Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

An Automatic Proof of Euler's Formula

Jun Zhang

Abstract

In this information age, everything is digitalized. The encoding of functions and the automatic proof of functions are important. This paper will discuss the automatic calculation for Taylor expansion coefficients, as an example, it can be applied to prove Euler's formula automatically.

Keywords: function, coefficient, automatic proof.

1 Introduction

The expansion of Taylor series is a very old topic in both pure and applied mathematics that plays a crucial role in both fundamental theory and applications. Computer algebra systems provide an interactive environment to assist in solving many mathematical problems.

One way to define an analytic function $f(z)$ is in terms of its Taylor series expansion at $z = 0$,

$$f(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_n z^n + \cdots$$

Quite a few theorems exist about how to find the coefficient a_n of a general term $a_n z^n$ in the expansion, which we shall denote $[z^n]f(z)$. Under some conditions, we have Taylor's formula [1]:

$$a_n = [z^n]f(z) = \frac{f^{(n)}(0)}{n!}.$$

This is a very nice formula and can be quite useful in finding a specific term such as $[z^3]f(z)$. However, for an arbitrary number n (usually considered to be very large), we cannot use the formula directly to determine $[z^n]f(z)$.

Ravenscroft implemented a Maple package called `genfunc` that can calculate $[z^n]f(z)$ for any rational function $f(z)$ [2]. Rational functions, however, are very well structured and easy to handle. As shown by Ravenscroft, every nontrivial rational generating function $F(z)$ encodes a sequence that is defined by a homogeneous linear recurrence with constant coefficients [3]. So finding $[z^n]f(z)$ reduces to solving a linear homogeneous recurrence with constant coefficients which, in turn, reduces to solving a corresponding polynomial equation.

If $f(z)$ is not a rational function, it is difficult in practice to calculate $[z^n]f(z)$. In many cases, an exact expansion of $[z^n]f(z)$ is impossible to find or too complicated to be of practical value. In such instances, we often have to settle for an asymptotic representation of $[z^n]f(z)$. Sadly and perhaps surprisingly, as Bruno Salvy stated a decade ago, "Current symbolic computation systems generally lack facilities for manipulating asymptotic expansion computations of a form more complex than the first terms of Taylor series or Puiseux expansions (involving fractional powers)[4]". This situation has not changed significantly since then.

This work is to provide an approach to calculate Taylor coefficients of functions, as an example, it can be applied to prove Euler's formula automatically.

2 Laplace Method

Assume that the function f is defined for $0 \leq t < \infty$. We write the Laplace transform as

$$F(s) = \text{Ł}\{f(t)\} = \int_0^\infty e^{-st} f(t)dt.$$

We shall refer to $f(t)$ as the original function and to $F(s)$ as the Laplace transform of the function $f(t)$. We also refer to $f(t)$ as the inverse Laplace transform of $F(s)$. The symbol Ł denotes the Laplace transformation. The function e^{-st} is called the kernel of the transformation. In our work, we think of s as a real variable. If the integral converges for all s greater than some s_0, then $F(s)$ is well defined and we say that the transform exists.

Now, let us look at some examples:

Example. Compute the Laplace transform of $f(t) = e^{2t}$.

$$\begin{aligned}\int_0^\infty e^{-st} f(t)dt &= \int_0^\infty e^{-st} e^{2t} dt = \int_0^\infty e^{-(s-2)t} dt \\ &= \lim_{b\to\infty} \int_0^b e^{-(s-2)t} dt = \lim_{b\to\infty} -\left[\frac{e^{-(s-2)t}}{s-2} \bigg|_0^b \right] \\ &= \lim_{b\to\infty} \left[\frac{1}{s-2} - \frac{e^{-(s-2)b}}{s-2} \right].\end{aligned}$$

This limit exists only when $s > 2$. Hence,

$$\int_0^\infty e^{-st} f(t)dt = \frac{1}{s-2}, \qquad s > 2. \qquad \diamondsuit$$

Now, let us consider the integral

$$\hat{f}(x) = \frac{1}{x} \int_0^\infty e^{-t/x} f(t)dt.$$

This is just the Laplace transform in which the variable x of the generating function has been replaced by its reciprocal.

3 Expansion Theory

We present the main theorem for our work based on Laplace transformation. See [5] for a proof.

Theorem 3.1. If

1. $f(t)$ is bounded and continuous for $0 < t < \infty$,
2. $\hat{f}(x) = \frac{1}{x} \int_0^\infty e^{-t/x} f(t)dt$, and
3. $\hat{f}(x) = \sum_{n=0}^\infty a_n x^n, \qquad 0 < x < \rho,$

then we have

$$f(x) = \sum_{n=0}^{\infty} a_n \frac{x^n}{n!}, \qquad 0 < x < \infty.$$

This theorem can serve as an alternate way to calculate the general term of a Taylor series expansion. Let us look at several examples.

Example. Consider $f(t) = \sin(t)$.

$$\begin{aligned}
\hat{f}(x) &= \frac{1}{x}\int_0^{\infty} e^{-t/x}\sin(t)dt \\
&= -\int_0^{\infty} \sin(t)de^{-t/x} \\
&= -\lim_{b\to\infty}\left[e^{-t/x}\sin(t)\Big|_0^b\right] + \int_0^{\infty} e^{-t/x}\cos(t)dt \\
&= -\lim_{b\to\infty}\left[xe^{-t/x}\cos(t)\Big|_0^b\right] - x\int_0^{\infty} e^{-t/x}\sin(t)dt \\
&= x - x^2\hat{f}(x),
\end{aligned}$$

so we have

$$\hat{f}(x) = \frac{x}{x^2+1} = x - x^3 + x^5 - \cdots, \qquad 0 < x < 1.$$

By Theorem 3.1,

$$f(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots,$$

is the series expansion for $\sin(x)$, as predicted. ♢

4 Automatic Calculation

Based on the above discussion, we can implement a procedure in Maple called "coefficient" as following:

```
with(inttrans);
with(genfunc);
```

```
coefficient := proc (y, x)
   local tem1, tem2;
   tem1 :=laplace(y, x, s);
   tem1 := subs (s = 1/t, tem1);
   tem1 := tem1/t;
   tem2 := rgf_expand(tem1, t, n));
   tem2 := (simplify(tem2))/n!;
   return tem2
end
```

Example 1. Consider $f(t) = \sin(t)$. By applying the above "coefficient" procedure in Maple, we have an answer

$$[t^n]f(t) = \sin(n\pi/2)/n!.$$

Example 2. Consider $f(t) = \cos(at)\sin(bt)$, where a and b are nonzero real constants ($a \neq b$). Apply the "coefficient" procedure in Maple, we get an answer equivalent to

$$\begin{aligned}[t^n]f(t) = {} & \frac{1}{4(a-b)n!}((I(a-b))^n + (-I(a-b))^n) \\ & -\frac{1}{4(a+b)n!}((I(a+b))^n + (-I(a+b))^n).\end{aligned}$$

where I is the imaginary number such that $I^2 = -1$.

Example 3. Consider $f(t) = e^{Ix} - \cos(x) - I\sin(x)$. By applying the above "coefficient" procedure in Maple, we have an answer

$$[t^n]f(t) = 0.$$

Since $f(x)$ is analytic, and all its Taylor's expansion coefficients are 0, we proved the Euler's formula $e^{Ix} = \cos(x) + I\sin(x)$.

5 Conclusion

This method provides a way to calculate the general coefficients of Taylor's expansion. It works for all the functions such that their Laplace transforms are rational. There is a wide range of functions satisfying such a condition, including the examples above, e^x, $\sin^k(z)$, $\cos^k(z)$, etc, where k is an natural number.

More advanced algorithms were developed in [6]. The algorithms developed in [6] can be used to calculate the coefficients for a much wide range of functions beyond rational functions, and return exact solutions. This paper provides an alternative solution with simpler implementation.

References

[1] R. Johnsonbaugh, W. E. Pfaffenberger. *Foundations of mathematical analysis.* Marcel Dekker, 1981.

[2] R.A. Ravenscroft, *Rational generating function applications in Maple,* in Maple V: Mathematics and Its Application, R. Lopez (ed.), Birkhaser: 1994.

[3] R.A. Ravenscroft and E. A. Lamagna. *Symbolic Summation with Generating Functions. Proceedings of the International Symposium on Symbolic and Algebraic Computation,* Association for Computing Machinery, pp.228–233. 1989.

[4] B. Salvy, *Examples of Automatic Asymptotic Expansion,* Technical Report RT-0114, INRIA Rocquencourt, France: 1989.

[5] D.V. Widder. *An Introduction to TRANSFORM THEORY.* Academic Press, 1971.

[6] J. Zhang, *ALGORITHMS FOR SERIES COEFFICIENTS,* Ph. D dissertation. University of Rhode Island, USA, 2001.

J. Zhang,

Received February 10, 2005

Department of Math and Computer Science
Troy University
Troy, AL 36081, USA
E–mail: *zhang@troy.edu*

Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

A generalization of the chromatic polynomial of a cycle

Julian A. Allagan

Abstract

We prove that if an edge of a cycle on n vertices is extended by adding k vertices, then the the chromatic polynomial of such generalized cycle is:

$$P(H_k, \lambda) = (\lambda - 1)^n \sum_{i=0}^{k} \lambda^i + (-1)^n (\lambda - 1).$$

1 Introduction

We consider simple finite graphs and assume that the basic definitions from graph and hypergraph theory (see, for example, [1, 3, 4]) are familiar to the reader.

Proper coloring of a graph $G = (V, \mathcal{E})$, is a mapping $f : V(G) \to \{1, 2, \dots, \lambda\}$ which is defined as an assignment of distinct colors from a finite set of colors $[\lambda]$ to the *vertices* of G in such a way that *adjacent vertices* have different colors. Such notion has been extended in 1966 by P. Erdös and A. Hajnal to the coloring of a hypergraph [2]. Thus, in general case, the *proper coloring* of a hypergraph $H = (V, \mathcal{E})$ is the labelling of the vertices of H in such a way that every *hyperedge* $E \in \mathcal{E}$ has at least two vertices of distinct colors.

The function $P(H, \lambda)$ counts the mappings $f : V(H) \to [\lambda]$ that properly color H using colors from the set $[\lambda] = \{1, 2, \dots, \lambda\}$. Thus, we define *the chromatic polynomial* of a hypergraph H as the number of all proper colorings of H using at most λ colors [3].

Let $C_n = (V, \mathcal{E})$ be a cycle on n vertices, $n \geq 3$, where $V = \{v_1, v_2, \ldots, v_n\}$. Consider an edge $E = \{v_1, v_2\}$ of C_n. We sequentially increase the size of E by adding k pendant vertices (a vertex is called *pendant* if its degree is one) from the set $S_k = \{x_1, x_2, x_3, \ldots, x_k\}$, $k \geq 1$. Notice that E becomes a hyperedge E', containing $k + 2 \geq 3$ vertices. We compute the chromatic polynomial of the obtained hypergraph $H_k = (V \cup S_k, \mathcal{E}')$, where k is the number of pendant vertices added.

2 Proof of the formula

Theorem 1. *The chromatic polynomial of the hypergraph H_k has the following form:*

$$P(H_k, \lambda) = (\lambda - 1)^n \sum_{i=0}^{k} \lambda^i + (-1)^n(\lambda - 1).$$

Proof. Induction on the number of pendant vertices k. Observe that

$$P(H_0, \lambda) = (\lambda-1)^n\lambda^0+(-1)^n(\lambda-1) = (\lambda-1)^n+(-1)^n(\lambda-1) = P(C_n, \lambda)$$

what is the chromatic polynomial of any cycle on n vertices, see [4, p.229].

The idea of proof consists in the following procedure: we apply to H_k, $k \geq 1$, the connection-contraction algorithm which is a special case of the splitting-contraction algorithm for mixed hypergraphs, see [3, p.30]. In any proper coloring of H, the vertices v_1, and x_1 either have different colors or have the same color. In the first case, we connect x_1 and v_1 by an edge; in the second case, we contract the edge $\{x_1, v_1\}$ and in this way identify the vertices x_1 and v_1. After removing of an exterior hyperedge containing vertices x_1, v_1, we obtain two graphs and some isolated vertices and compute the chromatic polynomial as a sum of two chromatic polynomials of the respective graphs.

Consider the case $k = 1$. We obtain that

$$P(H_1, \lambda) = P(T_{n+1}, \lambda) + P(H_0, \lambda),$$

where T_n is a tree on n vertices; it is well known that

$$P(T_n, \lambda) = \lambda(\lambda - 1)^{n-1}.$$

Since $P(H_0, \lambda) = P(C_n, \lambda) = (\lambda - 1)^n + (-1)^n(\lambda - 1)$ we obtain

$$P(H_1, \lambda) = \lambda(\lambda - 1)^n + (\lambda - 1)^n + (-1)^n(\lambda - 1) =$$

$$= (\lambda - 1)^n(\lambda + 1) + (-1)^n(\lambda - 1).$$

Consider the case $k = 2$. Using the same procedure we obtain a tree, a cycle and one isolated vertex. Therefore

$$P(H_2, \lambda) = P(T_{n+1}, \lambda)\lambda + P(H_1, \lambda).$$

Notice that the chromatic polynomial of the independent vertex set $P(S_k, \lambda) = \lambda^k$ because each isolated vertex can be assigned λ colors. Using $P(H_1, \lambda) = (\lambda - 1)^n(\lambda + 1) + (-1)^n(\lambda - 1)$ we establish the following equality:

$$P(H_2, \lambda) = \lambda(\lambda - 1)^n\lambda + (\lambda - 1)^n(\lambda + 1) + (-1)^n(\lambda - 1) =$$

$$= (\lambda - 1)^n(\lambda^2 + \lambda + 1) + (-1)^n(\lambda - 1).$$

Let us assume that our formula for the chromatic polynomial of $P(H_j, \lambda)$ is true for any number $j \geq 1$ of pendant vertices. We now prove that

$$P(H_{j+1}, \lambda) = (\lambda - 1)^n(\lambda^{j+1} + \lambda^j + \ldots + \lambda^1 + \lambda^0) + (-1)^n(\lambda - 1).$$

Consider $j + 1$ number of pendant vertices from the set $S_{j+1} = \{x_1, x_2, \ldots, x_j, x_{j+1}\}$ added to the edge $E = \{v_1, v_2\}$ of the cycle $C_n = (V, \mathcal{E})$. The edge $E = \{v_1, v_2\}$ becomes a hyperedge $E' = \{v_1, v_2, x_1, x_2, \ldots, x_j, x_{j+1}\} \in \mathcal{E}'$ of the new graph $H_{j+1} = (V \cup S_{j+1}, \mathcal{E}')$. Applying the algorithm as described in the previous cases to H_{j+1} yields the following chromatic polynomial equality:

$$P(H_{j+1}, \lambda) = P(T_{n+1}, \lambda)P(S_j, \lambda) + P(H_j, \lambda).$$

By the induction hypothesis,

$$P(H_j,\lambda) = (\lambda-1)^n(\lambda^j+\lambda^{j-1}+\ldots+\lambda^1+\lambda^0)+(-1)^n(\lambda-1);$$

also, $P(S_j,\lambda)=\lambda^j$. Therefore the following equality holds:

$$P(H_{j+1},\lambda)=$$
$$=\lambda(\lambda-1)^n\lambda^j+(\lambda-1)^n(\lambda^j+\lambda^{j-1}+\ldots+\lambda^1+\lambda^0)+(-1)^n(\lambda-1)=$$
$$=(\lambda-1)^n(\lambda^{j+1}+\lambda^j+\ldots+\lambda^1+\lambda^0)+(-1)^n(\lambda-1).$$

Consequently,

$$P(H_k,\lambda)=(\lambda-1)^n\sum_{i=0}^{k}\lambda^i+(-1)^n(\lambda-1)$$

holds for any number $k\geq 1$ of pendant vertices added to an edge of C_n.

□

Acknowledgements

The author expresses his sincere gratitude to Professor Vitaly Voloshin for multilateral help.

References

[1] C. Berge. *Graphs and Hypergraphs.* North Holland, 1989.

[2] P. Erdös, A. Hajnal. *On chromatic number of graphs and set-systems.* Acta Math. Acad. Sci. Hung., N.17, 1966, pp.61–99.

[3] V.Voloshin. *Coloring Mixed Hypergraphs: Theory, Algorithms and Applications.* American Mathematical Society 2002.

[4] D. West. *Introduction to Graph Theory.* Prentice Hall, 2001.

J.A. Allagan,

Received February 10, 2005

Department of Mathematics & Physics,
Troy University
Troy, Alabama
E–mail: *aallagan@hotmail.com*

Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

Nash equilibria sets in mixed extension of $2 \times 2 \times 2$ games

Valeriu Ungureanu, Ana Botnari

Abstract

We describe the Nash equilibria set as an intersection of graphs of players' best responses. The problem of Nash equilibria set construction for three-person extended $2 \times 2 \times 2$ games is studied.

Mathematics Subject Classification 2000: 90C05, 90C10, 90C29, 90C31.

Keywords and phrases: Noncooperative game; Nash equilibrium, Nash equilibria set, graph of best responses.

1 Introduction and preliminary results

The problem of the Nash equilibria set construction is rarely encountered in literature. There are diverse explanations of this fact. The main reason is the complexity of this problem [1].

We consider a noncooperative game:

$$\Gamma = \langle N, \{X_i\}_{i \in N}, \{f_i(x)\}_{i \in N} \rangle,$$

where $N = \{1, 2, ..., n\}$ is a set of players, X_i is a set of strategies of player $i \in N$ and $f_i : X \to R$ is a player's $i \in N$ payoff function defined on the Cartesian product $X = \times_{i \in N} X_i$. Elements of X are named outcomes of the game (situations or strategy profiles).

The outcome $x^* \in X$ of the game is the Nash equilibrium [3] (shortly NE) of Γ if

$$f_i(x_i, x^*_{-i}) \leq f_i(x^*_i, x^*_{-i}), \forall x_i \in X_i, \ \forall i \in N,$$

where

$$x^*_{-i} = (x^*_1, x^*_2, ..., x^*_{i-1}, x^*_{i+1}, ..., x^*_n),$$

$$x^*_{-i} \in X_{-i} = X_1 \times X_2 \times ... \times X_{i-1} \times X_{i+1} \times ... \times X_n,$$

$$(x_i, x^*_{-i}) = (x^*_1, x^*_2, ..., x^*_{i-1}, x_i, x^*_{i+1}, ..., x^*_n) \in X.$$

There are diverse alternative formulations of a Nash equilibrium [1] as:

- a fixed point of the best response correspondence;
- a fixed point of a function;
- a solution of a non-linear complementarity problem;
- a solution of a stationary point problem;
- a minimum of a function on a polytope;
- a semi-algebraic set.

We study the Nash equilibria set as an intersection of graphs of players' best responses [4], i.e. intersection of the sets:

$$Gr_i = \{(x_i, x_{-i}) \in X : x_{-i} \in X_{-i}, x_i \in \text{Arg} \max_{x_i \in X_i} f_i(x_i, x_{-i})\},\ i \in N.$$

Theorem 1. *The outcome $x^* \in X$ is a Nash equilibrium if and only if $x^* \in \bigcap\limits_{i \in N} Gr_i$.*

The proof follows from the definition of the Nash equilibrium.

Corollary. $NE(\Gamma) = \bigcap\limits_{i \in N} Gr_i$.

If all strategy sets $X_i, i \in N$, are finite, then a mixed extension of Γ is

$$\Gamma_m = \langle M_i, f^*_i(\mu), i \in N \rangle,$$

where

$$f_i^*(\mu) = \sum_{x \in X} f_i(x)\mu_1(x_1)\mu_2(x_2)\ldots\mu_n(x_n),$$

$$\mu = (\mu_1, \mu_2, ..., \mu_n) \in M = \times_{i \in N} M_i,$$

M_i is a set of mixed strategies of the player $i \in N$.

Theorem 2. *If X is a finite set, then the set $NE(\Gamma_m)$ is a nonempty compact subset of the M. Moreover, it contains the set $NE(\Gamma)$:*

$$NE(\Gamma) \subset NE(\Gamma_m) \neq \emptyset.$$

One of the simplest solvable problems of the NE set determination is the similar problem in the mixed extension of two-person 2×2 game [1, 2, 4]. In this paper the class partition of all three-person $2 \times 2 \times 2$ games is considered and the NE set is determined for mixed extension of the games of each class.

2 Main results

Consider a three-person matrix game Γ with matrices:

$$A = (a_{ijk}),\ B = (b_{ijk}),\ C = (c_{ijk}),\ \ i = \overline{1,2},\ j = \overline{1,2},\ k = \overline{1,2}.$$

The game $\Gamma_m = \langle \{1, 2, 3\}; X, Y, Z;\ f_1, f_2, f_3 \rangle$ is the mixed extension of Γ, where

$$X = \{\mathbf{x} = (x_1, x_2) \in R^2 : x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0\},$$

$$Y = \{\mathbf{y} = (y_1, y_2) \in R^2 : y_1 + y_2 = 1, y_1 \geq 0, y_2 \geq 0\},$$

$$Z = \{\mathbf{z} = (z_1, z_2) \in R^2 : z_1 + z_2 = 1, z_1 \geq 0, z_2 \geq 0\},$$

$$f_1(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \textstyle\sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} a_{ijk} x_i y_j z_k,$$

$$f_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \textstyle\sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} b_{ijk} x_i y_j z_k,$$

$$f_3(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \textstyle\sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} c_{ijk} x_i y_j z_k.$$

By substitutions:

$$x_1 = x,\ x_2 = 1 - x,\ x \in [0, 1];$$

$$y_1 = y,\ y_2 = 1 - y,\ y \in [0, 1];$$

$$z_1 = z,\ z_2 = 1 - z,\ z \in [0, 1],$$

the game Γ_m is reduced to the equivalent normal form game:

$$\Gamma'_m = \langle\{1, 2, 3\}; [0, 1], [0, 1], [0, 1]; \varphi_1, \varphi_2, \varphi_3\rangle.$$

where

$$\begin{array}{l} \varphi_1(x, y, z) = \\ ((a_{111} - a_{211})yz + (a_{112} - a_{212})y(1 - z) + (a_{121} - a_{221})(1 - y)z + \\ (a_{122} - a_{222})(1 - y)(1 - z))x + \\ ((a_{211} - a_{221})z + (a_{212} - a_{222})(1 - z))y + (a_{221} - a_{222})z + a_{222}; \end{array}$$

$$\begin{array}{l} \varphi_2(x, y, z) = \\ ((b_{111} - b_{121})xz + (b_{112} - b_{122})x(1 - z) + (b_{211} - b_{221})(1 - x)z + \\ (b_{212} - b_{222})(1 - x)(1 - z))y + \\ ((b_{121} - b_{221})z + (b_{122} - b_{222})(1 - z))x + (b_{221} - b_{222})z + b_{222}; \end{array}$$

$$\begin{array}{l} \varphi_3(x, y, z) = \\ ((c_{111} - c_{112})xy + (c_{121} - c_{122})x(1 - y) + (c_{211} - c_{212})(1 - x)y + \\ (c_{221} - c_{222})(1 - x)(1 - y))z + \\ ((c_{112} - c_{212})y + (c_{122} - c_{222})(1 - y))x + (c_{212} - c_{222})y + c_{222}. \end{array}$$

Thus, Γ_m is reduced to the game Γ'_m on the unit cube.

If $NE(\Gamma'_m)$ is known, then it is easy to construct the set $NE(\Gamma_m)$.

Basing on properties of strategies of each player of the initial pure strategies game Γ, diverse classes of games are considered and for every class the set of $NE(\Gamma'_m)$ is determined.

Proposition 1. *If all players have equivalent strategies, then*

$$NE(\Gamma'_m) = [0, 1]^3.$$

Remark. *In the case, considered in proposition 1, players have the following linear payoff functions:*

$\varphi_1(x,y,z) = ((a_{211} - a_{221})z + (a_{212} - a_{222})(1-z))y + (a_{221} - a_{222})z + a_{222}$,

$\varphi_2(x,y,z) = ((b_{121} - b_{221})z + (b_{122} - b_{222})(1-z))x + (b_{221} - b_{222})z + b_{222}$,

$\varphi_3(x,y,z) = ((c_{112} - c_{212})y + (c_{122} - c_{222})(1-y))x + (c_{212} - c_{222})y + c_{222}$.

Every player doesn't influence on his payoff function, but his strategy is essential for payoff values of the rest of the players.

Proposition 2. *If all the players have dominant strategies in* Γ, *then* $NE(\Gamma'_m)$ *contains only one point:*

$$NE(\Gamma'_m) = \begin{cases} (0,0,0) & \textit{if strategies (2,2,2) are dominant;} \\ (0,0,1) & \textit{if strategies (2,2,1) are dominant;} \\ (0,1,0) & \textit{if strategies (2,1,2) are dominant;} \\ (0,1,1) & \textit{if strategies (2,1,1) are dominant;} \\ (1,0,0) & \textit{if strategies (1,2,2) are dominant;} \\ (1,0,1) & \textit{if strategies (1,2,1) are dominant;} \\ (1,1,0) & \textit{if strategies (1,1,2) are dominant;} \\ (1,1,1) & \textit{if strategies (1,1,1) are dominant.} \end{cases}$$

Proof. It is easy to observe that graphs coincide with facets of unite cube.

For first player:

$$\operatorname{Arg}\max_{x\in[0,1]} \varphi_1(x,y,z) = \begin{cases} \{1\} & \text{if the 1-st strategy is dominant in } \Gamma, \\ \{0\} & \text{if the 2-nd strategy is dominant in } \Gamma, \end{cases}$$

$\forall (y,z) \in [0,1]^2$. Hence,

$$Gr_1 = \begin{cases} 1 \times [0,1] \times [0,1] & \text{if the 1-st strategy is dominant,} \\ 0 \times [0,1] \times [0,1] & \text{if the 2-nd strategy is dominant.} \end{cases}$$

For second player:

$$\operatorname{Arg}\max_{y\in[0,1]} \varphi_2(x,y,z) = \begin{cases} \{1\} & \text{if the 1-st strategy is dominant in } \Gamma, \\ \{0\} & \text{if the 2-nd strategy is dominant in } \Gamma, \end{cases}$$

$\forall (x,z) \in [0,1]^2$. So,

$$Gr_2 = \begin{cases} [0,1]{\times}1 \times [0,1] & \text{if the 1-st strategy is dominant,} \\ [0,1]{\times}0 \times [0,1] & \text{if the 2-nd strategy is dominant.} \end{cases}$$

For third player:

$$\operatorname{Arg}\max_{z\in[0,1]} \varphi_3(x,y,z) = \begin{cases} \{1\} & \text{if the 1-st strategy is dominant in } \Gamma, \\ \{0\} & \text{if the 2-nd strategy is dominant in } \Gamma, \end{cases}$$

$\forall (x,y) \in [0,1]^2$. Hence,

$$Gr_3 = \begin{cases} [0,1]{\times}[0,1] \times 1 & \text{if the 1-st strategy is dominant,} \\ [0,1]{\times}[0,1] \times 0 & \text{if the 2-nd strategy is dominant.} \end{cases}$$

Consequently, the NE set contains only one vertex of unit cube. □

Proposition 3. *If the first and the second players have dominant strategies and the third player has incomparable strategies, then:*

$$NE(\Gamma'_m) = \begin{cases} (1,1,0) & \textit{if (1,1,}\cdot\textit{) are dominant and } c_{111} < c_{112}, \\ (1,1,1) & \textit{if (1,1,}\cdot\textit{) are dominant and } c_{111} > c_{112}, \\ 1\times 1\times [0,1] & \textit{if (1,1,}\cdot\textit{) are dominant and } c_{111} = c_{112}, \\ (0,0,0) & \textit{if (2,2,}\cdot\textit{) are dominant and } c_{221} < c_{222}, \\ (0,0,1) & \textit{if (2,2,}\cdot\textit{) are dominant and } c_{221} > c_{222}, \\ 0\times 0\times [0,1] & \textit{if (2,2,}\cdot\textit{) are dominant and } c_{221} = c_{222}, \\ (1,0,0) & \textit{if (1,2,}\cdot\textit{) are dominant and } c_{121} < c_{122}, \\ (1,0,1) & \textit{if (1,2,}\cdot\textit{) are dominant and } c_{121} > c_{122}, \\ 1\times 0\times [0,1] & \textit{if (1,2,}\cdot\textit{) are dominant and } c_{121} = c_{122}, \\ (0,1,0) & \textit{if (2,1,}\cdot\textit{) are dominant and } c_{211} < c_{212}, \\ (0,1,1) & \textit{if (2,1,}\cdot\textit{) are dominant and } c_{211} > c_{212}, \\ 0\times 1\times [0,1] & \textit{if (2,1,}\cdot\textit{) are dominant and } c_{211} = c_{212}. \end{cases}$$

Similarly the NE set can be constructed in two other possible cases:

- *players 1 and 3 have dominant strategies, and player 2 has incomparable strategies;*

- *players 2 and 3 have dominant strategies, and player 1 has incomparable strategies.*

So, the NE set is either one vertex of a unit cube or one edge of this cube.

Proposition 4. *If the first and the second players have dominant strategies and the third one has equivalent strategies, then*

$$NE(\Gamma'_m) = \begin{cases} 1 \times 1 \times [0,1] & \textit{if (1,1,}\cdot\textit{) are dominant,} \\ 0 \times 0 \times [0,1] & \textit{if (2,2,}\cdot\textit{) are dominant,} \\ 1 \times 0 \times [0,1] & \textit{if (1,2,}\cdot\textit{) are dominant,} \\ 0 \times 1 \times [0,1] & \textit{if (2,1,}\cdot\textit{) are dominant.} \end{cases}$$

Similarly the NE set can be constructed in the following cases:

- *players 1 and 3 have dominant strategies, and player 2 has equivalent strategies;*

- *players 2 and 3 have dominant strategies, and player 1 has equivalent strategies.*

Thus, the NE set is an edge of unit cube.

Proposition 5. *If the first and the second players have equivalent strategies, and the third player has dominant strategy, then*

$$NE(\Gamma'_m) = \begin{cases} [0,1]\times[0,1]\times 1 & \textit{if the 1-st strategy is dominant,} \\ [0,1]\times[0,1]\times 0 & \textit{if the 2-nd strategy is dominant.} \end{cases}$$

Similarly the NE set can be constructed in the following cases:

- *players 1 and 3 have equivalent strategies, and player 2 has dominant strategy;*

- *players 2 and 3 have equivalent strategies, and player 1 has dominant strategy.*

In such a way, the NE set is a facet of a unit cube.

Proposition 6. *If the first player has equivalent strategies, the second player has dominant strategy and the third player has incomparable strategies, then*

$$NE(\Gamma'_m) = Gr_3,$$

where

$$Gr_3 = [0,1]^3 \cap \left\{ \begin{array}{ll} \left\{ \begin{array}{l} [0; -\frac{\gamma_2}{\gamma_1}) \times 1 \times 0 \cup \\ -\frac{\gamma_2}{\gamma_1} \times 1 \times [0,1] \cup \\ (-\frac{\gamma_2}{\gamma_1}; 1] \times 1 \times 1 \end{array} \right. & \text{if } \gamma_1 > 0, \\ \left\{ \begin{array}{l} [0; -\frac{\gamma_2}{\gamma_1}) \times 1 \times 1 \cup \\ -\frac{\gamma_2}{\gamma_1} \times 1 \times [0,1] \cup \\ (-\frac{\gamma_2}{\gamma_1}; 1] \times 1 \times 0 \end{array} \right. & \text{if } \gamma_1 < 0, \\ [0{,}1] \times 1 \times 0 & \text{if } \gamma_1 = 0,\ \gamma_2 < 0, \\ [0{,}1] \times 1 \times 1 & \text{if } \gamma_1 = 0,\ \gamma_2 > 0, \\ [0{,}1] \times 1 \times [0,1] & \text{if } \gamma_1 = \gamma_2 = 0, \end{array} \right.$$

$\gamma_1 = c_{111} - c_{112} - c_{211} + c_{212}$, $\gamma_2 = c_{211} - c_{212}$, $\gamma_3 = c_{112} - c_{212}$, $\gamma_4 = c_{212}$

if the 1-st strategy of the second player is dominant,
and

$$Gr_3 = [0,1]^3 \cap \left\{ \begin{array}{ll} \left\{ \begin{array}{l} [0; -\frac{\gamma_6}{\gamma_5}) \times 0 \times 0 \cup \\ -\frac{\gamma_6}{\gamma_5} \times 0 \times [0,1] \cup \\ (-\frac{\gamma_6}{\gamma_5}; 1] \times 0 \times 1 \end{array} \right. & \text{if } \gamma_5 > 0, \\ \left\{ \begin{array}{l} [0; -\frac{\gamma_6}{\gamma_5}) \times 0 \times 1 \cup \\ -\frac{\gamma_6}{\gamma_5} \times 0 \times [0,1] \cup \\ (-\frac{\gamma_6}{\gamma_5}; 1] \times 0 \times 0 \end{array} \right. & \text{if } \gamma_5 < 0, \\ [0{,}1] \times 0 \times 0 & \text{if } \gamma_5 = 0,\ \gamma_6 < 0, \\ [0{,}1] \times 0 \times 1 & \text{if } \gamma_5 = 0,\ \gamma_6 > 0, \\ [0{,}1] \times 0 \times [0,1] & \text{if } \gamma_5 = \gamma_6 = 0, \end{array} \right.$$

$\gamma_5 = c_{121} - c_{122} - c_{221} + c_{222}$, $\gamma_6 = c_{221} - c_{222}$, $\gamma_7 = c_{122} - c_{222}$, $\gamma_8 = c_{222}$

if the 2-nd strategy of the second player is dominant.

Proof. If the 1-st strategy of the second player is dominant, then

$$\varphi_3(x,y,z) = (x(c_{111}-c_{112})+(1-x)(c_{211}-c_{212}))z+(c_{112}-c_{212})x+c_{212} =$$

$$= (\gamma_1 x + \gamma_2)z + \gamma_3 x + \gamma_4.$$

From this the truth of proposition follows evidently.

If the 2-nd strategy of the second player is dominant, then

$$\varphi_3(x,y,z) = (x(c_{121}-c_{122})+(1-x)(c_{221}-c_{222}))z+(c_{122}-c_{222})x+c_{222} =$$

$$= (\gamma_5 x + \gamma_6)z + \gamma_7 x + \gamma_8.$$

From this the truth of the second part of the proposition results. □

Similarly the NE set can be constructed in the following cases:

- *player 1 has equivalent strategies, player 3 has dominant strategy, and player 2 has incomparable strategies;*
- *player 2 has equivalent strategies, player 1 has dominant strategy, and player 3 has incomparable strategies;*
- *player 2 has equivalent strategies, player 3 has dominant strategy, and player 1 has incomparable strategies;*
- *player 3 has equivalent strategies, player 1 has dominant strategy, and player 2 has incomparable strategies;*
- *player 3 has equivalent strategies, player 2 has dominant strategy, and player 1 has incomparable strategies.*

Proposition 7. *If the first and the second players have incomparable strategies and the third player has dominant strategy, then*

$$NE(\Gamma'_m) = Gr_1 \cap Gr_2,$$

where

$$Gr_1 = [0,1]^3 \cap \left\{ \begin{array}{ll} \left\{ \begin{array}{l} 0{\times}[0; -\frac{\alpha_2}{\alpha_1}) \times 1 \cup \\ {[0,1]}{\times} -\frac{\alpha_2}{\alpha_1} \times 1 \cup \\ 1 \times (-\frac{\alpha_2}{\alpha_1}; 1] \times 1 \end{array} \right. & \text{\textit{if} } \alpha_1 > 0, \\ \left\{ \begin{array}{l} 1 \times [0; -\frac{\alpha_2}{\alpha_1}) \times 1 \cup \\ {[0,1]}{\times} -\frac{\alpha_2}{\alpha_1} \times 1 \cup \\ 0 \times (-\frac{\alpha_2}{\alpha_1}; 1] \times 1 \end{array} \right. & \text{\textit{if} } \alpha_1 < 0, \\ 0 \times [0,1] \times 1 & \text{\textit{if} } \alpha_1 = 0,\ \alpha_2 < 0, \\ 1 \times [0,1] \times 1 & \text{\textit{if} } \alpha_1 = 0,\ \alpha_2 > 0, \\ {[0,1]}{\times}[0,1] \times 1 & \text{\textit{if} } \alpha_1 = \alpha_2 = 0, \end{array} \right.$$

$$Gr_2 = [0,1]^3 \cap \left\{ \begin{array}{ll} \left\{ \begin{array}{l} {[0; -\frac{\beta_2}{\beta_1})} \times 0 \times 1 \cup \\ -\frac{\beta_2}{\beta_1} \times [0,1] \times 1 \cup \\ (-\frac{\beta_2}{\beta_1}; 1] \times 1 \times 1 \end{array} \right. & \text{\textit{if} } \beta_1 > 0, \\ \left\{ \begin{array}{l} {[0; -\frac{\beta_2}{\beta_1})} \times 1 \times 1 \cup \\ -\frac{\beta_2}{\beta_1} \times [0,1] \times 1 \cup \\ (-\frac{\beta_2}{\beta_1}; 1] \times 0 \times 1 \end{array} \right. & \text{\textit{if} } \beta_1 < 0, \\ {[0,1]}{\times}0 \times 1 & \text{\textit{if} } \beta_1 = 0,\ \beta_2 < 0, \\ {[0,1]}{\times}1 \times 1 & \text{\textit{if} } \beta_1 = 0,\ \beta_2 > 0, \\ {[0,1]}{\times}[0,1] \times 1 & \text{\textit{if} } \beta_1 = \beta_2 = 0, \end{array} \right.$$

$\alpha_1 = a_{111} - a_{211} - a_{121} + a_{221}, \alpha_2 = a_{121} - a_{221}, \alpha_3 = a_{211} - a_{221}, \alpha_4 = a_{221}$,
$\beta_1 = b_{111} - b_{121} - b_{211} + b_{221}, \beta_2 = b_{211} - b_{221}, \beta_3 = b_{121} - b_{221}, \beta_4 = b_{221}$
if the 1-st strategy of the third player is dominant,
and

$$Gr_1 = [0,1]^3 \cap \left\{ \begin{array}{ll} \left\{ \begin{array}{l} 0 \times [0; -\frac{\alpha_6}{\alpha_5}) \times 0 \cup \\ {[0,1]}{\times} -\frac{\alpha_6}{\alpha_5} \times 0 \cup \\ 1 \times (-\frac{\alpha_6}{\alpha_5}; 1] \times 0 \end{array} \right. & \text{\textit{if} } \alpha_5 > 0, \\ \left\{ \begin{array}{l} 1 \times [0; -\frac{\alpha_6}{\alpha_5}) \times 0 \cup \\ {[0,1]}{\times} -\frac{\alpha_6}{\alpha_5} \times 0 \cup \\ 0 \times (-\frac{\alpha_6}{\alpha_5}; 1] \times 0 \end{array} \right. & \text{\textit{if} } \alpha_5 < 0, \\ 0 \times [0,1] \times 1 & \text{\textit{if} } \alpha_5 = 0,\ \alpha_6 < 0, \\ 1 \times [0,1] \times 0 & \text{\textit{if} } \alpha_5 = 0,\ \alpha_6 > 0, \\ {[0,1]}{\times}[0,1] \times 0 & \text{\textit{if} } \alpha_5 = \alpha_6 = 0, \end{array} \right.$$

$$Gr_2 = [0,1]^3 \cap \begin{cases} \begin{cases} [0; -\frac{\beta_6}{\beta_5}) \times 0 \times 0 \cup \\ -\frac{\beta_6}{\beta_5} \times [0,1] \times 0 \cup \\ (-\frac{\beta_6}{\beta_5}; 1] \times 1 \times 0 \end{cases} & \textit{if } \beta_5 > 0, \\ \begin{cases} [0; -\frac{\beta_6}{\beta_5}) \times 1 \times 0 \cup \\ -\frac{\beta_6}{\beta_5} \times [0,1] \times 0 \cup \\ (-\frac{\beta_6}{\beta_5}; 1] \times 0 \times 0 \end{cases} & \textit{if } \beta_5 < 0, \\ [0,1] \times 0 \times 0 & \textit{if } \beta_5 = 0,\ \beta_6 < 0, \\ [0,1] \times 1 \times 0 & \textit{if } \beta_5 = 0,\ \beta_6 > 0, \\ [0,1] \times [0,1] \times 0 & \textit{if } \beta_5 = \beta_6 = 0, \end{cases}$$

$\alpha_5 = a_{112} - a_{212} - a_{122} + a_{222}, \alpha_6 = a_{122} - a_{222}, \alpha_7 = a_{212} - a_{222}, \alpha_8 = a_{222},$

$\beta_5 = b_{112} - b_{122} - b_{212} + b_{222}, \beta_6 = b_{212} - b_{222}, \beta_7 = b_{122} - b_{222}, \beta_8 = b_{222}$
if the 2-nd strategy of the third player is dominant.

Proof. If the 1-st strategy of the third player is dominant, then

$$\varphi_1(x,y,z) = (y(a_{111} - a_{211}) + (1-y)(a_{121} - a_{221}))x + (a_{211} - a_{221})y + a_{221} =$$

$$= (\alpha_1 y + \alpha_2)x + \alpha_3 y + \alpha_4,$$

$$\varphi_2(x,y,z) = (x(b_{111} - b_{121}) + (1-x)(b_{211} - b_{221}))y + (b_{121} - b_{221})x + b_{221} =$$

$$= (\beta_1 x + \beta_2)y + \beta_3 x + \beta_4.$$

From the above the truth of the proposition follows.

If the 2-nd strategy of the third player is dominant, then

$$\varphi_1(x,y,z) = (y(a_{112} - a_{212}) + (1-y)(a_{122} - a_{222}))x + (a_{212} - a_{222})y + a_{222} =$$

$$(\alpha_5 y + \alpha_6)x + \alpha_7 y + \alpha_8,$$

$$\varphi_2(x,y,z) = (x(b_{112} - b_{122}) + (1-x)(b_{212} - b_{222}))y + (b_{122} - b_{222})x + b_{222} =$$

$$(\beta_5 x + \beta_6)y + \beta_7 x + \beta_8.$$

From this the truth of the second part of the proposition results. □

Similarly the NE set can be constructed in the following cases:

- *players 1 and 3 have incomparable strategies, player 2 has dominant strategy;*

- *players 2 and 3 have incomparable strategies, player 1 has dominant strategy.*

Proposition 8. *If the first and the second players have equivalent strategies and the third player has incomparable strategies, then*

$$NE(\Gamma'_m) = Gr_3,$$

where

$$Gr_3 = [0,1]^3 \cap \{X_< \times Y_< \times 0 \cup X_= \times Y_= \times [0,1] \cup X_> \times Y_> \times 1\},$$

$$X_< \times Y_< = \{(x,y) : x \in [0,1],\, y \in [0,1],\, \gamma_1 xy + \gamma_2 x + \gamma_3 y + \gamma_4 < 0\},$$

$$X_= \times Y_= = \{(x,y) : x \in [0,1],\, y \in [0,1],\, \gamma_1 xy + \gamma_2 x + \gamma_3 y + \gamma_4 = 0\},$$

$$X_> \times Y_> = \{(x,y) : x \in [0,1],\, y \in [0,1],\, \gamma_1 xy + \gamma_2 x + \gamma_3 y + \gamma_4 > 0\}.$$

$$\gamma_1 = c_{111} - c_{112} - c_{121} + c_{122} - c_{211} + c_{212} + c_{221} - c_{222},$$

$$\gamma_2 = c_{121} - c_{122} - c_{221} + c_{222}, \gamma_3 = c_{211} - c_{212} - c_{221} + c_{222}, \gamma_4 = c_{221} - c_{222}.$$

Proof. The truth of the proposition results from the following representation of the cost function:

$$\varphi_3(x,y,z) = (xy(c_{111} - c_{112}) + x(1-y)(c_{121} - c_{122}) + (1-x)y(c_{211} - c_{212}) +$$

$$+(1-x)(1-y)(c_{221} - c_{222}))z + (y(c_{112} - c_{212}) + (1-y)(c_{122} - c_{222}))x +$$

$$+(c_{212} - c_{222})y + c_{222} =$$

$$= (\gamma_1 xy + \gamma_2 x + \gamma_3 y + \gamma_4)z + \gamma_5 xy + \gamma_6 x + \gamma_7 y + \gamma_8,$$

where

$$\gamma_5 = c_{112} - c_{212} - c_{122} + c_{222}, \gamma_6 = c_{122} - c_{222}, \gamma_7 = c_{212} - c_{222}, \gamma_8 = c_{222}.$$

□

Similarly the NE set can be constructed in the following cases:

- *players 1 and 3 have equivalent strategies, player 2 has incomparable strategies;*

- *players 2 and 3 have equivalent strategies, player 1 has incomparable strategies.*

Proposition 9. *If the first and the second players have incomparable strategies and the third player has equivalent strategies, then*

$$NE(\Gamma'_m) = Gr_1 \cap Gr_2,$$

where

$$Gr_1 = [0,1]^3 \cap \{0 \times Y_< \times Z_< \cup [0,1] \times Y_= \times Z_= \cup 1 \times Y_> \times Z_>\},$$

$$Gr_2 = [0,1]^3 \cap \{X_< \times 0 \times Z_< \cup X_= \times [0,1] \times Z_= \cup X_> \times 1 \times Z_>\},$$

$$Y_< \times Z_< = \{(y,z) \,:\, y \in [0,1],\, z \in [0,1],\, \alpha_1 yz + \alpha_2 y + \alpha_3 z + \alpha_4 < 0\},$$

$$Y_= \times Z_= = \{(y,z) \,:\, y \in [0,1],\, z \in [0,1],\, \alpha_1 yz + \alpha_2 y + \alpha_3 z + \alpha_4 = 0\},$$

$$Y_> \times Z_> = \{(y,z) \,:\, y \in [0,1],\, z \in [0,1],\, \alpha_1 yz + \alpha_2 y + \alpha_3 z + \alpha_4 > 0\},$$

$$X_< \times Z_< = \{(x,z) \,:\, x \in [0,1],\, z \in [0,1],\, \beta_1 xz + \beta_2 x + \beta_3 z + \beta_4 < 0\},$$

$$X_= \times Z_= = \{(x,z) \,:\, x \in [0,1],\, z \in [0,1],\, \beta_1 xz + \beta_2 x + \beta_3 z + \beta_4 = 0\},$$

$$X_> \times Z_> = \{(x,z) \,:\, x \in [0,1],\, z \in [0,1],\, \beta_1 xz + \beta_2 x + \beta_3 z + \beta_4 > 0\},$$

$$\alpha_1 = a_{111} - a_{211} - a_{112} + a_{212} - a_{121} + a_{221} + a_{122} - a_{222},$$

$$\alpha_2 = a_{112} - a_{212} - a_{122} + a_{222}, \alpha_3 = a_{121} - a_{221} - a_{122} + a_{222}, \alpha_4 = a_{122} - a_{222},$$

$$\beta_1 = b_{111} - b_{121} - b_{112} + b_{122} - b_{211} + b_{221} + b_{212} - b_{222},$$

$$\beta_2 = b_{112} - b_{122} - b_{212} + b_{222}, \beta_3 = b_{211} - b_{221} - b_{212} + b_{222}, \beta_4 = b_{212} - b_{222}.$$

Proof. The truth of the proposition results from the following representation of the payoff functions:

$$\varphi_1(x,y,z) = (yz(a_{111} - a_{211}) + y(1-z)(a_{112} - a_{212}) + (1-y)z(a_{121} - a_{221}) +$$

$$+(1-y)(1-z)(a_{122} - a_{222}))x +$$

$$+(z(a_{211} - a_{221}) + (1-z)(a_{212} - a_{222}))y + (a_{221} - a_{222})z + a_{222} =$$

$$= (\alpha_1 yz + \alpha_2 y + \alpha_3 z + \alpha_4)x + \alpha_5 yz + \alpha_6 y + \alpha_7 z + \alpha_8,$$

$$\varphi_2(x,y,z) = (xz(b_{111}-b_{121})+x(1-z)(b_{112}-b_{122})+(1-x)z(b_{211}-b_{221})+$$
$$+(1-x)(1-z)(b_{212}-b_{222}))y+$$
$$+(z(b_{121}-b_{221})+(1-z)(b_{122}-b_{222}))x+(b_{221}-b_{222})z+b_{222}=$$
$$= (\beta_1 xz + \beta_2 x + \beta_3 z + \beta_4)y + \beta_5 xz + \beta_6 x + \beta_7 z + \beta_8.$$

□

Similarly the NE set can be constructed in the following cases:

- *players 1 and 3 have incomparable strategies, player 2 has equivalent strategies;*

- *players 2 and 3 have incomparable strategies, player 1 has equivalent strategies.*

Proposition 10. *If all players have incomparable strategies, then*

$$NE(\Gamma'_m) = Gr_1 \cap Gr_2 \cap Gr_3,$$

where

$$Gr_1 = [0,1]^3 \cap \{0 \times Y_< \times Z_< \cup [0,1] \times Y_= \times Z_= \cup 1 \times Y_> \times Z_>\},$$

$$Gr_2 = [0,1]^3 \cap \{X_< \times 0 \times Z_< \cup X_= \times [0,1] \times Z_= \cup X_> \times 1 \times Z_>\},$$

$$Gr_3 = [0,1]^3 \cap \{X_< \times Y_< \times 0 \cup X_= \times Y_= \times [0,1] \cup X_> \times Y_> \times 1\},$$

the components of the Gr_1, Gr_2, Gr_3 are defined as above.

Proof. The truth of proposition results from the following representation of the payoff functions:

$$\varphi_1(x,y,z) = (yz(a_{111}-a_{211}) + y(1-z)(a_{112}-a_{212})+$$
$$+(1-y)z(a_{121}-a_{221}) + (1-y)(1-z)(a_{122}-a_{222}))x+$$
$$+(z(a_{211}-a_{221}) + (1-z)(a_{212}-a_{222}))y + (a_{221}-a_{222})z + a_{222} =$$

$$= (\alpha_1 yz + \alpha_2 y + \alpha_3 z + \alpha_4)x + \alpha_5 yz + \alpha_6 y + \alpha_7 z + \alpha_8,$$

$$\varphi_2(x, y, z) = (xz(b_{111} - b_{121}) + x(1 - z)(b_{112} - b_{122}) +$$

$$+(1 - x)z(b_{211} - b_{221}) + (1 - x)(1 - z)(b_{212} - b_{222}))y +$$

$$+(z(b_{121} - b_{221}) + (1 - z)(b_{122} - b_{222}))x + (b_{221} - b_{222})z + b_{222} =$$

$$= (\beta_1 xz + \beta_2 x + \beta_3 z + \beta_4)y + \beta_5 xz + \beta_6 x + \beta_7 z + \beta_8,$$

$$\varphi_3(x, y, z) = (xy(c_{111} - c_{112}) + x(1 - y)(c_{121} - c_{122}) +$$

$$+(1 - x)y(c_{211} - c_{212}) + (1 - x)(1 - y)(c_{221} - c_{222}))z +$$

$$+(y(c_{112} - c_{212}) + (1 - y)(c_{122} - c_{222}))x + (c_{212} - c_{222})y + c_{222} =$$

$$= (\gamma_1 xy + \gamma_2 x + \gamma_3 y + \gamma_4)z + \gamma_5 xy + \gamma_6 x + \gamma_7 y + \gamma_8.$$

□

3 Conclusions

The NE set can be described as an intersection of graphs of players' best responses.

The solution of the problem of NE set construction in the mixed extension of the 2 × 2 × 2 game illustrates that the NE set is not necessarily convex even in convex game. Moreover, the NE set is frequently disconnected. Thus, new conceptual methods "which derive from the theory of semi-algebraic sets are required for finding all equilibria" [1]. In this article we make an attempt to give an idea of such a method.

References

[1] McKelvey R.D., McLenan A., *Computation of equilibria in finite games,* Handbook of Computational Economics, 1, Amsterdam, Elsevier, 1996, pp. 87-142.

[2] Moulen H., *Théorie des jeux pour l'éqonomie et la politique,* Paris, 1981 (in Russian: Games Theory, Moscow, Mir, 1985, 200 p.).

[3] Nash J.F., *Noncooperative game,* Annals of Mathematics, 54, 1951, pp. 280-295.

[4] Sagaidac M., Ungureanu V., *Operational research,* Chişinău, CEP USM, 2004, 296 p. (in Romanian).

V. Ungureanu, A. Botnari,

Received March 18, 2005

State University of Moldova,
60, A. Mateevici str.,
Chişinău, MD−2009, Moldova.
E–mail: *valungureanu@usm.md*

Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

The minimum cost multicommodity flow problem in dynamic networks and an algorithm for its solving

Maria A. Fonoberova, Dmitrii D. Lozovanu

Abstract

The dynamic version of the minimum cost multicommodity flow problem that generalizes the static minimum cost multicommodity flow problem is formulated and studied. This dynamic problem is considered on directed networks with a set of commodities, time-varying capacities, fixed transit times on arcs, and a given time horizon. We assume that cost functions, defined on edges, are nonlinear and depend on time and flow and the demand function also depends on time. The corresponding algorithm, based on reducing the dynamic problem to a static problem on a time-expanded network, to solve the minimum cost dynamic multicommodity flow problem is proposed and some details concerning its complexity are discussed.

Mathematics Subject Classification 2000: 90B10, 90C35, 90C27.

Keywords and phrases: dynamic networks, multicommodity flows, dynamic flows, flows over time, minimum cost flows.

1 Introduction

Multicommodity flows are among the most important and challenging problems in network optimization, due to the large size of these models in real world applications. Many product distribution, scheduling planning, telecommunication, transportation, communication, and management problems can be formulated and solved as multicommodity flow problems (see, for example, [1]). The multicommodity flow problem consists of shipping several different commodities from their respective sources to their sinks through a given network so that the total flow going through each edge does not exceed its capacity. No commodity ever transforms into another commodity, so that each one has its own flow conservation constraints, but they compete for the resources of the common network. Considered multicommodity network flow problem requires to find the minimum cost flow of a set of commodities through a network, where the arcs have an individual capacity for each commodity, and a mutual capacity for all the commodities.

While there is substantial literature on the static multicommodity flow problem, hardly any results on multicommodity dynamic flows are known, although the dynamic multicommodity flows are much more closer to reality than the static ones. In considered dynamic models the flow requires a certain amount of time to travel through each arc, it can be delayed at nodes, flow values on arcs and the network parameters can change with time. Dynamic flows are widely used to model different network-structured, decision-making problems over time (see, for example, [2, 3]), but because of their complexity, dynamic flow models have not been investigated as well as classical flow models.

In this paper we study the dynamic version of the minimum cost multicommodity flow problem on networks with time-varying capacities of edges. We assume that cost functions, defined on edges, are nonlinear and depend on time and flow and the demand function also depends on time. The minimum cost multicommodity dynamic flow problem asks for a feasible flow over time with given time horizon, satisfying all supplies and demands with minimum cost. We propose an algorithm for solving this problem, which is based on reducing the dynamic problem to the classical static problem on a time-expanded network.

2 Problem formulation

We consider a directed network $N = (V, E, K, w, u, \tau, d, \varphi)$ with set of vertices V, set of edges E and set of commodities K that must be routed through the same network. Each edge $e \in E$ has a nonnegative time-varying capacity $w_e^k(t)$ which bounds the amount of flow of each commodity $k \in K$ allowed on each arc $e \in E$ in every moment of time $t \in \mathbb{T}$. We also consider that every arc $e \in E$ has a nonnegative time-varying capacity for all commodities, which is known as the mutual capacity $u_e(t)$. Moreover, each edge $e \in E$ has an associated positive transit time τ_e which determines the amount of time it takes for flow to travel from the tail to the head of that edge. The underlying network also consists of demand function d: $V \times K \times \mathbb{T} \rightarrow R$ and cost function φ: $E \times R_+ \times K \times \mathbb{T} \rightarrow R_+$, where $\mathbb{T} = \{0, 1, 2, \ldots, T\}$.

The demand function $d_v^k(t)$ satisfies the following conditions:

a) there exists $v \in V$ for every $k \in K$ with $d_v^k(0) < 0$;

b) if $d_v^k(t) < 0$ for a node $v \in V$ for commodity $k \in K$ then $d_v^k(t) = 0$, $t = 1, 2, \ldots, T$;

In order for the flow to exist we require that $\sum\limits_{t \in \mathbb{T}} \sum\limits_{v \in V} d_v^k(t) = 0, \forall k \in$ $\in K$. Nodes $v \in V$ with $\sum\limits_{t \in \mathbb{T}} d_v^k(t) < 0$, $k \in K$ are called sources for commodity k, nodes $v \in V$ with $\sum\limits_{t \in \mathbb{T}} d_v^k(t) > 0$, $k \in K$ are called sinks for commodity k and nodes $v \in V$ with $\sum\limits_{t \in \mathbb{T}} d_v^k(t) = 0$, $k \in K$ are called intermediate for commodity k. We denote by V_-^k, V_+^k and V_0^k the set of sources, sinks and intermediate nodes for commodity k, respectively. The sources are nodes through which flow enters the network and the sinks are nodes through which flow leaves the network. The sources and sinks are sometimes called terminal nodes, while the intermediate nodes are called non-terminals.

To model transit costs, which may change over time, we define the cost function $\varphi_e^k(x_e^k(t), t)$ with the meaning that flow of commodity k of value $\xi = x_e^k(t)$ entering edge e at time t will incur a transit cost of $\varphi_e^k(\xi, t)$. We consider the discrete time model, in which all times are integral and bounded by horizon T. Time is measured in discrete steps, so that if one unit of flow leaves node u at time t on arc $e = (u, v)$, then one unit of flow arrives at node v at time $t + \tau_e$, where τ_e is the transit time of arc e. The time horizon (finite or infinite) is the time until which the flow can travel in the network and defines the makespan $\mathbb{T} = \{0, 1, \ldots, T\}$ of time moments we consider.

We start with the definition of static multicommodity flows. A static multicommodity flow x on $N = (V, E, K, w, u, d, \varphi)$ assigns to every arc $e \in E$ for each commodity $k \in K$ a non-negative flow value x_e^k such that the following flow conservation constraints are obeyed:

$$\sum_{e \in E^+(v)} x_e^k - \sum_{e \in E^-(v)} x_e^k = d_v^k, \ \forall\, v \in V, \ \forall\, k \in K,$$

where $E^+(v) = \{(u, v) \,|\, (u, v) \in E\}, \ \ E^-(v) = \{(v, u) \,|\, (v, u) \in E\}$.

The multicommodity flow x satisfies the demands if one-commodity flow x^k, $\forall k \in K$ satisfies the demands d_v^k for all $v \in V$.

Multicommodity flow x is called feasible if it obeys the mutual capacity constraints:

$$\sum_{k \in K} x_e^k \leq u_e, \ \forall\, e \in E \tag{1}$$

and individual capacities of every arc for each commodity:

$$0 \leq x_e^k \leq w_e^k, \ \forall\, e \in E, \ \forall k \in K. \tag{2}$$

Constraints (1) and (2) are called weak and strong forcing constraints, respectively.

The total cost of the static multicommodity flow x is defined as follows:

$$c(x) = \sum_{k \in K} \sum_{e \in E} \varphi_e^k(x_e^k).$$

A feasible dynamic flow on $N = (V, E, K, w, u, \tau, d, \varphi)$ is a function x: $E \times K \times \mathbb{T} \to R_+$ that satisfies the following conditions:

$$\sum_{\substack{e \in E^+(v) \\ t - \tau_e \geq 0}} x_e^k(t - \tau_e) - \sum_{e \in E^-(v)} x_e^k(t) = d_v^k(t), \ \forall\, t \in \mathbb{T}, \ \forall\, v \in V, \ \forall k \in K; \tag{3}$$

$$\sum_{k \in K} x_e^k(t) \leq u_e(t), \ \forall\, t \in \mathbb{T}, \ \forall e \in E; \tag{4}$$

$$0 \le x_e^k(t) \le w_e^k(t), \quad \forall\, t \in \mathbb{T},\ \forall\, e \in E,\ \forall k \in K; \tag{5}$$

$$x_e^k(t) = 0,\ \forall\, e \in E,\ t = \overline{T - \tau_e + 1, T},\ \forall k \in K. \tag{6}$$

Here the function x defines the value $x_e^k(t)$ of flow of commodity k entering edge e at time t. It is easy to observe that the flow does not enter edge e at time t if it will have to leave the edge after time T; this is ensured by condition (6). Capacity constraints (5) mean that in a feasible dynamic flow, at most $w_e^k(t)$ units of flow of commodity k can enter the arc e at time t. Mutual capacity constraints (4) mean that in a feasible dynamic flow, at most $u_e(t)$ units of flow can enter the arc e at time t. Conditions (3) represent flow conservation constraints.

The total cost of the dynamic multicommodity flow x is defined as follows:

$$c(x) = \sum_{t=0}^{T} \sum_{k \in K} \sum_{e \in E} \varphi_e^k(x_e^k(t), t). \tag{7}$$

The minimum-cost multicommodity dynamic flow problem is to find a feasible flow that minimizes the objective function (7).

It is easy to observe that if $\tau_e = 0, \forall\, e \in E$ and $T = 0$ then the formulated problem becomes the static minimum cost multicommodity flow problem.

3 The main results

In this paper we propose an approach for solving the formulated problem, which is based on its reduction to a static flow problem. We show that the minimum cost multicommodity flow problem on dynamic network N can be reduced to the minimum cost static flow problem on auxiliary static network N^T; we name it the time-expanded network. In such a way, a dynamic flow problem in a given network with transit times on the arcs can be transformed into an equivalent static flow problem in the corresponding time-expanded network. A discrete dynamic flow in the given network can be interpreted as a static flow in the corresponding time-expanded network. The advantage of this approach is that it turns the problem of determining an optimal flow over time into a classical static network flow problem in the time-expanded network.

The time-expanded network is a static representation of the dynamic network. Such a time-expanded network contains copies of the node set of the underlying network for each discrete interval of time, building a time layer. Copies of an arc of the considered network join copies of its end-nodes in time layers whose distances equal the transit time of that arc. We define this network as follows:

1. $V^T\colon = \{v(t)\,|\,v \in V,\ t \in \mathbb{T}\}$;
2. $E^T\colon = \{e(t) = (v(t), w(t+\tau_e))\,|\,e = (v,w) \in E,\ 0 \le t \le T - \tau_e\}$;
3. $u^T_{e(t)}\colon = u_e(t)$ for $e(t) \in E^T$;
4. ${w^k_{e(t)}}^T\colon = w^k_e(t)$ for $e(t) \in E^T,\ k \in K$.
5. ${\varphi^k_{e(t)}}^T({x^k_{e(t)}}^T)\colon = \varphi^k_e(x_e(t), t)$ for $e(t) \in E^T,\ k \in K$;
6. ${d^k_{v(t)}}^T\colon = d^k_v(t)$ for $v(t) \in V^T,\ k \in K$.

The essence of the time-expanded network is that it contains a copy of the vertices of the dynamic network for each time $t \in \mathbb{T}$, and the transit times and flows are implicit in the edges linking those copies.

Let $e(t) = (v(t), w(t + \tau_e)) \in E^T$ and let $x^k_e(t)$ be a flow of commodity $k \in K$ on the dynamic network N. The corresponding function on the time-expanded network N^T is defined as follows:

$${x^k_{e(t)}}^T = x^k_e(t),\ \forall k \in K.$$

Using the method from [4, 5] it can be proved that the set of feasible flows on the dynamic network N corresponds to the set of feasible flows on the time-expanded network N^T and that any dynamic flow corresponds to a static flow in the time-expanded network of equal cost, and vice versa. In such a way, for each minimum-cost flow in the dynamic network there is a corresponding minimum-cost flow in the static network and vice-versa.

Therefore, the minimum cost multicommodity flow problem on dynamic networks can be solved by static flow computations in the corresponding time-expanded network. If the cost function of dynamic network is linear with regard to flow, then the cost function of the time-expanded network will be linear. In this case we can apply well-known methods for minimum cost flow problems, including linear programming algorithms, combinatorial algorithms, as well as other developments, like [6]. If there is exactly one source and the cost function of the dynamic network is concave with regard to flow, then the cost function of the time-expanded network will be concave. If the cost function of dynamic network is convex with regard to flow, then the cost function of the time-expanded network will be convex. In this case we can apply methods from convex programming and the specialization of such methods for minimum cost flow problems.

4 The algorithm

Let the dynamic network N be given. Our object is to solve the minimum cost multicommodity flow problem on N. Proceedings are following:

1. Building the time-expanded network N^T for the given dynamic network N.
2. Solving the classical minimum cost multicommodity flow problem on the static network N^T, using one of the known algorithms (see, for example, [7, 8, 9, 10, 11]).
3. Reconstructing the solution of the static problem on N^T to the dynamic problem on N. □

The complexity of this algorithm depends on the complexity of the algorithm used for the minimum cost multicommodity flow problem in static networks.

References

[1] Assad, A. *Multicommodity network flows: A survey.* Networks **8**, (1978), pp. 37–92.

[2] Aronson, J. *A survey of dynamic network flows.* Ann. Oper. Res. **20**, (1989), pp. 1–66.

[3] Powell, W., Jaillet, P., Odoni, A. *Stochastic and dynamic networks and routing.* In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, Network Routing, volume 8 of Handbooks in Operations Research and Management Science, chapter 3, North Holland, Amsterdam, The Netherlands, (1995), pp. 141-295.

[4] Fonoberova, M., Lozovanu, D. *The optimal flow in dynamic networks with nonlinear cost functions on edges.* The Bulletin of Academy of Sciences of Moldova. Mathematics **3(46)**, (2004), pp. 21–27.

[5] Fonoberova, M., Lozovanu, D. *The maximum flow in dynamic networks.* Computer Science Journal of Moldova **3(36)**, (2004), pp. 387–396.

[6] Goldberg, A. *An efficient implementation of a scaling minimum-cost flow algorithm.* J. Algorithms **22(1)**, (1997), pp. 1–29.

[7] Castro, J., Nabona, N. *An implementation of linear and nonlinear multicommodity network flows.* European Journal of Operational Research Theory and Methodology **92**, (1996), pp. 37–53.

[8] McBride, R. *Progress made in solving the multicommodity flow problems.* Siam J. Optim. **8(4)**, (1998), pp. 947–955.

[9] Castro, J. *A specialized interior-point algorithm for multicommodity network flows.* Siam J. Optim. **10(3)**, (2000), pp. 852-877.

[10] Fleisher, L. *Approximating multicommodity flow independent of the number of commodities.* Siam J. Discrete Math. **13(4)**, (2000), pp. 505-520.

[11] Castro, J. *Solving Difficult Multicommodity Problems with a Specialized Interior-Point Algorithm.* Annals of Operations Research **124**, (2003), pp. 35-48.

M.A. Fonoberova, D.D. Lozovanu,

Received January 4, 2005

Institute of Mathematics and Computer Science,
5 Academiei str.
Chişinău, MD2028, Moldova.
E–mails: *mashaf83@yahoo.com*, *lozovanu@math.md*

Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

On a k-clique-join of a class of partitionable graphs

Mihai Talmaciu

Abstract

We call a graph G *O-graph* if there is an optimal coloring of the set of vertices and an optimal (disjoint) covering with cliques such that any class of colors intersects any clique. In this paper, it has been established the relation to $[p, q, r]$-*partite graphs* and the fact that the O-graphs admit a k-*clique-join*.

Key Words: perfect graphs, (α,ω)-partitionable graphs, [p,q,r]-partite graphs, k-clique-join.

American Mathematical Society (2000): 05C17.

1 Introduction.

Throughout this paper $G = (V, E)$ is a simple (i.e. finite, undirected, without loops and multiple edges) graph with vertex set $V = V(G)$ and edge set $E = E(G)$, with $\alpha = \alpha(G) \geq 2$ and $\omega = \omega(G) \geq 2$. $\overline{G}$ designates the complement of G. If $e = xy \in E$, we shall also write $x \sim y$, and $x \not\sim y$ whenever x, y are not adjacent in G. If $A \subseteq V$, then $G[A]$ (or $[A]$, or $[A]_G$) is the subgraph of G induced by $A \subseteq V$. By $G - W$ we mean the graph $(V, E - W)$, whenever $W \subseteq E$. For $A, B \subset V$, $A \cap B = \emptyset$, the set $\{ab | a \in A, b \in B, ab \in E\}$ will be denoted by (A, B), and we write $A \sim B$ whenever $ab \in E$ holds for any $a \in A$ and $b \in B$.

By P_n, C_n and K_n we mean a chordless path on $n \geq 3$ vertices, the chordless cycle on $n \geq 3$ vertices, and the complete graph on $n \geq 1$ vertices. A *hole* is a chordless cycle of length at least four; an *antihole* is the complement of such a cycle. A *Berge* graph is a graph which contains no odd hole and no odd antihole.

A *stable* set in G is a set of mutually non-adjacent vertices, and the *stability number* of G, denoted by $\alpha(G)$, is the cardinality of a maximum stable set.

By $S(G)$ we shall denote the family of all maximal stable sets of G, and $S_\alpha(G) = \{S|S \in S(G), |S| = \alpha(G)\}$. A *clique* in G is a subset A of $V(G)$ that induces a complete subgraph in G, and $C(G) = S(\overline{G})$, $\omega(G) = \alpha(\overline{G})$, while $C_\omega(G) = S_\alpha(\overline{G})$. Clearly, $S_\alpha(G) \subseteq S(G)$ and $C_\omega(G) \subseteq C(G)$ are true for any graph G.

The *chromatic number* and the *clique covering number* of G (i.e. the chromatic number of $\overline{G}$) will be denoted respectively, by $\chi(G)$ and $\theta(G)$. The *density* of G is the size of a largest clique in G, i.e., $\omega(G)= \alpha(\overline{G})$.

A graph G is *perfect* if $\alpha(H) = \theta(H)$ (or, equivalently, $\chi(H) = \omega(H)$) holds for any induced subgraph H of G.

Definition. *A graph G is called (α,ω)-partitionable (see Golumbic, [6], Olaru, [8]), if for any $v \in V(G)$, G-v admits a partition of α ω-cliques and a partition of ω α-stable sets.*

Properties referring to the (α,ω)-partitionable graphs can be found in (Chvatal, Graham, Perold, Whitesides, [4], also see Golumbic, [6], Olaru, [8]) and are given by:

Theorem. *Let G be a graph with n vertices, and $\alpha = \alpha(G)$ and $\omega = \omega(G)$. If G is (α,ω)-partitionable then the following statements hold:*

(i) n= $\alpha\omega + 1$;

(ii) G has exactly n ω-cliques and n α-stable sets;

(iii) Each vertex of G belongs to exactly α ω-cliques and to exactly ω α-stable sets;

(iv) Each ω-clique intersects exactly n-1 α-stable sets and is disjoint from exactly one and each α-stable set intersects exactly n-1 ω-cliques and is disjoint from exactly one.

From the previous Definition and Theorem a question is asked, what properties do the graphs that admit an optimal coloring and covering with cliques have, such that any clique intersects any class of colors. We call this type of graphs, O-graphs (to be seen [10]). We hope that this class of graphs makes a step towards a well-characterization of the graphs that admit a k-clique-join ([12]).

On the web page from [12] there are open problems concerning Perfect Graphs.

Perfect graphs have proved to be one of the most stimulating and fruitful concepts of modern graph theory: there are three books ([6], [2], [9]) and nearly six hundred papers ([5]) on the subject. The origin of this development was the Strong Perfect Graph Conjecture ([1]):

a graph is perfect if and only if neither it nor its complement
contains a chordless cycle whose length is odd and at least five.

There are theorems that elucidate the structure of objects in some class C by showing that every object in C has either a prescribed and relatively transparent structure or one of prescribed structural faults, along which it can be decomposed. M. Conforti, G. Cornuejols and K. Vuskovic proved that

every square-free Berge
(meaning Berge graph containing no hole of length four)
either belongs to one of two basic classes
(bipartite graphs and line-graphs of bipartite graphs),
or else it has one of two structural faults (star-cutset or 2-join).

Therefore every square-free Berge graph is perfect.

In 2002, M. Chudnovsky and P. Seymour, as well as, N. Robertson and R. Thomas announced that they had completed the proof of the Strong Perfect Graphs Conjecture. Their structural theorem asserts that

every Berge graph either belongs to one of five basic classes
(namely: bipartite graphs, their complements, line-graphs of
bipartite graphs, and their complements, double split graphs)
or else it has one of four structural faults
(namely: 2-join, 2-join in the complement, M-join,
a balanced skew partition).

Therefore every Berge graph is perfect (namely the Strong Perfect Graph Conjecture became, in May 2002, the Strong Perfect Graph

Theorem by Maria Chudnovsky, Neil Robertson, Paul Seymour, Robin Thomas ([3])).

2 The Results.

In the beginning we show that any O-graph admits a partition of the set of vertices in α ω-cliques and one in ω α-stable sets.

Definition 1. *A graph G is called O-graph if there is an optimal coloring $(S_1, ..., S_p)$ of vertices and optimal covering with cliques $(Q_1, ..., Q_r)$ such that any class of colors intersects any clique (i.e. $S_i \cap Q_j \neq \emptyset$, $(1 \leq i \leq p,\ 1 \leq j \leq r)$).*

We specify that in an optimal covering $(Q_1, ..., Q_r)$ with cliques of an O-graph, $S_i \cap Q_j \neq \emptyset$ $(1 \leq i < j \leq n)$.

If G is O-graph, we denote with $Q(G)$ and, respectively, $I(G)$ the covering set with cliques, respectively colorings of G with the property that any covering from $Q(G)$ and any coloring from $I(G)$ satisfies the condition from Definition 1.

We remark that G is O-graph if and only if $\overline{G}$ is O-graph and *any even cycle is O-graph* and *any even chain is O-graph.*

Lemma 1. *If G=(V,E) is an O-graph with n vertices, then for any p-coloration from I(G) and any r-covering from Q(G) the following statements hold:*

1) $p = \omega(G) (= \omega)$; $r = \alpha(G) (= \alpha)$ and

2) $n = \alpha\omega$.

Proof. We denote with $S = (S_1, ..., S_p)$ and $C = (Q_1, ..., Q_r)$ a p-coloration from $I(G)$ and respectively a r-covering from $Q(G)$. We prove that $|S_i| = \alpha$ and $|Q_j| = \omega$, $\forall i = 1, ..., p, \forall j = 1, ..., r$. Let S_i be fixed. Because $|S_i \cap Q_j| = 1$, $\forall j = 1, .., r$, we have $|S_i| \geq r$, that means that $\alpha \geq |S_i| \geq r$. Because $(Q_1, ..., Q_r)$ is a covering with cliques, it results that $|S_i| \leq r$. So we have $|S_i| = r$. We have, for any stable set S, $|S| \leq r$. If $|S| = \alpha$, then, in particularly $\alpha \leq r$. Because $\alpha \geq r$, we obtain $r = \alpha$. As conclusion, we have $|S_i| = \alpha$, $\forall i = 1, ..., p$ and we prove $|Q_j| = \omega$, $\forall j = 1, ..., r$ the same way. Because S is a partition of $V(G)$, we obtain $n = |V(G)| = \sum_{i=1}^{\omega} |S_i| = \alpha\omega$.

Corollary 1. *A graph G is O-graph if and only if there is a partition of the set of vertices in ω stable sets with α elements and a partition in α cliques with ω elements.*

Proof. The direct statement results from Lemma 1, and a graph with the property from the Corollary is obviously an O-graph.

Corollary 2. *If G is O-graph then:*

for any clique Q from an optimal covering with cliques of G: $\alpha(G-Q)=\alpha(G)-1$*;*

for any stable set S from an optimal coloring of G: $\omega(G-S)=\omega(G)-1$.

Corollary 3. *For any O-graph G, any class of colors from any optimal coloring intersects any clique from any optimal covering with cliques.*

Proof. We suppose that there exists a clique Q_j and a class of colors S_i, disjoint. Then $|Q_j|<\omega$ or $|S_i|<\alpha$, but $|V(G)|=\sum_{j=1}^{\alpha}|Q_j|=\sum_{i=1}^{\omega}|S_i|$, so $|V(G)|<\alpha\omega$, contradicting Lemma 1.

Remark 1. *If G is a (α,ω)-partitionable graph of order n then for any vertex v, G-v is O-graph.*

Proof. G being (α,ω)-partitionable, results that, for any $v\in V(G)$, G-v admits a partition of α ω-cliques and a partition of ω α-stable sets. If there is a clique Q_j from an optimal covering with cliques of G-v disjoint of a class S_j of colors from an optimal coloring of G-v then $|Q_j|<\omega$ or $|S_i|<\alpha$, but $|V(G)|-1=\sum_{j=1}^{\alpha}|Q_j|=\sum_{i=1}^{\omega}|S_i|$, so $|V(G)|-1<\alpha\omega$, contradicting that G is (α,ω)-partitionable.

Next, it is established a theorem of characterization of O-graphs, it is given an example of non-perfection of an O-graph and is shown in which condition an O-graph is perfect. For this it is given the definition of $[p,q,r]$-partite graphs.

Definition 2. *An [p,q,r]-partite graph ([11]) is a graph whose set of vertices, V, is partitioned in p independent sets $S_1,...,S_p$, each containing exactly q vertices, and $S_i\cup S_j$ contains exactly r independent edges, for $1\le i<j\le p$.*

Theorem 1. *Let G=(V,E) be a graph with $\alpha = \alpha(G)$ and $\omega = \omega(G)$. G is O-graph if and only if G is [ω, α, α]- partite and $\overline{G}$ is [α, ω, ω]-partite.*

Proof. Let G be an O-graph and $(S_1, ..., S_\omega)$ a partition of G in ω α-stable sets, and $(Q_1, ..., Q_\alpha)$ a partition in α ω-cliques with $S_i \cap Q_k \neq \emptyset, (1 \leq i \leq \omega, 1 \leq k \leq \alpha)$. We must show that $\forall i, j, i = 1, ..., \omega, j = 1, ..., \omega$ with $i \neq j$ $S_i \cup S_j$ admits a maximum matching with α elements. We denote with $\{x_k^i\} = S_i \cap Q_k$, $(1 \leq i \leq \omega, 1 \leq k \leq \alpha)$. For $1 \leq k, l \leq \alpha$, $k \neq l$ we have $x_k^i \neq x_l^i$ $(1 \leq i \leq \omega)$ because $Q_k \cap Q_l = \emptyset$. Therefore $S_i = \{x_1^i, ..., x_\alpha^i\}$ $(1 \leq i \leq \omega)$. For $1 \leq k \leq \alpha$, we have $x_k^i x_k^j \in E(G)$, because $\{x_k^i, x_k^j\} \subseteq Q_k$, $\forall i, j, i = 1, ..., \omega, j = 1, ..., \omega$ with $i \neq j$; so the set of edges $\{x_k^i x_k^j | k = 1, ..., \alpha\}$ is a matching in $[S_i \cup S_j]$ $\forall i, j, i = 1, ..., \omega, j = 1, ..., \omega$ with $i \neq j.$ Because $\overline{G}$ is O-graph with $\alpha(\overline{G}) = \omega$, $\omega(\overline{G}) = \alpha$, it results that $\overline{G}$ is $[\alpha, \omega, \omega]$-partite graph. Let G be $[\omega, \alpha, \alpha]$ -partite graph and $\overline{G}$ $[\alpha, \omega, \omega]$-partite graph. It results that there is a partition of V in $S = (S_1, ..., S_\omega)$ α -stable sets and in $C = (Q_1, ..., Q_\alpha)$ with Q_i cliques and $|Q_i| = \omega$, that means G is O-graph.

An example of O-graph which is not perfect is the reunion of a four disjoint four-cliques, adding four edges such that form an induced C_5.

Corollary 4. *A graph G is perfect O-graph if and only if G is* $[\omega, \alpha, \alpha]$*-partite,* $\{C_{2l+1}, \overline{C}_{2l+1}\}$*-free* $(l \geq 2)$ *and* $\overline{G}$ *is* $[\alpha, \omega, \omega]$*-partite graph.*

Proof. We suppose that G is $[\omega, \alpha, \alpha]$-partite and $\overline{G}$ is $[\alpha, \omega, \omega]$-partite graph. From Theorem 1 it results that G is O-graph. We suppose, on the contrary, that a minimal contraexample is minimal imperfect. Then G is unbreakable (Chvatal, to be seen [7]). So any vertex x of G is in a disk ([7]) (i.e. $x \in C_k$ or $x \in \overline{C}_k$, $k \geq 5$). Because G is $\{C_{2l+1}, \overline{C}_{2l+1}\}$-free $(l \geq 2)$, it results that $G \cong C_{2p}$ or $G \cong \overline{C}_{2p} (p \geq 3)$, a contradiction. So G is perfect O-graph. If G is O-graph it results that G is $[\omega, \alpha, \alpha]$-partite and $\overline{G}$ is $[\alpha, \omega, \omega]$-partite. If G contains $\{C_{2p+1}, (p \geq 2)$ as an induced subgraph then G is not perfect.

Next, it is shown that an O-graph admits a (k, α)-clique-join and the way in which an O-graph is obtained (built).

In [12] it is asked to be found a characterization for a k-clique-join.

Definition 3. *Let k be positive integer. A k-clique-join ([12]) of a graph G=(V,E) is a set of pairs* $\{(A_0, B_0), (A_1, B_1), ..., (A_k, B_k)\}$*, where* $\{A_0, B_0\}$ *is a partition of V, both* A_0 *and* B_0 *contain at least one* ω*-clique, and* $A_i \subseteq A_0, B_i \subseteq B_0 (i = 1, ..., k)$ *(not necessarily disjoint), moreover*

(i) If $x \in A_i$ *and* $y \in B_i$*, then* $xy \in E$

(ii) If K is an ω*-clique of G that intersects both* A_0 *and* B_0 *then there exists i so that* $K \subseteq A_i \cup B_i$

Definition 4. *Let k, s be positive integers. A k-clique-join* $\{(A_0, B_0), (A_1, B_1), ..., (A_k, B_k)\}$ *of a graph G=(V,E) is called (k,s)-clique-join if* $A_{i_1} \cup B_{i_1}$*,...,*$A_{i_{s-2}} \cup B_{i_{s-2}}$*,*$A_0 - \cup_{l=1}^{s-2} A_{i_l}$*,* $B_0 - \cup_{l=1}^{s-2} B_{i_l}$ *are s disjoint* ω*-cliques, for all* $i_1, ..., i_{s-2}$ *of the set* $\{1, ..., k\}$*.*

Theorem 2. *Let G be a graph with* $\alpha = \alpha(G)$*,* $\omega = \omega(G)$ *and* $\alpha \geq 2$*. G is O-graph if and only if there are k positive integers such that G admits a* (k, α)*-clique-join.*

Proof. Let $G = (V, E)$ be an O-graph and $(S_1, ..., S_\omega)$ a partition in ω α-stable sets of G and $(Q_1, ..., Q_\alpha)$ a disjoint covering with α ω-cliques of G with $S_i \cap Q_j \neq \phi$ $(1 \leq i \leq \omega,\ 1 \leq j \leq \alpha)$. Next, we define a (k, α)-clique-join $\{(A_0, B_0), (A_1, B_1), ..., (A_k, B_k)\}$. Because $\alpha \geq 2$, there are at least two disjoint ω-cliques. Without restricting the generality, we take $Q_1 \subseteq A_0, Q_2 \subseteq B_0, Q_1, Q_2$, the two ω-cliques. More, we consider $A_0 = Q_1 \cup \cup_{i=3}^{\alpha} M_i$, $B_0 = Q_2 \cup \cup_{i=3}^{\alpha} (Q_i - M_i)$, where $M_i \subset Q_i (M_i \neq Q_i)(3 \leq i \leq \alpha)$. We denoted $A_l = M_{l+2}, B_l = Q_{l+2} - M_{l+2} (1 \leq l \leq \alpha - 2)$ and we obtained:

$\{A_0, B_0\}$ is a partition of V, A_0 (B_0) contains at least a ω-clique Q_1 (Q_2);

If $x \in A_i$ and $y \in B_i (1 \leq i \leq \alpha - 2)$, then $xy \in E$ (because A_i is totally adjacent to B_i $(A_i \sim B_i)$);

If $K = Q_i$ for some i $(1 \leq i \leq \alpha)$ then (ii) is hold from Definition 3. Because a disjoint reunion of α ω-cliques is O-graph, it results that $k \geq \alpha - 2$. $\{(A_0, B_0), (A_1, B_1), ..., (A_k, B_k)\}$ is a k-clique-join for G with $k = \alpha - 2$. If K' is an ω-clique of G that intersects both A_0 and B_0, but $K' = Q_i$, $\forall i = 1, ..., \alpha$, then we denote $A_s = K' \cap A_0$, $B_s = K' \cap B_0$ for some s. We have $K' = A_s \cup B_s$ ((ii) holds from Definition 3), and if $x \in A_s$ and $y \in B_s$ then $xy \in E$ ((i) holds from Definition 3). For each such ω-clique K', we add the pair (A_s, B_s) to the previous k-clique-join and add one to k.

So $\{(A_0, B_0), (A_1, B_1), ..., (A_k, B_k)\}$ is a (k, α)-clique -join of G.

Reverse, we suppose that $\{(A_0, B_0), (A_1, B_1), ..., (A_k, B_k)\}$ is a (k, α)-clique -join of G and we show that G is a $[\omega, \alpha, \alpha]$-partite graph and $\overline{G}$ is a $[\alpha, \omega, \omega]$-partite graph. Without restricting the generality, we consider $A_1 \cup B_1, ..., A_{\alpha-2} \cup B_{\alpha-2}, A_0 - \cup_{i=1}^{\alpha-2} A_i, B_0 - \cup_{i=1}^{\alpha-2} B_i$ the α disjoint ω-clique, and let $Q_{l+2} = A_l \cup B_l$ be for $1 \leq l \leq \alpha - 2$, $Q_1 = A_0 - \cup_{i=1}^{\alpha-2} A_i$, $Q_2 = B_0 - \cup_{i=1}^{\alpha-2} B_i$. We denote $Q_j = \{x_j^1, ..., x_j^\omega\}(1 \leq j \leq \alpha)$. Because $Q_j (1 \leq j \leq \alpha)$ is a ω-clique, we have: $\forall x \in V - Q_j, \exists y \in Q_j$ so that $xy \notin E$. For $1 \leq j, t \leq \alpha, j \neq t$, we suppose $x_j^i x_t^i \notin E$. We denote $S_i = \{x_1^i, ..., x_\alpha^i\}(1 \leq i \leq \omega)$. Because $x_j^i \in S_i \cap Q_j, x_t^i \in S_i \cap Q_t (1 \leq j, t \leq \alpha, j \neq t)$, it results that S_i is α-stable. Because $E([S_i \cup S_s]) = \{x_j^i x_j^s | 1 \leq j \leq \alpha\}$, it results that $[S_i \cup S_s]$ contains α independent edges $(1 \leq i, s \leq \omega, i \neq s)$. $\overline{G}$ is a $[\alpha, \omega, \omega]$-partite graph, because Q_j $(1 \leq j \leq \alpha)$ is a stable set in $\overline{G}$ with ω elements, S_i $(1 \leq i \leq \omega)$ is a clique in $\overline{G}$ with α elements and $E([Q_j \cup Q_t]_{\overline{G}}) = \{x_j^i x_t^i | 1 \leq i \leq \omega\}$ $(1 \leq j, t \leq \alpha, j \neq t)$.

Proposition 1. *Let G=(V,E) be a simple graph with $\alpha = \alpha(G)$ and $\omega = \omega(G)$. G is O-graph if and only if it can be obtained from a disjoint reunion of α ω-cliques adding an edge between each two vertices non contained in any class of a partition of V in ω α-stable sets and in any class of a partition of V in α ω-cliques.*

Proof. Let $G = (V, E)$ be a simple graph and $S = \{S_1, ..., S_\omega\}$ $(C = \{Q_1, ..., Q_\alpha\})$ a partition of V in ω α-stable sets (α ω-cliques). We denote $S_i = \{x_1^i, ..., x_\alpha^i\}(1 \leq i \leq \omega)$ and $Q_j = \{x_j^1, ..., x_j^\omega\}(1 \leq j \leq \alpha)$. Clearly, the graph $H = [\cup_{i=1}^{\alpha} Q_j]_G$ is O-graph with $\alpha(H) = \alpha$ and $\omega(H) = \omega$. Let H' be the graph obtained from H to which an edge $e = xy$ $(x = x_k^i,\ y = x_q^p)$ of G is added. Because x and y belong to two distinct α-stable sets of S, it results that $i \neq p$. If $k = q$ then the edge e would be added to a ω-clique Q_k, contradicting the fact that

G is a simple graph. The graph H' has $\alpha(H') = \alpha$, $\omega(H') = \omega$ and the same S partition (respectively C) in ω α-stable sets (respectively α ω-cliques) as H. So H' is O-graph. Repeating the above procedure of adding edges from G, we obtain the graph G and the fact that G is an O-graph.

Reverse, if we suppose that $G = (V, E)$ is O-graph with $\alpha = \alpha(G)$, $\omega = \omega(G)$ and we consider the disjoint covering with α ω -cliques $C = \{Q_1, ..., Q_\alpha\}$ and the covering with ω α-stable sets $S = \{S_1, ..., S_\omega\}$ and we apply the procedure:

```
begin
            H:=G;
            while ($\exists e = xy \in E$ with $\{x, y\} \not\subset Q_j$ $(1 \leq j \leq \alpha)$) do
                        H:=H-e;
            end;
```

we obtain that $[\cup_{j=1}^{\alpha} Q_j]_G$ is an O-graph.

Indeed, the partition $C = \{Q_1, ..., Q_\alpha\}$ with α ω -cliques (and the covering with ω α-stable sets $S = \{S_1, ..., S_\omega\}$) of G, by deleting edges e=xy with $\{x, y\} \not\subset Q_j$ $(1 \leq j \leq \alpha)$ (according to the above procedure) remains the same also for $H = [\cup_{j=1}^{\alpha} Q_j]_G$, that means that H is an O-graph.

References

[1] C. Berge, *Farburn von Graphen deren samtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung), Wissenschaftliche Zeitschrift*, Martin Luther Universitat Halle- Wittenberg, Mathematisch-Naturwissenschaftliche Reihe (1961), pp.114–115.

[2] C. Berge and V.Chvatal, eds, *Topics on Perfect Graphs*, North-Holland Mathematics Studies, 88. Annals of Discrete Mathematics, 21. North-Holland Publishing Co., Amsterdam-New York, 1984.

[3] Maria Chudnovsky, Neil Robertson, Paul Seymour, Robin Thomas, *Strong Perfect Graph Theorem* http://arxiv.org/PS-chache/math/pdf/0212/0212070.pdf.

[4] V.Chvatal, R.L.Graham, A.F.Perold, S.H.Whiteside, *Combinatorial designs related to the perfect graph conjecture*, Discr. Math. 26(1976), pp.83–92.

[5] V.Chvatal, *A bibliography on perfect graph*, http://www.cs.rutgers.edu/ chvatal/perfect/papers.html.

[6] M.C.Golumbic, *Algorithmic Graphs Theory and Perfect Graphs*, Computer Science and Applied Mathematics, Academic Press [Harcourt Brace Jonovich, Publishers], New York-London-Toronto, Ont., 1980.

[7] R.B.Hayward, *Disk in unbreakable graphs*, Graphs Combin. 11(1995) pp.249–254.

[8] E.Olaru, *On strongly stable graphs and some consequences for partitionable graphs*, Analele Stiintifice ale Universitatii "Al.I.Cuza" din Iasi, Tomul VII, 1998, pp.33–41.

[9] J.L. Ramirez-Alfonsin and B.A. reea, eds., *Perfect Graphs*, Wiley2001.

[10] M.Talmaciu, *Decomposition Problems in the Graph Theory with Applications in Combinatorial Optimization* - Ph.D., University "Al.I.Cuza" Iasi, Romania, 2002.

[11] Raphael Yuster, *Independent Transversals and Independent Coverings in Sparse Partite Graphs*, Department of Mathematics Raymond and Beverly Sackler Faculty of Exact Sciences Tel Aviv University, Tel Aviv, Israel, (electronic) 2001.

[12] ***, *Perfect Graphs, The American Institute of Mathematics*, http://www.aimath.org/WWN/perfectgraph/, 2002.

M. Talmaciu,

Received March 31, 2005

Department of Mathematics, University of Bacau,
Spiru Haret, 8, 600114, Bacau, Romania
E–mail: *mtalmaciu@ub.ro*

Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

On quasistability radius of a vector trajectorial problem with a principle of optimality generalizing Pareto and lexicographic principles*

Sergey E. Bukhtoyarov, Vladimir A. Emelichev

Abstract

A multicriterion linear combinatorial problem with a parametric principle of optimality is considered. This principle is defined by a partitioning of partial criteria onto Pareto preference relation groups within each group and the lexicographic preference relation between them. Quasistability of the problem is investigated. This type of stability is a discrete analog of Hausdorff lower semi-continuity of the multiple-valued mapping that defines the choice function. A formula of quasistability radius is derived for the case of the metric l_∞. Some known results are stated as corollaries.

Mathematics Subject Classification 2000: 90C05, 90C10, 90C29, 90C31.

Key words and phrases: vector trajectorial problem, Pareto set, set of lexicographically optimal trajectories, quasistability, quasistability radius.

1 Introduction

Traditionally stability of an optimization problem is understood as continuous dependence of solutions on parameters of the problem. The most general approaches to stability analysis of optimization problems are based on properties of multiple-valued mappings that define optimality principles [1–4].

*This work was supported by Fundamental Research Foundation of the Republic of Belarus "Mathematical structures 29" (Grant 913/28)

Mathematical analysis does not present methods sufficient to investigate stability of a discrete optimization problem. It is greatly due to complexity of discrete models, which can behave unpredictably under small variations of initial data [4, 5]. At the same time, if terminology of general topology is not used, then the formulation of a stability problem can be significantly simplified in the case of a space of acnodes. There are different types of stability of discrete optimization problems (e. g. [4–9]). Stability of a discrete problem in the broad sense means that there exists a neighborhood of the initial data in the space of problem parameters such that any problem with parameters from this neighborhood possesses some invariance with respect to the initial problem. In particular, upper (lower) semicontinuity of an optimal mapping is equivalent to nonappearance of new (preserving of initial) optimal solutions under "small" perturbations of the mapping parameters. So concepts of stability [4–8] and quasistability [6–8, 10, 11] of discrete optimization problems arise.

In this article we consider an n-criterion trajectorial linear problem with partitioning of criteria into groups according to given Pareto preference relation within each group and the lexicographic preference relation between them. Two special cases of such partitioning correspond to Pareto and lexicographic optimality principles. A formula for quasistability radius of this problem is derived for the case of independent perturbations of initial data in the metric l_∞. Some known results are stated as corollaries.

Note that similar formulas were derived earlier in [12–16] for stability and quasistability radii of vector trajectorial and game-theoretic problems with other parametric principles of optimality ("from Condorset to Pareto", "from Pareto to Slater", "from Pareto to Nash").

2 Basic definitions and notations

Let a vector criterion

$$f(t, A) = (f_1(t, A_1), f_2(t, A_2), \ldots, f_n(t, A_n)) \rightarrow \min_{t \in T}$$

with partial criterion

$$f_i(t, A_i) = \sum_{j \in N(t)} a_{ij}, \quad i \in N_n = \{1, 2, ..., n\}, \ n \geq 1,$$

be defined on a system of subsets (trajectories) $T \subseteq 2^E$, $|T| \geq 2$, $E = \{e_1, e_2, \ldots, e_m\}$, $m \geq 2$. Here $N(t) = \{j \in N_m : e_j \in t\}$, A_i is the i-th row of a matrix $A = [a_{ij}] \in \mathbf{R}^{n \times m}$. Put $f_i(\emptyset, A_i) = 0$.

Let $s \in N_n$, $\mathcal{I} = \{I_1, I_2, \ldots, I_s\}$ be a partitioning of the set N_n into s nonintersecting nonempty sets, i. e.

$$N_n = \bigcup_{r \in N_s} I_r,$$

where $I_r \neq \emptyset$, $r \in N_s$; $p \neq q \Rightarrow I_p \cap I_q = \emptyset$. To any such partitioning we put in correspondence the binary relation $\Omega^n_{\mathcal{I}}$ of strict preference in the space $\mathbf{R}^n$ between different vectors $y = (y_1, y_2, \ldots, y_n)$ and $y' = (y'_1, y'_2, \ldots, y'_n)$ as follows

$$y \ \Omega^n_{\mathcal{I}} \ y' \quad \Leftrightarrow \quad y_{I_k} \succ y'_{I_k},$$

where $k = \min\{i \in N_s : \ y_{I_i} \neq y'_{I_i}\}$; y_{I_k} and y'_{I_k} are the projections of the vectors y and y' correspondingly onto the coordinate axes of the space $\mathbf{R}^n$ with numbers from the group I_k; $\succ$ is a relation, which induces Pareto optimality principle in the space $\mathbf{R}^{|I_k|}$:

$$y_{I_k} \succ y'_{I_k} \quad \Leftrightarrow \quad y_{I_k} \neq y'_{I_k} \ \ \& \ \ y_{I_k} \geq y'_{I_k}.$$

The introduced binary relation $\Omega^n_{\mathcal{I}}$ determines ordering of the formed groups of criteria such that any previous group is significantly more important that any consequent group. This relation generates the set of $\mathcal{I}$-optimal trajectories

$$T^n(A, \mathcal{I}) = \{t \in T : \forall t' \in T \quad (f(t, A) \ \overline{\Omega^n_{\mathcal{I}}} \ f(t', A))\},$$

where $\overline{\Omega^n_{\mathcal{I}}}$ is the negation of the relation $\Omega^n_{\mathcal{I}}$.

It is evident that $T^n(A, \mathcal{I}_P)$, $\mathcal{I}_P = \{N_n\}$ $(s = 1)$, is Pareto set, i. e. the set of efficient trajectories

$$P^n(A) = \{t \in T : \ \forall t' \in T \quad (f(t, A) \ \overline{\succ} \ f(t', A))\},$$

and $T^n(A, \mathcal{I}_L)$, $\mathcal{I}_L = \{\{1\}, \{2\}, \dots, \{n\}\}$ $(s = n)$, is the set of lexicographically optimal trajectories

$$L^n(A) = \{t \in T : \ \forall t' \in T \quad (f(t, A) \,\overline{\vdash}\, f(t', A))\},$$

where $\vdash$ is the lexicographic order in the space $\mathbf{R}^n$. This order is defined as follows

$$y \vdash y' \quad \Leftrightarrow \quad y_k > y'_k,$$
$$k = \min\{i \in N_n : \ y_i \neq y'_i\}.$$

So under the parametrization of optimality principle we understand assigning the characteristic of binary relation that in special cases induces well-known Pareto and lexicographic optimality principles.

It is easy to show that the binary relation $\Omega^n_{\mathcal{I}}$ is antireflexive, asymmetric, transitive, and hence it is acyclic. And since the set T is finite, the set $T^n(A, \mathcal{I})$ is non-empty for any matrix A and any partitioning $\mathcal{I}$ of the set N_n.

Hereinafter by $Z^n(A, \mathcal{I})$ we denote the problem of finding the set $T^n(A, \mathcal{I})$.

Clearly, $T^1(A, \{1\})$ is the set of optimal trajectories of the scalar linear trajectorial problem $Z^1(A, \{1\})$, where $A \in \mathbf{R}^m$. Many extreme combinatoric problems on graphs, boolean programming and scheduling problems and others are reduced to $Z^1(A, \{1\})$ [7, 9, 10, 17]).

The following properties follow directly from the above definitions.

Property 1. *$T^n(A, \mathcal{I}) \subseteq P_1(A) \subseteq T$, where*

$$P_1(A) = \{t \in T : \ \forall t' \in T \quad (f_{I_1}(t, A) \,\overline{\succ}\, f_{I_1}(t', A))\}.$$

Property 2. *If $f_{I_1}(t, A) \succ f_{I_1}(t', A)$, then $f(t, A) \ \Omega^n_{\mathcal{I}} \ f(t', A)$.*

Property 3. *If $f(t, A) \ \Omega^n_{\mathcal{I}} \ f(t', A)$, then $f_{I_1}(t, A_i) \geq f_{I_1}(t', A_i)$.*

Property 4. *A trajectory $t \notin T^n(A, \mathcal{I})$ if and only if there exists a trajectory t' such that $f(t, A) \ \Omega^n_{\mathcal{I}} \ f(t', A)$.*

Property 5. *A trajectory $t \in T^n(A, \mathcal{I})$ if and only if for any trajectory t' the relation $f(t, A) \ \overline{\Omega^n_{\mathcal{I}}} \ f(t', A)$ holds.*

Denote

$$S_1(A) = \{t \in P_1(A) : \ \forall t' \in T \setminus \{t\} \quad \big(f_{I_1}(t, A) \neq f_{I_1}(t', A)\big)\}.$$

Property 6. $S_1(A) \subseteq T^n(A, \mathcal{I})$.

Proof. Assume the converse, i. e. $t \in S_1(A)$ and $t \notin T^n(A, \mathcal{I})$. Then according to property 4 there exists a trajectory $t' \neq t$ such that

$$f(t, A)\ \Omega^n_{\mathcal{I}}\ f(t', A).$$

Hence due to property 3 we have

$$f_{I_1}(t, A) \geq f_{I_1}(t', A).$$

Taking into account the inclusion $t \in P_1(A)$ we obtain

$$f_{I_1}(t, A) = f_{I_1}(t', A),$$

i. e. $t \notin S_1(A)$, which contradicts the assumption.

Property 7. $\forall\ t \in S_1(A) \quad \forall\ t' \in T \setminus \{t\} \quad \exists\ i \in I_1 \quad \big(f_i(t', A_i) > f_i(t, A_i)\big)$.

For any number $\varepsilon > 0$, define the set of perturbation matrixes

$$\mathcal{B}(\varepsilon) = \{B \in \mathbf{R}^{n \times m} : \ ||B|| < \varepsilon\},$$

where $||B|| = \max\{|b_{ij}| \ : \ (i, j) \in N_n \times N_m\}, \ B = [b_{ij}]$.

As in [8, 10, 14, 17], under the quasistability radius of the problem $Z^n(A, \mathcal{I})$ we understand the number

$$\rho^n(A, \mathcal{I}) = \begin{cases} \sup K^n(A, \mathcal{I}) & \text{if } K^n(A, \mathcal{I}) \neq \emptyset, \\ 0 & \text{if } K^n(A, \mathcal{I}) = \emptyset, \end{cases}$$

where

$$K^n(A, \mathcal{I}) = \{\varepsilon > 0 : \forall\ B \in \mathcal{B}(\varepsilon)\ (T^n(A, \mathcal{I}) \subseteq T^n(A + B, \mathcal{I}))\}.$$

3 Lemmas

For any trajectories t and t' we define the numbers

$$\Delta(t,t') = |(t \cup t') \setminus (t \cap t')|,$$

$$d^n(t,t',A) = \max_{i \in I_1} \frac{f_i(t',A_i) - f_i(t,A_i)}{\Delta(t,t')}.$$

Lemma 1. *If $d^n(t,t',A) \geq \varphi > 0$, then the following relation holds for any perturbation matrix $B \in \mathcal{B}(\varphi)$:*

$$f(t,A+B)\ \overline{\Omega^n_{\mathcal{I}}}\ f(t',A+B).$$

Proof. Directly from the definition of the number $d^n(t,t',A)$ we have

$$\exists k \in I_1 \quad \left(f_k(t',A_k) - f_k(t,A_k) \geq \varphi\Delta(t,t')\right). \tag{1}$$

Further suppose that the assertion of the lemma is false, i. e. there exists matrix $B^* = [b^*_{ij}] \in \mathcal{B}(\varphi)$ such that $f(t,A+B^*)\ \Omega^n_{\mathcal{I}}\ f(t',A+B^*)$. Then by virtue of property 3 and linearity of the functions $f_i(t,A),\ i \in N_n$, we derive

$$0 \geq f_i(t',A_i+B^*_i) - f_i(t,A_i+B^*_i) =$$

$$= f_i(t',A_i) - f_i(t,A_i) + f_i(t',B^*_i) - f_i(t,B^*_i) \geq$$

$$\geq f_i(t',A_i) - f_i(t,A_i) - ||B_i||\Delta(t,t') >$$

$$> f_i(t',A_i) - f_i(t,A_i) - \varphi\Delta(t,t'), \quad i \in I_1,$$

i. e.

$$\forall i \in I_1 \quad (f_i(t',A_i) - f_i(t,A_i) < \varphi\Delta(t,t')),$$

which contradicts (1).

Lemma 2. *Let $t \in T^n(A,\mathcal{I}),\ t' \in T \setminus \{t\}$. For any number $\alpha > d^n(t,t',A)$ there exists a matrix $B^* \in \mathbf{R}^{n\times m}$ with norm $||B^*|| = \alpha$ such that*

$$f(t,A+B^*)\ \Omega^n_{\mathcal{I}}\ f(t',A+B^*). \tag{2}$$

Proof. We construct the perturbation matrix $B^* = [b^*_{ij}] \in \mathbf{R}^{n\times m}$ by the formula

$$b^*_{ij} = \begin{cases} -\alpha & \text{if } i \in I_1,\ e_j \in t' \setminus t, \\ \alpha & \text{if } i \in I_1,\ e_j \in t \setminus t', \\ 0 & \text{otherwise.} \end{cases}$$

Then $||B^*|| = \alpha$ and

$$f_i(t', B^*_i) - f_i(t, B^*_i) = -\alpha\Delta(t,t'), \quad i \in I_1.$$

From here we get

$$\frac{1}{\Delta(t,t')}(f_i(t', A_i + B^*_i) - f_i(t, A_i + B^*_i)) = \frac{f_i(t', A_i) - f_i(t, A_i)}{\Delta(t,t')} - \alpha \leq$$

$$\leq d^n(t,t',A) - \alpha < 0, \quad i \in I_1,$$

i. e. $f_{I_1}(t, A + B^*) \succ f_{I_1}(t', A + B^*)$. This implies (2) by virtue of property 2.

4 Theorem

Theorem. *For any partitioning $\mathcal{I}$ of the set N_n, $n \geq 1$, into s groups, $s \in N_n$, the quasistability radius $\rho^n(A,\mathcal{I})$ of a problem $Z^n(A,\mathcal{I})$ is expressed by the formula*

$$\rho^n(A,\mathcal{I}) = \min_{t\in T^n(A,\mathcal{I})} \min_{t'\in T\setminus\{t\}} d^n(t,t',A). \tag{3}$$

Proof. Denote the right hand side of (3) by φ for short. Before proving the theorem we note that since the sets $T^n(A,\mathcal{I})$ and $T \setminus \{t\}$ are non-empty, the number φ is correctly defined and nonnegative.

First we prove the inequality

$$\rho^n(A,\mathcal{I}) \geq \varphi. \tag{4}$$

Without loss of generality assume that $\varphi > 0$ (otherwise inequality (4) is obvious). From the definition of the number φ, it follows that for any trajectories $t \in T^n(A,\mathcal{I})$ and $t' \neq t$ the inequalities

$$d^n(t,t',A) \geq \varphi > 0$$

hold. Applying lemma 1 we get

$$\forall\ B \in \mathcal{B}(\varphi) \quad \forall\ t \in T^n(A,\mathcal{I}) \quad \forall\ t' \in T \quad (f(t, A+B)\ \overline{\Omega^n_{\mathcal{I}}}\ f(t', A+B)).$$

Therefore $t \in T^n(A+B,\mathcal{I})$ by virtue of property 5. Thus we conclude

$$\forall\ B \in \mathcal{B}(\varphi) \quad (T^n(A,\mathcal{I}) \subseteq T^n(A+B,\mathcal{I})).$$

This formula proves (4).

It remains to show that

$$\rho^n(A,\mathcal{I}) \leq \varphi. \tag{5}$$

Let $\varepsilon > \alpha > \varphi$ and trajectories $t \in T^n(A,\mathcal{I})$, $t' \neq t$ be such that $d^n(t,t',A) = \varphi$. Then according to lemma 2 there exists a matrix B^* with norm $||B^*|| = \alpha$ such that (2) holds, i. e. $t \notin T^n(A+B^*,\mathcal{I})$. Hence we have

$$\forall \varepsilon > \varphi \quad \exists B^* \in \mathcal{B}(\varepsilon) \quad (T^n(A,\mathcal{I}) \not\subseteq T^n(A+B^*,\mathcal{I}))\,.$$

This proves inequality (5). Summarizing (4) and (5) we obtain (3).

5 Corollaries

Corollary 1 [10]. *The quasistability radius of the problem $Z^n(A,\mathcal{I}_P)$, $n \geq 1$, of finding Pareto set $P^n(A)$ is expressed by the formula*

$$\rho^n(A,\mathcal{I}_P) = \min_{t\in P^n(A)} \ \min_{t'\in T\setminus\{t\}} \ \max_{i\in N_n} \frac{f_i(t',A_i) - f_i(t,A_i)}{\Delta(t,t')}.$$

Corollary 2 [18]. *The quasistability radius of the problem $Z^n(A,\mathcal{I}_L)$, $n \geq 1$, of finding the set of lexicographically optimal trajectories $L^n(A)$ is expressd by the formula*

$$\rho^n(A,\mathcal{I}_L) = \min_{t\in L^n(A)} \ \min_{t'\in T\setminus\{t\}} \frac{f_1(t',A_1) - f_1(t,A_1)}{\Delta(t,t')}.$$

A problem $Z^n(A,\mathcal{I})$ is called quasistable if $\rho^n(A,\mathcal{I}) > 0$. Thus quasistability of a problem $Z^n(A,\mathcal{I})$ is the property of preserving optimality by all $\mathcal{I}$-efficient trajectories under small variation of matrix A. In other words, quasistability is a discrete analog of Hausdorff lower semi-continuity of the multiple-valued mapping that assigns the set of $\mathcal{I}$-efficient trajectories to each set of the problem parameters.

Corollary 3. *For any partitioning $\mathcal{I}$ of the set N_n, $n \geq 1$, into s groups, $s \in N_n$, the following statements are equivalent for a problem $Z^n(A,\mathcal{I})$, $n \geq 1$:*

(i) *the problem $Z^n(A,\mathcal{I})$ is quasistable,*

(ii) $\forall\, t \in T^n(A,\mathcal{I}) \quad \forall\, t' \in T \setminus \{t\} \quad \exists\, i \in I_1 \quad \big(f_i(t', A_i) > f_i(t, A_i)\big)$,

(iii) $T^n(A,\mathcal{I}) = S_1(A)$.

Proof. Equivalence of statements (i) and (ii) follows directly from the theorem.

The implication (ii) $\Rightarrow$ (iii) is proved by contradiction. Suppose that (ii) holds but (iii) does not.

From properties 1 and 6 we get

$$S_1(A) \subseteq T^n(A,\mathcal{I}) \subseteq P_1(A).$$

Then (since $T^n(A,\mathcal{I}) \neq S_1(A)$ is assumed) there exists a trajectory $t \in T^n(A,\mathcal{I}) \subseteq P_1(A)$ such that $t \notin S_1(A)$. It follows that there exists trajectory $t' \in P_1(A)$ such that

$$t' \neq t, \quad f_{I_1}(t, A) = f_{I_1}(t', A).$$

This contradicts statement (ii).

The implication (iii) $\Rightarrow$ (i) is obvious by virtue of property 7.

From corollary 3, we easily get the following known result (e. g. see [10]).

Corollary 4. *The problem $Z^n(A,\mathcal{I}_P)$, $n \geq 1$, of finding Pareto set $P^n(A)$ is quasistable if and only if $P^n(A) = S^n(A)$.*

Here $S^n(A)$ is Smale set [19], i. e. the set of strictly efficient trajectories:

$$S^n(A) = \{t \in P^n(A) : \ \forall t' \in T \setminus \{t\} \quad \big(f(t,A) \neq f(t',A)\big)\}.$$

Corollary 3 also implies

Corollary 5 [18]. *The problem $Z^n(A, \mathcal{I}_L)$, $n \geq 1$, of finding the set $L^n(A)$ of lexicographically optimal trajectories is quasistable if and only if*

$$|L^n(A)| = \left| \text{Arg} \min_{t \in T} \ f_1(t, A_1) \right| = 1.$$

References

[1] Belousov E. G., Andronov V. G. *Solvability and stability of polynomial programming problems.* Moscow: Izd. MGU, 1993, 172 p. (in Russian)

[2] Dubov Yu. A., Travkin S. I., Yakimets V. N. *Multicriterion models of forming and choosing variants of systems.* Moscow: Nauka, 1986, 296 p. (in Russian)

[3] Molodtsov D. A. *Stability of optimality principles.* Moscow: Nauka, 1987, 272 p. (in Russian)

[4] Sergienko I. V., Kozerackaja L. N., Lebedeva T. T. *Investigation of stability and parametric analisys of discrete optimization problems.* Kiev: Naukova Dumka, 1995, 169 p. (in Russian)

[5] Kozerackaja L. N., Lebedeva T. T., Sergienko I. V. *Investigation of stability of discrete optimization problems.* Kibernetika i Sistemniy Analiz, 1993, no. 3, pp. 78 –93. (in Russian)

[6] Lebedeva T. T., Sergienko T. I. *Comparative analysis of different types of stability with respect to constraints of a vector integer-optimization problem.* Cybernetics and Systems Analysis, 2004, v. 40, no. 1, pp. 52–57.

[7] Leontev V. K. *Stability in linear discrete problems.* Problems of Cybernetics. M.: Nauka, 1979, v. 35, pp. 169–185. (in Russian)

[8] Emelichev V. A., Girlich E., Nikulin Yu. V., Podkopaev D. P. *Stability and regularization of vector problems of integer linear programming.* Optimization, 2002, v. 51, no. 4, pp. 645–676.

[9] Sotskov Yu. N., Leontev V. K., Gordeev E. N. *Some concepts of stability analysis in combinatorial optimization.* Discrete Math. Appl., 1995, v. 58, no. 2, pp. 169–190.

[10] Emelichev V. A., Kravtsov M. K., Podkopaev D. P. *On the quasistability of trajectory problems of vector optimization.* Math. Notes, 1998, v. 63, no. 1, pp. 19–24.

[11] Emelichev V. A., Kuzmin K. G., Leonovich A. M. *On one type of stability of a vector combinatorial problem with Σ-MINMAX and Σ-MINMIN partial criteria.* Izv. Vuzov, Matem., 2004, no. 12, pp. 17 –27. (in Russian)

[12] Emelichev V. A., Bukhtoyarov S. E. *Stability of generally efficient situation in finite cooperative games with parametric optimality principle ("from Pareto to Nash").* Computer Science Journal of Moldova, 2003, v. 11, no. 3, pp. 316-323.

[13] Bukhtoyarov S. E., Emelichev V. A. *Parametrization of principle of optimality ("from Pareto to Slater") and stability of multicriterion trajectory problems.* Diskretniy Analiz i Issled. Operaciy, ser. 2, 2003, v. 10, no. 2, pp. 3–18. (in Russian)

[14] Bukhtoyarov S. E., Emelichev V. A. *On quasistability of vector trajectorial problem with the parametric principle of optimality.* Izv. Vuzov, Matem., 2004, no. 1, pp. 25–30 (in Russian).

[15] Bukhtoyarov S. E., Emelichev V. A., Stepanishina Yu. V. *Stability of discrete vector problems with the parametric principle of optimality.* Cybernetics and Systems Analysis, 2003, v. 39, no. 4, pp. 604-614.

[16] Emelichev V. A., Stepanishina Yu. V. *Multicriteria combinatorial linear problems : parametrization of the optimality principle and the stability of the effective solutions.* Discrete Math. Appl., 2001, v. 11, no. 5, pp. 435-444.

[17] Emelichev V. A., Kuz'min K. G., Leonovich A. M. *Stability in the combinatorial vector optimization problems.* Automation and Remote Control, 2004, v. 65, no. 2, pp. 227–240.

[18] Emelichev V. A., Berdysheva R. A. *On stability and quasistability of trajectorial problem of sequential optimization.* Dokl. Nazhion. Akad. Nauk Belarusi, 1999, v. 43, no. 3, pp. 41–44. (in Russian)

[19] Smale S. *Global analysis and economics, V. Pareto theory with constraints.* J. Math. Econom., 1974, v. 1, no. 3, pp. 213–221.

S.E. Bukhtoyarov, V.A. Emelichev,

Received April 7, 2005

Belarusian State University
ave. Fr. Skoriny, 4,
Minsk, 220050, Belarus.
E–mail : *emelichev@bsu.by*

Computer Science Journal of Moldova, vol.**13**, no.1(37), 2005

The solution of problem of objects classification as the method of restoration of objects images

Igor Mardare

Abstract

This paper deals with problems of restoration of images on incomplete information of objects. In present paper the solution of the problem of restoration of defective images by using classification of objects is suggested.

1 Introduction

An important property of human brain is the figurative perception of the world. This property allows on the basis of acquaintance with final number of objects to find out with the certain reliability an infinite number of their variations, for example, by the incomplete, deformed or defective images to restore the true image of object. Another interesting property of human brain is the classification of input information. This property means the ability of a brain to react to the infinite set of conditions of external world by finite number of reactions. A person breaks data into groups of similar, but not identical phenomena. Different persons, training on a various material of supervision, equally and independently from each other classify the same objects. This is the objective character of images.

2 Restoration of images with the help of Kohonen networks.

An image or class is considered to be the classification grouping, uniting certain group of objects by some sign. Objects of the same image can differ greatly enough from each other. For example, following portrayals can make the image of a coin: free of defects; with semieffaced emblem; with semieffaced value of a coin; with semieffaced face of emperor; with scratches, etc. Other examples of images – triangular prism $\boldsymbol{A}$=$\{\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\ \boldsymbol{a}_n\}$ and rectangular triangle $\boldsymbol{B}$=$\{\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\ \boldsymbol{b}_m\}$ – are presented in fig.1.

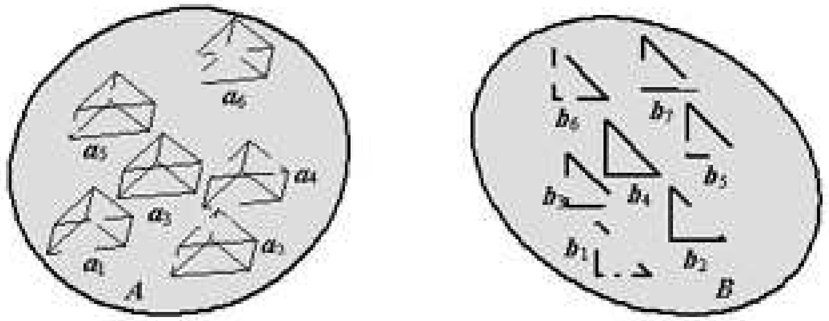


Figure 1. Images of triangular prism and rectangular triangle.

On the basis of the specified properties it is supposed to reduce problem of restoration of defective objects to problem of classification of objects. Thus, the problem of objects classification is a method of solution of the problem of restoration of defective images.

The problem of classification consists in division of objects into classes on the basis of vector of object parameters. Objects within one class are considered to be equivalent according to a criterion of division. Frequently, classes are unknown beforehand, and formed dynamically (for example, in Kohonen networks). In this case, classes depend on shown objects and the consequent addition of a new one demands the correction of system of classes.

Let objects be characterized by a vector of parameters $\boldsymbol{x}_n \in \boldsymbol{X}$, which has K components: $\boldsymbol{x}_n$=$(x_1,\ x_2,\ ...,\ x_K)$, where $\boldsymbol{X}$ is a space of objects. Set of classes $\boldsymbol{C}_q$=$\{\boldsymbol{C}_1,\ \boldsymbol{C}_2,\ ...,\ \boldsymbol{C}_Q\}$ is in the space of classes $\boldsymbol{C}$:$\{\boldsymbol{C}_1 \cup \boldsymbol{C}_2 \cup\ ...\ \cup\ \boldsymbol{C}_Q\} \subset \boldsymbol{C}_q$. Let's define nucleuses of classes $\boldsymbol{c}_q$=$\{\boldsymbol{c}_1,\ \boldsymbol{c}_2,\ ...,\ \boldsymbol{c}_Q\}$ in the space of classes $\boldsymbol{C}$ as true objects for its class. For example, if for the classification of geometrical figures the following parameters are chosen:

{quantity of corners, types of corners and their number, quantity of direct lines},

then the nucleus of the class "triangular prism" will have the following values of parameters:

{18 corners, 6 sharp corners, 12 right angles, 9 lines} (fig. 2).

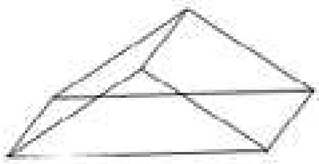


Figure 2. Nucleus of the class “triangular prism”.

It is possible to relate to this class the defective object with the following values of parameters:

{13 corners, 5 sharp corners, 9 right angles, 11 lines},

as from nucleuses "cylinder", "parallelepiped", "pyramid", "triangular prism" the parameters of considered object are most of all similar to the nucleus “triangular prism” (fig. 3). The quantitative estimation

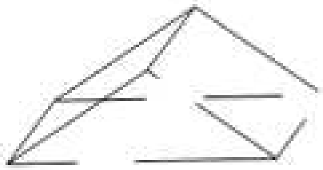


Figure 3. Defective object of class “triangular prism”.

of affinity of an object to a nucleus is defined by a measure of affinity d($\boldsymbol{x}_n$, $\boldsymbol{c}_q$) of the object and the nucleus of the class which is as less, as more this object is similar to the nucleus of the class. Measure of affinity of two nucleuses of classes is: d($\boldsymbol{c}_1$, $\boldsymbol{c}_2$). The Euclid measure represents geometrical distance between objects in many-dimensional space of attributes:

$$d(x,y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}.$$

A class is formed by group of vectors, the distance between which inside the group is less, than the distance up to the next groups. Inside classes objects should be closely connected among themselves, but objects of different classes should be far from each other (the requirement of compactness of classes). Distribution of objects inside classes should be uniform (fig. 4).

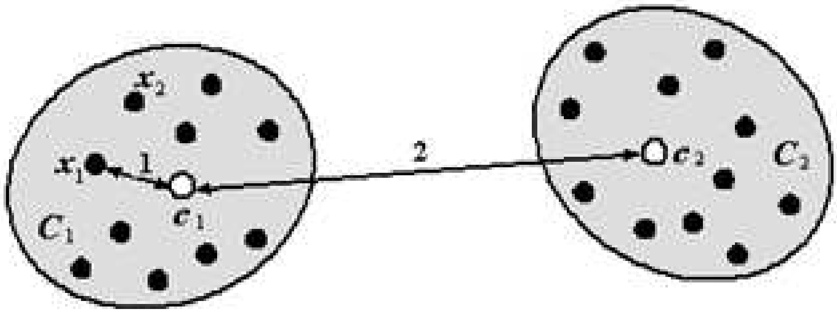


Figure 4. Distances between object and a nucleus (1) and two nucleuses (2).

The problem of classification for given number Q of classes is formulated as: to find Q nucleuses of classes $\{\boldsymbol{c}_q\}$ and to break objects $\{\boldsymbol{x}_n\}$ into classes $\{\boldsymbol{C}_q\}$, i.e. to construct the function q(n) so that to minimize a total measure of affinity for the whole set of input objects$\{\boldsymbol{x}_n\}$:

$$\min\{D = \sum_{n=1}^{N}\sum_{i=1}^{K}(x_{ni} - c_{q(n)i})^2\} \ .$$

The function q(n), which determines the class number by index n of set of objects $\{\boldsymbol{x}_n\}$, sets the splitting into classes and is the solution of the problem of classification. In the elementary case $\boldsymbol{X}=\boldsymbol{C}$, the space of objects $\boldsymbol{X}$ is broken into areas $\{\boldsymbol{C}_q\}$, and if $\boldsymbol{x}_n \in \boldsymbol{C}_q$, then q(n)=q and the object is ascribed to the class q.

Presence of nucleuses of classes allows using the neural networks trained without teacher (Kohonen networks) for restoration of defective images. For input data the vector of parameters of object $\boldsymbol{x}_n$ (the defective image) is used. The vector values of nucleuses of classes are given to synapse weights of neurons. Every neuron remembers one nucleus of a class, being responsible for definition of the objects in its class, and gives the sum $\boldsymbol{y}_i$ on the output. The total number of classes coincides with the number of Kohonen neurons. The neuron output value is the greater, the closer object is to the given nucleus of the class. The vector of defective image, input to the Kohonen network, makes active one of the neurons. Neuron with maximal output defines the class to which the object presented on input belongs.

Synapse weights of every neuron represent the n-dimensional vector-column $\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K]^T$, where K - is the dimension of input vectors. Before the beginning of training a network it is required to initialize weight coefficients of neurons. Usually to synapse weights of neurons the normalized small uniformly distributed random numbers are given initially.

The problem of training consists in teaching the network to activate the same neuron for similar input vectors. If the number of input vectors is equal to the number of nucleuses (neurons), then training is not required. It is enough to give to nucleuses the values of input vectors, and each vector will activate its own Kohonen neuron. If the number of classes is less than the number of input vectors, then the training will consist in consecutive correction of synapse weights of neurons. On each step of the training one of the vectors is selected randomly from initial data set, and then the search of the vector of neuron coefficients, which is most similar to it, is carried out. The most similar coefficient vector is defined by the neuron-winner which has the maximal output value. Similarity is understood as distance between vectors, calculated in Euklid space. For i-th neuron-winner we have:

$$|x_n - w_i| = \underset{j}{min}\{|x_n - w_j|\}.$$

Updating the weight coefficients is made according to the expression:

$$w_i^{t+1} = w_i^t + h_i^t(x_n^t - w_i^t) \ ,$$

where

w_i^{t+1} - new value of weight which the i-th neuron has gained;
w_i^t - previous value of this weight;
h_i^t - function of neuron neighborhood;
x_n^t - randomly chosen input vector on t-th iteration.

The training of Kohonen network with uniformly distributed random vectors of weights (nucleuses of classes) is graphically presented in fig. 5.

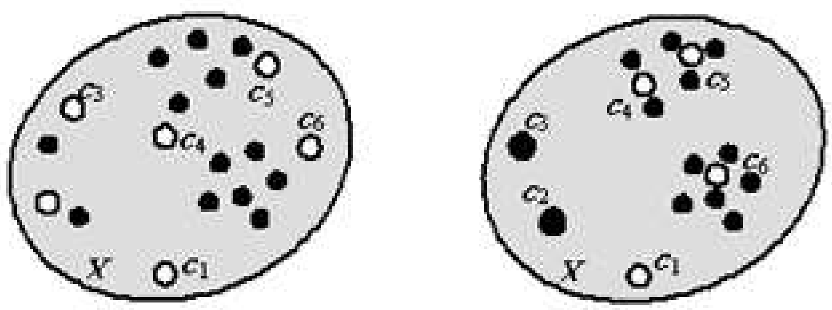


Figure 5. Training of Kohonen network: on the left - untrained network; on the right - trained network.

In the fields of space $\boldsymbol{X}$ in which the nucleuses are far from all training vectors, neuron $\boldsymbol{c}_1$ will never win, and its weights will not be corrected by training. In those areas where there are a lot of input vectors, the density of nucleuses is small, and the unlike objects will

activate the same neuron $\boldsymbol{c}_6$. These lacks are connected with the initial assignment of the uniformly distributed random numbers to the synapse weights of neurons. The problem is solved by allocation of nucleuses according to the density of input vectors. But the distribution of input vectors frequently happens to be unknown beforehand. In this case at training the method of convex combination is used, which allows to distribute the nucleuses of classes (vectors of weights) according to the density of input vectors in the space $\boldsymbol{X}$. Method is realized as follows:
- assignment of the identical initial value to all weights:

$$w_{ij} = \frac{1}{\sqrt{\dim \boldsymbol{X}}} \ ,$$

where dim $\boldsymbol{X}$ – diameter of a class;
- setting of training set $\{\boldsymbol{x}_n\}$ and carrying out the training with vectors:

$$\beta(t)x_n + \frac{1-\beta(t)}{\sqrt{dimX}},$$

where t – time of training;

β (t) – monotonously growing function in an interval [0,1].

In the beginning of training the function $\beta(t) = 0$ and all the vectors of weights and of training set have the same value (fig. 6.1). In the process of training the function $\beta(t)$ grows, the training vectors diverge from a point with coordinates $1\Big/\sqrt{\dim X}$ and approach to their true values $\boldsymbol{x}_n$ (fig. 6.2) which are reached at $\beta(t) = 1$. Each vector of weights grasps the group or one training vector and traces it in the process of growth of the function β. As a result in the network remains no untrained neuron and the density of weight vectors corresponds to the density of vectors of training set (fig. 6.3). The process of increasing of β demands many iterations that results in the increase of training time.

Method of convex combination gives correct distribution of density of nucleuses. And in network remain no untrained neurons which occur at usual training.

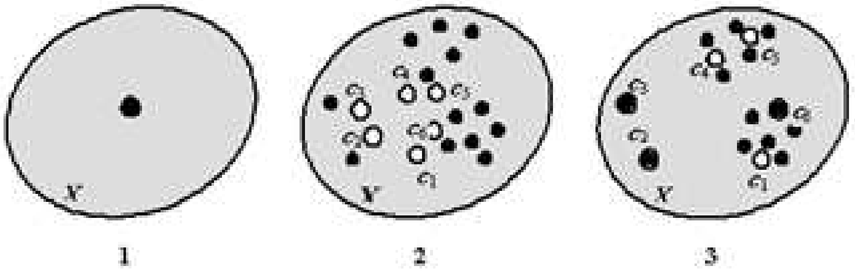


Figure 6. Training by a method of convex combination.

Kohonen Self Organizing Map is the competitive neural network in which neurons compete with each other for the right to be combined with the input vector in the best way. During self-organization the Kohonen map configures neurons according to topological representation of initial data, and vectors similar in the initial space, appear beside one another on the obtained map. Usually neurons are settled down in nodes of bidimentional network. At that neurons interact with each other. The value of interaction is defined by distance between neurons on the map. The class structure can be reflected by visualization of the distances between vectors of neuron weights. The values of distances define colors by which the node will be painted. Using gradation of gray color, the more is the distance, the darker the node is painted. For color palette the distance is defined according to a color scale (fig. 7). On presented Kohonen map two classes of objects are determined. By black points the vectors of defective images of objects, used at training, are marked. Empty cells mean the vectors of all possible defective images concerning given classes of objects. Thus, the analysis of Kohonen map allows to specify a priori with which defects the images can be restored.

3 Conclusions

From the point of view of the problem of restoration of defective images the application of method of convex combinations gives the following

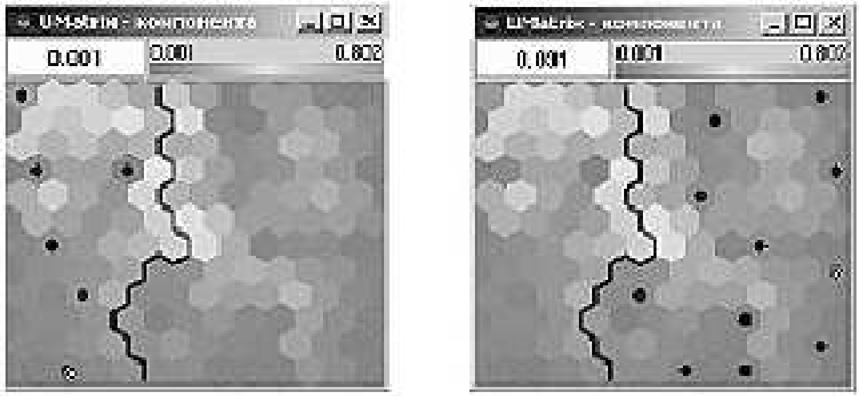


Figure 7. Kohonen maps and display of vectors of two clusters.

advantages:

- The method does not need obligatory presence of nucleus of class (true image). The nucleus of a class is formed during the network training on the basis of available defective images of the object, and can be specified in the process of appearance of new defective images.

- When only defective images of the object are present it is possible with sufficient degree of accuracy to specify a location of class nucleus (the true image of the object) by calculation of the center of gravity of the class. It is possible as, in the case of choice of Euclid measure of closeness, the nucleus of a class, that minimizes the sum of measures of closeness for objects of this class, coincides with the center of gravity of objects:

$$c_q = \frac{1}{N_q} \sum_{n:q(n)=q} x_n,$$

where N_q – number of objects $\boldsymbol{x}_n$ in the class q.

The carried out analysis of the problem of classification shows that the problem of restoration of defective image consists only in definition of belonging of an object to some class. The restored or true image will be the nucleus of the class.

References

[1] Vaintsvaig M.N., Poliacova M.P. *Mechanism of thinking and modelling of his work in real time. Intellectual processes and their modelling.* - M.: the Science, 1987. (rus).

[2] Mardare I. *Restoration of the image with use of a neural network.* International Informatization Academy. Branch of R.Moldova. Chisinau: Evrica, Acta Academia 2002 (pp.246-255). (rus).

[3] Sotnic S.L. *Rate of lectures in a subject "Bases of designing of systems with an artificial intellect".* - Dneprodzerzhinsk, 1997-2000. (rus).

[4] Staricov A. *Neural networks - the mathematical device* http://www.basegroup.ru/. (rus).

[5] Staricov A. *Bases of artificial neural networks.* http://www.basegroup.ru/. (rus).

[6] Wosserman F. *Neurocomputing technics: the theory and practice.* - M.: the World, 1992. - 237 p. (rus).

[7] *For A. Perception and recognition of images.* Translation with french - M.: Mechanical engineering, 1989. - 272p. (rus).

[8] Hunt E. *Artificial intellect.* - M.: The World, 1978. (rus).

[9] Cornea I., Mardare I. *Restoration of Images with Application of Neural Networks.* Proceedings of 2002 IEEE-TTTC. International Conference on Automation, Quality and Testing, Robotics. May 23–25, 2002, Cluj-Napoca, Romania (pp.95–98).

[10] Cornea I., Mardare I. *Methods of Artificial Intellect in Images Restoration.* CSCS-14. 14th International Conference on Control Systems and Computer Science. 2-5 july, 2003, Bucharest, Romania, (pp.141–143).

I. Mardare,

Received February 23, 2005

Department of Design and Manufacturing of Electronic Apparatus
Technical University of Moldova
Stefan Mare Av, 168, Chisinau
MD2012, Republic of MOLDOVA
Tel:(3732)237505
Fax:(3732)235236
E–mail: *mardarei@mail.md*, *imardare@adm.utm.md*

Generic Interfaces for Managing Web Data

Oleg Burlaca

Abstract

This paper discusses a generic user interface for managing web data that is incorporated in a content management system. The interface is created at run-time from a set of XML documents stored in database. We accentuate the importance of content analysis phase that leads to a well formed data model. Another important aspect is the use of context in the interface and the hierarchical model to represent multiple relationships between hierarchy items. The proposed event model acts like a glue between data management and application logic.

Keywords: User Interface, Database Access, Conceptual Modeling, Content Management System (CMS), XML.

1 Introduction

Significant research has addressed the conceptual modeling [1] and declarative specification [2] of data-intensive websites. Frameworks and paradigms are proposed to build websites from data sources with complex structure, but do not offer means or propose methodologies for managing the underlying data. In other words, there is a lack of tools that offer generic interfaces for managing intertwined (complex) data structures.

A practical implementation of methods and approaches described in this article is NeoSite (http://cms.neonet.md): a CMS developed by the author for structured web sites. Constantly been improved, it has been in use for the last three years to manage web sites of different complexity.

After the content analysis phase, we try to represent the data of a web site using a lightweight version of the entity relationship model introduced by Chen [3].

This article is structured as follows. The next section describes what incited us to develop a CMS that offers a generic interface. Section 3 explains why the content analysis phase is so important when developing structured web sites. Section 4 highlights some advantages of generic UI, describes some of the problems with web interfaces, reveals our approach to building generic interfaces, and accentuate on relationship and multirelations between entities. A brief outline of the event model that allows to tie the interface to application logic is provided in Section 5.

Section 6 gives an overview of related research and tools.

2 Historical Background

Previous experience of creating specialized content management systems showed that the most tedious and time consuming task was to develop handy user interfaces (UI) for content authors. Most database systems (we used MySql) contains rudimentary tools for manipulating and querying the data, that are usually insufficient for most users. For this reason, a customized UI to handle interaction between the user and the database was required for each web project.

As the number of web projects grew up, there was an urgent need for a generic Content Management System (CMS). Using our experience of creating tools for web site management, we started to build our own CMS that will encompass the most needed and reusable components. From our point of view, as developers, at that time, a CMS that offers a tight integration of WWW, DB and FTP services would be sufficient. From a content author and designer perspective, a CMS is a system that enables him to effectively and comfortably publish his content, change the design of the site, notify other CMS users, generate some reports for the administrative staff, etc.

In reality a CMS is a concept rather than a product. It is a concept that embraces a set of processes that will stay at the foundation of the next generation of web sites where content authors will have more privileges, duties and responsibility than designers and developers.

As we mentioned, developing custom UI was (besides information architecture process and conceptual modeling) the most time consuming stage. The two key factors were:

1) complexity and uniqueness of UI being developed;
2) the ever changing requirements of web projects (a new field was added to a DB table, the nature (attribute list) of a relation between entities has changed).

We desperately needed a high level interface to database; a system that will generate a UI from a set of rules: which fields should be available to the user, how relations are created between records etc. In other words, a generic UI to relational DB systems was needed.

We propose such a generic UI engine, implemented in our NeoSite CMS [4]. NeoSite is a web development environment where the data management facilities play a central role. The system responsibility is how the content is managed (created, updated, related) but not how it is used.

3 Content Analysis

To make a successful CMS for a particular web project, a research phase for content analysis is needed. It helps you reveal patterns and relationships within content and metadata that can be used to better structure, organize, and provide access to that content. The purpose of content analysis is to provide data that's critical to the development of a solid information architecture [5].

In this work we are mainly concerned with UI to structured web data. Examples of such sites are bookstores, electronic catalogs, e-news. These are sites that include pages composed of data whose overall structure is naturally hierarchical, but exhibits a modest degree of variation. In [6] such pages are called "data rich, ontological narrow, multiple-record web pages". Such sites are composed of index and entity pages. Index pages will be automatically generated from entity(leaf) pages. An entity page consists of:

1. internal content: it mainly appears on this page (a long text, a XML description of a product);
2. external content: attributes that are used to construct index pages, feed search engines;
3. relationships with other entity pages and the specification (attributes) of those relations.

Defining entity types, their relationships and organizing entity collections into a hierarchy that mimics the site structure is a creative process and requires information architects intuition, programmers point of view and customer involvement. After this work is carried out, a XML specification of the UI is elaborated and fed to NeoSite generic UI engine.

Hierarchy is ubiquitous in our lives. Because of its pervasiveness, users can easily understand web sites that use hierarchical organization models. They are able to develop a mental model of the site's structure and their location within that structure. This provides context that helps users feel comfortable.

4 Generic User Interfaces

Within the lifecycle of a software project there are often many changes required to the initial design as it progresses, and the developer's software tools are not able to easily cope with these changes. The UI needs to interact with the database and is sensitive to any changes occurred in the database. Such changes will cause the UI to stop working unless the same changes are also applied to the UI. Furthermore, the results of major changes to the underlying database and user interface may require substantial re-testing of the application.

The users desire is to view a multitude of data on a single complex screen. It is usually indulged by the developer even though this may not be desirable. Such complex screens are not scalable and difficult to change.

When developing a database system there are well-established rules of how data should be stored and accessed, called Relational Database

Theory. Michelangelo said: “Form relieves.” What if there will be a UI theory with strict rules? Maybe then, interfaces will be simpler and more consistent than a traditional application, but able to provide much more powerful functionality? A few obvious advantages of a generic interface are: reduced development time, scalability, no hand coding, less testing, consistent look and feel, standardization, less training.

4.1 Client application vs. Web interface

Almost all commercial CMSs uses a web interface to interact with the user. The central argument is that you don’t have to install special software on the client side. You can administer the site from any computer connected to the Internet that has a web browser installed. However, browser interface controls have limited functionality, compared to desktop applications. But the main problem is the stateless editing. HTTP is a stateless protocol and doesn’t support stateful interaction with a server. This problem can partially be solved by using a mix of cookies and server sessions imposed over the protocol, but in general, editing content in a web browser is not so pleasant nowadays. The problem becomes even worse when you have to edit a large collection of entities. For example, ERW – a system for handling complex databases through a web browser [7], “solves” this issue by “simulating remote procedure calls”. However, their todo list tells that the system will be redesigned from scratch; the first step in this direction is the creation of a working, independent framework for asynchronous remote-script callback invocation.

As you have probably guessed, we are using a client application for managing the site’s data. Besides installation issues (each user has to install the software on his machine), there are a lot of advantages that are evident when the site is managed by a few number of users.

4.2 The Hierarchy

After the content analysis phase, the identified entity types of the future web site will be hierarchically organized and presented to the user as a treeview. The hierarchy provides user with a decent navigation system, that simplifies the UI by minimizing the number of facilities used to find information. At the same time, the user will always have a clear understanding of his location within that structure due to breadcrumbs. In other words, the *hierarchy provides context*. Another feature is the *implicit relationship* (parent-child) between nodes. Instead of displaying a grid with two columns:

1. product title,
2. category of the product, sometimes a tree with two levels: category as parent with products as children is more appropriate.

However, hierarchies can be limiting from a navigation perspective. You are forced to move up and down and can't jump across branches (lateral navigation) or between multiple levels (vertical navigation). Additionally, some tasks, like editing multiple records with a single operation, will be inconvenient and time consuming when the hierarchy. Shortly, we must have the possibility to view the data from any angle. We tried to address this issue by introducing a special type of entity called "Data shortcut" described in 4.4.

4.3 Entity types

Entity type definitions are a set of XML documents that describe:

1) how an entity will be displayed in the treeview (icon, font); the associated editor plugin;
2) which database fields are available, and which editors to use for them;
3) which entity types can be related to each other and which attributes describes those relations;
4) triggered events and available actions when working with an entity.

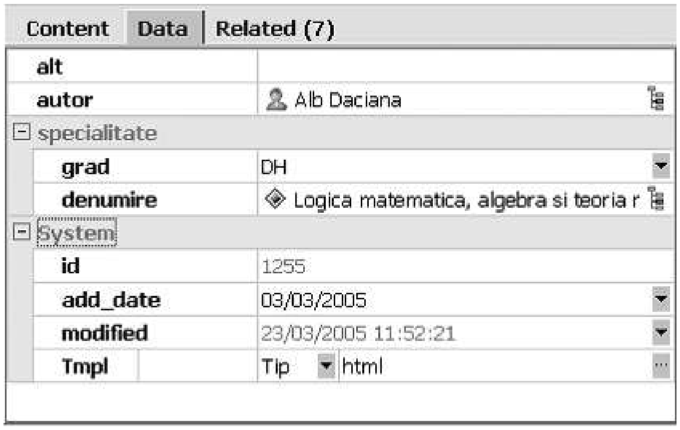
Figure 1 illustrates a snippet from the XML definition of an entity type that represents the db fields displayed. Figure 2 shows the generated UI.

Database fields that contain large texts and require specialized editors with syntax highlighting are described in the <memos> .. </memos> section.

```
...

<memos>
  <f>
      <title> content </title> <lang>en</lang>
      <db_field> content </db_field>
  </f>
</memos>

<linkage mode="tree">
     <types>
           <tip id="1">
                 <fields>
                        <f name="content" height="70" />
                 </fields>
           </tip>
     </types>
</linkage>

<fields>
    <f name="alt" />
    <f name="idx2" caption="autor" editor="TreeDrag" />
    <category caption="specialitate" />
    <f name="grad" caption="grad" width="30" editor="ComboBox">
         <items> <li>D</li> <li>DH</li> </items>
    </f>
    <f name="idx1" caption="denumire" editor="TreeDrag" />
    <category caption="System" expanded="0" />
    <f name="id" readonly="Y" />
    <f name="add_date"> <default>NOW()</default> </f>
    <f name="modified" readonly="Y" />
    <f name="_tmpl" />
</fields>

...
```

Figure 1. A snippet from the XML definition of an entity

Besides standard visual editors (ComboBox, CheckBox, DateEdit, TimeEdit...) available in the previous version of NeoSite [4], we have added two useful complex editors: Lookup and ImgEdit. The ImgEdit allows you to browse for an image on your computer for uploading via



alt	
autor	Alb Daciana
specialitate	
grad	DH
denumire	Logica matematica, algebra si teoria r
System	
id	1255
add_date	03/03/2005
modified	23/03/2005 11:52:21
Tmpl	Tip html

Figure 2. A piece of UI for db field editing

FTP. The uploading directory from the remote server can be obtained by invoking a server side script with the id of the item being edited. The resulting text written to db field is obtained by filling a template like “ <img width=”% w” height=”% h” src=”% src” alt=”% filename” />”.

Such automatizations help a lot when dealing with lots of images. It also relieves the content authors of burden. The Lookup editor displays dataset records within a dropdown window in a column-based format. These records are fetched based on a SQL Select. Figure 3 illustrates the definition of a lookup db field.

NeoSite has a modular architercture based on plugins. A plugin is a DLL library. Each entity type can have it’s own plugin for editing.

4.4 Data Shortcuts

It has been said that hierarchies can be limiting from a navigation perspective and have the possibility to view the data from any angle.

```
<f name="producer_id" editor="Lookup">
   <sql>SELECT id, title FROM site_en
        WHERE parent_id=5  ORDER BY title
   </sql>
   <columns show_header="1">
     <col name="title" caption="" />
     <col name="tel" caption="Phone" width="60" fixed="1" />
  </columns>
</f>
```

Figure 3. Definition of a lookup db field

For this reason, a special editor plugin was developed, called EdSql. It fetches a collection (described by a sql query) of entities from the hierarchy and has two purposes:

1) additional navigation pathway

2) editing multiple records with a single operation (see Figure 4).

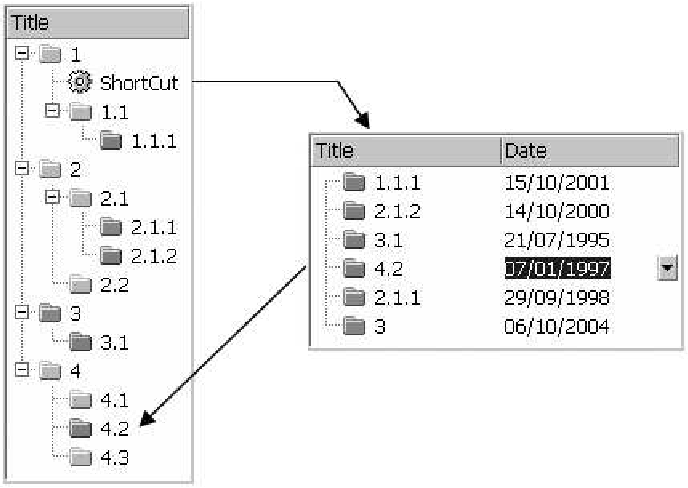
When double-clicking the ‘ShortCut’ node, a grid is displayed containing live data, the user can use grid inplace editing or press ‘Ctrl+Enter’ to find& focus the selected item in the hierarchy.

4.5 Relationships

The basic idea of Entity-Relationship modeling, introduced by Chen [3], is that using sets and relations we can model objects of the real world and their inter-relationships.

At first, when designing the database structure for a website, adjusting it to a relational normal form was a priority. In practice the theoretical model is often constrained by the way in which the data will be used and speed of access; compromises are introduced into the design.

Today we try to model site’s data using only hierarchies and relations between hierarchy nodes. It greatly reduces and simplifies the user interface. Instead of providing a highly customized interface, we



Title	Date
1.1.1	15/10/2001
2.1.2	14/10/2000
3.1	21/07/1995
4.2	07/01/1997
2.1.1	29/09/1998
3	06/10/2004

Figure 4. Data Shortcuts

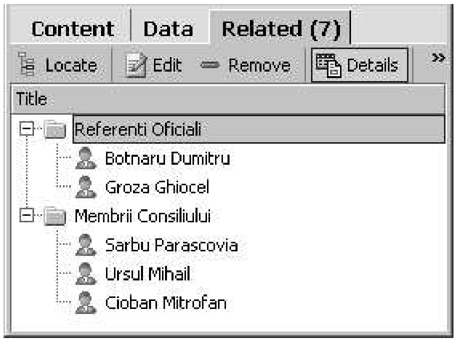
separate complex items into smaller parts, meaningfully organized in the hierarchy. If the context provided by the hierarchy is not sufficient to express the meaning of an item collection, explicit relations and relationship attributes are additionally used to incorporate more "data semantics".

Relations are defined using drag&drop operations.

4.6 Multiple Relationship

Two entities can be related "more than once". To store and display relations, the hierarchy model was used again: in order to relate an item multiple times, you'll have to drag&drop it to different parent nodes Figure 5. 'Referenti Oficiali' and 'Membrii Consiliului' items are abstract entities, automatically related to entities of 'thesis' type on their creation. Besides allowing multirelations, these abstract entities provide context, helping the user to better understand what a given relation means.

Previous version of NeoSite [4] was able to deal only with single



Referenti Oficiali
Botnaru Dumitru
Groza Ghiocel
Membrii Consiliului
Sarbu Parascovia
Ursul Mihail
Cioban Mitrofan

Figure 5. Hierarchical multirelation management

relations, but allowed the user to manage backward references. The practice showed that backward references are seldom used and only bothers the user with additional information.

5 Talking to the Logic

A generic solution will offer only basic functionality. The attempt to endow it with complex features that will encompass a large spectrum of tasks will end up in a flexible (maybe) but hardly manageable software creature. A generic framework should “connect”, not “implement” project specific or narrow features. To accomplish this objective, NeoSite offers a high level event and action mechanism. As we have mentioned in Section 4.3, an entity definition contains the description of its events (see Figure 6) and actions.

Entity action definition is similar to events. Actions are directly activated by the user through a run-time generated toolbar. Each of its buttons have an associated action. Buttons may have an icon specified in the entity definition.

Events are processed consecutively, the output of an event is one of the input parameters of the next event. This model adds workflow capabilities to our system.

```
<events>
   <OnCreate>
       <e type="sql">INSERT INTO site_en_related
                    (from_id, to_id, from_tip, to_tip)
                    VALUES ($id, 444, 10, 1)
       </e>
       <e type="http">
           <params>
                <url>http://neonet.md/cgi/build.cgi</url>
                <prms>id=$id</prms>
                <urgent>0</urgent>
                <method>get</method>
           </params>
       </e>
   </OnCreate>

   <OnCloseAfterCreate>  ...  </OnCloseAfterCreate>
   <OnOpen> ... </OnOpen>
   <OnRelateNode> ...  </OnRelateNode>
   ...
</events>
```

Figure 6. Entity events

6 Related Research and Tools

A generic query tool [8] that dynamically creates its user interface, based on xml configuration files, enables the user to query a metadata store through filters that impose search criteria on attributes. It can be seen as a complementary tool for NeoSite CMS, that shares the same metadata scheme described in a xml file.

Discovering a generic paradigm to manage complex data is a challenging task. The QSByE [6] interface for querying semistructured data allows representing complex objects with arbitrary hierarchy levels, presenting variations in their structure, as nested tables. Such approach can be used to provide an insight of a complex structure, but is not appropriate to edit it.

The Arepo Platform [9] is a set of development tools, used to generate a user interface directly from the database and supplementary metadata. Playing with the online demo, and investigating the docs on their site, we didn't find a way to specify relationships between records.

The closest approach to our tool is ERW [7]: a set of specifications and tools that makes it easy to create, modify and maintain via web a database described by an entity-relationship schema. It has a stable entity-relationship language (ERL) and the associated algorithm. ERW has a well established theoretical foundation, but provides a modest interface to the underlying gears. Because they use a web interface, it makes impracticable to edit large databases. A common "limitation" of both tools: ERW and NeoSite, is that it handles binary relations only. If you really need n-ary relations, you'll have to factor them. The main reason for this limitation is that it is very difficult to design a generic user interface for n-ary relations that will adapt to every situation.

7 Conclusions

This paper has discussed a hierarchical approach to manage entity multirelations in an intuitively clear and convenient way. It also proposed the "Data Shortcut" paradigm as a facility for navigation deep hierarchies and view data from different angles.

Even the grandest project depends on the success of the smallest components. Web projects that were implemented using the proposed approach and software (NeoSite CMS [4]) demonstrate the effectiveness of the proposed model for managing web data.

As future work, we intend to investigate how to build an interface for managing n-ary relationship.

References

[1] Stefano Ceri, Piero Fraternali, Maristella Mareta. *Conceptual Modeling of Data-Intensive Web Applications*, IEEE Internet Computing, August 2002, pag. 20-30

[2] Mary F. Fernandez, Daniela Florescu, Jaewoo Kang, Alon Y. Levy, Dan Suciu. *Catching the Boat with Strudel: Experiences with a Web-Site Management System*, SIGMOD Conference 1998: pp. 414-425

[3] P.P. Chen. *The entity-relationship model: towards a unified view of data*, ACM Trans, on Database Systems, 1(1), 1996, pp. 9-36

[4] Oleg Burlaca, *NeoSite: A Simple Content Management System*, Computer Science Journal of Moldova, vol.12, no.1(34), 2004, http://cms.neonet.md

[5] Louis Rosenfeld, Peter Morville, *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites*, O'Reilly, 2 edition, August, 2002. Chapter 10.4.2.

[6] Irna M.R. Evangelista Filha, Altigran S. da Silva, Alberto H. F. Laender, David W. Embley *Using Nested Tables for Representing and Querying Semistructured Web Data*, Springer-Verlag, 2002.

[7] Sebastiano Vigna. *ERW: Entities and relationships on the web*, Poster Proc. of Eleventh International World Wide Web Conference, Honolulu, USA, 2002. http://vigna.dsi.unimi.it/

[8] B. Verhoeven, E. Duval, H. Olivie, *A Generic Metadata Query Tool*, WebNet, pp. 1122 - 1127

[9] Arepo Solutions Ltd, http://www.arepo.com

O. Burlaca,

Received April 12, 2005

Institute of Mathematics and Computer Sciences,
Academy of Sciences, Moldova
5, Academiei Str., Kishinev,
MD2028, Moldova
E–mail: *oburlaca@neonet.md*

Variable Bit Permutations: Linear Characteristics and Pure VBP-Based Cipher

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

Abstract

This paper describes linear characteristics of the variable bit permutations (VBP) that are used in the form of the data-dependent permutations. This primitive suites well to the design of fast cheap-hardware-oriented ciphers. Because of the existence of one characteristic with bias 1/2 we discuss possibility to design a pure VBP-based block ciphers that are indistinguishable from a random transformation. We present design of the cipher which is based only on VBP, fixed permutations, and XOR operations. Performed analysis has shown that the designed pure VBP-based block cipher is secure against differential and linear attacks confirming the efficiency of the VBP as cryptographic primitive.

Key words: variable bit permutations, data-dependent permutations, linear analysis, fast block cipher

1 Introduction

Permutation networks (PNs) have been widely studied in the field of parallel processing and telephone switching systems [1] and they are very interesting to be used as cryptographic primitives. The PNs are well suited for cryptographic applications, since they allow one to specify and perform permutations at the same time. A variant of the symmetric cryptosystem based on the key-controlled PNs and Boolean functions is presented in [2]. Another cryptographic application of PNs is presented by the cipher ICE [3] in which a very simple PN is used to specify a key-dependent fixed permutation. Such use of PNs has been shown [4] to be not very effective against differential cryptanalysis. A more attractive approach is the use of PNs to perform variable bit permutations (VBP) implemented as data-dependent permutations (DDP) [5]. Efficiency of the use of data-dependent operations has been demonstrated by examples of ciphers RC5 [6], RC6 [7] and MARS [8], which are based on data-dependent rotations with 32 different modifications. The PNs can be used as controlled permutation (CP) boxes to perform DDP. It is easy to design CP boxes (CPBs) giving possibility to specify 2^{64} and more different modifications of the VBP performed on data subblocks [5] and subkeys [9].

This paper considers the linear characteristics of VBP, design of the pure VBP-based cipher oriented to cheap hardware implementation, and its security against differential and linear attacks. In section 2 we consider general design of the CP boxes. We also construct mutually inverse CP boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$ of the order $h = 2$, (see Definition 3) both of them having the same topology. In section 3 we consider algebraic and probabilistic properties of CP. Linear characteristics of the CP boxes are estimated in general case. In section 4 a pure VBP-based cipher DDP-64 using simple key scheduling is described. In section 5 the linear and differential analysis of DDP-64 is considered. We also propose to use switchable operations to avoid weak keys and homogeneity of the encryption in the case of simple key scheduling.

Notation

$\diamond$ Let $\{0,1\}^s$ denote the set of all binary vectors $U = (u_1, ..., u_s)$, where $\forall i \in \{1, ..., s\}$ $u_i \in \{0,1\}$.

$\diamond$ The Hamming weight $\varphi(U)$ of U be defined as the number of nonzero components of U and $\varphi'(U)$ denote the parity of $\varphi(U)$, i.e. $\varphi(U) \overset{def}{=} \sum_{i=1}^{s} u_i$, where $\varphi(U) \in \{0, 1, ..., s\}$ and $\varphi'(U) \overset{def}{=} \varphi(U) \bmod 2$.

$\diamond$ Let us fix i, $i \in \{1, ..., s\}$, and write E_i, for which $\varphi(E_i) = 1$ and $e_i = 1$.

$\diamond$ Let $E_0 = (0, ..., 0)$ and $D_0 = (1, ..., 1)$, i.e. $\varphi(E_0) = 0$ and $\varphi(D_0) = s$.

$\diamond$ Let $X \oplus Y$ denote the bit-wise XOR (EXCLUSIVE-OR) operation of the two vectors X and $Y :$ $X, Y \in \{0,1\}^s$.

$\diamond$ Let $X \otimes Y$ denote bit-wise AND operation of the two vectors

X and Y: $X, Y \in \{0,1\}^s$. For $c \in \{0,1\}$ and $X \in \{0,1\}^s$ we define $Y = c \cdot X$, where $y_i = c \cdot x_i$ $\forall i \in \{1, ..., s\}$.

$\diamond$ $\overline{U}$ denotes bit-wise complement of U, i.e. $\overline{U} \overset{def}{=} U \oplus D_0$ $\forall U \in \{0,1\}^s$.

$\diamond$ Let $\bullet$ denote the binary scalar product: $c = A \bullet X = \varphi'(A \otimes X)$ $(c \in \{0,1\})$.

$\diamond$ Let $Y = X^{\ggg k}$ denote rotation of the word X by k bits, where $\forall i \in \{1, ..., n-k\}$ we have $y_i = x_{i+k}$ and $\forall i \in \{n-k+1, ..., n\}$ we have $y_i = x_{i+k-n}$.

2 Design of fast CP boxes

Let $Y = \mathbf{F}(X, V)$ be the two-variable function $\mathbf{F}: \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n$.

Definition 1. *Function $\mathbf{F}(X, V)$ is called a CP box (or $\mathbf{P}_{n/m}$-box), if for each fixed V the function $\mathbf{F}(X, V)$ is a bijective mapping defined as bit permutation.*

For fixed V we have fixed bit permutation operation called CP modification or modification of VBP operation. We shall denote modifications as $\mathbf{F}_V$ or $\mathbf{P}_V$. We shall also use notation $\mathbf{P}_{n/m}^{(V)}$ for CPB with n-bit input, n-bit output, and m-bit control input . Thus, the notation $Y = \mathbf{P}_{n/m}(X, V) = \mathbf{P}_{n/m}^{(V)}(X)$ means $Y = \mathbf{P}_V(X)$.

In section 3.2 we use the following statements:

1. $\mathbf{F}_V(A) = B \Rightarrow \mathbf{F}_V(\overline{A}) = \overline{B}$.
2. $\mathbf{F}_V(A \otimes B) = \mathbf{F}_V(A) \otimes \mathbf{F}_V(B)$.
3. $\mathbf{F}_V(A) = B \Rightarrow \varphi(A) = \varphi(B)$ and $\varphi(A) \neq \varphi(B) \Rightarrow \mathbf{F}_V(A) \neq B$.
4. $\varphi(A \oplus B) = \varphi(A) + \varphi(B) - \varphi(A \otimes B)$. If $A \otimes B = E_0$, then $\varphi(A \oplus B) = \varphi(A) + \varphi(B)$.

While constructing CPBs it is preferable to use the layered topology of PNs, since it permits to design very fast CPBs. A layered CPB $\mathbf{P}_{n/m}$ (Fig. 1) can be represented as superposition of $s = 2m/n$ active layers separated with $s-1$ fixed permutations that are implemented in hardware as simple connections. Each active layer (Fig. 1b) in a CPB with n-bit input is represented by the set of $n/2$ elementary boxes $\mathbf{P}_{2/1}$ controlled with one bit v: $y_1 = x_{1+v}$ and $y_2 = x_{2-v}$ (see Fig. 1a). General structure of the layered CPB is shown in Fig. 1c.

In all figures in this paper the solid lines indicate data movement, while dotted lines indicate the controlling bits. We assume that in a layered CP box all elementary switching elements are consecutively numbered from left to right and from top to bottom and the ith bit of vector V controls the ith switching element $\mathbf{P}_{2/1}$. In accordance with the number of layers the vector V can be represented as concatenation of s vectors $V_1, V_2, ..., V_s \in \{0, 1\}^{n/2}$, i.e. $V = (V_1, V_2, ..., V_s) = V_1|V_2|...|V_s$.

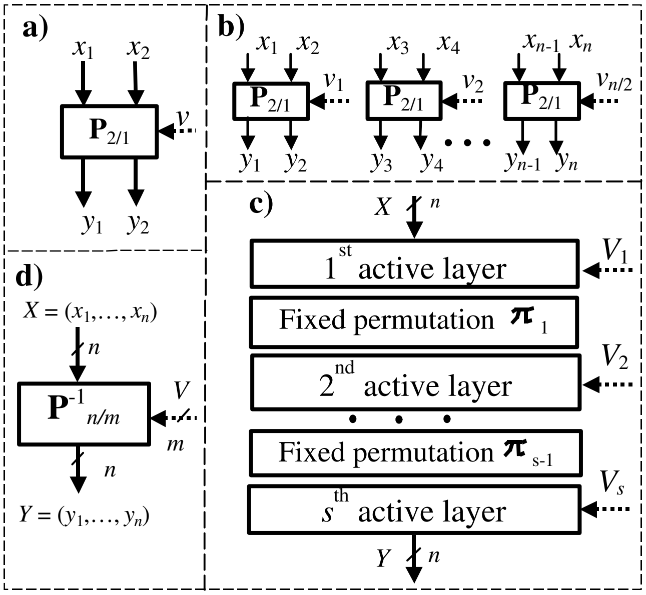


Figure 1. Notation of the $\mathbf{P}_{2/1}$- (a) and $\mathbf{P}^{-1}_{n/m}$-boxes (d), structure of one active layer (b) and general structure of the layered CP boxes (c)

The following two definitions we use according to [5].

Definition 2. *The CP boxes $\mathbf{P}_{n/m}$ and $\mathbf{P}^{-1}_{n/m}$ are mutual inverses, if for all possible values of the vector V the corresponding CP modifications $\mathbf{P}_V$ and $\mathbf{P}^{-1}_V$ are mutual inverses.*

Definition 3. *Suppose for arbitrary $h \leq n$ input bits $x_{\alpha_1}, x_{\alpha_2}, ..., x_{\alpha_h}$ and arbitrary h output bits $y_{\beta_1}, y_{\beta_2}, ..., y_{\beta_h}$ there is at least one value V which specifies a permutation $\mathbf{P}_V$ moving x_{α_i} to y_{β_i} for all $i = 1, 2, ..., h$. Such a $\mathbf{P}_{n/m}$-box is called a CP box of the order h.*

One active layer can be considered as the single-layer CPB $\mathbf{S}_n$. It is evidently that $\mathbf{P}_{2/1} = \mathbf{P}_{2/1}^{-1}$, therefore $\mathbf{S}_n = \mathbf{S}_n^{-1}$. A layered CPB $\mathbf{P}_{n/m}$ can be represented as superposition of the bit permutation operations: $\mathbf{P}_{n/m} = \mathbf{S}_n^{(V_1)} \circ \pi_1 \circ \mathbf{S}_n^{(V_2)} \circ \pi_2 \circ ... \circ \pi_{S-1} \circ \mathbf{S}_n^{(V_s)}$. The respective box $\mathbf{P}_{n/m}^{-1}$ has the following structure $\mathbf{P}_{n/m}^{-1} = \mathbf{S}_n^{(V_s)} \circ \pi_{s-1}^{-1} \circ \mathbf{S}_n^{(V_{s-1})} \circ \pi_{s-2}^{-1} \circ ... \circ \pi_1^{-1} \circ \mathbf{S}_n^{(V_1)}$. Thus, to construct inverse of the CP box $\mathbf{P}_{n/m}$ it is sufficient to number the boxes $\mathbf{P}_{2/1}$ from left to right and *from bottom to top* and to replace π_i by π_{s-i}^{-1}. We shall assume that in the boxes $\mathbf{P}_{n/m}^{-1}$ the switching elements $\mathbf{P}_{2/1}$ are consecutively numbered from left to right and from bottom to top, i.e. in the both $\mathbf{P}_{n/m}$ and $\mathbf{P}_{n/m}^{-1}$ the ith bit of V controls the ith elementary box $\mathbf{P}_{2/1}$. Note that the vector V_j corresponding to the jth active layer in the box $\mathbf{P}_{n/m}$ controls the $(s-j+1)$th active layer in $\mathbf{P}_{n/m}^{-1}$.

In the VBP-based ciphers described below there are used $\mathbf{P}_{32/96}$- and $\mathbf{P}_{32/96}^{-1}$-boxes. These boxes have the same structure comprising two subsequent cascades. The upper one consists of four boxes $\mathbf{P}_{8/12}$ (Fig. 2a) and the lower one consists of four parallel boxes $\mathbf{P}_{8/12}^{-1}$ (Fig. 2b). The cascades are separated with fixed permutational involution described as follows:

$$\begin{array}{c}(1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)(11,18)\\(12,26)(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32)\end{array}.$$

The structure of the boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}_{32/96}^{-1}$ is presented in Fig. 2c and 2d. One can show that both of these CP boxes have the second order and each of two superpositions $\left(\mathbf{P}_{32/96}^{(V)}\right)^{-1} \circ \mathbf{P}_{32/96}^{(V')}$ and $\mathbf{P}_{32/96}^{(V)} \circ \left(\mathbf{P}_{32/96}^{(V')}\right)^{-1}$ represent a twelve-layer CPB of the order $h = 32$.

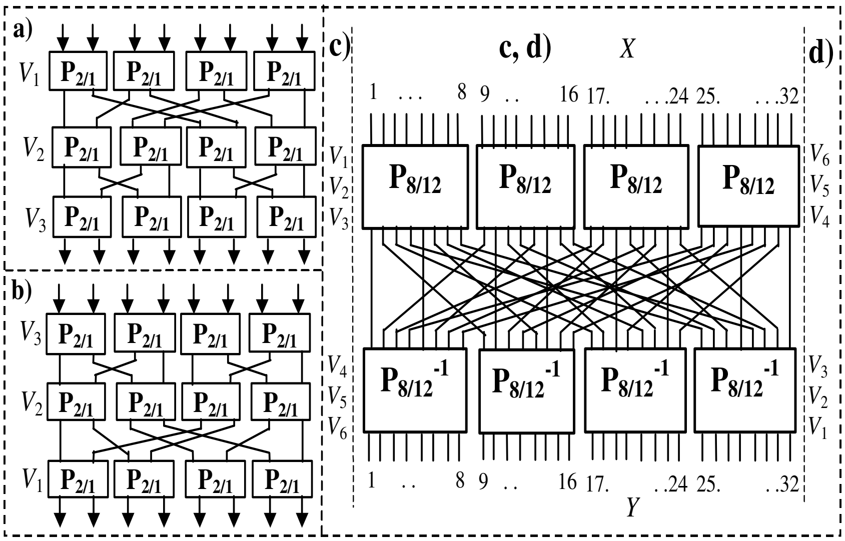


Figure 2. The first-order boxes $\mathbf{P}_{8/12}$ (a) and $\mathbf{P}_{8/12}^{-1}$ (b) and the second-order boxes $\mathbf{P}_{32/96}$ (c) and $\mathbf{P}_{32/96}^{-1}$ (d)

3 Properties of the controlled permutations

3.1 Terms of the linear cryptanalysis

Let $\mathbf{F} : \{0,1\}^r \to \{0,1\}^n, (r \geq n)$ be given. The resistance of the function against linear cryptanalysis (LCA) [10,11] is determined by the maximal value $|p|$, where

$$p = p_F = p_F(\Gamma u, \Gamma y) \overset{def}{=} \Pr_U(U \bullet \Gamma u \oplus Y \bullet \Gamma y = 0) - \frac{1}{2}, \qquad (1)$$

$U, \Gamma u \in \{0,1\}^r$, $Y \in \{0,1\}^n$, $\Gamma y \in \{0,1\}^n \setminus E_0$, and Γu, Γy are fixed vectors, that we called masks, and the value p is called the deviation (or bias). Similar to the notation used in [11] we describe linear characteristic (LC) of the function $\mathbf{F}$ as the combination $(\Gamma u, \Gamma y, p_F)$.

Let $Y = \mathbf{F}(X, V)$ be the two-variable function $\mathbf{F}$: $\{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$. Then for $U = X|V$, and $\Gamma u = \Gamma x|\Gamma v$ (1) is transformed in

$$p_F(\Gamma x, \Gamma y, \Gamma v) =$$
$$= p_F(\Gamma x|\Gamma v, \Gamma y) \overset{def}{=} \Pr_{X|V}(X \bullet \Gamma x \oplus V \bullet \Gamma v \oplus Y \bullet \Gamma y = 0) - \frac{1}{2}. \quad (2)$$

In particular, the value V corresponds to a subkey. For fixed value V the deviation has the form

$$p_{F_V}(\Gamma x, \Gamma y, \Gamma v) =$$
$$= p_{F_V}(\Gamma x|\Gamma v, \Gamma y) \overset{def}{=} \Pr_{X}(X \bullet \Gamma x \oplus V \bullet \Gamma v \oplus Y \bullet \Gamma y = 0) - \frac{1}{2}. \quad (3)$$

Below we shall also use the LC with the value p_{F_V}. According to different papers that dealt with the case of the uniformly distributed independent variables X and V the resistance of the function $\mathbf{F}$ against LCA can be estimated with the help of formulas using deviations p_{F_V}. For example, in [11] it has been derived the following formula:

$$LP^F_{max} \overset{def}{=}$$
$$\overset{def}{=} \max_{\Gamma x, \Gamma y \neq 0} LP^F(\Gamma x \to \Gamma y) \overset{def}{=} \max_{\Gamma x, \Gamma y \neq 0} \frac{1}{2^m} \sum_{V \in \{0,1\}^m} LP^{F_V}(\Gamma x \to \Gamma y),$$

where $\quad LP^{F_V}(\Gamma x \to \Gamma y) \overset{def}{=} (2\frac{\#\{X \in \{0,1\}^n : X \bullet \Gamma x = F_V(X) \bullet \Gamma y\}}{2^n} - 1)^2.$

Actually $$LP^{F_V}(\Gamma x \to \Gamma y) = (2p_{F_V}(\Gamma x, \Gamma y))^2, \quad (4)$$

where $$p_{F_V}(\Gamma x, \Gamma y) = \Pr_X(X \bullet \Gamma x \oplus F_V(X) \bullet \Gamma y = 0) - \frac{1}{2}. \quad (5)$$

The next section of the present paper considers the case of arbitrary distribution of the variable V including the case $V = const$. In this calculation of the deviation p it is necessary to use the total probability formula:

$$p_F(\Gamma x, \Gamma y, \Gamma v) = \sum_V p_{F_V}(\Gamma x, \Gamma y, \Gamma v) \cdot P_V, \quad (6)$$

where P_V is the probability of the given value V.

3.2 Linear characteristics

Analyzing LC of CP boxes we assume, that X and V are independent variables, and $P_X = \frac{1}{|\{X\}|} = \frac{1}{2^n}$, secondly, for masks Γx, Γy, Γv we will use identifications A, B, C, in accordance with [11]. Let

$$\lambda_F(A,B) \overset{def}{=} \sum_V \theta_{F_V}(A,B) \cdot P_V, \tag{7}$$

where $\quad \theta_{F_V}(A,B) \overset{def}{=} \begin{cases} 1, & \text{if } \mathbf{F}_V(A) = B; \\ 0, & \text{otherwise.} \end{cases}$

Note that $\lambda_F(A,B)$ is the probability that $\mathbf{F}(A){=}B$, i.e. $P_F(A{\to}B) \overset{def}{=} \lambda_F(A,B)$. Accordingly, let

$$\lambda_F(A,B,C) \overset{def}{=} \sum_V \theta_{F_V}(A,B,C) \cdot P_V, \tag{8}$$

where $\quad \theta_{F_V}(A,B,C) \overset{def}{=} \begin{cases} 1, & \text{if } \mathbf{F}_V(A) = B,\ C \bullet V = 0; \\ 0, & \text{otherwise.} \end{cases}$

Below we use the following statements:

1. $\varphi(A) \neq \varphi(B) \Rightarrow \mathbf{F}_V(A) \neq B\ \forall V \Rightarrow \lambda_F(A,B) = 0$.
2. $\varphi(A) = \varphi(B) \Rightarrow \lambda_F(\overline{A},\overline{B}) = \lambda_F(A,B)$.
3. $\lambda_F(A,B) = \lambda_F(A,B,E_0)$.
4. $\forall C\ \lambda_F(A,B,C) \leq \lambda_F(A,B)$.

These statements are quite evident and can be easy derived using general properties of permutations. Let us consider the function

$$\Psi_V(X) \overset{def}{=} \begin{cases} 1, & \text{if } A \bullet X \oplus B \bullet \mathbf{F}_V(X) \oplus C \bullet V = 0\ ; \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to see, that if $P_X = \frac{1}{|\{X\}|}$, then

$$p_{F_V}(A,B,C) = \frac{1}{|\{X\}|} \cdot \sum_X \Psi_V(X) - \frac{1}{2}. \tag{9}$$

Lemma. *Let X and V be independent variables and $P_X = \frac{1}{|\{X\}|}$. Then*

$$\frac{1}{|\{X\}|} \cdot \sum_X \Psi_V(X) = \begin{cases} \theta_{F_V}(A,B,C), & \textit{if } \mathbf{F}_V(A) = B; \\ 1/2, & \textit{if } \mathbf{F}_V(A) \neq B. \end{cases} \tag{10}$$

Proof. Let us consider two variants: $\mathbf{F}_V(A) = B$ and $\mathbf{F}_V(A) \neq B$.

Case 1. $\mathbf{F}_V(A) = B \Rightarrow \forall X:\ A \bullet X \oplus B \bullet \mathbf{F}_V(X) =$
$= A \bullet X \oplus \mathbf{F}_V(A) \bullet \mathbf{F}_V(X) = \varphi'(A \otimes X) \oplus \varphi'(\mathbf{F}_V(A) \otimes \mathbf{F}_V(X)) =$
$= \varphi'(A \otimes X) \oplus \varphi'(\mathbf{F}_V(A \otimes X)) = \varphi'(U) \oplus \varphi'(\mathbf{F}_V(U)) \equiv 0.$
Hence, $\forall X\ \Psi_V(X) = C \bullet V \oplus 1$ and

$$\frac{1}{|\{X\}|} \cdot \sum_{X, \mathbf{F}_V(A) = B} \Psi_V(X) = \begin{cases} 1, & \text{if } C \bullet V = 0; \\ 0, & \text{if } C \bullet V \neq 0. \end{cases} = \theta_{F_V}(A, B, C).$$

Case 2. $B \neq \mathbf{F}_V(A)$. Let $A':\ B = \mathbf{F}_V(A')$. It is obvious, that $A' \neq A$. Let $A^{(1)} = \overline{A'} \otimes A$, $A^{(12)} = A \otimes A'$ and $A^{(2)} = \overline{A} \otimes A'$. Since $A' = A^{(12)} \oplus A^{(2)}$ and ${}^{(12)} \otimes A^{(2)} = E_0$ are held, then $\varphi(A' \otimes X) = \varphi(A^{(12)} \otimes X \oplus A^{(2)} \otimes X) = \varphi(A^{(12)} \otimes X) + \varphi(A^{(2)} \otimes X)$ and $\varphi(A \otimes X) = \varphi(A^{(1)} \otimes X) + \varphi(A^{(12)} \otimes X)$.
According to Case 1, $\forall X$ the equation $A' \bullet X \oplus B \bullet \mathbf{F}_V(X) = 0$ is held, i.e. $B \bullet \mathbf{F}_V(X) = A' \bullet X$. Then $A \bullet X \oplus B \bullet \mathbf{F}_V(X) = A \bullet X \oplus A' \bullet X =$
$= \varphi'(A \otimes X) \oplus \varphi'(A' \otimes X) =$
$= \varphi'(A^{(1)} \otimes X) \oplus \varphi'(A^{(12)} \otimes X) \oplus \varphi'(A^{(12)} \otimes X) \oplus \varphi'(A^{(2)} \otimes X) =$
$= \varphi'(A^{(1)} \otimes X) \oplus \varphi'(A^{(2)} \otimes X) =$
$= \left(\sum_{i|a_i^{(1)}=1} x_i\right) \bmod 2 \oplus \left(\sum_{i|a_i^{(2)}=1} x_i\right) \bmod 2 \Rightarrow$

$$\Psi_V(X) = \left(\sum_{i|a_i^{(1)}=1} x_i\right) \bmod 2 \oplus \left(\sum_{i|a_i^{(2)}=1} x_i\right) \bmod 2 \oplus C \bullet V \oplus 1.$$

Let $t = \varphi(A^{(2)})$. The whole set of binary vectors $\{X\}$ ($|\{X\}| = 2^n$) can be represented as an association of 2^{n-t} disjoint subsets, each of which contains 2^t vectors differing only in the digits corresponding to the active (non-zero) bits of the mask $A^{(2)}$. Note that for each such subset $\left(\sum_{i|a_i^{(1)}=1} x_i\right) \bmod 2$ and $C \bullet V \oplus 1$ are constants and $\left(\sum_{i|a_i^{(2)}=1} x_i\right) \bmod 2$ is even in exactly one-half cases, i.e. for $2^{n-t} \cdot 2^{t-1} = 2^{n-1}$ values X $\Psi_V(X) = 0$ and for remaining 2^{n-1} values X $\Psi_V(X) = 1$ are held. Therefore, for each permutations $\mathbf{F}_V$: $\mathbf{F}_V(A) \neq$

B we have

$$\frac{1}{|\{X\}|} \cdot \sum_{X, \mathbf{F}_V(A) \neq B} \Psi_V(X) = \frac{2^{n-1}}{2^n} = \frac{1}{2}. \qquad \square$$

Theorem 1. (About deviations of controlled permutations). *Let X and V be independent variables and $P_X = \frac{1}{|\{X\}|}$. Then*

$$2p_F(A,B,C) = 2\lambda_F(A,B,C) - \lambda_F(A,B) \tag{11}$$

Proof. In accordance with (6),(9) and (10) we have

$$2p_F(A,B,C) = 2\sum_V p_{F_V}(A,B,C) \cdot P_V =$$

$$= 2 \sum_{V, \mathbf{F}_V(A)=B} P_V \cdot p_{F_V}(A,B,C) + 2 \sum_{V, \mathbf{F}_V(A) \neq B} P_V \cdot p_{F_V}(A,B,C) =$$

$$= 2 \sum_{V, \mathbf{F}_V(A)=B} P_V \cdot \left(\theta_{F_V}(A,B,C) - \frac{1}{2}\right) + 2 \sum_{V, \mathbf{F}_V(A) \neq B} P_V \cdot \left(\frac{1}{2} - \frac{1}{2}\right) =$$

$$= 2\sum_V P_V \cdot \theta_{F_V}(A,B,C) - \sum_V P_V \cdot \theta_{F_V}(A,B) = 2\lambda_{F_V}(A,B,C) - \lambda_{F_V}(A,B).$$

$\square$

Corollary 1. $2p_F(A,B) = 2p_F(A,B,E_0) = P_F(A \to B)$.

Indeed,
$2p_F(A,B) = 2p_F(A,B,E_0) = 2\lambda_{F_V}(A,B,E_0) - \lambda_{F_V}(A,B) =$
$= 2\lambda_{F_V}(A,B) - \lambda_{F_V}(A,B) = \lambda_{F_V}(A,B) = p_F(A \to B)$. $\square$

Corollary 2. $\forall A, B \;\; 0 \leq 2p_F(A,B) \leq 1$.

Corollary 3. $\varphi(A) \neq \varphi(B) \Rightarrow p_F(A,B) = 0$.

Indeed,
$\varphi(A) \neq \varphi(B) \Rightarrow B \neq \mathbf{F}_V(A) \; \forall V \;\; \Rightarrow \lambda_F(A,B) = 0 \Rightarrow p_F(A,B) = 0$. $\square$

Corollary 4. $\forall C \; |p_F(A,B,C)| \leq p_F(A,B)$.

Indeed, $0 \leq \lambda_F(A,B,C) \leq \lambda_F(A,B) \Rightarrow |2p_F(A,B,C)| =$ $= |2\lambda_F(A,B,C) - \lambda_F(A,B)| \leq \lambda_F(A,B) = 2p_F(A,B) \Rightarrow$ $\Rightarrow |p_F(A,B,C)| \leq p_F(A,B)$. □

Corollary 5. $\forall A,B \ \sum_{B:\varphi(B)=\varphi(A)} 2p_F(A,B) = 1$,
$\sum_{A:\varphi(A)=\varphi(B)} 2p_F(A,B) = 1$

Indeed, $\forall A,V \quad \exists! B : B = \mathbf{F}_V(A) \Rightarrow \sum_B \lambda_F(A,B) = 1$. Since $\sum_{B:\varphi(B)\neq\varphi(A)} \lambda_F(A,B) = 0$, then $\sum_B \lambda_F(A,B) =$ $= \sum_{B:\varphi(B)=\varphi(A)} \lambda_F(A,B) = 1 \Rightarrow \ \sum_{B:\varphi(B)=\varphi(A)} 2p_F(A,B) =$ $= \sum_{B:\varphi(B)=\varphi(A)} \lambda_F(A,B) = 1$. The second formula can be similarly derived. □

Corollary 6. $\forall A,B \ p_F(\overline{A},\overline{B}) = p_F(A,B)$. This is obvious, since $\lambda_F(\overline{A},\overline{B}) = \lambda_F(A,B)$. □

Corollary 7. Let $A = B = E_0$ or $A = B = D_0$. Then $2p_F(A,B) = 1$.

Corollary 8. Let $\varphi(A) = 1$. If $P_F(A \to B) = const \ \forall B \in \{0,1\}^n : \ \varphi(B) = 1$, then $2p_F(A,B) = P_F(A \to B) = \frac{1}{n}$, i.e. $p_F(A,B) = \frac{1}{2}(P_F(A \to B) = \frac{1}{2n}$.

Indeed, there exist exactly n of vectors $B \in \{0,1\}^n : \ \varphi(B) = 1$, therefore $P_F(A \to B) = const = 1/n$. This result was obtained in [12] for data-dependent rotations which are a particular case of the CP operations. □

Conclusion *The absolute value of the deviation of LC with a non-zero mask of the controlling vector does not exceed the value of the deviation of LC with a zero mask of V, the last being equal to half of the probability that a given input mask transforms into a given output mask.*

Since the theorem and its corollaries include the doubled value of deviation, it is reasonable to use the parameter $p' = |2p|$ which for zero mask of the controlling vector coincide with the probability $P_F(A \to B)$, i.e. $p' = |2p| = 2p_F(A,B) = P_F(A \to B)$.

The practical significance of the derived theorem and its corollaries lies in simplification of the calculation of LC:

1. To estimate LC of the CP boxes one can analyze only linear characteristics with zero mask of the controlling vector V.

2. The calculation of the deviation $p_F(A,B)$ is equivalent to the calculation of the probability $P_F(A \to B)$.

3. It is sufficient to analyze only deviations of LC for which $\varphi(A) \leq n/2$.

4. While designing CP boxes, the condition $\forall A, B \notin \{E_0, D_0\}$ $P_F(A \to B) \leq 1/n$ is to be satisfied.

Let us consider item 4. Let $\xi(t) \overset{def}{=} \max_{A,B:\varphi(A)=t} P_F(A \to B)$ be the function of the maximum value $P_F(A \to B)$ for given weight t. It is easy to see, that $\xi(n-t) = \xi(t)$, therefore it is enough to analyze this function only for $t = \{1, ..., n/2\}$. Since the number of different vectors B with weight $\varphi(B) = 1$ equals n, for $\varphi(A) = 1$ from corollary 5 one can obtain $\xi(t) \geq 1/n$. If $\forall A, B \in 0, 1^n : \varphi(A) = \varphi(B) = 1$ we have $P_F(A \to B) = const$, then $\xi(t) = 1/n$ is held. This is the case of the uniform CP boxes of the first order.

For arbitrary $t \leq n$ the number of different vectors B with weight t equals $\binom{n}{t}$. Thus, there are premises of the construction of CP boxes with monotonically descending function $\xi(t)$ for $t = \{1, ..., n/2\}$. For CP boxes of the hth order we have $P_F(A \to B) > 0$ $\forall A, B : \varphi(A) = \varphi(B) = h$. The approximately uniform CP boxes are characterized by the condition $P_F(A \to B) \approx const$ [5], hence for them we have $P_F(A \to B) \approx 1/q$, where $q = \binom{n}{h}$. A necessary condition for construction of such CP boxes is the inequality $2^m \geq \binom{n}{h}$, where m is the length of the controlling vector. However for weight 1 it is impossible to design a CP box with $p_F < \frac{1}{2n}$. While designing ciphers the VBP operations should be combined with other operations in order to thwart linear attacks using the LC with masks $A = B = (1, ..., 1)$ (see, for example, the use of the special nonlinear operation **G** in the cipher SPECTR-H64 [13]).

4 The block cipher DDP-64

While designing the single key cryptosystem DDP-64 our strategy was oriented to the extensive use of the controlled operations in the form of the CP box operations. This cryptographic primitive is fast and inexpensive while implementing in hardware. Our design criteria were the following:

1. The cryptosystem should be an iterated 64-bit cipher.
2. The cryptalgorithm should be able to perform encryption and decryption with simple and fast change of the sequence of the used subkeys.
3. The cipher should be fast in the case of frequent change of keys. For this reason we do not use precomputations.
4. Round transformation of data subblocks should be characterized by high parallelism.
5. The cipher should use only DDP as basic cryptographic primitive (therefore it is called DDP-64).

When designing a cipher based only on XOR and bit permutations (fixed and data-dependent ones) one of important problems is that combination of these operations usually gives ciphers which have the following property: "the parity of the plaintext + the parity of the key = the parity of the ciphertext". Such ciphers are not pseudorandom. To avoid this problem we have constructed a special operational box **F** that is based on fixed and data-dependent permutations. Structure of this box provides the arbitrary change of the oddness of the output. General structure of the encryption round proposed in [14] and implemented in the cipher SPECTR-H64 suites very well to satisfy our design criteria, therefore we have used SPECTR-H64 as a prototype when developing the pure VBP-based cipher DDP-64 having 64-bit input. The general encryption scheme of DDP-64 is described by the following formulas:

$$C = \mathbf{T}^{(e=0)}(M, K) \quad \text{and} \quad M = \mathbf{T}^{(e=1)}(C, K),$$

where M is the plaintext, C is the ciphertext ($M, C \in \{0,1\}^{64}$), K is the secrete key ($K \in \{0,1\}^{128}$), $\mathbf{T}$ is the transformation function, and

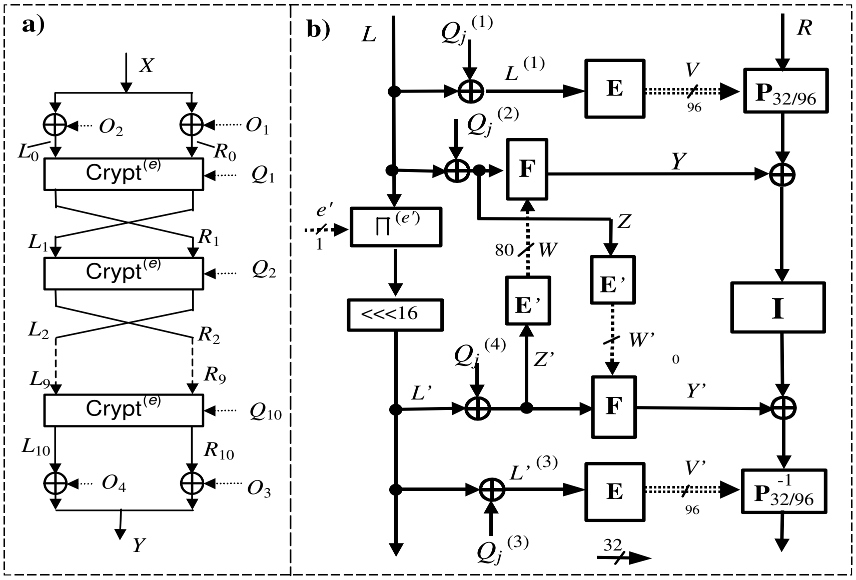


Figure 3. General structure of DDP-64 (a) and procedure $\mathbf{Crypt}^{(e)}$ (b)

$e \in \{0, 1\}$ is a parameter defining encryption ($e = 0$) or decryption ($e = 1$) mode. The secrete key is considered as concatenation of four 32-bit subkeys K_i, $i = 1, 2, 3, 4$: $K = (K_1, K_2, K_3, K_4,)$. DDP-64 uses no preprocessing to transform subkeys. Iterative structure of DDP-64 is shown in Fig. 3a and can be described as follows. First data block X is divided into two 32-bit subblocks L and R and initial transformation is performed as XORing subblocks with corresponding subkeys. Then 10 rounds with procedure $\mathbf{Crypt}^{(e)}$ followed by final transformation are performed. The structure of the procedure $\mathbf{Crypt}^{(e)}$ is shown in Fig. 3b.

4.1 Formation of the round keys

Each round key $Q_j = (Q_j^{(1)}, Q_j^{(2)}, Q_j^{(3)}, Q_j^{(4)}) \in \{0, 1\}^{32}$ is some e-dependent transposition of the subkeys K_1, K_2, K_3, K_4. Figure 4 and Table 1 specify round subkeys and their correspondence to the secret key. Subkeys K_i $(i = 1, ..., 4)$ are used directly in each round avoiding any processing them. The transposing subkeys K_1, K_2, K_3, K_4 is performed with two boxes $\mathbf{P}^{(e)}_{2\times 32/1}$. The box $\mathbf{P}^{(e)}_{2\times 32/1}$ is some single-layer CPB in which all elementary switching elements are controlled with the same bit e. The pairs (K_1, K_3) and (K_2, K_4) are inputs of the corresponding boxes $\mathbf{P}^{(e)}_{2\times 32/1}$. Four 32-bit outputs of two boxes $\mathbf{P}^{(e)}_{2\times 32/1}$ are the e-dependent subkeys O_i $(i = 1, 2, 3, 4)$. Thus, we have $O_i = K_i$, if $e = 0$, and $O_1 = K_3$, $O_2 = K_4$, $O_3 = K_1$, $O_4 = K_2$, if $e = 1$. Being free of any precomputing subkeys and using the same algorithm to perform encryption and decryption the cipher DDP-64 suites well to cheap hardware implementation.

The left data subblock combined with subkeys $Q^{(1)}_j$ and $Q^{(3)}_j$ is used to form the controlling vectors V and V' which specify the current modifications of the VBP performed on the right data subblock with boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$, respectively. The left data subblock combined with subkeys $Q^{(2)}_j$ and $Q^{(4)}_j$ is also transformed with two $\mathbf{F}$-boxes implementing special variant of VBP.

4.2 Switchable fixed permutations

Change of the ciphering mode is defined by swapping subkeys K_i with two single-layer boxes $\mathbf{P}^{(e)}_{2\times 32/1}$ (see Fig. 4a) and by switching the e'-dependent fixed permutation $\Pi^{(e')}$, where $e' \in \{0, 1\}$ and e' depends on e and on the round number j. The e'-dependent fixed permutation in the left branch of the cryptoscheme is used to prevent homogeneity of the encryption procedure in the case of the key having structure $K = (X, X, X, X)$. For this reason the schedule of the switching bit e' is non-periodic (see Table 1). The structure of the switchable operations $\Pi^{(e')}$ is shown in Fig. 4b. It is easy to see that we have $\Pi^{(0)} = \Pi$, $\Pi^{(1)} = \Pi^{-1}$, and $\Pi^{(e'\oplus 1)}(Y) = X$, if $Y = \Pi^{(e')}(X)$. The permutation Π is specified as follows:

$$(1,4,7,2,5,8,3,6)(9,12,15,10,13,16,11,14)$$
$$(17,20,23,18,21,24,19,22)(25,28,31,26,29,32,27,30).$$

Table 1. Specification of the round subkeys and switching bit e'

$j =$	1	2	3	4	5	6	7	8	9	10
$Q_j^{(1)} =$	O_3	O_2	O_1	O_4	O_3	O_3	O_4	O_1	O_2	O_3
$Q_j^{(2)} =$	O_4	O_3	O_2	O_1	O_2	O_2	O_1	O_2	O_3	O_4
$Q_j^{(3)} =$	O_1	O_4	O_3	O_2	O_1	O_1	O_2	O_3	O_4	O_1
$Q_j^{(4)} =$	O_2	O_1	O_4	O_3	O_4	O_4	O_3	O_4	O_1	O_2
$e_j'^{(e=0)} =$	1	0	1	1	0	1	1	1	0	1
$e_j'^{(e=1)} =$	0	1	0	0	0	1	0	0	1	0

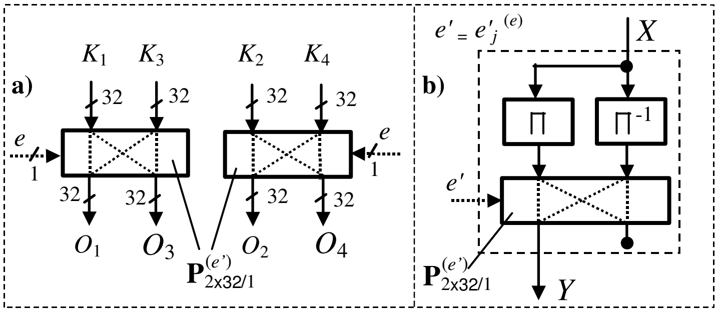


Figure 4. Swapping subkeys (a) and structure of the switchable fixed permutation (b)

4.3 Variable permutations

Variable permutations are performed with the CP boxes of the second order $\mathbf{P}^{(V)}_{32/96}$ and $\left(\mathbf{P}^{(V')}_{32/96}\right)^{-1}$ (see section 2) and **F**-boxes. The **F**-boxes represent special type of VBP. Construction of the **F**-boxes provides arbitrary change of the output vector weight. Indeed, depending on L, $Q_j^{(2)}$, and $Q_j^{(4)}$ eight of 32 input bits are replaced by bits of the constant $C = (10101010)$ while performing the operation **F**. Structure of the **F**-box is presented in Fig. 5. The **F**-box comprises two three-layer CP boxes $\mathbf{P}_{32/48}$ and $\mathbf{P}^{-1}_{32/48}$ separated with fixed permutation Π' which is described as follows:

(1,33)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,34,29,40)(10,35)(11,18)
(12,26)(14)(15,36,22,38)(16,30)(19,37)(20,27)(23)(24,31)(28,39)(32).

The 80-bit controlling vector $W = (W_1, W_2, W_3, W_4, W_5)$, where $W_i \in \{0,1\}^{16}$, of the **F**-box is divided into 48-bit controlling vector (W_1, W_2, W_3) of the CP box $\mathbf{P}_{32/48}$ and 32-bit part (W_4, W_5) of the controlling vector (W_6, W_4, W_5) of the $\mathbf{P}^{-1}_{32/48}$-box. The 16-bit vector W_6 is formed with the extension box "Ext" (Fig. 5a) using eight of the most significant bits of the output $H = (H_1, H_2, H_3, H_4, H_5)$, where $H_i \in \{0,1\}^8$, of the permutation Π': $W_6 = (H_5, H_5)$. The 80-bit controlling vector W is formed with the extension box $\mathbf{E}'$ input of which is the vector Z'. Relation between Z' and W is the following: $W_1 = Z'_l$, $W_2 = Z_l'^{\ggg 5}$, $W_3 = Z_l'^{\ggg 10}$, $W_4 = Z'_h$, $W_5 = Z_h'^{\ggg 5}$.

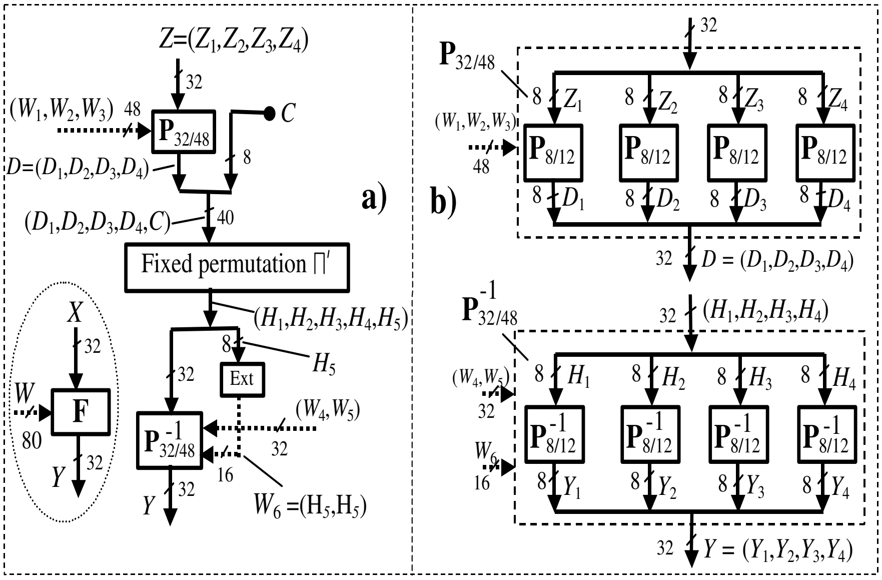


Figure 5. Structure of the **F**-box (a) and of the CP boxes $\mathbf{P}_{32/48}$ and $\mathbf{P}^{-1}_{32/48}$ (b)

The vectors W_1, W_2, and W_3 control the 1st, 2nd, and 3d active layers of the $\mathbf{P}_{32/48}$-box and the vectors W_4, W_5, and W_6 control the 1st, 2nd, and 3d active layers of the $\mathbf{P}^{-1}_{32/48}$-box, correspondingly. The vector D that is the output of $\mathbf{P}_{32/48}$ is concatenated with constant C forming the vector (D_1, D_2, D_3, D_4, C) at input of the fixed permutation Π'. At output of Π' the vector $(H_1, H_2, H_3, H_4, H_5)$, where $H_5 = (d_1, d_8, d_{10}, d_{15}, d_{19}, d_{22}, d_{28}, d_{29})$, is formed. Taking into account the structure of the $\mathbf{P}_{32/48}$-box one can see that superposition $\mathbf{P}_{32/80} \circ \Pi'$ moves arbitrary two bits of each byte Z_i of the vector $Z = (Z_1, Z_2, Z_3, Z_4)$ to H_5 with the same probability. Arbitrary single bit of each byte Z_i moves to H_5 with probability 2^{-2}. Thus, the vector H_5 is composed of eight bits of $Z = L \oplus Q^{(4)}_j$ which are replaced by 8 bits of C at the output of the $\mathbf{F}$-box. Depending on W different bits of Z are replaced, therefore the oddness of the output vector of the $\mathbf{F}$-box changes arbitrarily.

4.4 Permutational involutions

Rotation operation "$\ggg$ 16" performed on the left data subblock is used as permutational involution saving the "symmetric" use of the most significant (L_h) and least significant (L_l) halfs of L while performing two $\mathbf{F}$-box operations. The fixed permutation $\Pi^{(e')}$ has been selected to provide condition $(\Pi^{(e')}(L))^{\ggg 16} = \Pi^{(e')}(L^{\ggg 16})$ for $e' \in \{0, 1\}$ which is necessary for correct decryption. Permutational involution $\mathbf{I}$ in the right branch provides each bit at the input of the box $\mathbf{P}_{32/96}$ influences 31 bits at the output of the box $\mathbf{P}^{-1}_{32/96}$ even in the case $V = V'$ (without $\mathbf{I}$ in the case $V = V'$ each input bit of $\mathbf{P}_{32/96}$ influences only one output bit of $\mathbf{P}^{-1}_{32/96}$). The involution $\mathbf{I}$ is described with two rotations by eight bits: $Y = \mathbf{I}(X_1, X_2) = (X_1^{\ggg 8}, X_2^{\ggg 8})$, where $X_1, X_2 \in \{0, 1\}^{16}$. This permutation improves the resultant VBP corresponding to subsequently performed operations $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$. Indeed, even in the case $V = V'$ the superposition $\mathbf{P}^{(V)}_{32/96} \circ \mathbf{I} \circ \left(\mathbf{P}^{(V')}_{32/96}\right)^{-1}$ forms an effective CP box permutation all modifications of which are different permutational involutions. In general case we have $V \neq V'$, since the data are combined with different subkeys while forming the controlling vectors corresponding to the operations $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$. Investigating the role of the fixed permutation between two mutually inverse CP box operations we have performed many statistic experiments which have shown that the use of such permutation significantly improves the properties of the transformation performed with two mutually inverse CP boxes.

4.5 Formation of the controlling vectors V and V'

Controlling vectors corresponding to the boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$ are formed using the same extension box $\mathbf{E}$ implemented with simple connections. The inputs of the $\mathbf{E}$-boxes corresponding to the boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$ are $L^{(1)} = L \oplus Q^{(1)}_j$ and $L'^{(3)} = (L'^{\ggg 16}) \oplus Q^{(3)}_j$ (see Fig. 3b), where $L' = \left(\Pi^{(e')}(L)\right)^{\ggg 16}$, respectively. Let the 96-bit vectors $V = (V_1, V_2, V_3, V_4, V_5, V_6)$ and $V' = (V'_1, V'_2, V'_3, V'_4, V'_5, V'_6)$ be the outputs of the respective $\mathbf{E}$-boxes. The extension box provides the following relations:

$$V_1 = L^{(1)}_l, \quad V_2 = (L^{(1)}_l)^{\ggg 6}, \quad V_3 = (L^{(1)}_l)^{\ggg 12}, V_4 = L^{(1)}_h,$$
$$V_5 = (L^{(1)}_h)^{\ggg 6}, \quad V_6 = (L^{(1)}_h)^{\ggg 12},$$
$$V'_1 = L^{(3)}_l, \quad V'_2 = (L^{(3)}_l)^{\ggg 6}, \quad V'_3 = (L^{(3)}_l)^{\ggg 12}, V'_4 = L^{(3)}_h,$$
$$V'_5 = (L^{(3)}_h)^{\ggg 6}, \quad V'_6 = (L^{(3)}_h)^{\ggg 12}.$$

The extension box provides each bit of L influences three elementary boxes $\mathbf{P}_{2/1}$ in the CP box $\mathbf{P}_{32/96}$ and three $\mathbf{P}_{2/1}$-boxes in $\mathbf{P}^{-1}_{32/96}$. While designing the box $\mathbf{E}$ we have used the following criterion: *For all values of the controlling vector the permutation of each input bit of CPB must be defined by six different bits of L.* Due to realization of this criterion each bit of L influences exactly six bits of R while performing the CPB operation. It is easy to see that such distribution of the controlling bits provides that arbitrary input bit of the boxes $\mathbf{P}_{32/96}$ and $\mathbf{P}^{-1}_{32/96}$ moves to each output position with the same probability if L is a uniformly distributed random variable.

5 Discussion

Cipher DDP-64 presents an example of the pure VBP-based ciphers. The CP are extensively used in three different ways: (1) as VBP that are the basic cryptographic primitive, (2) as e-dependent swapping subkeys to change ciphering mode, and (3) in the switching permutation $\Pi^{(e')}$. Analogously to the VBP-based cipher SPECTR-H64 [13] the cryptosystem DDP-64 is fast in the case of frequent change of keys, since it is free of the key preprocessing. Avalanche effect spreads mostly when the changed bits are used as controlling ones, but not when they are transformed with the CP box operations (some avalanche connected with **F**-boxes is defined by the use of eight input bits as an internal controlling vector denoted earlier as W_6). In comparison with SPECTR-H64 the cipher DDP-64 has the following features:

1. It uses all secrete key in each round.

2. The DDP-64 is free of any additional nonlinear primirives (for example, the operation **G** in SPECTR-H64) and uses two **F**-boxes executed in parallel with the CP box operation $\mathbf{P}_{32/96}$. Each of two **F**-boxes is a special CP box generating at output the binary vector with arbitrary weight.

3. Round transformation includes special permutational involutions performed on the left and right data subblock and a switchable fixed permutation.

5.1 Some properties of VBP

For operations **F**, $\mathbf{P}_{32/96}$, and $\mathbf{P}^{-1}_{32/96}$ it is quite easy to calculate differential characteristics (DC) corresponding to differences with few number of active bits. Let Δ^U_h be the difference with arbitrary h active (non-zero) bits corresponding to some vector U. Let $\Delta_{h|i_1,...,i_n}$ be the difference with active bits corresponding to digits $i_1, ..., i_h$.

Avalanche effect corresponding to the operations $\mathbf{P}_{32/96}$, and $\mathbf{P}^{-1}_{32/96}$ is caused by the use of the data subblock L to define the values V and V'. Each bit of the left data subblock influences three bits of each of these controlling vectors. Each controlling bit influences two bits of the right data subblock. Thus, due to VBP performed on the right data subblock R with boxes $\mathbf{P}_{32/96}$, and $\mathbf{P}^{-1}_{32/96}$ one bit of L influences statistically about 12 bits of R. In the case when some difference with one active bit $\Delta^L_{1/i}$ passes the left branch of the cryptoscheme it influences three elementary switching elements permuting six different bits of the right data subblock. For example, if the input difference of the CP box $\mathbf{P}_{32/96}$ has no active bits (the case of zero difference), then the difference $\Delta^L_{1/i}$ can cause the generation at output of the $\mathbf{P}_{32/96}$-box the following differences: (1) Δ'_0 with probability 2^{-3}; (2) Δ'_2 with probability $3 \cdot 2^{-3}$; (3) Δ'_4 with probability $3 \cdot 2^{-3}$; (4) Δ'_6 with probability 2^{-3}. (Some other DC of the boxes $\mathbf{P}_{32/96}$ are presented in [15].)

Avalanche effect corresponding to the operations $\mathbf{F}$ relates to the use of the left data subblock to specify controlling vectors W and W'. Besides, avalanche spreads due to dependence of the output of "Ext"-box on L. Let consider the vector $L = (L_l, L_h)$ before the operation "$>\gg 16$". Each bit l_i, where $1 \leq i \leq 16$, of L_l influences three elementary boxes $\mathbf{P}_{2/1}$ of the $\mathbf{P}_{32/48}$-box in the lower $\mathbf{F}$-box and two boxes $\mathbf{P}_{2/1}$ of the $\mathbf{P}^{-1}_{32/48}$-box in the upper $\mathbf{F}$-box. Besides, with probability 2^{-2} (this probability corresponds to the event that l_i is moved to H_5) the bit l_i influences two boxes $\mathbf{P}_{2/1}$ of the $\mathbf{P}^{-1}_{32/48}$-box in the upper $\mathbf{F}$-box and with the same probability l_i influences two boxes $\mathbf{P}_{2/1}$ of the $\mathbf{P}^{-1}_{32/48}$-box in the lower $\mathbf{F}$-box. Analogous properties have all bits of L_h, since after the operation "$\ggg 16$" we have $(L_l, L_h)^{\ggg 16} = (L_h, L_l)$.

5.2 Security estimation

We have considered different types of attacks against DDP-64. Our results show that the differential cryptanalysis (DCA) is the most powerful attack. The iterative two-round DCs with differences (Δ^L_1, Δ^R_0) and (Δ^L_0, Δ^R_1) have the highest probability: $P(2) \approx P = 1.37 \cdot 2^{-17}$. The difference (Δ^L_1, Δ^R_0) passes eight and ten rounds of DDP-64 with probabilities $P(8) = P^4(2) \approx 1.79 \cdot 2^{-67}$ and $P(10) = P^5(2) \approx 1.23 \cdot 2^{-83}$. For some random cipher we have $P\left((\Delta^L_1, \Delta^R_0) \rightarrow (\Delta^L_1, \Delta^R_0)'\right) = 2^{-64} \cdot 2^5 = 2^{-59} > P(8) > P(10)$. Thus, DDP-64 with eight and ten rounds is undistinguishable from random cipher with differential attack using the most efficient two-round iterative characteristic.

Linear cryptanalysis (LCA) seems to be less efficient to attack DDP-64 as compared with DCA. Let denote the input mask as $A = (A^L, A^R)$ and the output mask as $B = (B^L, B^R)$. Linear attacks using masks $A = B = (1, 1, ..., 1)$ are prevented because of the use of two operations $\mathbf{F}$ changing arbitrary the oddness of their output. Using results of section 3 it is easy to find that the bias (deviation) of the linear characteristics (LC) with $z \leq 31$ active bits has value $b \leq 2^{-6}$ for each of the boxes $\mathbf{P}_{32/96}$, $\mathbf{P}^{-1}_{32/96}$, and $\mathbf{F}$, the maximal value corresponding to $z = 1$. Our linear analysis of DDP-64 has shown that among masks A^L, A^R, B^L, and B^R corresponding to individual subblocks and having weight less than 31 the masks with weight 1 have the maximal bias.

Analogously to consideration of the LC of the CP boxes, the LCA of the DDP-64 can be performed investigating the movement of the active bits through one or several encryption rounds. For LC with input mask $A = (A^L_{1|i}, A^R_{1|j})$ and output mask $B = (B^L_{1|i'}, B^R_{1|g})$, where indices indicate that we consider 32-bit masks with one active bit corresponding to ith and i'th (jth and gth) digits in the left (right) data subblocks at input and output respectively. It is easy to show that for arbitrary digits i, j, and g (digit i' is defined by digit i) the bias $b(1)$ of the one-round iterative LC (A, B, $b(1)$) is $b(1) = 0.56 \cdot 2^{-16}$. The last value is derived from the probability $p = 0.56 \cdot 2^{-15}$ that the jth bit of R is XORed two times with ith bit of L and then is moved to the gth digit at the output of the operation $\mathbf{P}^{-1}_{32/96}$. For r-round LC (A, B, $b(r)$) one can obtain $b(r) < 2^{-15r-1}$. For the random cipher LCs have bias $b \approx 2^{-32} > b(r) \gg 2^{-46} > b(3)$, therefore we can conclude that three-round DDP-64 is secure against LCA.

In spite of the simplicity of the key schedule the "symmetric" keys $K' = (X, Y, Y, X)$ and $K'' = (X, X, X, X)$ are not weak or semi-weak, since decryption requires switching the fixed permutation in the left branch of the cryptoscheme of DDP-64 (from Fig. 3 it is easy to see that $\mathbf{T}^{(e=0)}(C, K'') \neq M$, where $C = \mathbf{T}^{(e=0)}(M, K'')$). It seems to be difficult to calculate a semi-weak key-pair for DDP-64, if it is still possible. Slide attacks in the case of "symmetric" keys are also inefficient, since the encryption with DDP-64 is free of homogeneity (in the sense of [16]) due to the non-periodic schedule of the switching bit e' specifying the fixed permutation $\Pi^{(e')}$ performed on the left data subblock. This shows that the switchable operations can play sufficiently important role in the block ciphers which are free of the key preprocessing. For comparison one can remark that SPECTR-H64 which uses no switchable operations has weak keys (for all X its 256-bit key $K = (X, X, ..., X)$ is a weak one) and in the case of the weak key it seems vulnarable to slide attack.

Use of some strong key scheduling is a standard way to prevent weak keys and homogeneity in DDP-64, however this significantly encreases the hardware implementation cost.

5.3 Conclusion

Theoretic analysis of LC of the VBP operations conserving the weight of the transformed vector has shown that the principal problem in the design of VBP-based ciphers is to prevent LCA using masks $A = B = (1, 1, ..., 1)$. Examples of such attacks are proposed in [17,18]. In the known VBP-based ciphers SPECTR-H64 [13], SPECTR-128 [19], and Cobra-H64 [15] additional non-linear operations are used to thwart such variants of LCA. When developing a pure VBP-based cipher we have proposed a new type of the VBP operations (**F**-box operations) the use of which allows one to solve the mentioned problem without using additional non-linear operations.

The **F**-box operations have been used to design the cipher DDP-64. Presented analysis of DDP-64 illustrates efficiency of the use of VBP in the design of the block ciphers. The VBP thwarts well differential, linear, and other attacks allowing one to use comparatively small number of the encryption rounds. The efficiency of hardware implementation of the VBP-based ciphers is defined by the following factors: (1) VBP are efficient as cryptographic primitive, (2) property of the controllability of this primitive allows designing new advanced cryptoschemes, (3) VBP are fast and cheap in hardware [20]. Design of DDP-64 can be characterized as a design at bit level that defines low hardware implementation cost and high performance including the case of frequent change of keys.

Structure of DDP-64 suites well for detailed estimating DCs and LCs corresponding to differences and masks with few active bits. To attack DDP-64 the DCA is significantly more efficient than LCA. The DCA defines the minimum number of rounds for secure encryption with DDP-64.

We have also shown that in the case of the simple key scheduling the weak keys and homogeneity of the encryption can be prevented using switchable operations. Development of the simple and efficient switchable (e-dependent) operations is a new interesting item in the design of the ciphers that are free of precomputing the round keys.

References

[1] A.A. Waksman. *Permutation Network.* Journal of the ACM, vol. 15, no 1 (1968), pp. 159–163.

[2] M. Portz, *A generallized description of DES-based and Benes-based permutation generators.* LNCS, vol. 718 (1992), pp. 397–409.

[3] M. Kwan, *The design of the ICE encryption algorithm.* The 4th International Workshop, Fast Software Encryption - FSE '97 Proc. LNCS, vol. 1267 (1997), pp. 69–82.

[4] B. Van Rompay, L.R. Knudsen, V. Rijmen, *Differential cryptanalysis of the ICE encryption algorithm.* The 6th International Workshop, Fast Software Encryption - FSE'98 Proc. LNCS, vol. 1372 (1998), pp. 270–283.

[5] A.A. Moldovyan, N.A. Moldovyan, *A cipher based on data-dependent permutations.* Journal of Cryptology vol. 15, no. 1 (2002), pp. 61–72.

[6] R.L. Rivest, *The RC5 Encryption Algorithm.* The 2nd International Workshop, Fast Software Encryption - FSE'94 Proc. LNCS, vol. 1008 (1995), pp. 86–96.

[7] R.L. Rivest, M.J.B.Robshaw, R.Sidney, Y.L.Yin, *The RC6 Block Cipher.* 1st Advanced Encryption Standard Candidate Conference Proceedings, Venture, California, Aug. 20–22, 1998.

[8] C. Burwick, D.Coppersmith, E.D'Avingnon, R.Gennaro, Sh.Halevi, Ch.Jutla, Jr.S.M.Matyas, L.O'Connor, M.Peyravian, D.Safford, N.Zunic, *MARS - a Candidate Cipher for AES.* 1st Advanced Encryption Standard Candidate Conference Proceedings, Venture, California, Aug. 20–22, 1998.

[9] V.M. Maslovsky, A.A. Moldovyan, N.A. Moldovyan, *A method of the block encryption of discrete data.* Russian patent # 2140710. Bull. no 30 (1999).

[10] M. Matsui, *Linear Cryptanalysis Method for DES Cipher.* LNCS, vol. 765 (1994), pp. 386–397.

[11] M. Matsui, *New structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis.* LNCS, vol. 1267 (1996), pp. 205–218.

[12] B.S. Kaliski, Y.L. Yin, *On differential and linear cryptanalysis of the RC5 encryption algorithm.* The International conference, Advances in Cryptology - CRYPTO'95 Proc. LNCS, vol. 963 (1995), pp. 171-184.

[13] N.D. Goots, A.A. Moldovyan, N.A. Moldovyan, *Fast encryption algorithm SPECTR-H64.* International workshop, Methods, Models, and Architectures for Network Security - MMM-ANCS'01 Proc. LNCS, vol. 2052 (2001), pp. 275–286.

[14] L.E. Alekseev, T.G. Belkin, A.A. Moldovyan, N.A. Moldovyan, *A method of the iterated encryption of data blocks.* Russian patent # 2140714. Bull. no 30 (1999).

[15] N.A. Moldovyan, *Fast DDP-Based Ciphers: Design and Differential Analysis of Cobra-H64.* Computer Science Journal of Moldova. 2003, no. 2.

[16] A. Biryukov, D. Wagner, *Advanced Slide Attacks.* Advances in Cryptology - Eurocrypt'2000 Proc. LNCS, vol. 1807 (2000), pp. 589–606.

[17] Ch. Lee, D.Hong, Sun. Lee, San. Lee, S. Yang, J. Lim, *A chosen plaintext linear attack on block cipher CIKS-1.* LNCS, vol. 2513, pp. 456-468.

[18] Y.Ko, D.Hong, S.Hong, S.Lee, J.Lim, *Linear Cryptanalysis on SPECTR-H64 with Higher Order Differential Property.* International Workshop, Methods, Models, and Architectures for Network Security Proc. LNCS, vol. 2776 (2003), pp. 298–307.

[19] N.D. Goots, B.V.Izotov, A.A.Moldovyan, N.A.Moldovyan. *Modern cryptography: Protect Your Data with Fast Block Ciphers,* Wayne, A-LIST Publishing, 2003.- 400 p. (www.alistpublishing.com).

[20] N. Sklavos, A. A. Moldovyan, O. Koufopavlou *Encryption and Data Dependent Permutations: Implementation Cost and Performance Evaluation.* International workshop, Methods, Models,and Architectures for Network Security - MMM-ANCS'03 Proc. LNCS, vol. 2776 (2003), pp. 337–348.

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

Received July 22, 2003

Specialized Center of Program Systems “SPECTR”,
Kantemirovskaya str., 10,
St. Petersburg 197342, Russia;
ph./fax.7-812-2453743,
E–mail: *nmold@cobra.ru*

Abstracts of Dr. Thesis

Title: Head systems and applications to bio-informatics
Author: Serghei Verlan
Supervisor: Maurice Margenstern, director of LITA, professor at the University of Metz
University: University of Metz
Date of defence: 21 June 2004

The increasing interest for bioinformatics is not astonishing, because this discipline might potentially explain the functioning of some natural processes. One of relations between biology and computer science is the application of methods of computer science in biology. But we can investigate another relation trying to apply biological methods to solve computer science problems. This affirmation is based on the fact that a lot of biological processes, in particular DNA manipulations, may be viewed as information transformations. The study of biologically inspired systems is very exciting and can bring colossal gains.

From one hand, the theory of formal languages is grounded on rewriting operations. From the other hand, the nature uses different operations like copy and paste as well as different data structures. This is why it is very important to reconstruct old computational paradigms in this new framework.

The thesis is devoted to studying of H systems and their extensions.

H splicing systems were introduced by Thomas Head as a new language generating device. This abstract notion is good motivated biologically.

We considered two possible definitions of the splicing: 1-splicing and 2-splicing, and we studied for the first time the relation between classes of languages based on 1-splicing and on 2-splicing. We showed that one family is strictly included into another. We also found several non-trivial examples of splicing languages, and of regular languages that cannot be splicing languages.

We considered time-varying distributed H systems (or TVDH systems) that were introduced by Gh. Paun in 1998. Initially it was showed that 7 components are enough in order to generate all recursively enumerable languages. M. Margenstern and Yu. Rogozhin showed that TVDH systems with 2 components are able to do universal computations and that 2 components are enough to generate all recursively enumerable languages. The same authors showed that TVDH systems with one component are universal and that it is possible to generate all recursively enumerable languages with only one component in a sequential way.

We studied an extension of TVDH systems: enhanced time-varying distributed H systems, or ETVDH systems, which were introduced by M. Margenstern and Yu. Rogozhin in 2000. This model is a small modification of TVDH systems but it introduces more parallelism. Now one component is not enough in order to obtain a big computational power and, in this case, the generated language is limited by the family of regular languages. But two components are sufficient to generate all recursively enumerable languages. We studied the same problem in the parallel context and found solutions, at first with 4 components, after that with 3 components, and, finally, with 2 components that is a minimum.

We considered another extension of H systems which is inspired simultaneously by H systems and distributed grammars: test tube systems (or TT systems), introduced by E. Csuhaj-Varju, L. Kari and G. Paun. The computation in such system consists of two iteratively repeated steps: a computation step and a communication step. The number of tubes necessary to obtain the computational power of a Turing machine was established to two, while systems having one tube can generate only regular languages.

We proposed two variants of these systems: test tube systems with alternating filters and modified test tube systems. These variants differ from the original definition by the communication protocol. We showed that in the first variant two tubes are enough in order to produce all recursively enumerable languages. Moreover, it is possible to formulate rules in such a way that the second tube will be used as a garbage collector only having no associated rules. Therefore the universality is obtained with “one tube and a half”! This result shows also that H systems are already powerful enough and we can go from the regularity to the universality after small modifications.

For most of the systems we showed their equivalence with a formal grammar or a Turing machine. We analyzed previous proofs in the area of splicing systems and developed a new method, the method of directing molecules, which permits to decrease surprisingly the complexity of corresponding systems. We applied this method to TVDH systems, ETVDH systems, test tube systems with alternating filters, modified test tube systems and splicing membrane systems. In all these cases the new method permitted to simplify considerably the proofs.

We studied membrane systems, or P systems, which are a model of computing inspired by the structure and the functioning of a living cell. It was introduced in 1998 by Gh. Paun. We considered several models of computing issued from the combination of these systems and the splicing operation. We showed the structural aspect introduced by the presence of membranes combined with the splicing operation, which gives big computational power to systems.

We showed that the original definition of splicing P systems is not complete, and we proposed several variants in order to improve this lack.

We considered other variants of splicing P systems, non-extended splicing P systems and splicing P systems with immediate communication, and we showed how it is possible to decrease the number of membranes. We presented a final solution, as we showed a frontier between the decidability and undecidability for these systems.

We considered a variant of membrane systems where rules are assigned to membranes. We considered two types of rules: splicing rules and cutting/recombination rules. We showed that one membrane is sufficient in order to generate all recursively enumerable languages.

The thesis was written in French and English. See additional details at `http://lita.sciences.univ-metz.fr/~verlan/`.