 31, 2019

Faculty of Information Technology, Brno University of Technology
Brno, Czech Republic
E–mails: [1]meduna@fit.vutbr.cz, [2]ikrcmar@fit.vutbr.cz, [3]ikovari@fit.vutbr.cz, [4]ibenickova@fit.vut.cz

Computer Science Journal of Moldova, vol.**27**, no.3(81), 2019

On Anonymization of Cocks' Identity-based Encryption Scheme

Anca-Maria Nica and Ferucio Laurenţiu Ţiplea

Abstract

Cocks' identity-based encryption (IBE) scheme is the first IBE scheme that avoids the use of bilinear maps. Based on quadratic residues and due to its simplicity, the scheme gained much attention from researchers. Unfortunately, the scheme is not anonymous in the sense that the cryptotexts may reveal the identities for which they have been computed. Several anonymous variants of it have then been proposed.

In this paper we revise Joye's approach to the anonymization of the Cocks' IBE scheme. Due to some recent results on the distribution of quadratic residues, we present a very simple and direct approach that leads to Joye's scheme.

Keywords: identity-based encryption, quadratic residue, indistinguishability

1 Introduction

In 1984 Adi Shamir proposed the idea of identity-based encryption [14], which is a special case of public-key encryption. This model avoids the public-key infrastructure and the trust chain for public keys. It uses instead a string which uniquely identifies the receiver and computes his public key based on it, using a publicly known hash function.

In his paper [14], Shamir showed how one can sign using the identity-based paradigm but, regarding IBE, only seventeen years later the first solutions were proposed. Thus, in 2001, Boneh and Franklin [5] designed an IBE scheme relying on bilinear maps. In the same year, Cocks [8] created a pairing-free IBE scheme using quadratic residues.

Despite its cryptotext expansion, Cocks' scheme was a starting point for several quadratic residues (QR) based IBE schemes by virtue of its simplicity and elegance [17].

It was shown in [4] that Cocks' scheme does not provide anonymity of the receiver's public key in the sense of Bellare et al. [3]. From this point on, several ideas and schemes have been proposed in order to obtain anonymous variants. The first one was due to Di Crescenzo and Saraswat [9] in 2007, but it is quite impractical because it uses four keys per bit of plaintext. This is also the first public key encryption scheme with keyword search (PEKS) which is not based on bilinear maps.

Starting from the idea in [8] but using quadratic residues in a lighter and deeper way, Boneh, Gentry, and Hamburg [6] proposed an IBE scheme which hides a message not in an integer but in a Jacobi symbol. Moreover, the scheme is anonymous (please see also [16], [17] for supplementary discussion on this very important scheme).

Two years later, Ateniese and Gasti proposed an interesting solution to the anonymization of Cocks' scheme, that reaches universal anonymity [1]. This means that the anonymization process is independent from encryption and can be done using only the public key of the receiver. Moreover their variant is also parallelizable, as well as Cocks' IBE scheme. Unfortunately, Ateniese and Gasti's scheme has much larger cryptotexts than Cocks' scheme.

A variant of [1] was considered in 2010 by Aouinatou and Belkasmi. It has better performances and complexity that the original one [1] while keeping the security in [8].

In 2014, Clear, Tewari, and McGoldrick [7] designed a variant of Cocks' IBE scheme which, beside the fact that it is universally anonymous, it keeps the time complexity close to the initial scheme. Thus, it is faster than the scheme in [1] (as it is shown in Table 1 in [7]) and, further more, outputs shorter cryptotexts than Cocks' scheme.

In the same year, another scheme starting from [1] and providing universal anonymity was proposed by Schipor [13]. Compared to the universal anonymous variant of Ateniese and Gasti [1], the ciphertext expansion of Schipor's solution is considerably smaller and the scheme

is faster.

Joye's [11] solution to the anonymization of Cocks' scheme is probably the most elegant and efficient one. Joye showed that Cocks ciphertexts are squares in a torus-like algebraic structure, and form a quasi-group. From this, he obtained an anonymous variant of Cocks' scheme.

In this paper we re-consider Cocks' scheme, we discuss Galbraith's test in a very precise way based on the recent results in [18], and we present Joye's scheme for Cocks anonymization in a very direct and simply way. We appreciate that this makes more understandable the anonymization process behind Joye's solution.

2 Preliminaries

We recall now the basic notation and terminology that is to be used in the paper.

The set of integers is denoted by $\mathbb{Z}$. If $n, a, b \in \mathbb{Z}$, then a and b are called *congruent modulo* n, denoted $a \equiv b \bmod n$ or $a \equiv_n b$, if n divides $a - b$. The remainder of the integer division of a by n, assuming $n \neq 0$, is denoted $(a)_n$, respectively. Positive integers $n = pq$ that are product of two distinct primes p and q will be usually called *RSA integers* or *RSA moduli*.

Given a positive integer n, $\mathbb{Z}_n$ stands for the set of remainders modulo n, and $\mathbb{Z}_n^*$ is the subset of integers in $\mathbb{Z}_n$ that are co-prime to n. An integer a co-prime with n is a *quadratic residue modulo* n if $a \equiv_n x^2$, for some integer x; the integer x is called a *square root* of a modulo n.

Let p be a prime. The *Legendre symbol* of an integer a modulo p, denoted $\left(\frac{a}{p}\right)$, is 1 if a is a quadratic residue modulo p, 0 if p divides a, and -1 otherwise. The *Jacobi symbol* extends the Legendre symbol to composite moduli. If $n = p_1^{e_1} \cdots p_m^{e_m}$ is the prime factorization of the positive integer n, then the Jacobi symbol of a modulo n is

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \cdots \left(\frac{a}{p_m}\right)^{e_m}.$$

For the sake of simplicity we will use the terminology of Jacobi symbol

in both cases (prime or composite moduli). For details regarding basic properties of the Jacobi symbol the reader is referred to [12], [15].

Given a positive integer n and a subset $A \subseteq \mathbb{Z}_n^*$, $QR_n(A)$ ($QNR_n(A)$, $J_n^+(A)$, $J_n^-(A)$) stands for the set of quadratic residues (quadratic non-residues, integers with the Jacobi symbol 1, integers with the Jacobi symbol -1, respectively) modulo n from A. When $A = \mathbb{Z}_n^*$, the notation will be simplified to QR_n (QNR_n, J_n^+, J_n^-, respectively). When n is a prime, $QR_n(A) = J_n^+(A)$ and $QNR_n(A) = J_n^-(A)$.

For a set A, $a \leftarrow A$ means that a is uniformly at random chosen from A. If $\mathcal{A}$ is a probabilistic algorithm, then $a \leftarrow \mathcal{A}$ means that a is an output of $\mathcal{A}$ for some given input.

The asymptotic approach to security makes use of security parameters, denoted by λ in our paper. A positive function $f(\lambda)$ is called *negligible* if for any positive polynomial $poly(\lambda)$ there exists n_0 such that $f(\lambda) < 1/poly(\lambda)$, for any $\lambda \geq n_0$.

Let $RSAgen(\lambda)$ be a probabilistic polynomial time algorithm that, given a security parameter λ, outputs a triple (n, p, q), where $n = pq$ is an RSA modulus. The *quadratic residuosity* (QR) *assumption* holds for $RSAgen(\lambda)$ if the distance

$$|P(\mathcal{D}(a, n) = 1 \ : \ (n, p, q) \leftarrow RSAgen(\lambda),\ a \leftarrow QR_n) -$$

$$P(\mathcal{D}(a, n) = 1 \ : \ (n, p, q) \leftarrow RSAgen(\lambda),\ a \leftarrow J_n \setminus QR_n)|,$$

as a function of λ, is negligible for all probabilistic polynomial-time algorithms $\mathcal{D}$.

An IBE scheme consists of four probabilistic polynomial time (PPT) algorithms, Setup, KeyGen, Enc, and Dec. The Setup(λ) algorithm outputs the public parameters PP together with the master secret msk, having as input the security parameter λ. The KeyGen(PP, msk, ID) algorithm outputs the secret key for the identity ID. The third algorithm, Enc(PP, m), computes the ciphertext of the message m for a given identity, while the last algorithm, Dec(sk_{ID}, c), decrypts the cryptotext c using the secret key sk_{ID} of the identity ID.

Regarding the security, an IBE scheme $\mathcal{S}$ is said to be ANON-IND-ID-CPA secure if the advantage of any efficient PPT adversary

$\mathcal{A}$ against the scheme $\mathcal{S}$ is negligible when the following game is considered:

Setup: the challenger $\mathcal{C}$ gives the public parameters outputted by the SETUP algorithm to the adversary.

Phase 1: the adversary queries the challenger for the secret keys corresponding to the identities that $\mathcal{A}$ chooses. In response, $\mathcal{C}$ runs the KEYGEN algorithm and sends $\mathcal{A}$ the secret keys. The adversary also can issue decryption queries where he sends a pair (ID_i, c_i) and receives the corresponding plaintext decrypted by the DEC algorithm.

Challenge: in this phase $\mathcal{A}$ sends two challenge identities ID_0^* and ID_1^* to $\mathcal{C}$, different from all the identities used in the previous phase, and two equal-length messages m_0 and m_1. The challenger chooses at random a bit $b \in \{0, 1\}$ and encrypts the pair (ID_b^*, m_b). Then he sends the ciphertext to $\mathcal{A}$.

Phase 2: the adversary is allowed to issue the same types of queries as in *Phase 1*, with the restriction that the identities he chooses have to be different from ID_0 and ID_1 which were used in the *Challenge* phase.

Guess: the adversary tries to guess the pair that was encrypted by the challenger in the *Challenge* phase. Thus it sends a bit b' to the challenger.

The advantage of the adversary in the above game is computed as

$$Adv_{\mathcal{A},\mathcal{S}}(\lambda) = \left| P[b = b'] - \frac{1}{2} \right|,$$

where λ is the security parameter.

3 Cocks' IBE scheme and anonymization

The first pairing-free IBE scheme was proposed by Cocks and it is based on quadratic residues [8]. The scheme is IND-ID-CPA secure in the random oracle model (ROM) under the Quadratic Residuosity Assumption

(QRA) modulo a large RSA integer. Cocks' IBE scheme is defined by four probabilistic algorithms, as it is described in Algorithm 1 on page 289.

It was mentioned in [4, Section 4] that the scheme is not anonymous in the sense that the outputted cryptotexts contain information about the receiver so one can check if the ciphertext was encrypted for a specific identity. The tool which helps to decide this is Galbraith's test (GT), which was briefly described in [1], [4].

In order to understand Galbraith's test, we will turn our attention to *Cocks ciphertexts*, and mainly follow the approach in [18].

Working in an RSA group $\mathbb{Z}_n^*$, we will consider an integer a, which has the Jacobi symbol modulo n equal to 1 (a corresponds to some identity ID, but, for simplicity, we will call a the identity). Then, a cryptotext computed for the identity a has the form $c = t + at^{-1} \bmod n$, for some $t \in \mathbb{Z}_n^*$. That is, t is a solution to the degree two congruence $t^2 - ct + a \equiv 0 \bmod n$. It is useful then to denote [18]:

$$\begin{aligned} C_n(a) &= \{(t + at^{-1})_n \mid t \in \mathbb{Z}_n^*\}, \\ C_n^*(a) &= C_n(a) \cap \mathbb{Z}_n^*, \\ C_n^{e_1,e_2}(a) &= \left\{c \in \mathbb{Z}_p^* \mid \left(\tfrac{c^2-4a}{p}\right) = e_1,\ \left(\tfrac{c^2-4a}{q}\right) = e_2\right\}, \end{aligned}$$

where n is the product of two primes p and q, and $e_1, e_2 \in \{-1, 0, 1\}$.

We recall below some results in [18]. First of all, we clearly have:

Corollary 1 ([18]). *Let n be an RSA modulus and $a \in \mathbb{Z}_n^*$. Then the set $C_n^*(a)$ is a union of the sets $C_n^{0,0}, C_n^{1,0}, C_n^{0,1}$ and $C_n^{1,1}$.*

This union is pictorially described by Figure 1. This diagram gives us a clear understanding of the output given by Galbraith's test.

Notation 3.1 ([18]). *Let p be an odd prime, $p \bmod 4 = i$, $i \in \{1, 3\}$ and a an integer co-prime with p, then we define:*

$$\tau_{p,a}^i = \begin{cases} 1, & \text{if } (p)_4 = i \text{ and } (a)_p \in QR_p \\ 0, & \text{otherwise} \end{cases}$$

Algorithm 1 Cocks' IBE scheme

procedure SETUP(λ):
  $(p, q, n) \leftarrow RSA_{gen}(\lambda)$, where $n = pq$;
  $e \leftarrow J_n^+ \setminus QR_n$;
  choose a hash function $h : \{0,1\}^* \to J_n^+$;
  PP: n, e, h; ▷ the public parameters
  msk: p, q; ▷ the master secret key
  return (PP, msk).
end procedure

procedure KEYGEN(msk, ID):
  $a = h(ID)$; ▷ the identity
  if $a \in QR_n$ **then**
    $r \leftarrow a^{1/2}$
  else $r \leftarrow (ea)^{1/2}$
  end if;
  return r. ▷ the secret key of the identity ID
end procedure

procedure ENC(PP, ID, m): ▷ $m \in \{\pm 1\}$
  $a = h(ID)$;
  $t_1, t_2 \leftarrow \mathbb{Z}_n^*$ such that $(\frac{t_1}{n}) = (\frac{t_2}{n}) = (\frac{m}{n})$;
  $c_1 = t_1 + at_1^{-1} \bmod n$ and $c_2 = t_2 + eat_2^{-1} \bmod n$;
  return (c_1, c_2).
end procedure

procedure DEC($PP, r, (c_1, c_2)$):
  if $r^2 \equiv h(ID) \bmod n$ **then** $c = c_1$
  else $c = c_2$
  end if;
  $m = \left(\frac{c+2r}{n}\right)$;
  return m.
end procedure

and

$$\bar{\tau}^i_{p,a} = \begin{cases} 1, & \text{if } (p)_4 = i \text{ and } (a)_p \in QNR_p \\ 0, & \text{otherwise,} \end{cases}$$

where $i = 1, 3$.

Corollary 2 ([18]). *Let* p, q *be two odd primes,* $n = pq$ *an RSA modulus,* $a \in \mathbb{Z}_n^*$, $k_1 = p \text{ div } 4$, *and* $k_2 = q \text{ div } 4$. *Then,*

1. $|C_n^*(a)| = |C_p^*((a)_p)| \cdot |C_q^*((a)_q)|$;
2. $|C_n^{0,0}(a)| = 4(\tau^1_{p,a} + \tau^3_{p,a})(\tau^1_{q,a} + \tau^3_{q,a})$;
3. $|C_n^{0,1}(a)| = 4(\tau^1_{p,a} + \tau^3_{p,a})(k_2 - \tau^1_{q,a})$;
4. $|C_n^{1,0}(a)| = 4(\tau^1_{q,a} + \tau^3_{q,a})(k_1 - \tau^1_{p,a})$;
5. $|C_n^{1,1}(a)| = 4|QR_n(a + QR_n)| = 4(k_1 - \tau^1_{p,a})(k_2 - \tau^1_{q,a})$;
6. $|C_n(a)| = |C_p((a)_p)| \cdot |C_q((a)_q)|$.

Now we get the following useful result which will help us to compute a probability in order to analyze Galbraith's test.

Theorem 1 ([18]). *Let* n *be an RSA modulus, the product of odd primes* p *and* q, *and* $a \in \mathbb{Z}_n^*$. *Then the set* $G_n(a)$ *is partitioned by the sets* $C_n^{1,1}(a)$ *and* $C_n^{-1,-1}(a)$, *and its cardinal is:* $4|QR_n(a + J_n^+)|$.

The Galbraith's test of $c \in \mathbb{Z}_n^*$ w.r.t. a, denoted $GT_{n,a}(c)$, is

$$GT_{n,a}(c) = \left(\frac{c^2 - 4a}{n}\right).$$

Let $G_n(a) = \{c \in \mathbb{Z}_n^* \mid GT_{n,a}(c) = 1\}$. Clearly,

$$G_n(a) = C_n^{1,1}(a) \cup C_n^{-1,-1}(a).$$

An integer $c \in \mathbb{Z}_n^*$ passes Galbraith's test w.r.t. n and a if $GT_{n,a}(c) = 1$ (or, $c \in G_n(a)$). Not all Cocks ciphertexts pass Galbraith's test, but most of them do. The diagram in Figure 1 shows clearly which Cocks ciphertexts pass Galbraith's test.

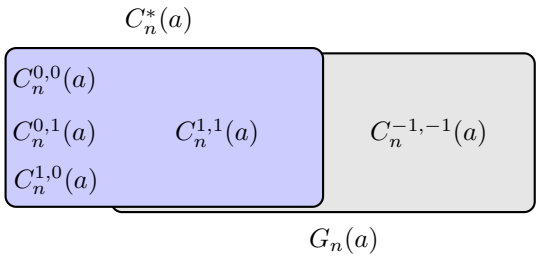

Figure 1. The sets C_n^* and $G_n(a)$

Galbraith's test for anonymity is then described in Algorithm 2.

Algorithm 2 Galbraith's Test

`Input` : RSA modulus n, $a \in J_n^+$, and $c \in \mathbb{Z}_n^*$
`Output` : 1 / 0
if $\left(\frac{c^2-4a}{n}\right) = 1$ **then**
  1 ▷ $c \in C_n^*(a)$ `with prob.` $\frac{1}{2} - \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$
else
  0 ▷ $c \in C_n^*(a)$ `with negl. prob.`
end if

We note that if $\left(\frac{c^2-4a}{n}\right) = -1$, then $c \notin C_n^*(a)$ (with probability 1). However, if $\left(\frac{c^2-4a}{n}\right) = 1$, then c is a Cocks ciphertext with probability overwhelming closed to 1/2. More precise results are offered in [18] by the following probability:
$P(c \in C_n^*(a) : c \leftarrow G_n(a)) = \frac{|C_n^{1,1}(a)|}{|G_n(a)|} = \frac{4|QR_n(a+QR_n)|}{4|QR_n(a+J_n^+)|} = \frac{1}{2} - \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$.

The *key-privacy test* consists of the repeated application of Galbraith's test with values sampled from either $G_n(a)$ (and $G_n(ea)$) or $G_n(b)$ (and $G_n(eb)$), where e is as in Cocks' IBE scheme.

4 Another view on Joye's scheme

Through his work, Joye [11] showed that Cocks' scheme is homomorphic. He analyzed its cryptotexts and defined their precise algebraic structure using a cyclotomic polynomial and an algebraic torus. He defined a special multiplicative group with a certain operation, then he described how it works for a prime modulus in order to use these results for defining it in the case of an RSA modulus. Finally he considered the operation above over Cocks' ciphertexts, using the discriminant $\delta = c^2 - 4a$ of the equation $t^2 - ct + a$ in the unknown t, where $c, a \in \mathbb{Z}_n^*$, and more specifically, he considered c as a Cocks ciphertext and a an identity in the same context.

If we look again to the set of Cocks ciphertexts, whose main part consists of $C_n^{1,1}(a)$, we may imagine the following very simple method to anonymize them: given $c \in C_n^{1,1}(a)$, modify it into c' on a random basis such that $GT_{n,a}(c') = \pm 1$. As we have to decrypt c', the ciphertext c must be altered in such a way so that the receiver be able to reverse it. From this point of view, the simplest method seems to choose and fix d such that $GT_{n,a}(d) = -1$, and then to look for a binary operation $\circ$ on $\mathbb{Z}_n^*$ such that

$$GT_{n,a}(c \circ d) = GT_{n,a}(c) \cdot GT_{n,a}(d).$$

If such a binary operation is found, then we may take $c' = c \circ d$ (but, once again, we flip a coin to decide whether we keep c or compute c').

Under these circumstances we define the operation

$$u \circ v = \frac{uv + 4a}{u + v} \bmod n,$$

for all $u, v \in \mathbb{Z}_n^*$ with $(u + v, n) = 1$.

Although this operation depends on n and a, for the sake of simplicity, we will simply denote it by $\circ$. Its basic properties are listed below.

Proposition 1. *Let $u, v, w \in \mathbb{Z}_n^*$ and $a \in J_n^+$. Then :*

1. *When defined,* $\circ$ *is associative*
$$u \circ (v \circ w) = (u \circ v) \circ w.$$
2. *If* $(u+v, n) = 1$ *and* $(v^2 - 4a, n) = 1$*, then*
$$(u \circ v) \circ (-v) = u$$
(*remark that* $v \circ (-v)$ *is not defined*).
3. $GT_{n,a}(u \circ v) = GT_{n,a}(u) \cdot GT_{n,a}(v)$*, provided that* $u \circ v$ *is defined.*
4. $u \circ u \in G_n(a)$.

Proof. In order to prove (1) we simply apply the operation $\circ$ and do the computation
$$u \circ \frac{vw + 4a}{v + w} = \frac{uv + 4a}{u + v} \circ w.$$
Getting the same result on both sides, we are done
$$\frac{uvw + 4a(u + v + w)}{uv + uw + vw + 4a}.$$
The second property follows from the simple computation below
$$\begin{aligned}
(u \circ v) \circ (-v) &= \frac{\frac{uv+4a}{u+v} \cdot (-v) + 4a}{\frac{uv+4a}{u+v} + (-v)} \\
&= \frac{(u(-v^2) + 4a(u+v-v)) \cdot (u+v)}{(u+v) \cdot (uv + 4a - uv - v^2)} \\
&= \frac{u(4a - v^2)}{4a - v^2} \\
&= u.
\end{aligned}$$
For (3) we have:
$$\begin{aligned}
GT_{n,a}(u \circ v) &= \left(\frac{(u^2 - 4a)(v^2 - 4a)(u+v)^{-2}}{n} \right) \\
&= \left(\frac{u^2 - 4a}{n} \right) \left(\frac{v^2 - 4a}{n} \right) \\
&= GT_{n,a}(u) \cdot GT_{n,a}(v).
\end{aligned}$$
Given that $GT_{n,a}(u \circ u) = GT_{n,a}(u) \cdot GT_{n,a}(u)$ from (3), it immediately follows that $GT_{n,a}(u \circ u)$ is 1, so u is in $G_n(a)$. □

Proposition 1(3) says that $u \circ v$ passes Galbraith's test if and only if both u and v pass Galbraith's test or both do not pass Galbraith's test (provided that $u \circ v$ is defined), and Proposition 1(4) says that $u \circ u$ always passes Galbraith's test.

Fortunately, this is all we need to obtain Joye's approach to the anonymization of Cocks' IBE scheme, as we can see in Algorithm 3 on page 295.

The correctness of the scheme in Algorithm 3 simply follows from Proposition 1. As with respect to security, we have the following result.

Theorem 2. *Cocks' AnonIBE scheme is ANON-IND-ID-CPA secure in the random oracle model under the QRA.*

Proof. Any adversary $\mathcal{A}$ against Cocks' AnonIBE scheme can be transformed into an adversary $\mathcal{A}'$ against Cocks' IBE scheme, with an advantage greater than or equal to the advantage of $\mathcal{A}$. Therefore, as Cocks' IBE scheme is IND-ID-CPA, Cocks' AnonIBE scheme must be.

To prove that Cocks' AnonIBE scheme is anonymous, consider the sets

$$\mathcal{D}_0 = G_n(a_0) \cup (G_n(a_0) \circ d)$$

and

$$\mathcal{D}_1 = G_n(a_1) \cup (G_n(a_1) \circ d),$$

where a_0 and a_1 are two public keys and d is as in Cocks' AnonIBE scheme. We prove that $c \in \mathcal{D}_0$ iff $c \in \mathcal{D}_1$.

Let $c \in \mathcal{D}_0$. Assume first that, if $GT_{n,a_1}(c) = 1$, then $c \in \mathcal{D}_1$. Otherwise, $GT_{n,a_1}(c \circ (-d)) = 1$, which shows that $c \circ (-d) \in G_n(a_1)$. But then, $(c \circ (-d)) \circ d \in G_n(a_1) \circ d$, which leads to $c \in G_n(a_1) \circ d \subseteq \mathcal{D}_1$.

Assume now that $GT_{n,a_0}(c) = -1$. Then, $GT_{n,a_0}(c \circ (-d)) = 1$, which proves that $c \circ (-d) \in G_n(a_0) \subseteq \mathcal{D}_0$. The above argument proves then $c \circ (-d) \in \mathcal{D}_1$, from which it follows that $c \in \mathcal{D}_1$.

As a conclusion, if $c \in \mathcal{D}_0$, then $c \in \mathcal{D}_1$. Due to the symmetry of this relation, we get the theorem. □

Algorithm 3 Cocks' AnonIBE scheme

procedure SETUP(λ):
$PP = (n, e, d, h)$
▷ where n and e are as in Cocks' IBE scheme
$d \leftarrow \mathbb{Z}_n^*$ and $h : \{0,1\}^* \to J_n^+$ are chosen so that $GT_{n,a}(d) = -1 = GT_{n,ea}(d)$, for any output a of h
$msk = (p, q)$
return (PP, msk).
end procedure

procedure EXT(msk, ID):
$a = h(ID)$;
▷ private key: random square root r of a or ea
return r.
end procedure

procedure ENC(PP, ID, m):
$a = h(ID)$;
$t_0, t_1 \leftarrow \mathbb{Z}_n^*$ with $J_n(t_0) = m = J_n(t_1)$;
$c_0 \leftarrow \{u, u \circ d\}$ where $u = t_0 + at_0^{-1} \bmod n$;
$c_1 \leftarrow \{v, v \circ d\}$ where $v = t_1 + eat_1^{-1} \bmod n$;
return (c_0, c_1).
end procedure

procedure DEC($(c_0, c_1), r$):
set $b \in \{0,1\}$ such that $e^b a \equiv_n r^2 \bmod n$;
return $m = \begin{cases} J_n(c_b + 2r), & \text{if } GT_{n,e^b a}(c_b) = 1 \\ J_n(c_b \circ (-d) + 2r), & \text{otherwise} \end{cases}$
end procedure

5 Conclusion

In this paper we achieved the anonymization in Joye's work [11] in a significantly easier way. The main resource used in order to attain this was the new results on quadratic residues in [18]. This extensive study facilitates a better understanding of the cryptotexts outputted by Cocks' IBE scheme and their structure. Furthermore, [18] facilitates a detailed analysis of Galbraith's test, capturing the essence of the anonymization process regarding Cocks' IBE cryptotexts. This allowed us to bring clarity and capture the essence in Joye's variant.

References

[1] Giuseppe Ateniese and Paolo Gasti, "Universally anonymous IBE based on the quadratic residuosity assumption," in *CT-RSA 2009* (Lecture Notes in Computer Science, vol. 5473), Springer, 2009, pp. 32–47.

[2] Rkia Aouinatou and Mostafa Belkasmi, "Efficient anonymity for Cocks' scheme," in *Proceedings of 5th International Symposium on I/V Communications and Mobile Networks, ISIVC 2010*, 2010, pp. 1–4. Preprint on IACR Cryptology ePrint Archive. Report 2011/684, 2016. https://eprint.iacr.org/2011/684.

[3] Mihir Bellare, Dennis Hofheinz, and Scott Yilek, "Possibility and impossibility results for encryption and commitment secure under selective opening," in *Advances in Cryptology - EUROCRYPT 2009, Proceedings of 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, (Cologne, Germany, April 26-30, 2009), 2009, pp. 1–35.

[4] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano, "Public key encryption with keyword search," in *EUROCRYPT 2004* (Lecture Notes in Computer Science, vol. 3027), Springer, 2004, pp. 506–522.

[5] Dan Boneh and Matthew K. Franklin, "Identity-based encryption from the Weil pairing," in *CRYPTO 2001*, (Lecture Notes in Computer Science, vol. 2139), Springer, 2001, pp. 213–229.

[6] Dan Boneh, Craig Gentry, and Michael Hamburg, "Space-efficient identity based encryption without pairings," in *FOCS 2007, IEEE Computer Society*, 2007, pp. 647–657.

[7] Michael Clear, Hitesh Tewari, and Ciarán McGoldrick, "Anonymous IBE from quadratic residuosity with improved performance," in *AFRICACRYPT 2014* (Lecture Notes in Computer Science, vol. 8469), Springer, 2014, pp. 377–397.

[8] Clifford Cocks, "An identity based encryption scheme based on quadratic residues," in *IMACC 2001* (Lecture Notes in Computer Science, vol. 2260), Springer, 2001, pp. 360–363.

[9] Giovanni Di Crescenzo and Vishal Saraswat, "Public key encryption with searchable keywords based on Jacobi symbols," in *Progress in Cryptology – INDOCRYPT 2007, Proceedings of 8th International Conference on Cryptology in India*, (Chennai, India, December 9-13, 2007), 2007, pp. 282–296.

[10] Ryotaro Hayashi and Keisuke Tanaka, "Universally anonymizable public-key encryption," in *Proceedings of the 11th international conference on Theory and Application of Cryptology and Information Security, ASIACRYPT'05*, Berlin, Heidelberg: Springer-Verlag, Dec 2005, pp. 293–312.

[11] Marc Joye, "Identity-based cryptosystems and quadratic residuosity," in *PKC 2016* (Lecture Notes in Computer Science, vol. 9614), Springer, 2016, pp. 225–254.

[12] Melvyn B. Nathanson, *Elementary Methods in Number Theory*, New York: Springer, 2000.

[13] Gheorghe A. Schipor, "On the anonymization of Cocks IBE scheme," in *Cryptography and Information Security in the Balkans*

- *First International Conference, BalkanCryptSec 2014, Revised Selected Papers*, (Istanbul, Turkey, October 16-17, 2014), 2014, pp. 194–202.

[14] Adi Shamir, "Identity-based cryptosystems and signature schemes," in *CRYPTO 1984* (Lecture Notes in Computer Science, vol. 196), Springer, 1985, pp. 47–53.

[15] Victor Shoup, *A Computational Introduction to Number Theory and Algebra*, 2nd edition, New York, NY, USA: Cambridge University Press, 2009.

[16] George Teşeleanu, Ferucio Laurenţiu Ţiplea, Sorin Iftene, and Anca-Maria Nica, "Boneh-Gentry-Hamburg's identity-based encryption schemes revisited," Preprint on IACR Cryptology ePrint Archive. Report 2016/516, 2016. https://eprint.iacr.org/2016/516.

[17] Ferucio Laurenţiu Ţiplea, Sorin Iftene, George Teşeleanu, and Anca-Maria Nica, "Security of identity-based encryption schemes from quadratic residues," in *Innovative Security Solutions for Information Technology and Communications - 9th International Conference, SECITC 2016, Revised Selected Papers*, (Bucharest, Romania, June 9-10, 2016), 2016, pp. 63–77.

[18] Ferucio Laurenţiu Ţiplea, Sorin Iftene, George Teşeleanu, and Anca-Maria Nica, "On the distribution of quadratic residues and non-residues modulo composite integers and applications to cryptography," Preprint on IACR Cryptology ePrint Archive. Report 2019/638, https://eprint.iacr.org/2019/638.pdf, 2019. Submitted to *Applied Mathematics and Computation*, 2019.

Anca-Maria Nica[1], Ferucio Laurenţiu Ţiplea[1] Received September 20, 2019

[1]Department of Computer Science, "Al.I.Cuza" University of Iaşi
E–mail: contact@ancamarianica.ro, ferucio.tiplea@uaic.ro

Computer Science Journal of Moldova, vol.**27**, no.3(81), 2019

Post-quantum commutative encryption algorithm

A.A. Moldovyan, D.N. Moldovyan, N.A. Moldovyan

Abstract

To provide possibility to design the commutative encryption algorithms on the basis of new versions of the hidden discrete logarithm problem, the term "commutativity" is interpreted in the extended sense. Namely, the encryption algorithm is called commutative, if the double encryption on two different keys produces the ciphertext that can be correctly decrypted using the keys in arbitrary order. The introduced commutative encryption method is characterized in using the single-use random subkeys. This feature defines probabilistic nature of the encryption process. A candidate for post-quantum commutative encryption algorithm is proposed, using the computations in the 6-dimensional finite non-commutative associative algebra with a large set of the right-sided global units. The proposed algorithm is used as the base of the post-quantum no-key protocol.

Keywords: commutative encryption, post-quantum cryptoscheme, no-key protocol, finite non-commutative algebra, associative algebra, homomorphism

MSC 2000: 94A60, 16Z05, 14G50, 11T71, 16S50.

1 Introduction

The most widely used public-key cryptoschemes are based on the computational difficulty of the factoring problem (FP) and the discrete logarithm problem (DLP). However, both the FP and the DLP can be solved in polynomial time on a quantum computer [1]. This means that cryptographic schemes based on the FP and on the DLP will not

be secure in the coming era of quantum computing [2]. Future practice needs post-quantum public-key cryptoschemes. A cryptographic algorithm or protocol is called post-quantum, if it runs efficiently on classical computers but will resist quantum attacks [3],[4], i. e., attacks performed with using hypothetic quantum computers.

Currently, post-quantum cryptographic research efforts are mainly focused on developing practical public-key cryptoschemes based on the computationally difficult problems different from the FP and DLP. In the frame of the competition announced by NIST (in the end of 2016) on development of the post-quantum public-key cryptoschemes suitable as candidates for new cryptographic standards several dozen of signature schemes, public encryption algorithms, and public key-agreement protocols have been selected for further research [5].

The commutative encryption algorithms possessing security to the known-plaintext attacks are attractive for different practical applications. The known commutative ciphers are based on the computational difficulty of the DLP, therefore they do not provide security against quantum attacks. Development of the post-quantum versions of the commutative ciphers is also a challenge in the area of applied and theoretic cryptography. However, the problem of the development of the post-quantum commutative encryption algorithms has practically remained outside the attention of researchers.

The first attempt to solve the noted problem relates to the work [6] in which a commutative cipher was proposed, based on the hidden discrete logarithm problem (HDLP) defined in a finite quaternion algebra. In the recent paper [7] it is shown that that version of the HDLP can be polynomially reduced to the DLP in a finite field.

In the present paper a new form of the HDLP is applied to design the post-quantum commutative encryption algorithm suitable as the base primitive of the post-quantum no-key protocols. The used form of the HDLP is formulated in the 6-dimensional finite non-commutative associative algebra (FNAA) containing a large set of the global right-sided units. The introduced encryption method represents a specific probabilistic transformation, therefore it has been required to extend the interpretation of the notion of commutative encryption. Namely,

we call the encryption algorithm commutative, if the double encryption on two different keys produces the ciphertext that can be correctly decrypted with using the keys in arbitrary order. Such type of commutative ciphers can be used in the no-key encryption protocols.

2 Preliminaries

2.1 Algebraic carriers of the HDLP

In a finite m-dimensional vector space defined over a finite field, for example, $GF(p)$ there are defined two standard operations: addition of two vectors and multiplication of some given vector A by a scalar $d \in GF(p)$. The vector space with the additionally defined operation of multiplying two arbitrary vectors which is distributive relatively the addition operation is called the m-dimensional finite algebra. If the multiplication operation (denoted as $\circ$) is non-commutative and associative, then we have an m-dimensional FNAA. Suppose $\mathbf{e}_0$, $\mathbf{e}_1$, ... $\mathbf{e}_{m-1}$ are the basis vectors. The vector A is denoted in the following two forms: $A = (a_0, a_1, \ldots, a_{m-1})$ and $A = a_0\mathbf{e}_0 + a_1\mathbf{e}_1 + \cdots + a_{m-1}\mathbf{e}_{m-1}$, where $a_0, a_1, \ldots, a_{m-1} \in GF(p)$ are called coordinates.

Usually the multiplication operation of two vectors A and $B = \sum_{j=0}^{m-1} b_j\mathbf{e}_j$ is defined with the formula

$$A \circ B = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} a_i b_j (\mathbf{e}_i \circ \mathbf{e}_j),$$

in which products of different pairs of basis vectors $\mathbf{e}_i \circ \mathbf{e}_j$ are to be substituted by a single-component vector indicated in the so called basis vector multiplication table (BVMT). Every cell of the BVMT contains some single-component vector $\lambda\mathbf{e}_k$, where $\lambda \in GF(p)$ is called structural coefficient. If $\lambda = 1$, then the content of the cell is denoted as $\mathbf{e}_k$. One usually assumes that the left operand $\mathbf{e}_i$ defines the row and the right one $\mathbf{e}_j$ defines the column. The intersection of the ith row and jth column defines the cell indicating the value of the product $\mathbf{e}_i \circ \mathbf{e}_j$. To define associative multiplication one should compose the

BVMT that defines associative multiplication of all possible triples of the basis vectors $(\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k)$:

$$(\mathbf{e}_i \circ \mathbf{e}_j) \circ \mathbf{e}_k = \mathbf{e}_i \circ (\mathbf{e}_j \circ \mathbf{e}_k) .$$

2.2 Forms of the HDLP

The DLP is defined in a finite cyclic group Γ as follows: $Y' = G^x$, where G is a generator of the group and the value x is unknown natural number. Finding the value x, when the values G and Y' are known, is called DLP. The HDLP is defined so that one of the values G and Y' or both of them are hidden (masked). Thus, it is supposed the cyclic group Γ is a subset of elements of some algebraic structure called carrier of the HDLP. The FNAAs suite well for defining different versions of the HDLP. The exponentiation operation G^x is the base operation in the HDLP. The operations used to mask the values G and Y' are called the masking operations. To provide possibility to design a public-key cryptoscheme on the basis of HDLP one should use the masking operations that are mutually commutative with the base exponentiation operation. Therefore, the automorphism-map operations and the homomorphism-map operations are attractive to be applied as masking operations. A particular form of the HDLP is defined by the concrete set of the used masking operations.

The FNAAs are of significant interest as algebraic carriers of the HDLP and the cryptoschemes on its base. Different types of the FNAAs are used to define different forms of the HDLP. For the first time the HDLP was defined in the finite algebra of quaternions [6],[8] as follows:

$$Y = Q^w \circ G^x \circ Q^{-w} = \alpha\left(G^x\right), \tag{1}$$

where $Q \circ G \neq G \circ Q$; $\alpha(V)$ is the automorphism-map operation (V takes on all values in the quaternion algebra). The form of HDLP described by the formula (1) was applied to design a public key-agreement scheme and commutative encryption algorithm [6],[8]. However, reducibility of the first form of the HDLP to the DLP in the finite field $GF(p^2)$ was shown in the paper [7].

Recently [9],[10] several new FNAAs and new versions of the HDLP were introduced and used to develop the post-quantum digital signature protocols. For example, in the digital signature scheme defined in the FNAA containing global two-sided unit the public key represents the triple of vectors (Y, Z, T) defined as follows [9]:

$$Y = Q \circ G^x \circ Q^{-1}, \quad Z = H \circ G \circ H^{-1}; \quad T = Q \circ E \circ H^{-1}, \qquad (2)$$

where $Q \circ G \neq G \circ Q$; $H \circ G \neq H \circ Q$; E is a randomly selected vector from the set of local units related to the non-invertible vector G. The HDLP consists in finding the value x in the case, when only the public key is known.

In the signature scheme defined in the FNAA containing a large set of global left-sided units the public key represents the pair of vectors (Y, Z) defined as follows [10]:

$$Y = H \circ G^x \circ D, \quad Z = J \circ G \circ T, \qquad (3)$$

where $D \circ G \neq G \circ D$; $D \circ H = L_1$; $D \circ J = L_2$; $T \circ J = L_3$; L_1, L_2, and L_3 are global left-sided units. The HDLP consists in finding the value x in the case, when only the values Y and Z are known. In each of the last two versions of the HDLP no element of the base finite cyclic group is known, therefore the method [7] for reducing the HDLP to the DLP in a finite field does not work.

2.3 Commutative encryption

Encryption function (algorithm) F is called commutative, if for arbitrary encryption keys K_1 and $K_2 \neq K_1$ it satisfies the following condition

$$F_{K_1}[F_{K_2}(M)] = F_{K_2}[F_{K_1}(M)], \qquad (4)$$

where M is a plaintext. Commutative ciphers resisting the known plaintext attack can be used to implement Shamir's no-key protocol (also called Shamir's three-pass protocol [11]) described as follows. Suppose Alice wishes to send the secret message M to Bob, using a public channel and no shared key. For this purpose they can use the following protocol:

1. Alice selects at random the key K_1 and encrypts the message M using a commutative encryption function F: $C_1 = F_{K_1}(M)$. Then she sends the ciphertext C_1 to Bob.

2. Bob selects at random the key K_2 and encrypts the ciphertext C_1 : $C_2 = F_{K_2}(C_1)$. Then he sends the ciphertext C_2 to Alice.

3. Alice decrypts the ciphertext C_2 obtaining the ciphertext $C_3 = F^{-1}_{K_1}(C_2)$. Then she sends the ciphertext C_3 to Bob.

Having received the ciphertext C_3 Bob computes the value $M' = F^{-1}_{K_2}(C_3)$. Due to commutativity of the encryption function F we have $M' = M$:

$$\begin{aligned} M' &= F^{-1}_{K_2}(C_3) = F^{-1}_{K_2}[F^{-1}_{K_1}(C_2)] = F^{-1}_{K_2}[F^{-1}_{K_1}[F_{K_2}(C_1)]] = \\ &= F^{-1}_{K_2}[F^{-1}_{K_1}[F_{K_2}(F_{K_1}(M))]] = F^{-1}_{K_2}[F^{-1}_{K_1}[F_{K_1}(F_{K_2}(M))]] = \\ &= F^{-1}_{K_2}[F_{K_2}(M)] = M. \end{aligned}$$

This protocol provides security to passive attacks (an adversary only intercepts the ciphertexts sent via a channel), if the used commutative encryption function F is secure to the know-plaintext attack. The appropriate commutative encryption function is provided by the exponentiation cipher proposed by Pohlig and Hellman in [12], which is described as follows. Suppose p is a 2464-bit prime such that number $(p-1)$ contains a large prime divisor q, for example, $p = 2q + 1$.

To generate an encryption/decryption key (e, d) one selects a random number e (having size equal to 256 bit or more) that is mutually prime with the value $(p-1)$ and then computes the value $d = e^{-1} \bmod p - 1$. The encryption is defined by the formula

$$C = M^e \bmod p.$$

The next formula defines the decryption operation:

$$M = C^d \bmod p.$$

The Pohlig-Hellman commutative encryption algorithm is as secure as the DLP modulo p is difficult. In this paper we use the notion of the commutativity in the extended sense, namely, we call the encryption

algorithm commutative, if the consecutive encryption of the paintext with using two different keys produces the ciphertext which can be decrypted correctly using the keys in arbitrary order. The ciphers possessing such property can be used for imlementation of the three-pass no-key protocol.

The indicated interpretation of the notion of the commutative encryption covers the commutative encryption functions defined by the formula (4) Only deterministic encryption functions can be commutative in the sense defined by the formula (4). In the proposed extended sense of commutativity both the deterministic encryption functions and the probabilistic encryption functions can be commutative.

3 Algebraic carrier of the proposed post-quantum commutative cipher

The BVMT shown as Table 1 defines the 6-dimensional FNAA which contains p^3 different global right-sided units that can be computed from the following vector equation:

$$A \circ X = A, \tag{5}$$

where $X = (x_0, x_1, \ldots, x_5)$ is the unknown. Using Table 1 one can represent the equation (5) in the form of the following system of six linear equations with six unknowns $x_0, x_1, x_2, x_3, x_4, x_5$:

$$\begin{cases} \tau a_0 x_1 + a_0 x_4 + a_2 x_2 + \tau a_2 x_5 + a_4 x_0 + \tau a_4 x_3 = a_0; \\ \tau a_1 x_1 + a_1 x_4 + a_3 x_2 + \tau a_3 x_5 + a_5 x_0 + \tau a_5 x_3 = a_1; \\ a_0 x_0 + \tau a_0 x_3 + \tau a_2 x_1 + a_2 x_4 + a_4 x_2 + \tau a_4 x_5 = a_2; \\ a_1 x_0 + \tau a_1 x_3 + \tau a_3 x_1 + a_3 x_4 + a_5 x_2 + \tau a_5 x_5 = a_3; \\ a_0 x_2 + \tau a_0 x_5 + a_2 x_0 + \tau a_2 x_3 + \tau a_4 x_1 + a_4 x_4 = a_4; \\ a_1 x_2 + \tau a_1 x_5 + a_3 x_0 + \tau a_3 x_3 + \tau a_5 x_1 + a_5 x_4 = a_5 \end{cases} \tag{6}$$

Performing the variable substitution $u_1 = \tau x_1 + x_4$, $u_2 = x_2 + \tau x_5$, and $u_3 = x_0 + \tau x_3$ one can represent the system (6) in the form of the

following two systems of linear equations:

$$\begin{cases} u_1a_0 + u_2a_2 + u_3a_4 = a_0; \\ u_1a_2 + u_2a_4 + u_3a_0 = a_2; \\ u_1a_4 + u_2a_0 + u_3a_2 = a_4; \end{cases} \tag{7}$$

$$\begin{cases} u_1a_1 + u_2a_3 + u_3a_5 = a_1; \\ u_1a_3 + u_2a_5 + u_3a_1 = a_3; \\ u_1a_5 + u_2a_1 + u_3a_3 = a_5. \end{cases} \tag{8}$$

Table 1

The BVMT defining the FNAA containing p^3 global right-sided units

$\circ$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_0$	$\mathbf{e}_2$	$\tau\mathbf{e}_0$	$\mathbf{e}_4$	$\tau\mathbf{e}_2$	$\mathbf{e}_0$	$\tau\mathbf{e}_4$
$\mathbf{e}_1$	$\mathbf{e}_3$	$\tau\mathbf{e}_1$	$\mathbf{e}_5$	$\tau\mathbf{e}_3$	$\mathbf{e}_1$	$\tau\mathbf{e}_5$
$\mathbf{e}_2$	$\mathbf{e}_4$	$\tau\mathbf{e}_2$	$\mathbf{e}_0$	$\tau\mathbf{e}_4$	$\mathbf{e}_2$	$\tau\mathbf{e}_0$
$\mathbf{e}_3$	$\mathbf{e}_5$	$\tau\mathbf{e}_3$	$\mathbf{e}_1$	$\tau\mathbf{e}_5$	$\mathbf{e}_3$	$\tau\mathbf{e}_1$
$\mathbf{e}_4$	$\mathbf{e}_0$	$\tau\mathbf{e}_4$	$\mathbf{e}_2$	$\tau\mathbf{e}_0$	$\mathbf{e}_4$	$\tau\mathbf{e}_2$
$\mathbf{e}_5$	$\mathbf{e}_1$	$\tau\mathbf{e}_5$	$\mathbf{e}_3$	$\tau\mathbf{e}_1$	$\mathbf{e}_5$	$\tau\mathbf{e}_3$

One can easily see that the systems (7) and (8) are simultaneously satisfied for arbitrary vector A with the values of the unknowns $u_1 = 1$, $u_2 = 0$, and $u_3 = 0$. Thus, every vector $X = (x_0, x_1, x_2, x_3, x_4, x_5)$ coordinates of which satisfy the conditions

$$\begin{cases} \tau x_1 + x_4 = u_1 = 1; \\ x_2 + \tau x_5 = u_2 = 0; \\ x_0 + \tau x_3 = u_3 = 0; \end{cases} \Rightarrow \begin{cases} x_4 = 1 - \tau x_1; \\ x_2 = -\tau x_5; \\ x_0 = -\tau x_3 \end{cases} \tag{9}$$

is one of the solutions of the system (6) for every possible value A, i. e., every of such vectors X represent a right-sided unit acting on all elements of the considered FNAA (in this sense such units are global).

From the conditions (9) we get the following formula describing the set of p^3 different global right-sided units contained in the algebra:

$$R = (r_0, r_1, r_2, r_3, r_4, r_5) = (-\tau x_3,\ x_1,\ -\tau x_5,\ x_3,\ 1 - \tau x_1,\ x_5)\,, \tag{10}$$

where $x_1, x_3, x_5 = 0, 1, \ldots p-1$.

Evidently the considered FNAA contains neither global left-sided unit nor global two-sided unit. However, the algebra contains local left-sided units acting in the frame of some subsets of the elements. The local left-sided unit corresponding to some 6-dimensional vector A can be computed as solution of the following vector equation:

$$X \circ A = A. \tag{11}$$

The vector equation (11) can be represented in the form of the following two systems each of which contains three equations:

$$\begin{cases} (\tau a_1 + a_4)\, x_0 + (a_2 + \tau a_5)\, x_2 + (a_0 + \tau a_3)\, x_4 = a_0; \\ (a_0 + \tau a_3)\, x_0 + (\tau a_1 + a_4)\, x_2 + (a_2 + \tau a_5)\, x_4 = a_2; \\ (a_2 + \tau a_5)\, x_0 + (a_0 + \tau a_3)\, x_2 + (\tau a_1 + a_4)\, x_4 = a_4; \end{cases} \tag{12}$$

$$\begin{cases} (\tau a_1 + a_4)\, x_1 + (a_2 + \tau a_5)\, x_3 + (a_0 + \tau a_3)\, x_5 = a_1; \\ (a_0 + \tau a_3)\, x_1 + (\tau a_1 + a_4)\, x_3 + (a_2 + \tau a_5)\, x_5 = a_3; \\ (a_2 + \tau a_5)\, x_1 + (a_0 + \tau a_3)\, x_3 + (\tau a_1 + a_4)\, x_5 = a_5. \end{cases} \tag{13}$$

The main determinant of the systems (12) and (13) is the same and equal to the value

$$\begin{aligned} \Delta_A = (a_0 + \tau a_3)^3 + (\tau a_1 + a_4)^3 + (a_2 + \tau a_5)^3 - \\ - 3\,(a_0 + \tau a_3)\,(\tau a_1 + a_4)\,(a_2 + \tau a_5)\,. \end{aligned} \tag{14}$$

If $\Delta_A \neq 0$, then there exists the single solution $X = L_A$ that represents the local left-sided unit of the vector A.

Proposition 1. Suppose the vector A is such that $\Delta_A \neq 0$. Then there exists some integer ω such that $A^\omega = L_A$ and the vector L_A is simultaneously one of the global right-sided units.

Proof. Let us consider the sequence of the vectors $A, A^2, \ldots A^h, \ldots A^k, \ldots$. For all integer values i one has $A^i \neq O$, where $O = (0,0,0,0,0,0)$, since $\Delta_A \neq 0$. Due to finiteness of the considered algebras and condition $\Delta_A \neq 0$ the indicated sequence is periodic, i. e., for some integer h and minimum integer $k > h$ we have the following:

$$A^k = A^h \Rightarrow A^{k-h} \circ A^h = A^h \Rightarrow \Big(A^{k-h} \circ A - A\Big) \circ A^{h-1} = O \Rightarrow$$
$$A^{k-h} \circ A - A = O \Rightarrow A^{k-h} \circ A = A \Rightarrow A^{k-h} = L_A,$$
$$A^{k-h} \circ A = A \Rightarrow A \circ A^{k-h} = A \Rightarrow A^{k-h} = R_A = L_A = E_A,$$

where R_A is one of the global right-sided units described by the formula (10); L_A is the local left-sided unit related to the vector A; E_A is the local two-sided unit related to A. Thus, Proposition 1 holds true.

The single local two-sided unit E_A corresponds to the vector A such that $\Delta_A \neq 0$, since $E_A = L_A$ and we have the single local left-sided unit relating to the vector A. The value $\omega = k - h$ such that $A^\omega = E_A$ can be called the local order of the vector A. Proposition 1 shows $A^{\omega-1} \circ A = A \circ A^{\omega-1} = E_A$, i. e., the vector $A^{\omega-1}$ is the inverse value of the vector A relatively the local two-sided unit E_A.

In the next section a new method for commutative encryption is proposed, at the development of which the following propositions have been used.

Proposition 2. If the vector equation $A \circ X = B$ has solution $X = S$, then p^3 different values $X_i = R_i \circ S$, where R_i takes on all values from the set (10), also are solutions of the given equation.

Proof. $A \circ (R_i \circ S) = (A \circ R_i) \circ S = A \circ S = B$. The Proposition 2 is proven.

Proposition 3. If $A \circ B = R$, where R is a global right-sided unit, then the equality $A^i \circ B^i = R$ holds true for arbitrary natural value i.

Proof.

$A^i \circ B^i = \big(A^{i-1} \circ R\big) \circ B^{i-1} = A^{i-1} \circ B^{i-1} = A^{i-2} \circ B^{i-2} = \ldots$

$\cdots = A \circ B = R$. The Proposition 3 is proven.

Proposition 4. If $A \circ B = R$, where R is a global right-sided unit, then the map defined by the formula $\psi(X) = B \circ X \circ A$, where the

vector X takes on all values in the considered algebra, represents a homomorphism.

Proof. Suppose X_1 and X_2 are arbitrary two vectors. Then we have

$$\psi(X_1 \circ X_2) = B \circ (X_1 \circ X_2) \circ A = B \circ (X_1 \circ R \circ X_2) \circ A = \\ = (B \circ X_1 \circ A) \circ (B \circ X_2 \circ A) = \psi(X_1) \circ \psi(X_2);$$

$$\psi(X_1 + X_2) = B \circ (X_1 + X_2) \circ A = (B \circ X_1 \circ A) + (B \circ X_2 \circ A) = \\ = \psi(X_1) + \psi(X_2).$$

The Proposition 4 is proven.

Proposition 5. The homomorphism-map operation $\psi(X) = B \circ X \circ A$, where $A \circ B = R$, and the exponentiation operation X^i are mutually commutative, i. e., the equality $B \circ X^i \circ A = (B \circ X \circ A)^i$ holds true.

Proof. Due to Proposition 4 we have $\psi(X^i) = (\psi(X))^i$, i. e., $B \circ X^i \circ A = (B \circ X \circ A)^i$. The Proposition 5 is proven.

Proposition 6. Suppose the vector R is an arbitrary global right-sided unit and the vector X takes on all values in the considered FNAA. Then the map defined by the formula $\varphi(X) = R \circ X$ is a homomorphism.

Proof. Suppose X_1 and X_2 are arbitrary two 6-dimensional vectors. Then we have

$$\varphi(X_1 \circ X_2) = R \circ (X_1 \circ X_2) = (R \circ X_1) \circ (R \circ X_2) = \varphi(X_1) \circ \varphi(X_2);$$

$$\varphi(X_1 + X_2) = R \circ (X_1 + X_2) = R \circ X_1 + R \circ X_2 = \varphi(X_1) + \varphi(X_2).$$

The Proposition 6 is proven.

Proposition 7. The homomorphism-map operation $\varphi(X) = R \circ X$, where R is a global right-sided unit, and the exponentiation operation X^i are mutually commutative, i. e., the equality $R \circ X^i = (R \circ X)^i$ holds true.

Proof. Due to Proposition 6 we have $\varphi(X^i) = (\varphi(X))^i$, i. e., $R \circ X^i = (R \circ X)^i$. The Proposition 7 is proven.

4 Probabilistic commutative encryption algorithm

Suppose the FNAA described in Section 3 is defined over the field $GF(p)$, where the prime $p = 2q + 1$; q is a 256-bit prime, and the encrypted message is represented in the form of the vector $T = (t_0, t_1, \ldots, t_5)$ such that $\Delta_T \neq 0$. (For some given message probability that $\Delta_A = 0$ is negligible. Besides, one can modify the message to satisfy the condition $\Delta_T \neq 0$.) The local two-sided unit E_T relating to the vector T can be computed from the vector equation

$$X \circ T = T. \tag{15}$$

The vector E_T is contained in the set of global right-sided units (10). The algebra contains different vectors relating to some fixed value R from the set (10). All vectors such that the local two-sided unit of every of them is equal to R compose a finite group. The vector T is contained in a single of the mentioned finite groups and it generates a cyclic subgroup having the order ω that is a divisor of the value $p^2 - 1$. Therefore, the alternative method for computing the value E_T relates to using the formula

$$E_T = T^{p^2-1}. \tag{16}$$

However, finding the value E_T as solution of the vector equation (15) has significantly lower computational complexity. From (16) one can see that the message T can be encrypted and correctly decrypted with using the following two formulas:

$$C = T^e; \quad T = C^d, \tag{17}$$

where e and d are non-negative integers such that $ed \equiv 1 \bmod p^2 - 1$.

The formula (17) defines the commutative encryption function, the security of which to the known-plaintext attack is based on computational difficulty of the DLP. To develop a post-quantum commutative cipher one can use the masking operations, namely, the homomorphism-map operations ψ and φ.

4.1 Using the homomorphism map φ as encryption operations

The encryption of the message T with using the φ operation is performed with using the single-use subkey representing a randomly selected global right-sided unit R and is described as follows:

1. Compute the local two-sided unit E_T relating to the vector T.

2. Generate a random global right-sided unit R and compute the vector $C = R \circ T^e$.

3. Output the ciphertext in the form of the pair of two vectors (E_T, C).

This encryption procedure represents probabilistic ciphering process due to using a randomly selected single-use subkey R. The output ciphertext is two times larger in size than the input message T. The decryption is performed in accordance with the formula

$$T = E_T \circ C^d.$$

Correctness proof of this decryption formula is as follows:

$$E_T \circ C^d = E_T \circ (R \circ T^e)^d = E_T \circ R \circ T^{ed} = E_T \circ T = T.$$

To consider the commutativity property of the introduced encryption function one should define the process of encrypting the message on two different keys. Evidently, the first element of the ciphertext, i. e., the vector E_T should not be transformed in the frame of the second encryption. Therefore, the encryption of the message T on the key (e_1, d_1) and then on the key (e_2, d_2) produces the ciphertext

$$C_{12} = (E_T, R_2 \circ T^{e_1 e_2}),$$

where R_2 is a global right-sided unit selected at random at the second encryption as the single-use subkey. The encryption of the message T on the key (e_2, d_2) and then on the key (e_1, d_1) produces the ciphertext

$$C_{21} = (E_T, R_1 \circ T^{e_2 e_1}),$$

where R_1 is a random global right-sided unit selected as the single-use subkey at the second encryption. Due to probabilistic nature of the

encryption function in the considered two cases we have got different output ciphertexts. However, one can easily show both the ciphertext C_{21} and the ciphertext C_{12} are decrypted correctly using the keys (e_1, d_1) and (e_2, d_2) in arbitrary order. Thus, the described encryption function is commutative.

4.2 Using the homomorphism map ψ as encryption operation

To implement encryption of the message T with using the ψ operation one should apply two common parameters representing the vectors A and B such that $A \circ B = R_0$, where R_0 is some fixed global right-sided unit, and additional subkey representing a non-negative integer $t < p^2 - 1$, i. e., the encryption key represents the triple of the natural numbers (e, d, t). The encryption procedure includes the following steps:

1. Compute the local two-sided unit E_T relating to the vector T.
2. Compute the vector $C = B^t \circ T^e \circ A^t$.
3. Output the ciphertext in the form of the pair of two vectors (E_T, C).

This encryption algorithm is a deterministic procedure; the ciphertext is two times larger in size than the source message though. The decryption is performed in accordance with the formula

$$T = E_T \circ A^t \circ C^d \circ B^t.$$

Correctness proof of this decryption formula is as follows:

$$E_T \circ A^t \circ C^d \circ B^t = E_T \circ A^t \circ \left(B^t \circ T^{ed} \circ A^t\right) \circ B^t =$$
$$= E_T \circ R_0 \circ T \circ R_0 = E_T \circ T = T.$$

Let us consider the case of encryption of the message T on two independent keys (e_1, d_1, t_1) and (e_2, d_2, t_2). Like in the case of the encryption algorithm described in the previous subsection the first element of the ciphertext, i. e., the vector E_T should not be transformed in the frame of the second encryption. Encryption of the message T on the first key

and then on the second key produces the ciphertext

$$C_{12} = \left(E_T, B^{t_1+t_2} \circ T^{e_1e_2} \circ A^{t_1+t_2}\right).$$

Encryption of the message T on the second key and then on the first key produces the ciphertext

$$C_{21} = \left(E_T, B^{t_2+t_1} \circ T^{e_2e_1} \circ A^{t_2+t_1}\right) = C_{12}.$$

Thus, we have the same output ciphertext in the both cases, i. e., the described encryption algorithm is commutative.

4.3 Post-quantum commutative cipher and no-key protocol on its basis

In this section the post-quantum version of the commutative cipher is proposed, which is based on the HDLP defined with using both the masking φ-map operation and the masking ψ-map operation. The encryption procedure is performed using the key (e, d, t) as follows:

1. Compute the local two-sided unit E_T relating to the input message T.

2. Generate a random global right-sided unit and compute the vector $C = R \circ B^t \circ T^e \circ A^t$.

3. Output the ciphertext in the form of the pair of two vectors (E_T, C).

This encryption algorithm is a probabilistic procedure. The decryption is performed in accordance with the formula

$$T = E_T \circ A^t \circ C^d \circ B^t.$$

Correctness proof of this decryption formula is as follows:

$$E_T \circ A^t \circ C^d \circ B^t = E_T \circ A^t \circ \left(R \circ B^t \circ T^{ed} \circ A^t\right) \circ B^t =$$
$$= E_T \circ R_0 \circ T \circ R_0 = E_T \circ T = T.$$

Encrypting the message T on the key (e_1, d_1, t_1) and then on the key (e_2, d_2, t_2) produces the ciphertext

$$C_{12} = \left(E_T, R_2 \circ B^{t_2+t_1} \circ T^{e_1e_2} \circ A^{t_1+t_2}\right),$$

where R_2 is a random global right-sided unit used at the second encryption as the single-use subkey. Encryption of the message T on the key (e_2, d_2, t_2) and then on the key (e_1, d_1, t_1) produces the ciphertext

$$C_{21} = \left(E_T, R_1 \circ B^{t_1+t_2} \circ T^{e_2 e_1} \circ A^{t_2+t_1}\right),$$

where R_1 is a random global right-sided unit used at the second encryption as the single-use subkey.

In the considered two cases of double encryption we have different output ciphertexts. However, each of the ciphertexts is decrypted correctly, when performing the double decryption with using the keys in arbitrary order. Thus, the proposed post-quantum encryption algorithm is commutative. It can be used as the base encryption function in the following post-quantum no-key protocol.

1. Alice selects her local key (e_A, d_A, t_A), generates at random the single-use subkey R_A, computes the two-sided local unit relating to the vector T, and encrypts the message M:

$$C_1 = R_A \circ B^{t_A} \circ T^{e_A} \circ A^{t_A}.$$

Then she sends the ciphertext (E_T, C_1) to Bob.

2. Bob selects his local key (e_B, d_B, t_B), generates at random the single-use subkey R_B, and encrypts the vector C_1:

$$C_2 = R_B \circ B^{t_B} \circ C_1^{e_B} \circ A^{t_B}.$$

Then he sends the vector C_2 to Alice.

3. Alice generates at random the single-use subkey R'_A and decrypts the vector C_2 obtaining the ciphertext

$$C_3 = R'_A \circ A^{t_A} \circ C_2^{d_A} \circ B^{t_A}.$$

Then she sends the vector C_3 to Bob.

Having received the ciphertext C_3 Bob computes the value

$$M = E_T \circ A^{t_B} \circ C_3^{d_B} \circ B^{t_B}.$$

Correctness proof of this decryption formula is as follows:

$$
\begin{aligned}
C_2 &= R_B \circ B^{t_B} \circ C_1^{e_B} \circ A^{t_B} = \\
&= R_B \circ B^{t_B} \circ \left(R_A \circ B^{t_A} \circ T^{e_A} \circ A^{t_A}\right)^{e_B} \circ A^{t_B} = \\
&= R_B \circ B^{t_B} \circ \left(R_A \circ B^{t_A} \circ T^{e_A e_B} \circ A^{t_A}\right) \circ A^{t_B} = \\
&= R_B \circ B^{t_B} B^{t_A} \circ T^{e_A e_B} \circ A^{t_A} A^{t_B} \;\; \Rightarrow \\
C_3 &= R'_A \circ A^{t_A} \circ \left(R_B \circ B^{t_B} B^{t_A} \circ T^{e_A e_B} \circ A^{t_A} A^{t_B}\right)^{d_A} \circ B^{t_A} = \\
&= R'_A \circ R_0 \circ B^{t_B} \circ T^{e_A e_B d_A} \circ A^{t_B} \circ R_0 = \\
&= R'_A \circ B^{t_B} \circ T^{e_B} \circ A^{t_B} \quad \Rightarrow \\
E_T \circ A^{t_B} \circ C_3^{d_B} \circ B^{t_B} &= E_T \circ A^{t_B} \circ \left(R'_A \circ B^{t_B} \circ T^{e_B d_B} \circ A^{t_B}\right) \circ B^{t_B} = \\
&= E_T \circ R_0 \circ T \circ R_0 = E_T \circ T = T.
\end{aligned}
$$

5 Conclusion

Interpreting the notion of commutative encryption in the wider sense, for the first time the probabilistic commutative encryption algorithm has been developed. The 6-dimensional FNAA containing p^3 different global right-sided units, which is defined over the finite ground field $GF(p)$, have been used as the algebraic carrier of the proposed post-quantum probabilistic commutative cipher based on the HDLP. The base encryption operation is the exponentiation operation complemented with two different masking homomorphism-map operations. One more novel feature of the commutative encryption method is the application of the single-use subkeys selected at random from the set of the global right-sided units.

On the basis of the introduced commutative cipher a post-quantum no-key protocol have been developed that seems more attractive for practical applications than the recently proposed one [13].

References

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on quantum computer," *SIAM Journal of Computing*, vol. 26, pp. 1484–1509, 1997.

[2] S. Y. Yan, *Quantum Attacks on Public-Key Cryptosystems*, Springer, 2014, 207 p.

[3] Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. NIST PQCrypto project. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[4] *Post-Quantum Cryptography. 9th International Conference, PQCrypto 2018 Proceedings*, Fort Lauderdale, FL, USA, April 9-11, 2018, (Lecture Notes in Computer Science, vol. 10786), 2018.

[5] PQC Standardization Process: Second Round Candidate Announcement. https://csrc.nist.gov/news/2019/pqc-standardization-process-2nd-round-candidates

[6] D.N. Moldovyan, "Non-Commutative Finite Groups as Primitive of Public-Key Cryptoschemes," *Quasigroups and Related Systems*, vol. 18, no. 2, pp. 165–176, 2010.

[7] A. S. Kuzmin, V. T. Markov, A. A. Mikhalev, A. V. Mikhalev, and A. A. Nechaev, "Cryptographic Algorithms on Groups and Algebras," *Journal of Mathematical Sciences*, vol. 223, no. 5, pp. 629–641, 2017.

[8] D. N. Moldovyan and N. A. Moldovyan, "Cryptoschemes over hidden conjugacy search problem and attacks using homomorphisms," *Quasigroups Related Systems*, vol. 18, no. 2, pp. 177–186, 2010.

[9] A. A. Moldovyan and N. A. Moldovyan, "Post-quantum signature algorithms based on the hidden discrete logarithm problem," *Computer Science Journal of Moldova*, vol. 26, no. 3(78), pp. 301–313, 2018.

[10] N. A. Moldovyan and A. A. Moldovyan, "Finite Non-commutative Associative Algebras as carriers of Hidden Discrete Logarithm Problem," *Bulletin of the South Ural State University. Ser. Mathematical Modelling, Programming & Computer Software (Bulletin SUSU MMCS)*, vol. 12, no. 1, pp. 66–81, 2019.

[11] A. J. Menezes, P. C. Oorschot, and S. A. Vanstone, *Applied cryptography*, New York, London: CRC Press, 1996, 780 p.

[12] M. E. Hellman and S. C. Pohlig, "Exponentiation Cryptographic Apparatus and Method," U.S. Patent # 4,424,414. 3 Jan. 1984.

[13] N. A. Moldovyan, A. A. Moldovyan, and V. A. Shcherbacov, "Post-quantum No-key Protocol," *Buletinul Academiei de Stiinte a Republicii Moldova. Matematica*, no. 3(85), pp. 115–119, 2017.

A.A. Moldovyan, D.N. Moldovyan, N.A. Moldovyan, Received April 15, 2019

St. Petersburg Institute for Informatics and Automation
of Russian Academy of Sciences,
14 Liniya, 39, St. Petersburg 199178, Russia
E–mail: maa1305@yandex.ru; mdn.spectr@mail.ru; nmold@mail.ru;
http://www.spiiras.nw.ru

Computer Science Journal of Moldova, vol.**27**, no.3(81), 2019

A new multi-offspring crossover operator for genetic algorithm to facilitate the traveling salesman problem

Abid Hussain, Salman A. Cheema

Abstract

This research work provides a detailed working principle and official analysis of a multi-offspring crossover operator. The proposed operator explains the true theory of survival-of-fittest using the foundation of evolutionary theories of biology and ecological theories of mathematics. We found a considerable improvement because the proposed operator enhances the opportunity of having better offspring, which results in highly competitive population. Simulation results of this operator with other competitor crossover operators for one of the combinatorial optimization problems, i.e. traveling salesman problem, are obviously showing its pros at better accuracy level. Moreover, the t-test and performance index (PI) establishes the improved significance and accuracy levels of the proposed operator. Preferable results of this operator not only confirm its advantages over the others, but also show long run survival of a generation having a number of offspring more than the number of parents with the help of mathematical ecology theory.

Keywords: Traveling salesman problem, genetic algorithms, crossover operators, multi-offspring, performance index.

1 Introduction

The traveling salesman problem (TSP) is one of the most famous benchmark, significant, historic and very important combinatorial optimization problem [1]. TSP was documented by Euler in 1759, his interest

was how to get rid of the knight's tour problem [2]. It is the fundamental problem in the fields of computer science, engineering, operations research, discrete mathematics and graph theory etc. To find the shortest tour that visits each city by a salesman in a given list exactly once and then returns to the starting city is the required goal. TSP has extremely large search spaces and is very difficult to solve, so it is called a typical non-deterministic polynomial (NP-hard) problem [3]–[5]. The given n cities, a distance matrix $C = [c_{ij}]_{n\times n}$ is searched for a permutation $\lambda : \{0, ..., n-1\} \longrightarrow \{0, ..., n-1\}$, where c_{ij} is the distance from city i to city j, which minimizes the traveled distance, $f(\lambda, C)$:

$$f(\lambda, C) = \sum_{i=0}^{n-1} d(c_{\lambda(i)}, c_{\lambda(i+1)}) + d(c_{\lambda(n)}, c_{\lambda(1)}), \tag{1}$$

where $\lambda(i)$ represents the location of city i in each tour, $d(c_i, c_j)$ is the distance between city i and city j and (x_i, x_j) is a specified position of each city in a tour in the plane, and the Euclidean distances of the distance matrix C between the city i and the city j are expressed as:

$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{2}$$

For n cities, there are $(n-1)!$ possible ways to find the tour after fixing the starting city for asymmetric distances, i.e. $c_{ij} \neq c_{ji}$ and its half for symmetric ones, i.e. $c_{ij} = c_{ji}$. This type of problems cannot be solved using traditional optimization approaches like gradient-based methods. To achieve the optimal solution within a reasonable amount of time, heuristic approaches are efficient at handling the NP-hard problems [6]. Apart from its theoretical approach, TSP is widely used as a model in many fields such as vehicle routing problem [7], physical mapping problems [8], constructing phylogenetic trees [9], machine flow shop scheduling [10] and so on.

TSP is a very carefully studied problem and received considerable attention over the last few decades. A lot of algorithms have been presented to solve this problem by researchers. These algorithms are generally divided into two classes: exact and approximate algorithms. The

branch-and-bound [11] and cutting planes [12] are the two examples of exact algorithms which are excessively time consuming especially in large scale problems. Approximate algorithms are further classified into heuristic and meta-heuristic algorithms. There are three classes of the heuristic algorithm: tour construction, tour improvement and composite methods. To add an unvisited city to the solution at each step and try to shorten the initial solution are tour construction and tour improvement methods respectively. Finally, the composite method is the combination of these two algorithms. In the last three decades, it has started a new stage of study about optimization problems with the appearance of meta-heuristic algorithms. These search algorithms have also been applied on TSP; 2-opt [6], particle swarm optimization [13], simulated annealing [14], ant colony optimization [15], neural network [16], tabu search [17], and genetic algorithms (GAs) [18]–[21].

We used genetic algorithm (GA) in this research to solve the TSP. In the literature of GAs, the pair of parents generated the pair of offspring. For a long run survival and diversity of species, it should be desirable to create more offspring than parents. In this research, we provide a multi-offspring crossover operator to handle this problem and give a detail discussion how this can be done. The simulation results with benchmarks show that the proposed operator indeed generates improved performance as compared to other traditional operators in the TSP application.

The rest of this article is presented as follows: in Section 2, we present the overview of GAs for TSP. Theoretical background with implementation of the proposed operator is presented in Section 3. The performance evaluation based on simulation study and conclusion are in Sections 4 and 5 respectively.

2 Genetic algorithms for TSP

Genetic algorithms (GAs) are derivative free stochastic approaches which are based on biological evolutionary processes proposed by John Holland [22]. GA is used to search the optimal or near to optimal solutions of various optimization problems. It is one of the main branches of

evolutionary algorithms, where the natural process from biology discipline is mimicked which was originally described by Darwin [23]. GAs are iterative processes and iterations are called generations [24]. In 1985, GAs were first time used to solve TSP by Goldberg [25]. A simple and pure GA for TSP can be defined in the following subsections.

2.1 Chromosome representation

In literature, there have been various representations to encode feasible solutions to solve the TSP using the GAs. Some of them are binary, path, adjacency, real and matrix representations of the chromosomes. A detailed study of these approaches is presented by Larranaga et al. [2]. Other than the path, all techniques have complex nature to complete a legal tour for the next generation. A path representation is desired because it is the most natural and elaborated way to represent a tour. For example, a nine-city tour $7 \longrightarrow 2 \longrightarrow 6 \longrightarrow 1 \longrightarrow 4 \longrightarrow 8 \longrightarrow 3 \longrightarrow 9 \longrightarrow 5$ can be represented simply as (7 2 6 1 4 8 3 9 5), and the total distance of this path is

$$D_{72} + D_{26} + D_{61} + D_{14} + D_{48} + D_{83} + D_{39} + D_{95} + D_{57}$$

2.2 Initial population

A traveling track in the population can be represented by the sequence of all cities, and an array of integers is used to indicate the sequence of cities. For example, there is a TSP with nine cities so all 9! tracks are non-repeating integers from 1 to 9 (both included), such as 7 2 6 1 4 8 3 9 5 is one of them. More generally, if a TSP having 'm' cities and 'n' is the population size, it means we generate 'n' randomly chosen sequences with each of them containing 'm' non-repeating integers.

2.3 Fitness evaluation

The evaluation of individuals in a population is based upon some specific fitness criteria. It turns from simple to complex if more and more parameters are added to evaluate an individual. The easiest way to

evaluate the fitness of an individual is summing up the lengths of its track [26],[27]. This kind of fitness evaluator is used if all solutions are feasible and no other constraints are given (like the number of routes, the maximal length of the route and similar).

2.4 Selection operator

Selection scheme is a procedure in which individuals are selected on the base of their fitness for mating. There are several methods usually used for the TSP and these are equally useful for their respective problems. The commonly used methods are tournament [28], roulette wheel [22] and rank based [29] methods. Among the above methods, the most popular one is tournament, where randomly selected individuals compete with one another and the best one is picked for the next phase. Generally there are two types of tournament methods, i.e. binary and more than two individuals competing at the same time [30]. We have used binary tournament selection (BTS) in this research, where two candidates are selected at random and the better of them is nominated as a parent. For t-tournament selection, the probability after rank 1 to the best individual can be defined as:

$$P_i = N^{-t}((N - i + 1)^t - (N - i)^t), \tag{3}$$

where $i \in \{1, 2, ...N\}$, N and t are the population and the tournament sizes respectively. A roulette wheel selection (RWS) is also used in this study to choose the individuals for mating process. For this purpose, first we calculate the fitness of all individuals using the following rule:

$$f_i = \beta(1 - \beta)^i; \quad i = 1, 2, ..., N,$$

where f_i is the i^{th} individual of sorted population in ascending order, and $\beta \in (0, 1)$ and generally suitable within the range of 0.01 to 0.3 [31]. After this, we assign the selection probability of i^{th} individual to be calculated as follows:

$$P_i = \frac{f_i}{\sum_{i=1}^{N} f_i}. \tag{4}$$

The next selection approach which we have used in this study is the linear ranked selection (LRS). This LRS is an alternative approach of RWS, when fitness values are far away from each other in a population. The selection probability is linearly assigned to the individuals after rank 1 to the worst individual as:

$$p_i = \frac{1}{N}(\eta^- + (\eta^+ - \eta^-)\frac{i-1}{N-1}); \qquad i \in \{1, 2, ..., N\}. \tag{5}$$

Here $\frac{\eta^-}{N}$ and $\frac{\eta^+}{N}$ are the probabilities of the worst and the best chromosomes to be selected respectively. All the individuals get a different rank even if they have same fitness value. The conditions $\eta^+ = 2 - \eta^-$ and $\eta^- \geq 0$ must be fulfilled. All the above mentioned selection procedures have been used in various applications, see for example [32].

2.5 Crossover operator

Crossover is the process of mating selected candidates for the purpose of producing further offspring. Since these offspring share the characteristics of their parents, hence there must be a rule to make a combination of these characteristics. The type of crossover is strongly dependent on the genotype representation of the population. In TSP, the most used method is path-based crossover technique. To generate offspring from a combination of the pair of paths, any random point is selected in such a way that it takes the first portion from one parent and the second portion from the other parent. This approach is called a 1-point crossover. As a result, there may emerge some infeasible solutions which can be avoided by putting a constraint that only feasible solutions can enter into coming generation [33]. Whereas in 2-point crossover (2PX), the expansion proceeds with a random selection of two points. Since the middle portion is exchanged here [34],[35], hence it may also become infeasible. To overcome this problem, Choi et al. [26] randomly insert missing bits in the middle portion. Another approach to counter this problem is a Partially-mapped crossover (PMX), where the indexes are exchanged within the cut-points portion and missing bits are replaced with their mapping index in the other parent [25]. The order crossover

(OX) is the next more commonly used operator of 2PX, in which the central part (from point a to point b, where $a < b$) is taken from one parent and other parent is selected from $b+1$ point circularly onward to complete the legal offspring [36]. The existing crossover approaches are less useful because they don't take the important nodes into account. It means that some tracks may be divided in half and others may not even be copied. This can dramatically affect in the new generation by generating new random tracks irrespective of the good ones. Next crossover operation is named cycle crossover(CX), which is unusual in its nature and detects a set of cycles between two selected candidates in their chromosomes and later it duplicates these cycles one by one to offspring [37].

The multi-offspring genetic algorithm (MO-GA) was proposed by Wang et al. [31]. It builds offspring preserving the relative order of bits of one parent by choosing a sub-tour of other parent. First of all the parent's track is divided into three regions after applying two random cut points. This scheme produces four offspring in two stages. In the first stage, its work is similar as OX, where the middle portion (from point a to point b, where $a < b$) is taken from one parent and the other parent is selected from $b + 1$ point circularly onward to complete the legal offspring. In the second stage, exchange regions 1 and 2 or simply de-track the original tracks of parents. Find region 3 elements from the first new parent, delete the corresponding ones in the second new parent, and vice versa. After this the regions 3 of both parents are replaced with each other for producing offspring. We also proposed a new multi-offspring crossover operator in this study which is presented in Section 4.

2.6 Mutation operator

Mutation operator expands the search space by a small degree. It is rare happening in nature and similar as that in GAs, where the mutation probability factor is predefined by the researchers. Since mutation causes extraordinary results to genetic principle, hence its factor must be small. It is the simplest operator in GAs which can be used in

various ways. A fractional change in genotype may result in a reasonable change in the next generation. Dealing with TSP, the simplest mutation can be completed by random exchange of any two cities in a route as an exchange mutation (EM) [38]. The other approach is insertion mutation (IM) which shifts the selected city from its original place and inserts it on another random place [39],[40]. Another popular approach is inversion mutation (INM), where a specific sub-track of a chromosome is inverted [39],[41]. If only two cities in a chromosome are inverted, then it is called the swap mutation (SM) [42].

2.7 Termination criteria

Generally, the termination criteria vary with the nature of research. It may be time dependent process as done with evolving the limited time only 1800 seconds [43]. Another option is the limitation of generations as done [27] with maximum of 1000 while [44] used up to 50,000 generations. The other significant process of evaluation is terminating when candidates stay the same for 30% of the generation limit [45]. Researchers [44] stopped the evolution process when almost 95% of the population is converged to one unique solution. This research uses two criteria to stop the process, i.e. a tour shorter than the current optimum trip has not been found during 300 consecutive iterations otherwise to complete the process with a maximum number of generations, i.e. 5000.

3 Theoretical background of MO-OX

3.1 Biological theory foundation

In the literature, the GA for TSP explains a pair of parents producing a pair of offspring [21],[46]. However the biological evolution proceeds in such a way that it generates equal or less than a pair of parents, which is uncommon. Since having the risk of extinction the generation or being less competitive, it is desirable to have the number of offspring more than two in genetic. When the number of offspring is larger than

2 from two better parents, there is more competition among them to survive, with increases chances of better offspring.

3.2 Mathematical ecological theory foundation

The probability of extinction of species can be illustrated by supposing that a pair of species has only two offspring at first. The probability of population size that leads to "0" at a certain time "t" can be defined as:

$$P_0(t|i=1) = \frac{\eta\ exp((\theta-\eta)t) - \eta}{\theta\ exp((\theta-\eta)t) - \eta}, \tag{6}$$

where i denotes the size of the initial population, η and θ are the mortality and fertility rates respectively. This population can be extinct with time elapsing by the following probability:

$$P_0(t) = [P_0(t|i=1)]^i = [\frac{\eta\ exp((\theta-\eta)t) - \eta}{\theta\ exp((\theta-\eta)t) - \eta}]^i. \tag{7}$$

As t$\longrightarrow \infty$, three scenarios of equation (7) can be observed:

(1) When the mortality rate equals the fertility rate, i.e. $\eta = \theta$, equation (7) is expressed as a series of exponential terms. Letting $\theta - \eta = r$, then $P_0(t)$ as t$\longrightarrow \infty$ should be

$$P_0(t) = [\frac{\eta(1 + rt + r^2t^2/2! + ...) - \eta}{\theta(1 + rt + r^2t^2/2! + ...) - \eta}]^i. \tag{8}$$

If η equals to θ, then r$\longrightarrow$0, so we are eliminating r^2 and higher terms and get the following expression:

$$P_0(t) \longrightarrow [\frac{\eta rt}{(\theta + \theta rt) - \eta}]^i \longrightarrow [\frac{\eta rt}{(\theta - \eta) + \theta rt}]^i \longrightarrow [\frac{\eta rt}{(r + \theta rt}]^i. \tag{9}$$

Hence,

$$\lim_{t\to\infty} [\frac{\theta t}{1 + \theta t}]^i = 1. \tag{10}$$

It means that the species should become extinct with 100% probability (shown in equation (10)), even when the mortality rate equals to the fertility rate.

(2) If the mortality rate is greater than fertility rate in each generation, i.e. $\eta > \theta$. Letting $\theta - \eta = -r$, then the exponential terms of equation (7) would be "0" when t$\longrightarrow \infty$ and we get:

$$P_0(t) \longrightarrow [\frac{\eta\ exp(-rt) - \eta}{\theta\ exp(-rt) - \eta}]^i, \tag{11}$$

and finally it leads to

$$\lim_{t\to\infty}(P_0(t)) = 1.$$

Hence, in this scenario with growing the time, species should become extinct.

(3) If the mortality rate is less than fertility rate in each generation, i.e. $\eta < \theta$, the equation (7) as t$\longrightarrow \infty$ can be expressed as follows:

$$P_0(t) \longrightarrow [\frac{\eta\ exp(t) - \eta}{\theta\ exp(t) - \eta}]^i. \tag{12}$$

Hence,

$$\lim_{t\to\infty}(P_0(t)) = [\frac{\eta}{\theta}]^i.$$

According to equation (12), there is no guarantee that such a population will never become extinct because a finite probability exists for the extinction. However, if the mortality rate is much lower than the fertility rate, then we have the least probability of biological extinction.

According to mathematical ecological theory in biology, if $\theta = \eta$ for a certain population, the probability of population extinction is 1. Thus, the probability distribution of biological population size depends on the product of fertility rate and time when the biological initial population is known. Hence, it is important to improve the fertility rate of species to get better individuals in possible shortest time.

3.3 Multi-offspring crossover procedure

The new extension involves producing four offspring, two from each technique of direct ordering, as well as the other one being from revers ordering of the parents tracks, so we suggested multi-offspring order crossover (MO-OX). Consider an example of the two parents tours (with randomly two cut points marked by "|"):

$$P_1 = (9 \quad 4 \quad 5 \mid 2 \quad 8 \quad 1 \mid 6 \quad 7 \quad 3) \quad \textit{and}$$

$$P_2 = (3 \quad 6 \quad 1 \mid 9 \quad 7 \quad 8 \mid 2 \quad 4 \quad 5).$$

The MO-OX is utilized in two stages to generate four offspring. In Stage 1, the crossover operation is as follows:

Stage 1: The offspring are produced in the following way. First, the bits are copied down between the cuts with similar way into the offspring, which gives:

$$O_1 = (\times \quad \times \quad \times \mid 2 \quad 8 \quad 1 \mid \times \quad \times \quad \times) \quad \textit{and}$$

$$O_2 = (\times \quad \times \quad \times \mid 9 \quad 7 \quad 8 \mid \times \quad \times \quad \times).$$

After this, starting from the second cut point of one parent, the bits from the other parent are copied in the same order omitting existing bits. As the sequence of the bits in the second parent from the second cut point is:

$$2 - 4 - 5 - 3 - 6 - 1 - 9 - 7 - 8$$

after removal of bits 2, 8 and 1, which already exist in the first offspring, the new sequence is:

$$4 - 5 - 3 - 6 - 9 - 7.$$

To complete the first offspring, this new sequence is placed starting from the second cut point:

$$O_1 = (6 \quad 9 \quad 7 \mid 2 \quad 8 \quad 1 \mid 4 \quad 5 \quad 3).$$

Analogously, we complete the second offspring as well:

$$O_2 = (5 \quad 2 \quad 1 \mid 9 \quad 7 \quad 8 \mid 6 \quad 3 \quad 4).$$

Stage 2: In Stage 2, the offspring are produced in the following way. First, the bits are copied down between the cuts with similar way into the offspring, which gives:

$$O_1 = (\times \ \times \ \times \mid 2 \ \ 8 \ \ 1 \mid \times \ \times \ \times) \quad \textit{and}$$

$$O_2 = (\times \ \times \ \times \mid 9 \ \ 7 \ \ 8 \mid \times \ \times \ \times).$$

After this, starting from the first cut point of one parent, the bits from the other parent are copied in the reverse order omitting existing bits. The reverse sequence of the bits in the second parent from the first cut point is as follows:

$$1 - 6 - 3 - 5 - 4 - 2 - 8 - 7 - 9.$$

After removal of bits 2, 8 and 1, which are already in the first offspring, the new sequence is:

$$6 - 3 - 5 - 4 - 7 - 9.$$

This sequence is placed in the first offspring starting from the first cut point with reverse sequence as:

$$O_1 = (5 \ \ 3 \ \ 6 \mid 2 \ \ 8 \ \ 1 \mid 9 \ \ 7 \ \ 4).$$

Analogously, we complete the second offspring as well:

$$O_2 = (3 \ \ 4 \ \ 5 \mid 9 \ \ 7 \ \ 8 \mid 2 \ \ 1 \ \ 6).$$

The newer scheme, by combining both OX and ROX can be termed as a multi-offspring order crossover (MO-OX). Hence, we differentiate MO-OX in the Algorithm 1.

Algorithm 1 The Pseudo-code of MO-OX

```
N ⟵ no. of cities
i ⟵ 1
while (i <= N) do
    P1 ⟵ a sequence of the first parent
    P2 ⟵ a sequence of the second parent
    cut1 ⟵ random point at the same location on parents
    cut2 ⟵ another random point at the same location on parents
    offspring1 ⟵ P1(cut1 to cut2)          %child by OX
    offspring2 ⟵ P2(cut1 to cut2)          %child by ROX
    offspring3 ⟵ P1(cut1 to cut2)          %child by RX
    offspring4 ⟵ P2(cut1 to cut2)          %child by ROX
 % For offspring1 in clockwise direction
    i ⟵ cut2, j ⟵ i
    while (length of offspring1 ! = length of P1) do
        if find((offspring1 == P1(i)))     %if value already exists in offspring1
            i = i + 1
        else
            offspring1(j) = P1(i)
            i ⟵ i + 1; j ⟵ j + 1;
        end if
        if (i > length of P1)
            i ⟵ 1
        end if
        if (j > length of P1)
            j ⟵ 1
        end if
    end while
    %Similarly to offspring3 from P2%
 % For offspring2 in anticlockwise direction
    i ⟵ cut1, j ⟵ i
    while (length of offspring2 ! = length of P1) do
        if find ((offspring2 == P1(i)))    %if value already exists in offspring2
            i = i - 1
        else
            offspring2(j) = P1(i)
            i ⟵ i - 1; j ⟵ j - 1;
        end if
        if(i==0)
            i ⟵ length of P1
        end if
        if(j==0)
            j ⟵ length of P1
        end if
    end while
    %Similarly to offspring4 from P2%
end while
```

4 Performance evaluation

4.1 Computational testing methodology

To evaluate the performances of the proposed MO-OX, computational experiments have been performed by using six benchmark instances which are taken from the traveling salesman problem library (TSPLIB) [47]. The test benchmarks are Euclidean, two-dimensional symmetric and asymmetric problems with 42, 53, 171, 180, 443, and 532 cities. In our simulation experiments, all GA programs were implemented in MATLAB version R2015a. Table 1 shows the state-of-the-art parametric configuration of the GA, and complete methodology for all the experiments performed in this research is shown in Figure 1. Moreover, we used two stopping criteria for our simulation experiments, i.e. attaining the maximum number of generations and if the tour, shorter than the current optimal tour, is not being found during the last 300 consecutive generations.

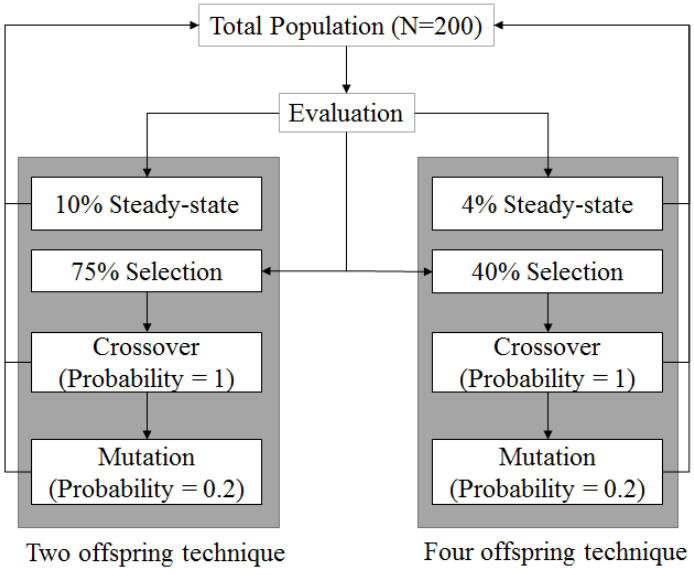

Figure 1. The methodology flow chart for simulation study

Table 1. Parametric configuration for GA

Parameter	Value
Population size	200
Selection criteria	BTS, RWS, RBS
Crossover rate	100%
Mutation method	EM, INM
Mutation rate	20%
Maximum generation	5000
Replacement in GA	Steady-state GA

Since GAs belong to the class of stochastic search algorithms, we employ statistical hypothesis testing using the two independent samples t-test. The equation (13) presents the general expression for the applied t-test such as:

$$t = \frac{\bar{x_1} - \bar{x_2}}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \tag{13}$$

where,

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2},$$

and $\bar{x_1}$ and $\bar{x_2}$ are the averages of 30 trails for the proposed and rival operators respectively. Similarly, s_1^2 and s_2^2 are standard deviations (S.D) of the proposed and contemporary method, respectively. A statistical significant difference is then quantified at $\alpha = 0.05$ (95% confidence) level of significance, under the null hypothesis stating that "MO-OX and the selected operator converge equally fast under given settings".

We also used the performance index (PI) criteria which is widely used to compare the newly proposed techniques for population-based heuristic algorithms [48]–[51]. The relative performances using PI are calculable as:

$$PI = \frac{1}{Np} \sum_{i=1}^{Np} (k_1 \alpha_1^i + k_2 \alpha_2^i + k_3 \alpha_3^i),$$

where,

$$\alpha_1^i = \frac{A^i}{MA^i}$$
$$\alpha_2^i = \frac{S^i}{MS^i}$$
$$\alpha_3^i = \frac{R^i}{MR^i}; \qquad for\ i = 1, 2, ..., N_p$$

and

N_p**:** Total number of problems analyzed

A^i**:** Average values of objective function of i^{th} problem for all competing operators

MA^i**:** Minimum of average value of objective function of i^{th} problem obtained through all competing operators

S^i**:** Standard deviations of objective function of i^{th} problem for all competing operators

MS^i**:** Minimum of Standard deviation of objective function of i^{th} problem obtained through all competing operators

R^i**:** Relative errors of objective function of i^{th} problem for all competing operators

MR^i**:** Minimum of relative error of objective function of i^{th} problem obtained through all competing operators.

Where k_1, k_2 and k_3 are the weights of aforementioned criteria, such that $k_1 + k_2 + k_3 = 1$ for all $0 \leq k_1, k_2, k_3 \leq 1$. Following the [49], the weights are assigned as:

$$case(a): \quad k_1 = w,\ k_2 = k_3 = (1-w)/2,\ 0 \leq w \leq 1$$
$$case(b): \quad k_2 = w,\ k_1 = k_3 = (1-w)/2,\ 0 \leq w \leq 1$$
$$case(c): \quad k_3 = w,\ k_2 = k_3 = (1-w)/2,\ 0 \leq w \leq 1,$$

such that two criteria are given equal weights. For demonstration purposes, we explain in all the PI's Figures with respect to three cases: case (a), the S.D and R.E are given the same weights, whereas in case (b), mean and R.E have the same weights and in the last case (c), same weights have been assigned to mean and S.D.

4.2 Simulation-based results and discussions

This section examines the performance comparisons between the proposed MO-OX and the other most used approaches for permutation based crossover operators. For a better evaluation, we compared the proposed MO-OX to MO-GA, OX, PMX and CX in parallel with three selection and two mutation operators. The selection operators which are used in this study are binary tournament selection (BTS), roulette wheel selection (RWS) and linear rank selection (LRS). The exchange mutation (EM) and inversion mutation (INM) are used for random changes in each generation. So there are six groups of experiments conducted as five crossovers examined with each pair of three selection and two mutation operators.

Table 2 summarizes the results of five competing crossover operators with BTS and EM. The results are compared on the basis of average, S.D and R.E in percentage (%) values. The significant improvements in the results of MO-OX with respect to each other approach are indicated through t-values. The proposed operator indicated less than average values for all six benchmarks with low S.D as well. According to the critical value (t = - 2.00), all computed t-values are less than -2.00 for all six benchmark instances except of one operator for rbg443 instance and one for att532 instance, which is MO-GA. The bold t-test values have shown the significantly improved performance by the proposed operator. The other two values of t-statistic which are not bold (non-significant), but are negative, indicate a better average performance by the proposed operator. In other words, the simulation results found by the MO-OX are statistically significantly better than the solution quality by the other four crossover approaches (MO-GA, OX, PMX and CX). For the given setting of experiment, the relevant graphs of PI for each case are shown in Figure 2, where weights are presented on horizontal axis and PI stays at vertical axis. A vividly superior

performance of the proposed MO-OX can be observed as compared to the existing alternatives in all three cases.

As it was mentioned previously, we also have tested the performance of the MO-OX with the same selection method BTS and new mutation approach INM in Table 3. These results are comparable with the Table 2 results based on average, S.D and R.E performance for all used benchmarks. Based on our simulation results, we can say that BTS has given better results with EM other than with INM. MO-OX statistically outperforms (bold t-test values) than the other four crossover approaches for five out of six benchmark instances ($t \leq$ - 2.00). The only two operators (MO-GA and OX) are non-significant with the proposed operator for the benchmark rbg443. For this experiment, the PI for each case are shown in Figure 3, where we also observed a superior performance by the newly proposed operator for all the given benchmarks.

We continue our simulation study to check the performance of the proposed operator along with other crossover methods and different techniques of selection and mutation. Likewise, in Table 4, we tested the performance of the proposed operator MO-OX with the pair of RWS (selection operator) and EM (mutation operator). The simulation results summarize the lower average, S.D and R.E values for all benchmarks. Based on statistical perspective, the MO-OX outperforms (bold t-test values) all the other crossover methods for all six benchmark instances ($t \leq$ - 2.00) except in one case when compared with MO-GA for the problem rbg443. Through the indicator PI, the MO-OX performs better than all other competing operators (see Figure 4).

Table 5 shows the simulation results of the crossover operators but with another mutation approach, i.e. INM, which only differs from Table 4 approaches. The conclusion based on our results indicated that RWS is better performing with EM other than INM. The MO-GA operator is insignificant with the proposed one only for the instance rbg443. Table 6 summarizes the results of all six used crossover operators along with LRS (selection operator) and EM (mutation operator). The experiment results demonstrated that the proposed approach has a significant improvement as compared to all other approaches. The

significance of such improvement is validated through t-test values (t $\leq$ - 2.00). We presented our more simulation results with LRS and INM in Table 7. The results are clearly in the favor of MO-OX in all measurement aspects. MO-GA is insignificant with the proposed one only for the instance ftv170. Finally, with all these changes in the GA's experiments, the PI performs outstanding in the favor of proposed crossover operator (see Figures 5 to 7).

In summary, the simulation results with the six TSP instances show that MO-OX is effective and gave significantly better results throughout the Tables 2 to 7 along with Figures 2 to 7. The main and first purpose to conduct the series of experiments in this study is to measure the performance of the proposed operator with all other competing crossover operators. The second purpose of the simulation is to detect the best against of selection and mutation operators which work better with these crossover operators. The binary tournament (BTS) and exchange mutation (EM) is the best pair to perform better with these crossover operators (Table 2). Finally, we can conclude our final remarks based on all simulation results of Tables 2 to 7 for minimizing the total travel distance with less variations and relative measures, and more performance indexes. These results suggest that the proposed MO-OX achieve significantly better solution quality for all used benchmark instances when compared to the other four alternative crossover approaches.

5 Conclusion

Unlike other meta-heuristic methods, GA exploits natural rules of selection, crossover and mutation, in search of optimal solutions. Inspired by the key role of crossover operator in the scheme of GA, this article suggested a new crossover operator, namely multi-offspring order crossover (MO-OX) operator. This new approach generates four offspring which shows that it can effectively enhance the search ability of the algorithm by producing more possible solutions. The performance of MO-OX is studied by comparing with four highly debated and commonly used existing alternatives (MO-GA, OX, PMX and CX), through rigorous simulation study. Based on various performance evaluation criteria,

i.e. average performance in 30 trails, S.D, R.E, two-tailed t-test and performance index, a superior performance of the proposed operator is observed.

Table 2. Comparison Results of Crossover Operators with BTS and EM

Instance	Optimal	Crossover	Average	R.E	S.D	t-test
		MO-OX	707	1.14	06	-
		MO-GA	721	3.15	15	**-4.67**
dantzig42	699	OX	722	3.29	23	**-3.30**
		PMX	739	5.72	33	**-5.14**
		CX	748	7.01	35	**-6.22**
		MO-OX	7114	3.03	109	-
		MO-GA	7329	6.14	152	**-6.19**
ft53	6905	OX	7387	6.98	197	**-6.52**
		PMX	7412	7.34	211	**-6.76**
		CX	7518	8.88	273	**-7.40**
		MO-OX	2932	6.42	175	-
		MO-GA	3061	11.11	208	**-2.56**
ftv170	2755	OX	3112	12.96	221	**-3.44**
		PMX	3088	12.09	198	**-3.18**
		CX	3191	15.83	277	**-4.26**
		MO-OX	1997	2.41	28	-
		MO-GA	2071	6.21	70	**-5.29**
brg180	1950	OX	2059	5.59	62	**-4.91**
		PMX	2053	5.28	72	**-3.90**
		CX	2087	7.03	95	**-4.89**
		MO-OX	3518	29.34	317	-
		MO-GA	3714	36.54	414	**-2.02**
rbg443	2720	OX	3609	32.68	343	-1.05
		PMX	3683	35.40	309	**-2.01**
		CX	3802	39.78	449	**-2.78**
		MO-OX	28962	4.61	466	-
		MO-GA	29271	5.72	573	**-2.25**
att532	27686	OX	29152	5.30	522	-1.46
		PMX	29436	6.32	487	**-5.97**
		CX	29629	7.02	627	**-4.60**

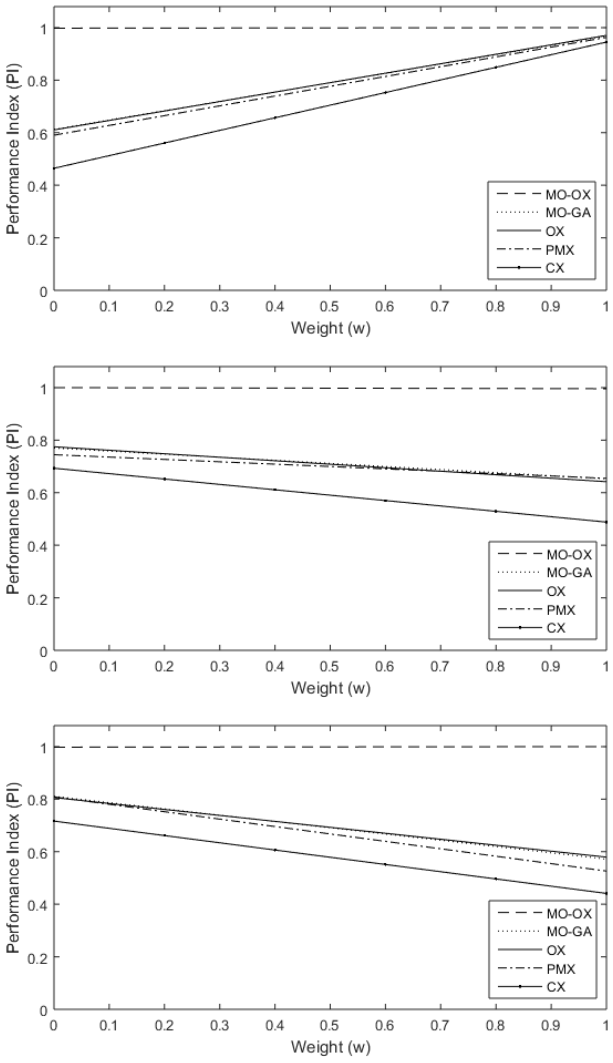

Figure 2. The display of performance index (PI) when: (a) $k_1 = w$ and $k_2 = k_3 = \frac{1-w}{2}$; (b) $k_2 = w$ and $k_1 = k_3 = \frac{1-w}{2}$; and (c) $k_3 = w$ and $k_1 = k_2 = \frac{1-w}{2}$

Table 3. Comparison Results of Crossover Operators with BTS and INM

Instance	Optimal	Crossover	Average	R.E	S.D	t-test
		MO-OX	718	2.72	14	-
		MO-GA	743	6.29	24	**-4.85**
dantzig42	699	OX	751	7.44	31	**-5.22**
		PMX	744	6.44	28	**-4.47**
		CX	759	8.58	33	**-6.16**
		MO-OX	7209	4.40	187	-
		MO-GA	7394	7.08	208	**-3.56**
ft53	6905	OX	7406	7.26	273	**-3.21**
		PMX	7493	8.52	298	**-4.35**
		CX	7527	9.01	330	**-4.51**
		MO-OX	2951	7.11	192	-
		MO-GA	3097	12.41	179	**-3.00**
ftv170	2755	OX	3176	15.28	253	**-3.81**
		PMX	3140	13.97	244	**-3.28**
		CX	3231	17.28	291	**-4.33**
		MO-OX	2018	3.49	54	-
		MO-GA	2131	9.28	101	**-5.31**
brg180	1950	OX	2099	7.64	88	**-4.22**
		PMX	2152	10.36	112	**-5.80**
		CX	2190	12.31	131	**-6.54**
		MO-OX	3573	31.36	350	-
		MO-GA	3648	34.12	395	-0.77
rbg443	2720	OX	3741	37.54	402	-1.70
		PMX	3788	39.26	374	**-2.26**
		CX	3856	41.76	471	**-2.60**
		MO-OX	29623	7.00	499	-
		MO-GA	29931	8.11	548	**-2.24**
att532	27686	OX	30111	8.76	573	**-3.46**
		PMX	30749	11.06	732	**-6.84**
		CX	31008	12.00	987	**-6.74**

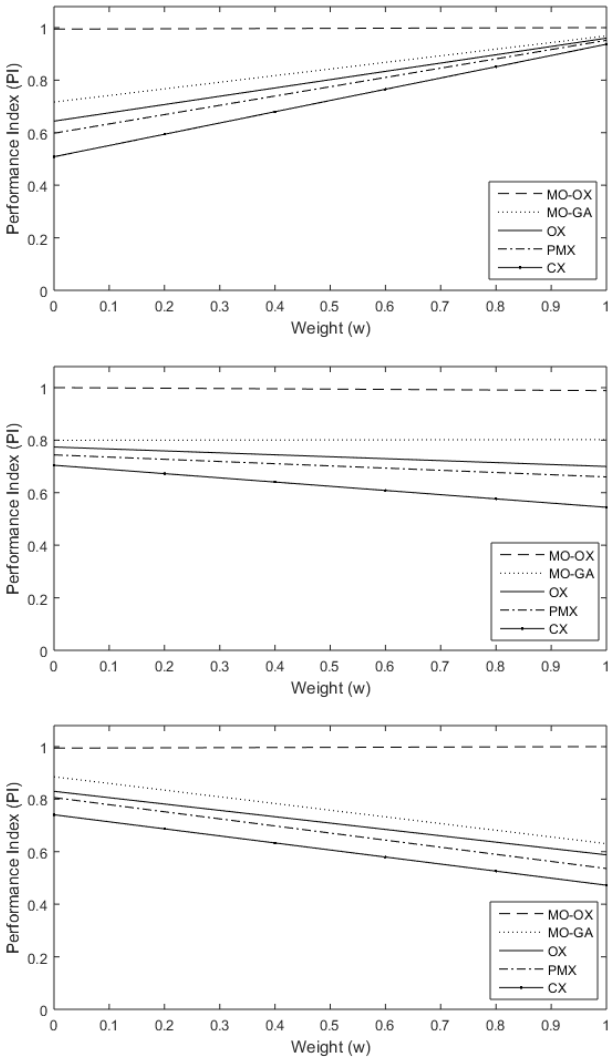

Figure 3. The display of performance index (PI) when: (a) $k_1 = w$ and $k_2 = k_3 = \frac{1-w}{2}$; (b) $k_2 = w$ and $k_1 = k_3 = \frac{1-w}{2}$; and (c) $k_3 = w$ and $k_1 = k_2 = \frac{1-w}{2}$

Table 4. Comparison Results of Crossover Operators with RWS and EM

Instance	Optimal	Crossover	Average	R.E	S.D	t-test
		MO-OX	710	1.57	08	-
		MO-GA	727	4.01	18	**-4.65**
dantzig42	699	OX	718	2.72	14	**-2.67**
		PMX	731	4.58	29	**-3.76**
		CX	756	8.15	40	**-6.07**
		MO-OX	7142	3.43	122	-
		MO-GA	7388	6.99	149	**-6.88**
ft53	6905	OX	7408	7.28	231	**-5.48**
		PMX	7453	7.94	260	**-5.83**
		CX	7485	8.40	279	**-6.07**
		MO-OX	2937	6.61	223	-
		MO-GA	3096	12.38	217	**-2.75**
ftv170	2755	OX	3153	14.45	263	**-3.37**
		PMX	3164	14.85	251	**-3.64**
		CX	3172	15.14	318	**-3.26**
		MO-OX	2009	3.03	057	-
		MO-GA	2121	8.77	103	**-5.12**
brg180	1950	OX	2095	7.44	084	**-4.56**
		PMX	2140	9.74	099	**-6.18**
		CX	2179	11.74	111	**-7.34**
		MO-OX	3550	30.51	374	-
		MO-GA	3693	35.77	452	-1.31
rbg443	2720	OX	3761	38.27	412	**-2.04**
		PMX	3742	37.57	384	-1.93
		CX	3783	39.08	477	**-2.07**
		MO-OX	29311	5.87	776	-
		MO-GA	29894	7.98	707	**-2.99**
att532	27686	OX	30040	12.48	689	**-3.78**
		PMX	29993	8.33	887	**-3.12**
		CX	30301	13.06	990	**-4.24**

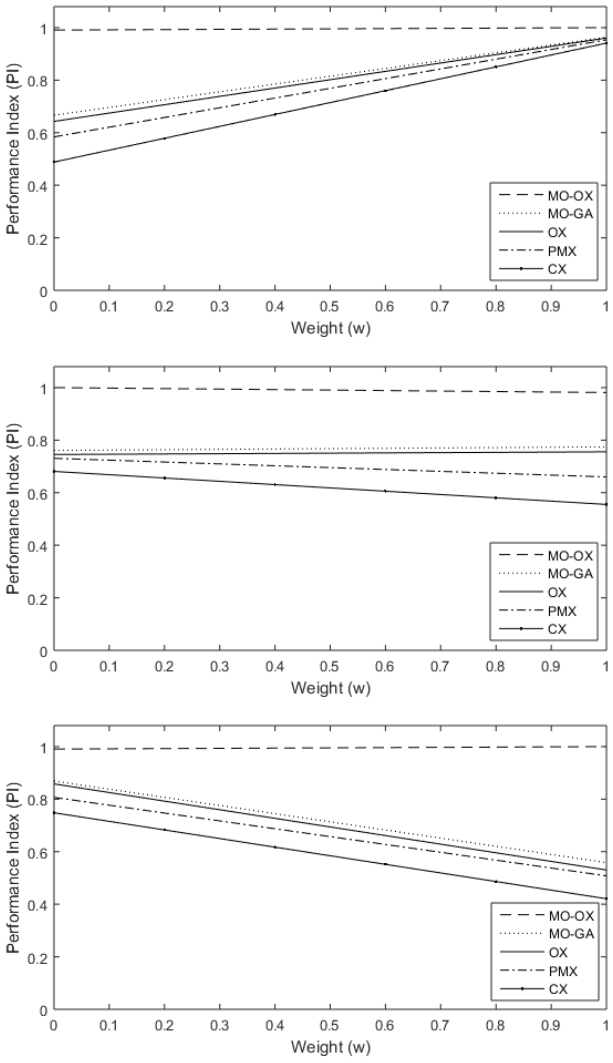

Figure 4. The display of performance index (PI) when: (a) $k_1 = w$ and $k_2 = k_3 = \frac{1-w}{2}$; (b) $k_2 = w$ and $k_1 = k_3 = \frac{1-w}{2}$; and (c) $k_3 = w$ and $k_1 = k_2 = \frac{1-w}{2}$

Table 5. Comparison Results of Crossover Operators with RWS and INM

Instance	Optimal	Crossover	Average	R.E	S.D	t-test
		MO-OX	719	2.86	12	-
		MO-GA	735	5.15	20	**-3.69**
dantzig42	699	OX	728	4.15	19	**-2.16**
		PMX	749	7.15	38	**-4.05**
		CX	744	6.44	31	**-4.05**
		MO-OX	7304	5.78	167	-
		MO-GA	7474	8.24	202	**-3.49**
ft53	6905	OX	7525	8.98	253	**-3.93**
		PMX	7582	9.80	288	**-4.50**
		CX	7483	8.37	312	**-2.72**
		MO-OX	3071	11.47	211	-
		MO-GA	3233	17.35	264	**-2.58**
ftv170	2755	OX	3210	16.52	283	**-2.12**
		PMX	3303	19.89	246	**-3.85**
		CX	3346	21.45	298	**-4.06**
		MO-OX	2024	3.79	069	-
		MO-GA	2132	9.33	113	**-4.39**
brg180	1950	OX	2168	11.18	142	**-4.91**
		PMX	2185	12.05	130	**-5.89**
		CX	2262	16.00	128	**-8.81**
		MO-OX	3591	32.02	383	-
		MO-GA	3689	35.63	456	-0.89
rbg443	2720	OX	3794	39.49	375	**-2.04**
		PMX	3799	39.67	394	**-2.04**
		CX	3851	41.58	459	**-2.34**
		MO-OX	29654	7.11	865	-
		MO-GA	30753	11.08	954	**-4.60**
att532	27686	OX	31631	14.25	876	**-8.64**
		PMX	31892	15.19	989	**-9.17**
		CX	31504	13.79	981	**-7.62**

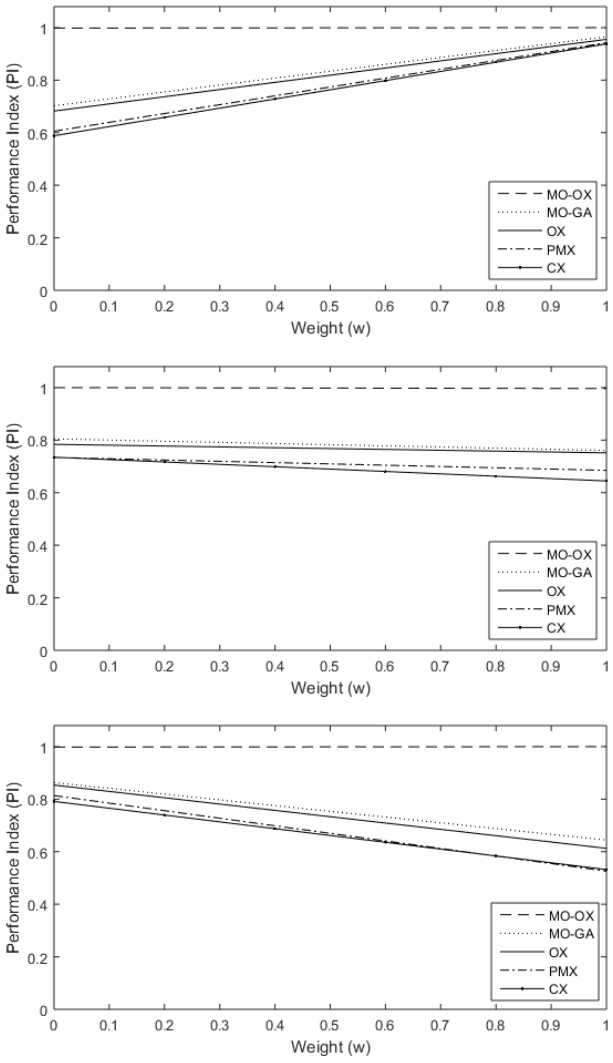

Figure 5. The display of performance index (PI) when: (a) $k_1 = w$ and $k_2 = k_3 = \frac{1-w}{2}$; (b) $k_2 = w$ and $k_1 = k_3 = \frac{1-w}{2}$; and (c) $k_3 = w$ and $k_1 = k_2 = \frac{1-w}{2}$

Table 6. Comparison Results of Crossover Operators with LRS and EM

Instance	Optimal	Crossover	Average	R.E	S.D	t-test
		MO-OX	725	3.72	18	-
		MO-GA	741	6.01	29	**-2.52**
dantzig42	699	OX	753	7.73	35	**-3.83**
		PMX	754	7.86	39	**-3.64**
		CX	760	8.73	52	**-3.43**
		MO-OX	7217	4.52	131	-
		MO-GA	7376	6.82	183	**-3.80**
ft53	6905	OX	7371	6.75	214	**-3.31**
		PMX	7439	7.73	241	**-4.36**
		CX	7502	8.65	298	**-4.71**
		MO-OX	2953	7.19	169	-
		MO-GA	3084	11.94	219	**-2.55**
ftv170	2755	OX	3059	11.03	204	**-2.15**
		PMX	3104	12.67	217	**-2.96**
		CX	3214	16.66	306	**-4.02**
		MO-OX	2001	2.62	32	-
		MO-GA	2093	7.33	65	**-6.84**
brg180	1950	OX	2114	8.41	79	**-7.14**
		PMX	2177	11.64	97	**-9.28**
		CX	2197	12.67	113	**-8.99**
		MO-OX	3573	31.36	392	-
		MO-GA	3794	39.49	416	**-2.08**
rbg443	2720	OX	3716	36.62	388	-1.40
		PMX	3791	39.38	405	**-2.08**
		CX	3865	42.10	483	**-2.53**
		MO-OX	29782	7.57	864	-
		MO-GA	30764	11.12	968	**-4.08**
att532	27686	OX	31081	12.26	892	**-5.63**
		PMX	30749	11.06	979	**-3.99**
		CX	31138	12.47	785	**-6.26**

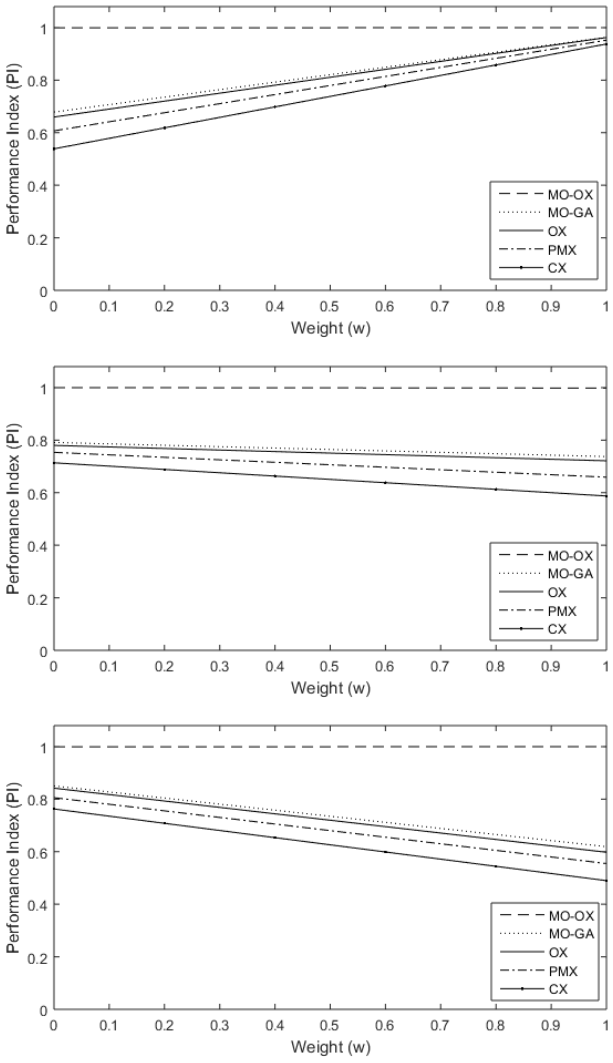

Figure 6. The display of performance index (PI) when: (a) $k_1 = w$ and $k_2 = k_3 = \frac{1-w}{2}$; (b) $k_2 = w$ and $k_1 = k_3 = \frac{1-w}{2}$; and (c) $k_3 = w$ and $k_1 = k_2 = \frac{1-w}{2}$

Table 7. Comparison Results of Crossover Operators with LRS and INM

Instance	Optimal	Crossover	Average	R.E	S.D	t-test
		MO-OX	735	5.15	20	-
		MO-GA	749	7.15	27	**-2.24**
dantzig42	699	OX	759	8.58	33	**-3.35**
		PMX	772	10.44	47	**-3.90**
		CX	770	10.16	59	**-3.03**
		MO-OX	7184	4.04	138	-
		MO-GA	7279	5.42	162	**-2.40**
ft53	6905	OX	7387	6.98	227	**-4.12**
		PMX	7421	7.47	249	**-4.48**
		CX	7490	8.47	213	**-6.49**
		MO-OX	2968	7.73	171	-
		MO-GA	3004	9.04	192	-0.75
ftv170	2755	OX	3143	14.08	246	**-3.15**
		PMX	3175	15.25	217	**-4.03**
		CX	3185	15.61	278	**-3.58**
		MO-OX	2012	3.18	36	-
		MO-GA	2077	6.51	77	**-4.12**
brg180	1950	OX	2114	8.41	86	**-5.89**
		PMX	2210	13.33	123	**-8.32**
		CX	2233	14.51	116	**-7.04**
		MO-OX	3547	30.40	443	-
		MO-GA	3796	39.56	487	**-2.04**
rbg443	2720	OX	3828	40.74	513	**-2.23**
		PMX	3841	41.21	519	**-2.32**
		CX	3854	41.69	468	**-2.57**
		MO-OX	30313	9.49	795	-
		MO-GA	30975	11.88	876	**-3.01**
att532	27686	OX	30844	11.41	922	**-2.35**
		PMX	31759	14.71	985	**-6.15**
		CX	31627	14.24	878	**-5.97**

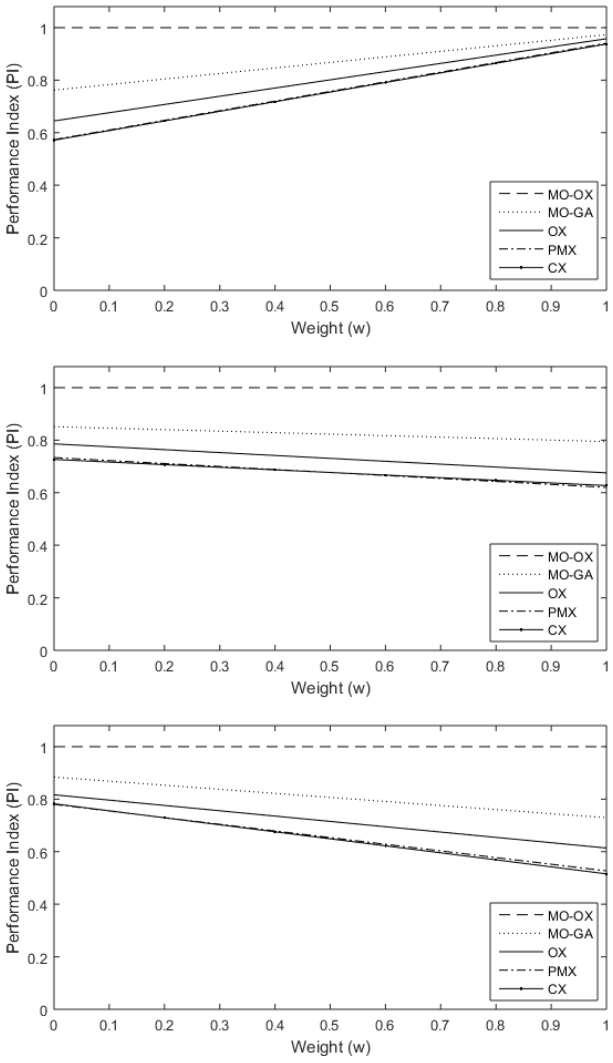

Figure 7. The display of performance index (PI) when: (a) $k_1 = w$ and $k_2 = k_3 = \frac{1-w}{2}$; (b) $k_2 = w$ and $k_1 = k_3 = \frac{1-w}{2}$; and (c) $k_3 = w$ and $k_1 = k_2 = \frac{1-w}{2}$

Data availability

The data used to support the findings of this manuscript are taken from the website of TSPLIB (https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/).

Disclosure statement

The author of this article declares that there is no conflicts of interest regarding the publication of this article.

References

[1] A. Philip, A. A. Taofiki, and O. Kehinde, "A genetic algorithm for solving traveling salesman problem," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 2, no. 1, pp. 26–29, 2011.

[2] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the traveling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129–170, 1999.

[3] M. R, Garey and D. S. Johnson, *Computers and Intractability – A Guide to NP-completeness,* San Francisco: W.H. Freeman and Company, 1979.

[4] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization,* Prentice Hall Englewood Cliffs, 1982.

[5] S. Singh and E. A. Lodhi, "Study of variation in tsp using genetic algorithm and its operator comparison," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, pp. 264–267, 2013.

[6] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[7] M. A. Mohammed, M. K. A. Ghani, R. I. Hamed, S. A. Mostafa, M. S. Ahmad, and D. A. Ibrahim, "Solving vehicle routing problem by using improved genetic algorithm for optimal solution," *Journal of Computational Science*, vol. 21, pp. 255–262, 2017.

[8] F. Alizadeh, R. M. Karp, L. A. Newberg, and D. K. Weisser, "Physical mapping of chromosomes: A combinatorial problem in molecular biology," *Algorithmica*, vol. 13, no. (1-2), pp. 52–76, 1995.

[9] C. Korostensky and G. H. Gonnet, "Using traveling salesman problem algorithms for evolutionary tree construction," *Bioinformatics*, vol. 16, no. 7, pp. 619–627, 2000.

[10] B. D. Corwin and A. O. Esogbue, "Two machine flow shop scheduling problems with sequence dependent setup times: A dynamic programming approach," *Naval Research Logistics (NRL)*, vol. 21, no. 3, pp. 515–524, 1974.

[11] G. Finke. "A two-commodity network flow approach to the traveling salesman problem," *Congresses Numeration*, vol. 41, pp. 167–178, 1984.

[12] P. Miliotis, "Using cutting planes to solve the symmetric traveling salesman problem," *Mathematical Programming*, vol. 15, no. 1, pp. 177–188, 1978.

[13] M. Cunkas and M. Y. Ozsaglam, "A comparative study on particle swarm optimization and genetic algorithms for traveling salesman problems," *Cybernetics and Systems: An International Journal*, vol. 40, no. 6, pp. 490–507, 2009.

[14] C. H. Song, K. Lee, and W. D. Lee, "Extended simulated annealing for augmented tsp and multi-salesmen tsp," in *Proceedings of the International Joint Conference on Neural Networks, 2003, IEEE*, vol. 3, 2003, pp. 2340–2343.

[15] V. Maniezzo and M. Roffilli, "Very strongly constrained problems: an ant colony optimization approach," *Cybernetics and Systems: An International Journal*, vol. 39, no. 4, pp. 395–424, 2008.

[16] S. Bhide, N. John, and M. R. Kabuka, "A boolean neural network approach for the traveling salesman problem," *IEEE Transactions on Computers*, vol. 42, no. 10, pp. 1271–1278, 1993.

[17] Y. He, Y. Qiu, G. Liu, and K. Lei, "A parallel adaptive tabu search approach for traveling salesman problems," in *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*, IEEE, 2005, pp. 796–801.

[18] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *European Journal of Operational Research*, vol. 140, no. 3, pp. 606–617, 2002.

[19] M. Bhattacharyya and A. K. Bandyopadhyay, "Comparative study of some solution methods for traveling salesman problem using genetic algorithms," *Cybernetics and Systems*, vol. 40, no. 1, pp. 1–24, 2008.

[20] Z. H. Ahmed, "Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator," *International Journal of Biometrics & Bioinformatics (IJBB)*, vol. 3, no. 6, pp. 96–105, 2010.

[21] A. Hussain, Y. S. Muhammad, M. N. Sajid, I. Hussain, A. M. Shoukry, and S. Gani, "Genetic algorithm for traveling salesman problem with modified cycle crossover operator," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1–7, 2017.

[22] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, 2nd ed., MIT press, 1992.

[23] C. Darwin, *On the Origin of Species by Means of Natural Selection Or the Preservation of Favored Races in the Struggle for Life*, H. Milford; Oxford University Press, 1859.

[24] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[25] D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in *Proceedings of an international conference on genetic algorithms and their applications*, Hillsdale, NJ: Lawrence Erlbaum, vol. 154, 1985, pp. 154–159.

[26] I. C. Choi, S. I. Kim, and H. S. Kim, "A genetic algorithm with a mixed region search for the asymmetric traveling salesman prob-

lem," *Computers & Operations Research*, vol. 30, no. 5, pp. 773–786, 2003.

[27] Z. Ursani, D. Essam, D. Cornforth, and R. Stocker, "Localized genetic algorithm for vehicle routing problem with time windows," *Applied Soft Computing*, vol. 11, no. 8, pp. 5375–5390, 2011.

[28] H. Muhlenbein, "Parallel genetic algorithms, population genetics and combinatorial optimization," in *Workshop on Parallel Processing: Logic, Organization, and Technology*, Springer, 1989, pp. 398–406.

[29] J. E. Baker, "Adaptive selection methods for genetic algorithms," in *Proceedings of an International Conference on Genetic Algorithms and their applications*, Hillsdale, New Jersey, 1985, pp. 101–111.

[30] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of Genetic Algorithms* (vol. 1), Elsevier, 1991, pp. 69–93.

[31] J. Wang, O. K. Ersoy, M. He, and F. Wang, "Multi-offspring genetic algorithm and its application to the traveling salesman problem," *Applied Soft Computing*, vol. 43, pp. 415–423, 2016.

[32] A. M. Aibinu, H. B. Salau, N. A. Rahman, M. N. Nwohu, and C. M. Akachukwu, "A novel clustering based genetic algorithm for route optimization," *Engineering Science and Technology, an International Journal*, vol. 19, no. 4, pp. 2022–2034, 2016.

[33] L. M. A. Drummond, L. S. Ochi, and D. S. Vianna, "An asynchronous parallel metaheuristic for the period vehicle routing problem," *Future Generation Computer Systems*, vol. 17, no. 4, pp. 379–386, 2001.

[34] K. Ganesh and T. T. Narendran, "Cloves: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up," *European Journal of Operational Research*, vol. 178, no. 3, pp. 699–717, 2007.

[35] M. M. S. Haghighi, M. H. Zahedi, and M. Ghazizadeh, "A multi level priority clustering ga based approach for solving heterogeneous vehicle routing problem (pcgvrp)," in *Innovations and Ad-*

vances in Computer Sciences and Engineering, Springer, 2010, pp. 331–335.

[36] L. Davis, "Applying adaptive algorithms to epistatic domains," in *IJCAI'85 Proceedings of the 9th international joint conference on Artificial intelligence*, vol. 1, pp. 162–164, 1985.

[37] I. M. Oliver, D. J. Smith, and J. R. C. Holland, "Study of permutation crossover operators on the traveling salesman problem," in *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms* (July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA), Hillsdale, NJ: L. Erlhaum Associates, 1987, pp. 224–230.

[38] W. Banzhaf, "The 'molecular' traveling salesman," *Biological Cybernetics*, vol. 64, no. 1, pp. 7–14, 1990.

[39] D. B. Fogel, "An evolutionary approach to the traveling salesman problem," *Biological Cybernetics*, vol. 60, no. 2, pp. 139–144, 1988.

[40] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, New York: Springer, 1996.

[41] D. B. Fogel, "Applying evolutionary programming to selected traveling salesman problems," *Cybernetics and Systems*, vol. 24, no. 1, pp. 27–36, 1993.

[42] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[43] J. Berger and M. Barkaoui, "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows," *Computers & Operations Research*, vol. 31, no. 12, pp. 2037–2053, 2004.

[44] M. Skok, D. Skrlec, and S. Krajcar, "The genetic algorithm method for multiple depot capacitated vehicle routing problem solving," in *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000, Proc. Fourth International Conference on, vol. 2*, IEEE, 2000, pp. 520–526.

[45] K. Q. Zhu, "A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows," in *Tools with Artificial Intelligence, 2003. Proc. 15th IEEE International Conference on*, IEEE, 2003, pp. 176–183.

[46] X. Mou, D. l. Xie, and W. Yan, "Research based on genetic algorithm traveling sealer problem of trajectory optimization," *Journal of System Simulation*, vol. 25, pp. 86–89, 2013.

[47] G. Reinelt, "Tsplib http://www. iwr. uni-heidelberg. de/groups/comopt/software". [Online]. Available: *TSPLIB95*, 1995.

[48] Bharti, "Controlled random search technique and their applications," PhD thesis, Department of Mathematics, University of Roorkee, India, 1994.

[49] C. Mohan and H. T. Nguyen, "A controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems," *Computational Optimization and Applications*, vol. 14, no. 1, pp. 103–132, 1999.

[50] K. Deep and M. Thakur, "A new crossover operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 895–911, 2007.

[51] M. Thakur, "A new genetic algorithm for global optimization of multimodal continuous functions," *Journal of Computational Science*, vol. 5, no. 2, pp. 298–311, 2014.

Received October 21, 2019

Abid Hussain
Department of Statistics, Quaid-i-Azam University, Islamabad, Pakistan.
E–mail: abid0100@gmail.com

Salman A. Cheema
School of Mathematical and Physical Sciences, University of Newcastle, Australia
E–mail: saqvn.cheema@gmail.com

Computer Science Journal of Moldova, vol.**27**, no.3(81), 2019

The concept of personal learning pathway for Intelligent Tutoring System GeoMe

Caftanatov Olesea

Abstract

This paper presents a study of designing personal learning pathways for our intelligent tutoring system "GeoMe". The purpose of the study is to define specific requirements for our application and conceptualize the workflow for personal learning pathways.

Keywords: Learning pathway, intelligent tutoring system, space repetition, forgetting curve, Leitner system.

1 Introduction

Digital culture is increasingly applied in e-learning; it also contributes to improve the educational process by adapting to the student's interests, capabilities and knowledge. Nowadays there are many different kinds of educational software for students based on adaptive learning, personalized learning or even personal learning pathway (PLP). Learning pathway can be described as a route, taken by a pupil through a range of e-learning activities, which allows learners to get new skills and build knowledge progressively. Clement [1] defines a learning pathway as "The sequence of intermediate steps from preconceptions to target model form what Scott (1991) and Niedderer and Goldberg (1995) have called a learning pathway. For any particular topic, such a pathway would provide both a theory of instruction and a guideline for teachers and curriculum developers."

For students personal learning paths are the best solution, because they can more effectively acquire and retain knowledge and skills that

will help them in real world. However, what about the elementary schoolchildren? For them it is harder to decide which learning model will be the most appropriate and effective. In this case, one of solutions can be designing an intelligent tutoring system (ITS) that will analyze the behavior and preferences of pupils and afterward will automatically recommend a personal learning pathway. So, in this paper we started with general description what ITS's are, followed by a short review of ITS "GeoMe"[2] and continued with philosophy behind of our personal learning pathway.

2 Intelligent Tutoring Systems

Intelligent Tutoring System is a part of a new breed of instructional computer programs with the aim to provide immediate support one-on-one and personalized feedback to learners. Canfield [3] defines ITS as a system that is able to diagnose and adapt to student's knowledge and skills. According to Dašić et al. [4] ITS intend to support and improve the teaching and learning process in a selected area of knowledge while respecting the individuality of a learner. Along the years, there are many definitions regarding what intelligent tutoring systems are, however, the common point is focused on using Artificial Intelligence (AI) techniques in order to track learner's needs and respond with an appropriate feedback.

Sleeman and Brown [5] coined the term "Intelligent Tutoring Systems" in 1982 in order to describe evolving tutoring systems and to distinguish them from the previous computer-aided instruction (CAI) systems. A little later, in 1988 the first conference dedicated to intelligent tutoring systems took place, where conceptions of ITS's were consolidated. However, the efforts in ITS first began with the development of what was called Intelligent Computer-Aided Instruction (ICAI) by Carbonell in 1970 [6], so he is called the *"father"* of intelligent systems for teaching and learning.

Carbonell's system was named SCHOLAR, whose goal was to communicate information regarding geography of South America to learners and to review their knowledge by maintaining a mixed-initiative dialogue with learners in a rather comfortable subset of English. It was

Carbonell's belief that the semantic net representation of the knowledge base used in this project was close to the internal knowledge structure of humans.

Research in AI and Cognitive Psychology fueled the new principles of intelligent tutoring systems. Thus, with time they evolved from very primitive form of ICAI into very progressive form, with significant development in their user interfaces. By using AI techniques, intelligent tutoring systems are now being used for a broad range of tasks, although, in a limited way.

For instance, some of the major tasks are: selecting appropriate teaching strategies; generating and solving problems; tracking learner's behaviour and progress; storing and retrieving data; recommending learning materials; diagnosing student's misconception; detecting learner's mood; offering immediate feedback and of course, carrying out a natural language dialogue with learners.

3 Overview of ITS "GeoMe"

GeoMe stands for "geometry for me". It is an ITS designed to help pupils in learning geometry by personalizing their learning paths. Geometry proving theorem is known to be very challenging for students to learn. Thus, almost all ITS proposed for geometry are dedicated to learning proof-writing with constructions, for instance: "Advanced Geometry Tutor" [7], "Advanced Geometry Proof Tutor" [8], ANGLE [9], AgentGeom [10], Geometry Explanation Tool [11] as new version of Geometry Cognitive Tutor.

Although many of these systems are used to provide supportive problem solving for advanced geometry, we intend to deliver learning material for elementary geometry to elementary schoolchildren. Therefore the knowledge model emphasizes the identification of basic shapes, properties of shapes, the shape's comparison etc.

Generally, ITS can take different views to implement the pedagogical criteria according to its educational scope that generates a classification into *specific* and *generic* ITS. According to [12], an ITS for ***generic domains*** is aimed to provide a framework to design and implement training proposal for multiple educational domains. GeoMe is

an ITS for ***specific domain***, it uses pedagogical criteria suitable for just one specific educational domain. Basically, we intend to develop a tool that will help additionally to assimilate information regarding geometry lessons.

Unfortunately, in process of developing application we got some limitations regarding ***time***: according to [13] pupils of four grade daily could spend only 1 hour doing their homework for all their subjects. Thus, for all math' tasks pupils should spend only 10 minutes. Therefore, the time that a pupil needs to spend in our application should be even less. Taking in account time limitations, we decided to divide the features of our intelligent tutoring system as follows:

- *a session compartment*, where a pupil daily should spend no more than 8 minutes. Generally, we will reserve 3 minutes for theory and 5 for practice, but, in order to not stress the learners we will not add time counter. If some of them will need a few more minutes to finish their tasks, then we'll let them do so;
- *an interactive tool for 3D* visualization of shapes, the goal is to assist the users when needed, see Figure 1;

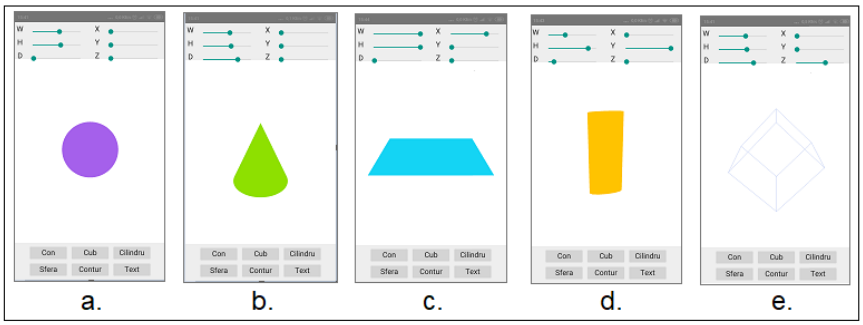

Figure 1. A few examples with interactive changing shapes: a) changing the circle's weight and height; b) changing triangle's depth, c) rotating square on axe X; d) rotating cylinder on axe Y; e) rotating cube's outline on axe Z

- *gamification compartment*, the goal is to help the users to rest while playing;
- *theoretical compartment*, the goal is also to assist the users when needed.

Even if the Ministry of Education, Culture and Research of the Republic of Moldova indicated that, for pupils in grades I-IV, the weekly volume of homework must not exceed 5 hours, we understand that all schoolchildren learn differently. Some of them can finish their tasks in the allocated time, but for the others it may take more efforts and more time.

In this regard, our task is not to overload pupil's homework by using our application, but finding a method that lets users spend less time studying while retaining the same amount of information. Well, one of the solutions can be applying the space repetition technique to session compartment. Regarding the other compartments, they should be designed as a way to rest while navigating in our application. It can be games or useful material that can be used not daily but wherever user wishes.

4 Spaced repetition

In order to make the most of session compartment we analyzed memory retention through spaced repetition. It is a method of reviewing material at systematic intervals. Spaced repetition technique is usually performed with flashcards. An ideal system of spaced repetitions allows user to review the material before it is forgotten, helping to retain information and transfer it from short-term memory to long-term memory.

4.1 Ebbinghaus' forgetting curve

We all know the phenomena when we ever tried to learn something new and been overwhelmed by the task, perhaps some of us succeeded in learning everything only to forget it all the next day. This aspect of human learning has been investigated during more than a century. Hermann Ebbinghaus, a German psychologist, first studied one of the

simple memory models, the exponential forgetting curve [14], in 1885. He identified two critical variables that determine the probability of recalling an item: ***reinforcement***, i.e., repeated exposure to the item, and ***delay***, i.e., time since the item was last reviewed.

According to Ebbinghaus' forgetting curve, there is a strong correlation between time and memory. In fact, forgetting occurs rapidly at first, then it slows down; this process can be seen in the graph from Figure 2.

Figure 2. Ebbinghaus' forgetting curve

Generally, we forget about 60 percent of what we have just processed within the first 20 minutes. Moreover, more than half of memory loss that occurs is within the first hour. Most of material that will be forgotten is done so within the first 8 hours. Thus, the main question is *how to disturb the forgetting process?* Well, the theory goes that if we test ourselves, just as we are about to forget the thing that we have learnt, our brain will hold on to the information for longer, see Figure 3.

Every time we test our new knowledge, our brain will hold on to it for longer and longer. Another important question is *how do we know when to test ourselves when we do not know when we will forget it?* Another theory goes that if we test first time ourselves on what we have learnt, then we can remember it for 5^2 seconds, thus next time we should take the second test between 20-25 seconds and we will

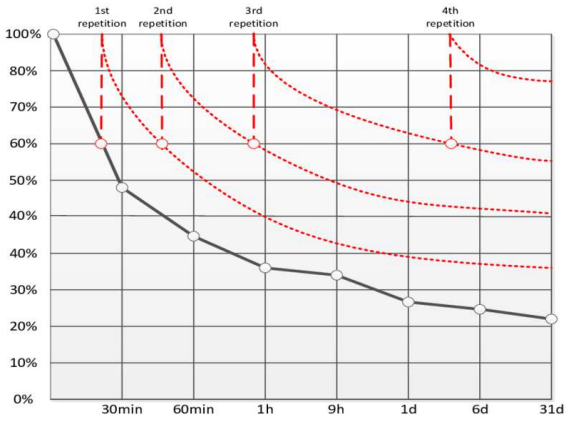

Figure 3. Ebbinghaus' forgetting curve and review cycle

remember it this time for 5^3 seconds. After the tenth time we will not need to test ourselves again for over a year.

Ebbinghaus's publication also includes an equation to approximate his forgetting curve:

$$S = 100 * \frac{1.84}{(log_{10}t)^{1.25} * 1.84}$$

S represents savings and is expressed as percentage. In other words, they are analogous to retention rate. **T** represents time in minutes. Savings of 100 percent would indicate that all items were still known from the first trial.

According to linguist Paul Pimsleur [15] by using audio reviewing for learning language, memory schedule should be as follows: 5 seconds, 225 seconds, 2 minutes, 10 minutes, 1 hour, 5 hours, 1 day, 5 days, 25 days, 4 months, 2 years. However, this approach is limited since the schedule is pre-recorded and cannot adapt to the learner's actual ability Another experiment have shown that if we memorize repeatedly within one hour, we will remember for one day, if we memorize one day later, we will remember for one week [16].

4.2 Leitner system

In our research we focus on one of the simplest and oldest spaced repetition methods, the Leitner system. Leitner [17] proposed a different repetition algorithm intended for use with flashcards. His system is more adaptive than Pimsleur method, since the spacing intervals can increase and decrease depending on students' performance. Figure 4 illustrates a popular variant of this method.

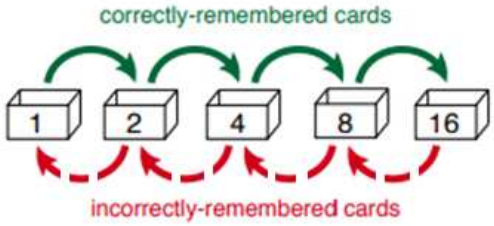

Figure 4. The Leitner System for flashcards

The main idea is to have a few boxes that will correspond to different practice intervals, such us the 1st day, the 2nd day and so on. Initially all cards will be placed in the 1st day box. When student practices, and if he remembers the correct answer, then that flashcard is promoted to the next box, otherwise, it will be demoted. Suppose, the student got the wrong answer from the 8th box, then that flashcard will be demoted to the 4th box.

5 Personal learning pathways

The main idea for our personal learning pathways is to give learners bits of information with different types of styles, repetitively at gradually increasing increments of time. In such way, learners would retain those bits of information for longer time periods each time. We intend to combine Ebbinghaus' memory models to Leitners' system.

Firstly, we have a collection with four learning styles for each task, such as textual format, visual, audio, sensorial. At the starting point, all tasks will be presented in textual format, afterword it will change by each repetition (see Figure 5).

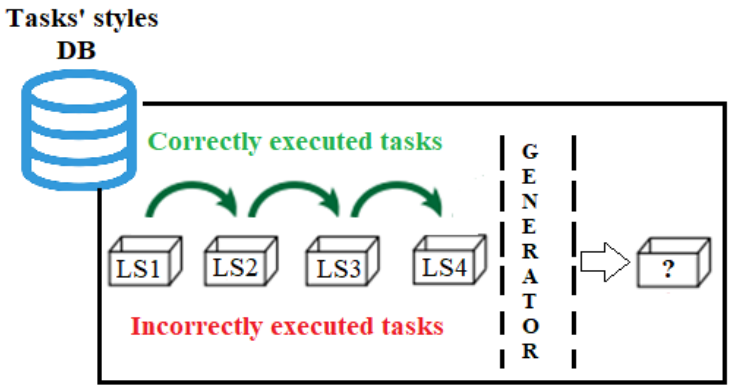

Figure 5. Task's learning style review cycle

It does not matter if previously the learner gave correct or incorrect answer, anyway, the same task at the new repetition will change its learning style. When the learner will walk through all four types of the same task, the generator will choose for the learner the most appropriated style based on statistics.

In such order, the learners will not be boring when repeating the same information. Moreover, studying the same information from different aspects will increase their memory retention.

Regarding schedule repetition, we believe that if the learner executed the task correctly in the 1st day, than he will remember how to solve it at least one more day, so the next time when the learner should repeat the same task should be on the 3rd day. On the 3rd day, if the learner answered correctly again, then the next repetition time should increase with one more day versus the previous one, so it will be on the 5th day.

In the case that the learner gave incorrect answer, the repetition will be right the next day. Taking into account that we have four

learning styles, the schedule repetition for 5 subjects in our learning sessions will be as it is shown in Figure 6.

List of subjects	*Learning sessions review cycle*			
Point & Lines	1	3	6	10
Angles	2	4	7	11
Triangles	3	5	8	12
Squares	4	6	9	13
Circles	5	7	10	14
Learning style	***Textual***	***Audio***	***Visual***	***Sensorial***

Figure 6. Schedule repetition

Another question is *when we will stop repeating the same tasks?* Our theory goes that if the learner executes correctly for the four consecutive sessions, then the learner's memory retention is 100 percent, and application stops repeating the same task. In the case the learner gives incorrect answers after two repetitions, then his memory retention is equal to 50 percent, and application will generate the same task until the learner will get 100 percent. Additionally, in the case, when the learner gives four consecutive incorrect answers, then human expert is involved. All these cases are shown in Figure 7.

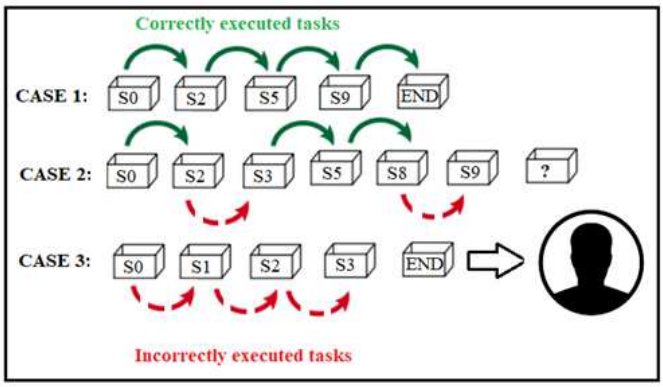

Figure 7. Task's review cycle examples

6 Conclusion

This paper presents a study of designing personal learning pathways for our intelligent tutoring system "GeoMe". Our main idea for designing personal learning pathways consists in adding different learning styles to schedule repetition based on Leitner system and Ebbinghaus' forgetting curve. In such way, we'll design personal learning pathways and we assume that it will adapt to learner's skills. The more correct answer the learners will give, the less tasks they will get daily and vice-versa the less correct answers, the more effort pupils should put in their learning process. Nevertheless, by repeating their lessons the memory retention will definitely increase. Moreover, on each repetition presenting material in different type of learning styles will help pupils to understand deeper their lessons.

References

[1] J. Clement, "Model based learning as a key research area for science education," *Inter. J. of Sci. Educ.*, vol. 22, no. 9, pp. 1041–1053, Sep. 2000.

[2] O. Caftanatov, "A new approach of designing the intelligent tutoring system GeoMe," *in Proc.The fifth Conf. of Mathematical Society of the Rep. of Moldova*, Chisinau, vol. 5, pp. 305–308, Sep. 2019.

[3] W.A. Canfield, "A web-based intelligent tutoring system," *Mathematics and Computer Education*, vol. 35, no. 2, pp. 152–158, 2001.

[4] P. Dašić *et al.*, "A review of intelligent tutoring systems in E-learning," *Fascicle of Management and Tech. Engineering.*, no. 3, pp. 85–90, Dec. 2016.

[5] D. Sleeman and J.S. Brown, "Introduction: Intelligent Tutoring Systems," in *Intelligent Tutoring Systems*, D. Sleeman and J.S. Brown, Eds. New York: Academic Press, pp. 1–11, 1982.

[6] J.R. Carbonell, "AI in CAI: An artificial-intelligence approach to computer-assisted instruction," *IEEE Tran. on Man-Machine Systems*, vol. MMS-11, no. 4, pp. 190–202, Dec. 1970.

[7] N. Matsuda and K. VanLehn, "Advanced Geometry Tutor: An intelligent tutor that teaches proof-writing with construction," in *AIED – 12th Int. Conf. on AI in Educ*, Jul. 2005, pp. 8.

[8] K. Wang and S. Zhendong, "Interactive, Intelligent Tutoring for Auxiliary Construction in Geometry Proofs," *ArXiv Journal*, vol. 1/1722.07254, pp. 11, Nov. 2017.

[9] K.R. Koedinger and J.R. Anderson, "Effective use of intelligent software in high school math classrooms," in *Proceedings of World Conf. on AI in Ed*, VA:AACE, pp. 8, 1993.

[10] P. Cobo, J.M. Fortuny, E. Puertas, and P.R. Richard, "AgentGeom: a multiagent system for pedagogical support in geometric proof problems," *Inter. J. of Comp.for Math. Learning*, pp. 57–79, Apr. 2007. Available: DOI: 10.1007/s10758-007-9111-5.

[11] V. Alven, A. Ogan, O. Popescu, C. Torrey, and K.Koedinger, "Evaluating the effectiveness of a tutorial dialogue system for self-explanation," *ITS2004, LNCS3220*, Springer-Verlag Berlin Heidelberg, pp. 443–454, 2004.

[12] M. Badaracco and L. Martínez, "An Intelligent tutoring system architecture for competency-based learning," *KES 2011, Part II, LNAI 6882*, Springer-Verlag Berlin Heidelberg, pp. 124–133, 2011.

[13] Ministry of Education, Culture and Research of the Republic of Moldova, "Instruction: Homework management in primary, secondary and high school," *Regulation*, pp. 1-26, Aug. 2018. Available on web site: http://particip.gov.md/proiectview.php?l=ro&idd=5622

[14] H. Roediger, "Remembering Ebbinghaus. A review of Memory: A Contribution to Experimental Psychology," *Contemporary Psy-*

chology: A Journal of Reviews, vol. 30, no. 7, pp. 519–523, Jul. 1985.

[15] P. Pimsleur, "A memory schedule," *Modern Language Journal*, vol. 51, no. 2, pp. 73–75, Feb. 1967.

[16] D.R. Bacon and K.A. Stewart, "How fast do students forget what they learn in consumer behavior?: A longitudinal study," *Journal of Marketing Education*, vol. 28, no. 3, pp. 181–192, Dec. 2006.

[17] S. Leitner, "That is how you learn to learn," *Applied Learning Psychology - A Path to Success*, Freiburg im Breisgau, Basel, Wien: Verlag, Herder, 1972. (in German).

Olesea Caftanatov[1,2] Received December 12, 2019

[1] Vladimir Andrunachievici Institute of Mathematics and Computer Science;
5, Academiei street, Chisinau, Republic of Moldova, MD 2028

[2] The State University "Dimitrie Cantemir"
3/2 Academiei Street, Chisinau, Republic of Moldova, MD-2028;

E–mail: olesea.caftanatov@math.md

Computer Science Journal of Moldova, vol.**27**, no.3(81), 2019

Author-Initiated Retraction: "A remark on the weak Turán's Theorem"

Nader Jafari Rad

The paper [1] has one main theorem, namely Theorem 3. In the last line of the theorem's proof it is assumed that the set S^* is independent. This assumption is not necessarily correct, and it is not possible to fix it, since the set S is randomly chosen. This mistake makes the main result of the paper wrong.

At the request of the author (Received October 21, 2019), the article has been retracted.

Editorial Board of Computer Science Journal of Moldova

References

[1] Nader Jafari Rad, "A remark on the weak Turán's Theorem," *Computer Sci. J. Moldova*, vol. 25, no. 3(75), 2017, pp. 256–259.

Nader Jafari Rad

Department of Mathematics, Shahed University
Tehran, Iran
E–mail: n.jafarirad@gmail.com