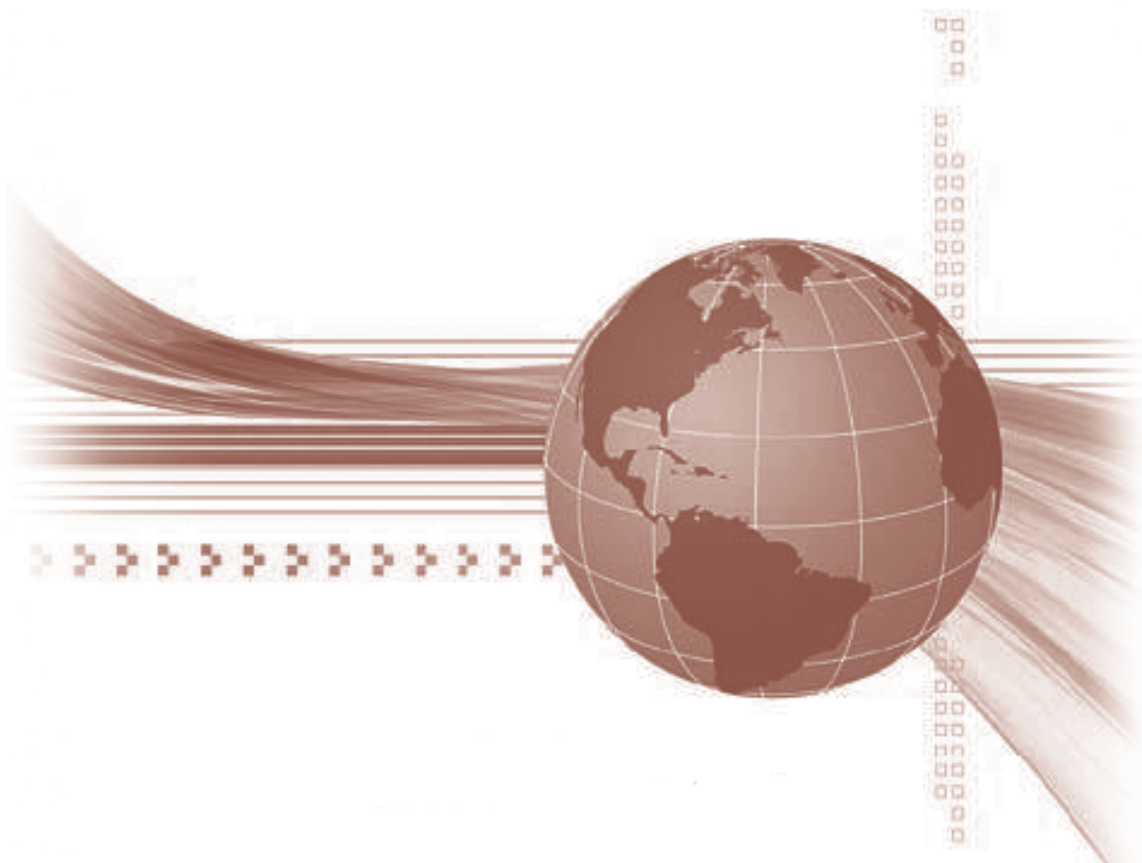




STUDIA UNIVERSITATIS
BABEŞ-BOLYAI



INFORMATICA

1/2009

S T U D I A
UNIVERSITATIS BABEȘ-BOLYAI
INFORMATICA

1

Redacția: 400084 Cluj-Napoca, Str. M. Kogălniceanu nr. 1 Tel: 405300

SUMAR – CONTENTS – SOMMAIRE

L. Aszalós, P. Balbiani, <i>Proving the Decidability of the PDL×PDL Product Logic</i>	3
M. Bakó, L. Aszalós, <i>Using Spreadsheets for Solving Logic Puzzles</i>	17
P. Bălan, <i>A Method with Random Modification of Gradient Components for Convex Models</i>	25
A. A. Badr, <i>Diagnosis of Images of Human Behaviour</i>	37
E. A. El-Dahshan, A.-B. M. Salem, T. H. Younis, <i>A Hybrid Technique for Automatic MRI Brain Images Classification</i>	55
M. Frențiu, H. F. Pop, <i>Overview of Fuzzy Methods for Effort Estimation by Analogy</i> ..	68
G. S. Cojocar, G. Czibula, I. G. Czibula, <i>A Comparative Analysis of Clustering Algorithms in Aspect Mining</i>	75
M. Oltean, <i>Unconventional Computing: A Short Introduction</i>	85
C. Groșan, <i>A Multiobjective Metaheuristic for Job-Shop Scheduling</i>	97
F. Muntenescu, <i>Fingerprint Matching on Small Images</i>	103

PROVING THE DECIDABILITY OF THE PDL×PDL PRODUCT LOGIC

LÁSZLÓ ASZALÓS AND PHILIPPE BALBIANI

ABSTRACT. The propositional dynamic logic (PDL) is an adequate tool to write down programs. In a previous article we used PDL to formulate cryptographic protocols as parallel programs. In these protocols at least two agents/individuals exchange messages, so we needed to use product logic to formulate the parallel actions. Ágnes Kurucz proved that $S5 \times S5 \times S5$ — which is the simplest triple product logic — is undecidable, hence it follows that $PDL \times PDL \times PDL$ is undecidable, too. It is easy to show that the PDL logic (without the star operator) is decidable, so it is an interesting problem, that the $PDL \times PDL$ product logic is decidable or not.

1. INTRODUCTION

Authentication protocols emerged from numerous works of computer scientists and their use has become common in the science and study of methods of exchanging keys. They are basically sequences of message exchanges, whose purpose is to assure users that communications do not leak confidential data. Indeed, there is a wide variety of protocols that have been specified and implemented, from protocols with trusted third party, to protocols with public key and, even more generally, hybrid protocols. The one drawback is that many of them have been shown to be flawed, from which one may explain the great deal of attention devoted to the formal verification of security properties of protocols. Examples of protocols can be found in [4].

Received by the editors: December 6, 2008.

2000 *Mathematics Subject Classification.* 03B45, 94A60.

1998 *CR Categories and Descriptors.* F.4.1 [**Mathematical logic**]: model logic – *product of propositional dynamic logics*; D.4.6 [**Security and protection**]: Authentication – *formal verification of authentication protocols* .

Key words and phrases. Mathematical logic, Decidability, $PDL \times PDL$ product logic, Formal verification.

This paper has been presented at the 7th Joint Conference on Mathematics and Computer Science (7th MaCS), Cluj-Napoca, Romania, July 3-6, 2008.

In the literature, the most popular logic-based formal approach to the analysis of authentication protocols is perhaps the modal BAN calculus introduced by Burrows, Abadi and Needham [3]. From the point of view of computer science, a virtue of BAN is that it allows static characterization of epistemic concepts. In spite of its success in finding flaws or redundancies in some well-known protocols, the effectiveness of BAN as a formal method for the analysis of authentication protocols has been a source of debate, see [9] for details. The problem with the BAN logic is that it explicitly excludes time. On the other hand there is no way to represent actions performed by users. Communication, by its nature, refers to time, and its properties are naturally expressed in terms of actions like sending and receiving messages. When devising a protocol, we usually think of some property that we want the protocol to satisfy. We are mainly interested in the correctness of a protocol with respect to epistemic properties between two users like the arranging of a secret key known only to them. Therefore, our emphasis is on the interplay between knowledge and action. This leads us to consider a language that allows to express notions of knowledge and actions in a straightforward way: the language of modal logic.

We can treat protocols as programs, so we used the propositional dynamic logic (PDL) [7] as a starting point. It allows for us to examine properties of the protocol using logic. Protocols are not just sole programs, but a set of programs. Usually two or three programs run parallel when a protocol executed: the program of Alice, of Bob and maybe program of Charlie, if we use the traditional names of the cryptography. To handle the parallel execution of programs, we developed the product logic $\text{PDL} \times \text{PDL}$, using the construction of Gabbay and Shehtman [5, 6].

We would use the logic $\text{PDL} \times \text{PDL}$ to examine real protocols, so the decidability of the logic is very important. From [8] we know that $\text{S5} \times \text{S5} \times \text{S5}$ — which is the simplest triple product logic — is undecidable, so the examination of $\text{PDL} \times \text{PDL} \times \text{PDL}$ unnecessary. The original PDL logic is decidable. What is the status of our construction which is between in PDL and $\text{PDL} \times \text{PDL} \times \text{PDL}$? We will prove in this article that $\text{PDL} \times \text{PDL}$ is decidable.

In the following section we introduce the logic, $\text{PDL} \times \text{PDL}$, and after we show the method of quasimodels developed by Wolter and Zakharyashev and explained in [5].

2. $\text{PDL} \times \text{PDL}$ LOGIC

The PDL logic is a logic of actions, so at first we define the set of actions. We have a finite set of atomic actions, its elements are denoted with π_i . Two

atomic actions are special: the sending and receiving messages. They are denoted with **send** and **rec**. For our proofs the structure of messages are indifferent. In our previous papers [1, 2] we discussed the structure of messages in detail. To construct complicated actions we can use the operators of test, sequence and selections, denoted by $?$, semicolon and \cup , respectively.

$$\alpha \rightleftharpoons \lambda \mid \pi_k \mid A? \mid \alpha; \beta \mid \alpha \cup \beta \mid \mathbf{send}(m) \mid \mathbf{rec}(m)$$

We can define the formulae based on the set of atomic formulae, by using the usual logical connectives and the modalities constructed from a pair of actions:

$$A \rightleftharpoons p_k \mid \neg A \mid A \vee B \mid \langle \alpha_1 \parallel \alpha_2 \rangle A$$

For the semantics, we use a variant of the Kripke model. We have two agents, so the global state is build up from local states. The model \mathcal{M} is a (W_1, W_2, r, R, V) tuple where W_1 and W_2 are the set of local states (possible worlds), r and R is a family of relations on W_i ($r_i, R_i \subseteq W_i \times W_i$), and V is a valuation on $W_1 \times W_2$ ($V(p_j) \subseteq W_1 \times W_2$). Given a model \mathcal{M} we define the relation $R_{\alpha \parallel \beta}$ and the $(s, t, c) \models_{\mathcal{M}} A$ truth-relation by a parallel induction for any states $s, s' \in W_1, t, t' \in W_2$, actions α, β and formula A as follows:

- $(s, t, c) R_{\lambda \parallel \lambda} (s', t', c')$ iff $s = s', t = t', c = c'$;
- $(s, t, c) R_{\pi_i \parallel \lambda} (s', t', c')$ iff $sr_i s', t = t', c = c'$;
- $(s, t, c) R_{\lambda \parallel \pi_i} (s', t', c')$ iff $s = s', tR_i t', c = c'$;
- $(s, t, c) R_{A? \parallel \lambda} (s', t', c')$ iff $s = s', t = t', c = c', (s, t, c) \models_{\mathcal{M}} A$;
- $(s, t, c) R_{\lambda \parallel A?} (s', t', c')$ iff $s = s', t = t', c = c', (s, t, c) \not\models_{\mathcal{M}} A$;
- $(s, t, c) R_{\mathbf{send}(m) \parallel \lambda} (s', t', c')$ iff $s = s', t = t'$, and if $c = (c_1, c_2)$, then $c' = (c_1, c_2 \star m)$;
- $(s, t, c) R_{\lambda \parallel \mathbf{send}(m)} (s', t', c')$ iff $s = s', t = t'$, and if $c = (c_1, c_2)$, then $c' = (c_1 \star m, c_2)$;
- $(s, t, c) R_{\mathbf{rec}(m) \parallel \lambda} (s', t', c')$ iff $s = s', t = t'$, and if $c' = (c_1, c_2)$, then $c = (m \star c_1, c_2)$;
- $(s, t, c) R_{\lambda \parallel \mathbf{rec}(m)} (s', t', c')$ iff $s = s', t = t'$, and if $c' = (c_1, c_2)$, then $c = (c_1, m \star c_2)$;
- $R_{\varphi; \alpha \parallel \psi; \beta} \rightleftharpoons (R_{\varphi \parallel \lambda} \circ R_{\alpha \parallel \psi; \beta}) \cup (R_{\lambda \parallel \psi} \circ R_{\varphi; \alpha \parallel \beta})$ where φ_i and ψ_j are atomic action, test, send or receive actions;
- $R_{\alpha(\alpha_1 \cup \alpha_2) \parallel \beta} \rightleftharpoons R_{\alpha(\alpha_1) \parallel \beta} \cup R_{\alpha(\alpha_2) \parallel \beta}$;
- $R_{\alpha \parallel \beta(\beta_1 \cup \beta_2)} \rightleftharpoons R_{\alpha \parallel \beta(\beta_1)} \cup R_{\alpha \parallel \beta(\beta_2)}$.
- $(s, t, c) \models_{\mathcal{M}} p_i$ iff $(s, t) \in V(p_i)$
- $(s, t, c) \models_{\mathcal{M}} \neg A$ iff $(s, t, c) \not\models_{\mathcal{M}} A$.
- $(s, t, c) \models_{\mathcal{M}} A \vee B$ iff $(s, t, c) \models_{\mathcal{M}} A$ or $(s, t, c) \models_{\mathcal{M}} B$.

- $(s, t, c) \models_{\mathcal{M}} \langle \alpha \parallel \beta \rangle A$, if there exists a triple (s', t', c') such that $(s, t, c) R_{\alpha_1 \parallel \alpha_2} (s', t', c')$ and $(s', t', c') \models_{\mathcal{M}} A$

We say that formula A is *satisfiable* in model \mathcal{M} if there is exists $s \in W_1$ and $t \in W_2$ such that $(s, t, (\varepsilon, \varepsilon)) \models_{\mathcal{M}} A$; and we say that formula A is *valid* in model \mathcal{M} if for all $s \in W_1$ and $t \in W_2$, $(s, t, (\varepsilon, \varepsilon)) \models_{\mathcal{M}} A$.

3. QUASIMODEL

To prove the decidability of the PDL \times PDL logic, we follow the method described in [5]. At first we need the concept of the subformula. The standard definition is not suitable for us, so we use a variant. The *Fischer-Ladner closure* of φ ($flc(\varphi)$) defined as

- if $\psi \vee \chi \in flc(\varphi)$ then $\psi \in flc(\varphi)$, $\chi \in flc(\varphi)$;
- if $\neg\psi \in flc(\varphi)$ then $\psi \in flc(\varphi)$;
- if $\langle \alpha \parallel \beta \rangle \psi \in flc(\varphi)$ then $\psi \in flc(\varphi)$;
- if $\langle \alpha(\alpha_1 \cup \alpha_2) \parallel \beta \rangle \psi \in flc(\varphi)$ then $\langle \alpha(\alpha_1) \parallel \beta \rangle \psi \in flc(\varphi)$, $\langle \alpha(\alpha_2) \parallel \beta \rangle \psi \in flc(\varphi)$;
- if $\langle \alpha \parallel \beta(\beta_1 \cup \beta_2) \rangle \psi \in flc(\varphi)$ then $\langle \alpha \parallel \beta(\beta_1) \rangle \psi \in flc(\varphi)$, $\langle \alpha \parallel \beta(\beta_2) \rangle \psi \in flc(\varphi)$;
- if $\langle \pi; \alpha \parallel \beta \rangle \psi \in flc(\varphi)$ then $\langle \pi \parallel \lambda \rangle \langle \alpha \parallel \beta \rangle \psi \in flc(\varphi)$, where π is an atomic action or a test;
- if $\langle \alpha \parallel \pi; \beta \rangle \psi \in flc(\varphi)$ then $\langle \lambda \parallel \pi \rangle \langle \alpha \parallel \beta \rangle \psi \in flc(\varphi)$, where π is an atomic action or a test;
- if $\langle \psi? \parallel \lambda \rangle \chi \in flc(\varphi)$ or $\langle \lambda \parallel \psi? \rangle \chi \in flc(\varphi)$ then $\psi \in flc(\varphi)$, and $\chi \in flc(\varphi)$.

Type t for φ is a Boolean saturated subset t of $flc(\varphi)$, satisfying the following conditions:

- (t_1) $\langle \lambda \parallel \lambda \rangle \psi \in t$ iff $\psi \in t$ for all $\langle \lambda \parallel \lambda \rangle \psi \in flc(\varphi)$;
- (t_2) $\langle \pi; \alpha \parallel \lambda \rangle \psi \in t$ iff $\langle \pi \parallel \lambda \rangle \langle \alpha \parallel \lambda \rangle \psi \in t$ for all $\langle \pi; \alpha \parallel \lambda \rangle \psi \in flc(\varphi)$;
- (t_3) $\langle \lambda \parallel \pi; \beta \rangle \psi \in t$ iff $\langle \lambda \parallel \pi \rangle \langle \lambda \parallel \beta \rangle \psi \in t$ for all $\langle \lambda \parallel \pi; \beta \rangle \psi \in flc(\varphi)$;
- (t_4) $\langle \pi; \alpha \parallel \pi'; \beta \rangle \psi \in t$ iff either $\langle \pi \parallel \lambda \rangle \langle \alpha \parallel \pi'; \beta \rangle \psi \in t$ or $\langle \lambda \parallel \pi' \rangle \langle \pi; \alpha \parallel \beta \rangle \psi \in t$ for all $\langle \pi; \alpha \parallel \pi'; \beta \rangle \psi \in flc(\varphi)$;
- (t_5) $\langle \alpha(\alpha_1 \cup \alpha_2) \parallel \beta \rangle \psi \in t$ iff either $\langle \alpha(\alpha_1) \parallel \beta \rangle \psi \in t$ or $\langle \alpha(\alpha_2) \parallel \beta \rangle \psi \in t$ for all $\langle \alpha(\alpha_1 \cup \alpha_2) \parallel \beta \rangle \psi \in flc(\varphi)$;
- (t_6) $\langle \alpha \parallel \beta(\beta_1 \cup \beta_2) \rangle \psi \in t$ iff either $\langle \alpha \parallel \beta(\beta_1) \rangle \psi \in t$ or $\langle \alpha \parallel \beta(\beta_2) \rangle \psi \in t$ for all $\langle \alpha \parallel \beta(\beta_1 \cup \beta_2) \rangle \psi \in flc(\varphi)$;
- (t_7) $\langle \psi? \parallel \lambda \rangle \chi \in t$ iff $\psi \in t$ and $\chi \in t$ for all $\langle \psi? \parallel \lambda \rangle \chi \in flc(\varphi)$;
- (t_8) $\langle \lambda \parallel \psi? \rangle \chi \in t$ iff $\psi \in t$ and $\chi \in t$ for all $\langle \lambda \parallel \psi? \rangle \chi \in flc(\varphi)$.

Modal depth of a formula φ ($md(\varphi)$) is defined as usual:

- $md(p_i) = md(\top) = 0$;
- $md(\neg\varphi) = md(\varphi)$;
- $md(\varphi \vee \psi) = \max(md(\varphi), md(\psi))$;
- $md([\alpha \parallel \beta]\varphi) = md(\langle \alpha \parallel \beta \rangle \varphi)$;
- $md(\langle \lambda \parallel \lambda \rangle \varphi) = md(\varphi)$;
- $md(\langle \alpha(\alpha_1 \cup \alpha_2) \parallel \beta \rangle \varphi) = \max(md(\langle \alpha(\alpha_1) \parallel \beta \rangle \varphi), md(\langle \alpha(\alpha_2) \parallel \beta \rangle \varphi))$;
- $md(\langle \alpha \parallel \beta(\beta_1 \cup \beta_2) \rangle \varphi) = \max(md(\langle \alpha \parallel \beta(\beta_1) \rangle \varphi), md(\langle \alpha \parallel \beta(\beta_2) \rangle \varphi))$;
- $md(\langle \pi; \alpha \parallel \beta \rangle \varphi) = md(\langle \alpha \parallel \pi; \beta \rangle \varphi) = 1 + md(\langle \alpha \parallel \beta \rangle \varphi)$.

An n -frame $\mathcal{F} = (W, R_1, \dots, R_n)$ is called *rooted*, if there is a $w_0 \in W$ such that $W = \{w \in W \mid w_0 R^* w\}$, where $R = \bigcup_{1 \leq j \leq n} R_j$. Such a w_0 is called a *root* of \mathcal{F} . A rooted frame $\mathcal{F} = (W, R_1, \dots, R_n)$ is said to be a *tree* if all the R_j are pairwise disjoint and for every $x \in W$, the set $W_x = \{y \in W \mid y R^* x\}$ is finite and linearly ordered by the reflexive and transitive closure R^* of the relation R (its restriction to W_x , to be more precise). \mathcal{F} is called *intransitive* if for any R_j, R_k ($1 \leq j, k \leq n$) we have $\forall x, y, z \in W (x R_j y \wedge y R_k z \rightarrow \neg x R_k z \wedge \neg x R_j z)$. A path of length l from x to y in \mathcal{F} is a sequence (x_0, \dots, x_l) such that $x_0 = x$, $x_l = y$ and $x_k R_j x_{k+1}$ for each $k < l$ and some j , $1 \leq j \leq n$. The length of the path from the root of \mathcal{F} to x is called the *co-depth* of x . The *depth* of \mathcal{F} is the maximum of co-depth of x ($x \in W$), if this maximum exists. By the *depth* of x in \mathcal{F} we understand the depth of the subtree of \mathcal{F} with root x . The *Quasistate candidate* for φ is a pair $((T, R_1, \dots, R_k), t)$, where (T, R_1, \dots, R_k) is a finite intransitive tree of depth $md(\varphi)$, and t is a labeling function associating with each $x \in T$ a type $t(x)$ for φ . $((T, R_1, \dots, R_k), t)$ is a *quasistate* for φ if

- (qm1) for all $x \in T$ and $\langle \lambda \parallel \pi_i \rangle \psi \in flc(\varphi)$: $\langle \lambda \parallel \pi_i \rangle \psi \in t(x)$ iff there exists a $y \in T$ such that $x R_i y$ and $\psi \in t(y)$.
- (qm1') for all $x_0, x_1, x_2 \in T$ such that $x_0 R_i x_1$, $x_0 R_i x_2$, and $x_1 \neq x_2$ the structures $((T^{x_1}, R_1^{x_1}, \dots, R_k^{x_1}), t^{x_1})$ and $((T^{x_2}, R_1^{x_2}, \dots, R_k^{x_2}), t^{x_2})$ are not isomorphic. (Two quasistate candidates $((T, <_1, \dots, <_n), t)$ and $((T', <'_1, \dots, <'_n), t')$ are called *isomorphic* if there is an isomorphism f between the trees $(T, <_1, \dots, <_n)$ and $(T', <'_1, \dots, <'_n)$ such that $t(x) = t'(f(x))$, for all $x \in T$.)

A *basic structure* for φ of depth m is a pair (\mathcal{F}, q) , such that $\mathcal{F} = (W, r_1, \dots, r_k)$ and q is a function associating with each world $w \in W$ and each message $c = (c_1, c_2)$ a quasistate $q(w, c) = ((T_w^c, R_{w,1}^c, \dots, R_{w,k}^c), t_w^c)$ for φ such that the depth of each $(T_w^c, R_{w,i}^c)$ is m . Let (\mathcal{F}, q) be a basic structure for φ of depth m and let $l \leq m$. An *l -run through* (\mathcal{F}, q) is a function ρ giving for each $w \in W$ and the list of messages c a point $\rho(w, c) \in T_w^c$ of co-depth l . Given a set \mathcal{R} of runs we denote by \mathcal{R}_l the set of all l -runs from \mathcal{R} . A run ρ is called

coherent, if for all lists of messages c , for all possible worlds $w \in W$ and for all formulae the following conditions are satisfied:

- $\langle \pi_i \parallel \lambda \rangle \psi \in flc(\varphi)$: if there exists a world $v \in W$ such that $wr_i v$ and $\psi \in t_v^c(\rho(v, c))$ then $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$;
- $\langle \mathbf{send}(m) \parallel \lambda \rangle \psi \in flc(\varphi)$: if $c' = (c_1, c_2 \star m)$ where $c = (c_1, c_2)$ and $\psi \in t_w^{c'}(\rho(w, c'))$ then $\langle \mathbf{send}(m) \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$;
- $\langle \lambda \parallel \mathbf{send}(m) \rangle \psi \in flc(\varphi)$: if $c' = (c_1 \star m, c_2)$ where $c = (c_1, c_2)$ and $\psi \in t_w^{c'}(\rho(w, c'))$ then $\langle \lambda \parallel \mathbf{send}(m) \rangle \psi \in t_w^c(\rho(w, c))$;
- $\langle \mathbf{rec}(m) \parallel \lambda \rangle \psi \in flc(\varphi)$: if $c' = (c_1, c_2)$ where $c = (m \star c_1, c_2)$ and $\psi \in t_w^{c'}(\rho(w, c'))$ then $\langle \mathbf{rec}(m) \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$;
- $\langle \lambda \parallel \mathbf{rec}(m) \rangle \psi \in flc(\varphi)$: if $c' = (c_1, c_2)$ where $c = (c_1, m \star c_2)$ and $\psi \in t_w^{c'}(\rho(w, c'))$ then $\langle \lambda \parallel \mathbf{rec}(m) \rangle \psi \in t_w^c(\rho(w, c))$.

In the previous definition the sign \star denotes the concatenation of messages.

A run ρ is called *w-saturated* for $w \in W$, if for all lists of messages c and for all formulae the following conditions are satisfied:

- $\langle \pi_i \parallel \lambda \rangle \psi \in flc(\varphi)$: if $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$ then there exists a world $v \in W$ such that $wr_i v$ and $\psi \in t_v^c(\rho(v, c))$;
- $\langle \mathbf{send}(m) \parallel \lambda \rangle \psi \in flc(\varphi)$: if $\langle \mathbf{send}(m) \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$ then $\psi \in t_w^{c'}(\rho(w, c'))$ where if $c = (c_1, c_2)$ then $c' = (c_1, c_2 \star m)$;
- $\langle \lambda \parallel \mathbf{send}(m) \rangle \psi \in flc(\varphi)$: if $\langle \lambda \parallel \mathbf{send}(m) \rangle \psi \in t_w^c(\rho(w, c))$ then $\psi \in t_w^{c'}(\rho(w, c'))$ where if $c = (c_1, c_2)$ then $c' = (c_1 \star m, c_2)$;
- $\langle \mathbf{rec}(m) \parallel \lambda \rangle \psi \in flc(\varphi)$: if $\langle \mathbf{rec}(m) \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$ then $\psi \in t_w^{c'}(\rho(w, c'))$ where if $c = (m \star c_1, c_2)$ then $c' = (c_1, c_2)$;
- $\langle \lambda \parallel \mathbf{rec}(m) \rangle \psi \in flc(\varphi)$: if $\langle \lambda \parallel \mathbf{rec}(m) \rangle \psi \in t_w^c(\rho(w, c))$ then $\psi \in t_w^{c'}(\rho(w, c'))$ where if $c = (c_1, m \star c_2)$ then $c' = (c_1, c_2)$.

A run is *saturated*, if it is *w-saturated* for all $w \in W$. $\mathcal{Q} = (\mathcal{F}, q, \mathcal{R}, \triangleleft)$ is a *PDL \times PDL-quasimodel* for φ if (\mathcal{F}, q) is a basic structure for φ of depth $m \leq md(\varphi)$ such that

(qm2) there exists a world $w_0 \in W$ and $\varphi \in t_{w_0}^{(\varepsilon, \varepsilon)}(x_0)$, where x_0 is the root of $(T_{w_0}^{(\varepsilon, \varepsilon)}, R_{w_0, 1}^{(\varepsilon, \varepsilon)}, \dots, R_{w_0, k}^{(\varepsilon, \varepsilon)})$.

\mathcal{R} is a set of coherent and saturated runs through (\mathcal{F}, q) and \triangleleft is a set of binary relation on \mathcal{R} satisfying the following conditions:

(qm3) for all $\rho, \rho' \in \mathcal{R}$, if $\rho \triangleleft_i \rho'$ then $\rho(w, c)R_{w, i}^c \rho'(w, c)$ for all $w \in W$ and lists of messages c .

(qm4) $\mathcal{R}_0 \neq \varepsilon$ and for all $l < m$, $\rho \in \mathcal{R}_l$, $w \in W$, for all lists of messages c , $x \in T_w^c$, for all $1 \leq i \leq k$, if $\rho(w, c)R_{w, i}^c x$ then there is $\rho' \in \mathcal{R}_{l+1}$ such that $\rho'(w, c) = x$ and $\rho \triangleleft_i \rho'$.

Lemma 1. *An \mathcal{ML}_2 formula φ satisfiable in a product frame $\mathcal{F} \times \mathcal{G}$ iff there is a PDL×PDL-quasimodel for φ based on \mathcal{F} .*

Proof. Let $(\mathcal{F}, q, \mathcal{R}, \triangleleft)$ be a PDL×PDL-quasimodel for φ based on \mathcal{F} , where $\mathcal{F} = (W, r_1, \dots, r_k)$. Take the product frame $\mathcal{F} \times (\mathcal{R}, \triangleleft)$, and define a valuation \mathcal{V} in it as follows: $\mathcal{V}(p_i) = \{(w, \rho, c) \mid p \in t_w^c(\rho(w, c))\}$ for every propositional variable p_i . Let \mathcal{M} be $(\mathcal{F} \times (\mathcal{R}, \triangleleft), \mathcal{V})$. By induction on the construction of $\psi \in flc(\varphi)$ we need to show that for every $(w, \rho, c) \in \mathcal{M}$ we have $(w, \rho, c) \models_{\mathcal{M}} \psi$ iff $\psi \in t_w^c(\rho(w, c))$.

- For variables this follows from the definition.
 - For Booleans, types are Boolean saturated sets.
 - $(w, \rho, c) \models_{\mathcal{M}} \langle \pi_i \parallel \lambda \rangle \psi$ (based on the definition of the semantics) iff there exists a world $w' \in W$ such that $wr_i w'$ and $(w', \rho, c) \models_{\mathcal{M}} \psi$. Then by induction hypothesis (IH) $\psi \in t_{w'}^c(\rho(w', c))$. ρ is saturated and coherent, so the previous holds iff $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$.
 - $(w, \rho, c) \models_{\mathcal{M}} \langle \lambda \parallel \pi_i \rangle \psi$ (based on the definition of the semantics) iff there exists a run $\rho' \in \mathcal{R}$ such that $\rho \triangleleft_i \rho'$ and $(w, \rho', c) \models_{\mathcal{M}} \psi$. Then by IH $\psi \in t_w^c(\rho'(w, c))$. According to (qm3), from $\rho \triangleleft_i \rho'$ we get $\rho(w, c)R_{w,i}^c \rho'(w, c)$. Finally based on (qm1) we get that $\langle \lambda \parallel \pi_i \rangle \psi \in t_w^c(\rho(w, c))$.
- In other direction let assume, that $\langle \lambda \parallel \pi_i \rangle \psi \in t_w^c(\rho(w, c))$ Then by (qm1) there exists a $x \in T_w^c$ such that $\rho(w, c)R_i x$ and $\psi \in t_w^c(x)$. According to (qm4) there exists $\rho' \in \mathcal{R}$ such that $\rho \triangleleft_i \rho'$ and $\psi \in t_w^c(\rho'(w, c))$. By IH we get $(w, \rho', c) \models_{\mathcal{M}} \psi$ and finally according to the definition of the semantics $(w, \rho, c) \models_{\mathcal{M}} \langle \lambda \parallel \pi_i \rangle \psi$.
- $(w, \rho, c) \models_{\mathcal{M}} \langle \mathbf{send}(m) \parallel \lambda \rangle \psi$ iff $(w, \rho, c') \models_{\mathcal{M}} \psi$ where if $c = (c_1, c_2)$ then $c' = (c_1, c_2 \star m)$ (by def.). Then by IH $\psi \in t_w^c(\rho(w, c'))$. ρ is saturated and coherent, so the previous holds iff $\langle \mathbf{send}(m) \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$.
 - $(w, \rho, c) \models_{\mathcal{M}} \langle \lambda \parallel \mathbf{send}(m) \rangle \psi$ iff $(w, \rho, c') \models_{\mathcal{M}} \psi$ where if $c = (c_1, c_2)$ then $c' = (c_1 \star m, c_2)$ (by def.). Then by IH $\psi \in t_w^c(\rho(w, c'))$. ρ is saturated and coherent, so the previous holds iff $\langle \lambda \parallel \mathbf{send}(m) \rangle \psi \in t_w^c(\rho(w, c))$.
 - $(w, \rho, c) \models_{\mathcal{M}} \langle \mathbf{rec}(m) \parallel \lambda \rangle \psi$ iff $(w, \rho, c') \models_{\mathcal{M}} \psi$ where if $c' = (c_1, c_2)$ then $c = (m \star c_1, c_2)$ (by def.). Then by IH $\psi \in t_w^c(\rho(w, c'))$. ρ is saturated and coherent, so the previous holds iff $\langle \mathbf{rec}(m) \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$.
 - $(w, \rho, c) \models_{\mathcal{M}} \langle \lambda \parallel \mathbf{rec}(m) \rangle \psi$ iff $(w, \rho, c') \models_{\mathcal{M}} \psi$ where if $c' = (c_1, c_2)$ then $c = (c_1, m \star c_2)$ (by def.). Then by IH $\psi \in t_w^c(\rho(w, c'))$. ρ is saturated and coherent, so the previous holds iff $\langle \lambda \parallel \mathbf{rec}(m) \rangle \psi \in t_w^c(\rho(w, c))$.

- $(w, \rho, c) \models_{\overline{\mathcal{M}}} \langle \psi? \parallel \lambda \rangle \chi$ iff $(w, \rho, c) \models_{\overline{\mathcal{M}}} \psi$ and $(w, \rho, c) \models_{\overline{\mathcal{M}}} \chi$. By IH this is true iff $\psi \in t_w^c(\rho(w, c))$ and $\chi \in t_w^c(\rho(w, c))$. But according to (t7) this is true iff $\langle \psi? \parallel \lambda \rangle \chi \in t_w^c(\rho(w, c))$
- $(w, \rho, c) \models_{\overline{\mathcal{M}}} \langle \lambda \parallel \psi? \rangle \chi$ iff $(w, \rho, c) \models_{\overline{\mathcal{M}}} \psi$ and $(w, \rho, c) \models_{\overline{\mathcal{M}}} \chi$. By IH this is true iff $\psi \in t_w^c(\rho(w, c))$ and $\chi \in t_w^c(\rho(w, c))$. But according to (t8) this is true iff $\langle \lambda \parallel \psi? \rangle \chi \in t_w^c(\rho(w, c))$
- $(w, \rho, c) \models_{\overline{\mathcal{M}}} \langle \alpha(\alpha_1 \cup \alpha_2) \parallel \beta \rangle \psi$ iff $(w, \rho, c) \models_{\overline{\mathcal{M}}} \langle \alpha(\alpha_1) \parallel \beta \rangle \psi$ or $(w, \rho, c) \models_{\mathcal{M}} \langle \alpha(\alpha_2) \parallel \beta \rangle \psi$ (by def.). By IH this is true iff $\langle \alpha(\alpha_1) \parallel \beta \rangle \psi \in t_w^c(\rho(w, c))$ or $\langle \alpha(\alpha_2) \parallel \beta \rangle \psi \in t_w^c(\rho(w, c))$. But according to (t5) this is true iff $\langle \alpha(\alpha_1 \cup \alpha_2) \parallel \beta \rangle \psi \in t_w^c(\rho(w, c))$
- $(w, \rho, c) \models_{\overline{\mathcal{M}}} \langle \alpha \parallel \beta(\beta_1 \cup \beta_2) \rangle \psi$ iff $(w, \rho, c) \models_{\overline{\mathcal{M}}} \langle \alpha \parallel \beta(\beta_1) \rangle \psi$ or $(w, \rho, c) \models_{\mathcal{M}} \langle \alpha \parallel \beta(\beta_2) \rangle \psi$ (by def.). By IH this is true iff $\langle \alpha \parallel \beta(\beta_1) \rangle \psi \in t_w^c(\rho(w, c))$ or $\langle \alpha \parallel \beta(\beta_2) \rangle \psi \in t_w^c(\rho(w, c))$. But according to (t6) this is true iff $\langle \alpha \parallel \beta(\beta_1 \cup \beta_2) \rangle \psi \in t_w^c(\rho(w, c))$
- $(w, \rho, c) \models_{\overline{\mathcal{M}}} \langle \pi_i; \alpha \parallel \lambda \rangle \psi$ iff there existst a world w' such that $wr_i w'$ and $(w', \rho, c) \models_{\overline{\mathcal{M}}} \langle \alpha \parallel \lambda \rangle \psi$ (by def.). Then by IH $\langle \alpha \parallel \lambda \rangle \psi \in t_{w'}^c(\rho(w', c))$. ρ is coherent and saturated, so $\langle \pi_i \parallel \lambda \rangle \langle \alpha \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$. According to (t2) this is true iff $\langle \pi_i; \alpha \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$.
- $(w, \rho, c) \models_{\overline{\mathcal{M}}} \langle \lambda \parallel \pi_i; \beta \rangle \psi$ iff there existst a run $\rho' \in \mathcal{R}$ such that $\rho \triangleleft_i \rho'$ and $(w, \rho', c) \models_{\overline{\mathcal{M}}} \langle \lambda \parallel \beta \rangle \psi$ (by def.). Then by IH $\langle \lambda \parallel \beta \rangle \psi \in t_w^c(\rho'(w, c))$. According to (qm3) $\rho(w, c) R_{w, i}^c \rho'(w, c)$, and by (qm1) $\langle \lambda \parallel \pi_i \rangle \langle \lambda \parallel \beta \rangle \psi \in t_w^c(\rho(w, c))$. According to (t3) this is true iff $\langle \lambda \parallel \pi_i; \beta \rangle \psi \in t_w^c(\rho(w, c))$.
- $(w, \rho, c) \models_{\mathcal{M}} \langle \pi_i; \alpha \parallel \pi_j; \beta \rangle \psi$ iff $(w, \rho, c) \models_{\mathcal{M}} \langle \pi_i \parallel \lambda \rangle \langle \alpha \parallel \pi_j; \beta \rangle \psi$ or $(w, \rho, c) \models_{\mathcal{M}} \langle \lambda \parallel \pi_j \rangle \langle \pi_i; \alpha \parallel \beta \rangle \psi$. Based on previous points of this proof we get that $\langle \pi_i \parallel \lambda \rangle \langle \alpha \parallel \pi_j; \beta \rangle \psi \in t_w^c(\rho(w, c))$ or $\langle \lambda \parallel \pi_j \rangle \langle \pi_i; \alpha \parallel \beta \rangle \psi \in t_w^c(\rho(w, c))$. According to (t4) this is true iff $\langle \pi_i; \alpha \parallel \pi_j; \beta \rangle \psi \in t_w^c(\rho(w, c))$.

Therefore by (qm2), φ is satisfied in \mathcal{M} .

For the other direction, suppose that φ is satisfied in a model \mathcal{M} based on the product $\mathcal{F} \times \mathcal{G}$ of frames $\mathcal{F} = (W, r_1, \dots, r_k)$ and $\mathcal{G} = (\Delta, R_1, \dots, R_k)$. By proposition 1.7 and 3.9 in [5] we may assume, that \mathcal{G} is an intransitive tree of depth $m \leq md(\varphi)$ and $(w_0, x_0, (\varepsilon, \varepsilon)) \models_{\overline{\mathcal{M}}} \varphi$ for some $w_0 \in W$ with x_0 being the root of \mathcal{G} . With every triple (w, x, c) where $w \in W$, $x \in \Delta$ and c is a lists of messages we associate the type $t(w, x, c) = \{\psi \in flc(\varphi) \mid (w, x, c) \models_{\overline{\mathcal{M}}} \psi\}$.

Fix w and c and define a binary relation \sim_w^c on Δ as follows:

- if $x, y \in \Delta$ of depth 0 then $x \sim_w^c y$ iff $t(w, x, c) = t(w, y, c)$.
- if $x, y \in \Delta$ of depth $l < md(\varphi)$ then $x \sim_w^c y$ iff $t(w, x, c) = t(w, y, c)$ and for all $z \in \Delta$ and for all $1 \leq i \leq k$

- if xR_iz then there exists a $z' \in \Delta$ such that yR_iz' and $z \sim_w^c z'$
- if yR_iz then there exists a $z' \in \Delta$ such that xR_iz' and $z \sim_w^c z'$.

Clearly \sim_w^c is an equivalence relation on Δ . Denote by $[x]_w^c$ the \sim_w^c -equivalence class of x , and put $\Delta_w^c \equiv \{[x]_w^c \mid x \in \Delta\}$, $s_w^c([x]_w^c) \equiv t(w, x, c)$ and $[x]_w^c R_{w,i}^c [y]_w^c$ if there exists a $y' \in \Delta_w^c$ such that $xR_iz'y'$. Then by the definition of \sim_w^c , r_w^c is well-defined, and the structure $((\Delta_w^c, r_w^c), s_w^c)$ clearly satisfies (qm1'). The map $x \mapsto [x]_w^c$ is a p-morphism from (Δ, r_2) to (Δ_w^c, r_w^c) , so it also satisfies (qm1). However (Δ_w^c, r_w^c) is not necessarily a tree.

The tree $(T_w^c, <_w^c)$ we need can be obtained from this structure:

$$T_w^c = \left\{ ([x_0]_w^c, \dots, [x_l]_w^c) \mid l \leq m, [x_0]_w^c r_{w,i_1}^c [x_1]_w^c \cdots [x_{l-1}]_w^c r_{w,i_{l-1}}^c [x_l]_w^c \right\}$$

If $u, v \in T_w^c$ then $u <_{w,i}^c v$ iff $u = ([x_0]_w^c, \dots, [x_l]_w^c)$, $v = ([x_0]_w^c, \dots, [x_l]_w^c, [x_{l+1}]_w^c)$ and $x_l R_i x_{l+1}$. $t_w^c([x_0]_w^c, \dots, [x_l]_w^c) \equiv t(w, x, c)$. It is easy to show that $((T_w^c, <_w^c), t_w^c)$ is a quasistate for φ for any $w \in W$ and messages c . Moreover $\varphi \in t_{w_0}^{(\varepsilon, \varepsilon)}([x_0]_{w_0}^{(\varepsilon, \varepsilon)})$. So by taking $q(w, c) \equiv ((T_w^c, <_{w,1}^c, \dots, <_{w,k}^c), t_w^c)$ for each $w \in W$ and each message c we obtain a basic structure (\mathcal{F}, q) for φ satisfying (qm2). We need to define runs through (\mathcal{F}, q) . To do this for each $l \leq m$ and each sequence (x_0, \dots, x_l) in Δ such that $x_0 R_{i_1} \cdots R_{i_l} x_l$, take the map $\rho : (w, c) \mapsto ([x_0]_w^c, \dots, [x_l]_w^c)$. It is easy to check that ρ is a coherent and a saturated l -run. Let \mathcal{R} be the set of all such runs. For $\rho, \rho' \in \mathcal{R}$ let $\rho \triangleleft_i \rho'$ iff $\rho(w, c) <_w^c \rho'(w, c)$ for all $w \in W$ and for all messages c . Then (qm3) holds by definition. It remains to prove (qm4). Let $\rho \in \mathcal{R}_l$, $v \in W$, c any messages and $z \in T_v^c$ be such that $\rho(v, c) <_v^c z$. We have to show that there is $\rho' \in \mathcal{R}_{l+1}$ such that $\rho \triangleleft_i \rho'$, and $\rho'(v, c) = z$. Since $\rho(v, c) <_{v,i}^c z$, we have $\rho(v, c) = ([x_0]_v^c, \dots, [x_l]_v^c)$ and $z = ([x_0]_v^c, \dots, [x_l]_v^c, [x_{l+1}]_v^c)$ for some x_1, \dots, x_l, x_{l+1} with $x_0 R_{j_1}^c x_1 \cdots R_{j_l}^c x_l$ and $[x_l]_v^c r_{v,i}^c [x_{l+1}]_v^c$. By the definition of $R_{v,i}^c$ there is $y \in [x_{l+1}]_v^c$ such that $x_l R_i y$. But then the map $\rho' : (w, c) \mapsto ([x_0]_w^c, \dots, [x_l]_w^c, [y]_w^c)$ is in \mathcal{R} . Thus $(\mathcal{F}, q, \mathcal{R}, \triangleleft)$ is a quasimodel for φ . \square

4. BLOCKS

A block for φ with root w is quadruple $\mathcal{B} = (\mathcal{F}, q, \mathcal{R}, \triangleleft)$ such that

- $\mathcal{F} = (\Delta, <)$ is a tree of depth less equal 1 with root w
- (\mathcal{F}, q) is a basic structure for φ of depth m for some $m < md(\varphi)$
- \mathcal{R} is a set of coherent and saturated runs through (\mathcal{F}, q)
- \triangleleft is a set of binary relation on \mathcal{R} satisfying (qm3) and (qm4)

A set \mathcal{S} of blocks for φ is called *satisfying*, if

- all blocks in \mathcal{S} are of the same depth m for some $m < md(\varphi)$

- \mathcal{S} contains a block satisfying (qm2), and
- for every block $\mathcal{B} = (\mathcal{F}, q, \mathcal{R}, \triangleleft)$ in \mathcal{S} with $\mathcal{F} = (\Delta, <)$ and every $v \in \Delta$, and every messages c there exists a block $\mathcal{B}' = (\mathcal{F}', q', \mathcal{R}', \triangleleft')$ in \mathcal{S} such that $q(v, c) = q'(w', c)$ for the root w' of \mathcal{B}'

Lemma 2. *There is a PDL \times PDL-quasimodel for φ iff there is a satisfying set of blocks for φ such that the number of quasistates in each block does not exceed $M(\varphi) = 1 + (md(\varphi) + 1) \cdot p(\varphi) \cdot |flc(\varphi)|$.*

In the previous lemma $p(\varphi)$ is a finite constant depending on the φ . Its precise definition is in the first chapter of [5].

Proof. We call a quadruple $(\mathcal{F}, q, \mathcal{R}, \triangleleft)$ a *weak quasimodel* for φ if the following conditions hold:

- (wq1) $\mathcal{F} = (W, r_1, \dots, r_k)$ is a finite frame and (\mathcal{F}, q) is a basic structure for φ satisfying (qm2).
- (wq2) \mathcal{R} is a set of runs through (\mathcal{F}, q) and \triangleleft_i is a binary relation on \mathcal{R} , satisfying (qm3) and (qm4).
- (wq3) for all messages c and for all $w, v \in W$ if $w \neq v$ and $wr_i v$ then there exists a block $\mathcal{B}_{w,v}^c = (\mathcal{F}_{w,v}^c, q_{w,v}^c, \mathcal{R}_{w,v}^c, \triangleleft_{w,v}^c)$ in \mathcal{S} with $\mathcal{F}_{w,v}^c = (\Delta, <)$ such that
 - $\Delta \subseteq W$, and $w, v \in \Delta$
 - for all $u \in \Delta$, $q(u, c) = q_{w,v}^c(u, c)$
 - for all $u, u' \in \Delta$ if $ur_i u'$ then $u <_i u'$
 - for all $\rho \in \mathcal{R}$ the restriction $\rho_{w,v}$ of ρ to Δ is a run in $\mathcal{R}_{w,v}^c$

Let $\mathcal{Q}_0 = (\mathcal{F}_0, q_0, \mathcal{R}_0, \triangleleft_0)$ be a block in \mathcal{S} with root w_0 for which (qm2) holds. Now \mathcal{Q}_0 is a weak quasimodel. Suppose now that we have already constructed $\mathcal{Q}_n = (\mathcal{F}_n, q_n, \mathcal{R}_n, \triangleleft_n)$ with $\mathcal{F}_n = (W_n, r_{1n}, \dots, r_{kn})$. For each $w \in W_n - W_{n-1}$ (where let $W_{-1} = w_0$) select a block $\mathcal{B}_w^c = (\mathcal{F}_w^c, q_w^c, \mathcal{R}_w^c, \triangleleft_w^c)$ from \mathcal{S} with w as root and $\mathcal{F}_w^c = (\Delta_w^c, <_w^c)$ such that $q_n(w, c) = q_w^c(w, c)$. The existence of such block follows from (wq3). We may assume that all the selected blocks are pairwise disjoint and $\Delta_w^c \cap W_n = \{w\}$. Define $(\mathcal{F}_{n+1}, q_{n+1})$ by taking

$$\begin{aligned} W_{n+1} &= W_n \cup \bigcup \{ \Delta_w^c \mid w \in W_n - W_{n-1} \}, \\ r_{n+1} &= r_n \cup \bigcup \{ <_w^c \mid w \in W_n - W_{n-1} \}, \\ \mathcal{F}_{n+1} &= (W_{n+1}, r_{n+1}), \end{aligned}$$

$$q_{n+1}(v, c) = \begin{cases} q_w^c(v, c), & \text{if } v \in \Delta_w^c, w \in W_n - W_{n-1} \\ q_n(v, c), & \text{if } v \in W_n \end{cases}$$

Now let $\rho \in \mathcal{R}_n$ and $\bar{s} = \{s \in \mathcal{R}_w^c \mid w \in W_n - W_{n-1} \text{ and } s(w, c) = \rho(w, c)\}$
 Define the extension $\rho \cup \bar{s}$ of ρ by taking for all $v \in W_{n+1}$

$$(\rho \cup \bar{s})(v, c) = \begin{cases} \rho_w^c(v, c), & \text{if } v \in \Delta_w^c, w \in W_n - W_{n-1} \\ \rho(v, c), & \text{if } v \in W_n \end{cases}$$

Let \mathcal{R}_{n+1} be the set of all such extensions and let $(\rho_1 \cup \bar{s}_1) \triangleleft_{n+1, i} (\rho_2 \cup \bar{s}_2)$ iff $\rho_1 \triangleleft_{n, i} \rho_2$ and $s_1 \triangleleft_{w, i}^c s_2$ for all $w \in W_n - W_{n-1}$. It can be checked that \mathcal{R}_{n+1} and \triangleleft_n satisfy (qm3) and (qm4), and $\mathcal{Q}_{n+1} = (\mathcal{F}_{n+1}, q_{n+1}, \mathcal{R}_{n+1}, \triangleleft_{n+1})$ is a weak quasimodel. The *limit quasimodel* defined as follows. Let $\mathcal{F} = (W, r)$, where $W = \bigcup_n W_n$, $r = \bigcup_n r_n$ and let $q = \bigcup_n q_n$. For each sequence of (ρ_0, ρ_1, \dots) , where $\rho_n \in \mathcal{R}_n$ and ρ_{n+1} is an extension of ρ_n let $\rho = \bigcup_n \rho_n$. Let \mathcal{R} is the set of all such runs. For $\rho, \rho' \in \mathcal{R}$ define $\rho \triangleleft_i \rho'$ iff $\rho \triangleleft_{n, i} \rho'$ for all n (where $\rho = \bigcup_n \rho'_n$).

We leave to the reader to show by using (wq1) and (wq3) that $(\mathcal{F}, q, \mathcal{R}, \triangleleft)$ is a quasimodel for φ . Here we show only that all runs in \mathcal{R} are coherent and saturated, i.e. for all $\rho \in \mathcal{R}$, $w \in W$, for all messages c and formula $\langle \pi_i \parallel \lambda \rangle \psi \in \text{flc}(\varphi)$: $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$ iff there exists a world v such that $wr_i v$ and $\psi \in t_v^c(\rho(v, c))$. Suppose that $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$ and let n such that $w \in W_n - W_{n-1}$. Then $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho_n(w, c))$, and by definition \mathcal{Q}_{n+1} there exists $v \in W_{n+1}$ for which $wr_{n+1} v$ and $\psi \in t_v^c(\rho_{n+1}(v, c))$. Conversely, suppose $wr_n v$ and $\psi \in t_v^c(\rho_n(v, c))$. Then it follows from (wq3) that $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$.

For the other direction of the proof, let us assume that we have a given quasimodel $\mathcal{Q} = (\mathcal{F}, q, \mathcal{R}, \triangleleft)$ for φ of depth $m \leq md(\varphi)$ with $\mathcal{F} = (W, r_1, \dots, r_k)$. Note first that we may assume each world w in \mathcal{F} to have arbitrarily many indistinguishable copies in \mathcal{Q} in the following sense. Say that two distinct worlds $w, w' \in W$ are *twins (in \mathcal{Q})* if

- for all messages c , $q(w, c) = q(w', c)$
- for all $v \in W$, $vr_i w$ iff $vr_i w'$ and $wr_i v$ iff $w'r_i v$,
- and for all runs $\rho \in \mathcal{R}$ and for all messages c , $\rho(w, c) = \rho(w', c)$.

To construct a satisfying set \mathcal{S} of blocks, we will associate with each $w \in W$ and each messages c a block $\mathcal{B}_w^c = (\mathcal{F}_w^c, q_w^c, \mathcal{R}_w^c, \triangleleft_w^c)$ with w as root, such that $q_w^c(w, c) = q(w, c)$, and put $\mathcal{S} = \{\mathcal{B}\}$. The resulting \mathcal{S} will clearly be a satisfying set of blocks for φ . Let $w \in W$ and c some lists of messages. First we define inductively sets of runs $\mathcal{P}_k \subseteq \mathcal{R}_k$, $k \leq m$:

- $\mathcal{P}_0 = \{\rho_0\}$

- Given \mathcal{P}_k , we construct \mathcal{P}_{k+1} as follows. For every run $\rho \in \mathcal{P}_k$ and every $x \in T_w^c$ with $\rho(w, c) <_w^c x$, select an $\rho' \in \mathcal{R}_{k+1}$ such that $\rho \triangleleft \rho'$ and $\rho'(w, c) = x$, and put it into \mathcal{P}_{k+1} . Such a run exists by (qm4).

Finally let $\mathcal{P} = \bigcup_{l \leq m} \mathcal{P}_k$. For every $\rho \in \mathcal{P}$ and every $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))$ we then let $Sat(\rho, c, \langle \pi_i \parallel \lambda \rangle \psi) = \{v \in W \mid wRv, \psi \in t_v^c(\rho(v, c))\}$. As ρ is saturated, $Sat(\rho, c, \langle \pi_i \parallel \lambda \rangle \psi) \neq \emptyset$. We select a finite subset $\Delta_w^c(\rho, \langle \pi_i \parallel \lambda \rangle \psi)$ of $Sat(\rho, c, \langle \pi_i \parallel \lambda \rangle \psi)$ in the following way. If $Sat(\rho, c, \langle \pi_i \parallel \lambda \rangle \psi) = \{w\}$, then $\Delta_w^c(\rho, \langle \pi_i \parallel \lambda \rangle \psi) = \{w\}$ as well. Otherwise let $\Delta_w^c(\rho, \langle \pi_i \parallel \lambda \rangle \psi)$ consist of a $v \neq w$ from $Sat(\rho, c, \langle \pi_i \parallel \lambda \rangle \psi)$ together with $m+1$ twins of v . We may assume that the obtained sets $\Delta_w^c(\rho, \langle \pi_i \parallel \lambda \rangle \psi)$ are pairwise disjoint. Now we define

- $\Delta_w^c = \{w\} \cup \bigcup \{\Delta_w^c(\rho, \langle \pi_i \parallel \lambda \rangle \psi) \mid \rho \in \mathcal{P}, \langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(\rho(w, c))\}$,
- for all $v, v' \in \Delta_w^c$, $vr_w^c v'$ iff $v = w$ and vRv' ,
- $\mathcal{F}_w^c = (\Delta_w^c, r_w^c)$ and
- for all $v \in \Delta_w^c$, $q_w^c = q(v)$.

Then \mathcal{F}_w^c is a tree of depth ≤ 1 and (\mathcal{F}_w^c, q_w^c) is a basic structure for φ . The cardinality of Δ_w^c is clearly bounded by $1 + (md(\varphi) + 1) \cdot p(\varphi) \cdot |flc(\varphi)|$.

It remains to define a set \mathcal{R}_w^c of coherent and saturated runs through (\mathcal{F}_w^c, q_w^c) and binary relations $\triangleleft_{w,i}^c$ on \mathcal{R}_w^c such that (qm3) and (qm4) hold. Let $v \in \Delta_w^c$, $v \neq w$ and suppose that ρ and ρ' are functions whose domain contains Δ_w^c and $\rho(w, c) = \rho'(w, c)$. Define a function $\rho +_v \rho'$ with domain Δ_w^c by taking, for all $z \in \Delta_w^c$

$$(\rho +_v \rho')(z, c) = \begin{cases} \rho(z, c), & \text{if } z = v \\ \rho'(z, c), & \text{if } z \neq v. \end{cases}$$

Using this ‘addition’ function we now define sets \mathcal{R}_{wl}^c of l -runs for every $l \leq m$. Let \mathcal{R}_{w0}^c be the restriction of ρ_0 to Δ_w^c . For $k > 0$, we put all the restrictions of runs from \mathcal{P}_l (to Δ_w^c) into \mathcal{R}_{wl}^c , and also add the functions $\rho_1 +_{v_1} (\rho_2 +_{v_2} (\dots (\rho_n +_{v_n} \rho) \dots))$, where $1 \leq n \leq l$, $\rho \in \mathcal{P}_l$, $\rho_1, \dots, \rho_n \in \mathcal{R}_l$ such that $\rho(w) = \rho_j(w)$, for $1 \leq j \leq n$, and v_1, \dots, v_n are pairwise distinct points in Δ_w^c different from w .

Obviously every run $s \in \mathcal{R}_w^c$ is coherent. We show that it is w -saturated. This is clear if s is a restriction of some run from \mathcal{P} . Otherwise, s is on the form $\rho_1 +_{v_1} (\rho_2 +_{v_2} (\dots (\rho_n +_{v_n} \rho) \dots))$, for some $n \leq m$. So, we modified the w -saturated run ρ at most m places. Take some formula $\langle \pi_i \parallel \lambda \rangle \psi \in t_w^c(s(w, c))$. Since we selected for Δ_w^c $m+1$ twins for each point in $Sat(\rho, c, \langle \pi_i \parallel \lambda \rangle \psi)$, there is still at least one v left to ‘saturate s with respect to $\langle \pi_i \parallel \lambda \rangle \psi$ ’, that is such that $\psi \in t_v^c(s(v, c))$.

Finally let $s = \rho_1 +_{v_1} (\rho_2 +_{v_2} (\dots (\rho_n +_{v_n} \rho) \dots))$ and $s' = \rho'_1 +_{v'_1} (\rho'_2 +_{v'_2} (\dots (\rho'_l +_{v'_l} \rho') \dots))$ be two runs in \mathcal{R}_w^c . If s or s' is a restriction of some run from \mathcal{P} , then we consider n or l to be 0, respectively. We let $s \triangleleft_{w,i}^c s'$ if the following hold:

- $s \in \mathcal{R}_{w,l}^c$ and $s' \in \mathcal{R}_{w,l+1}^c$, for some $l < m$,
- $\rho \triangleleft_i \rho'$,
- $n \leq l$ and $v_j = v'_j$ for all $1 \leq j \leq n$,
- for all $z \in \Delta_w^c$, $\rho_j(z, c) r_{w,i}^c \rho'_j(z, c)$ whenever $1 \leq j \leq n$ and $\rho(z, c) r_{w,i}^c \rho'_j(z, c)$ whenever $n+1 \leq j \leq l$.

Then (qm3) holds by definition. We show that (qm4) also holds. Suppose that $s = \rho_1 +_{v_1} (\rho_2 +_{v_2} (\dots (\rho_n +_{v_n} \rho) \dots))$, $z \in \Delta_w^c$, $x \in T_w^c$ and $s(z, c) R_{w,i}^c x$. We need a run $s' \in \mathcal{R}_w^c$ such that $s \triangleleft_{w,i}^c s'$ and $s'(z, c) = x$.

Case 1: $z = v_j$ for some $1 \leq j \leq n$. Then $s(z, c) = \rho_j(z, c) = v_j$ for some $\rho_j \in \mathcal{R}$. As the original quasimodel \mathcal{Q} satisfies (qm4), we have a run $\rho'_j \in \mathcal{R}$ such that $\rho_j \triangleleft_i \rho'_j$ and $\rho'_j(z, c) = x$. Similarly for all $l \neq j$, $1 \leq l \leq n$, take a run ρ'_l from \mathcal{R} such that $\rho_l \triangleleft_i \rho'_l$ and $\rho'_l(w, c) = \rho'_j(w, c)$. Finally take a run ρ' from \mathcal{P} such that $\rho \triangleleft_i \rho'$ and $\rho'(w, c) = \rho'_j(w, c)$. Such a run exists by the definition of \mathcal{P} . Then $s' = \rho'_1 +_{v_1} (\rho'_2 +_{v_2} (\dots (\rho'_n +_{v_n} \rho') \dots))$ is a run in \mathcal{R}_w^c as required.

Case 2: $z \neq v_j$ for any $1 \leq j \leq n$. Then $s(z, c) = \rho(z, c)$. Select a run ρ'_{n+1} from \mathcal{R} such that $\rho \triangleleft_i \rho'_{n+1}$ and $\rho'_{n+1}(z, c) = x$. For each j , $1 \leq j \leq n$, take a run ρ'_j from \mathcal{R} such that $\rho_j \triangleleft_i \rho'_j$ and $\rho'_j(w, c) = \rho'_{n+1}(w, c)$. Finally, take a run ρ' from \mathcal{P} such that $\rho \triangleleft_i \rho'$ and $\rho'(w, c) = \rho'_{n+1}(w, c)$. Then $s' = \rho'_1 +_{v_1} (\rho'_2 +_{v_2} (\dots (\rho'_{n+1} +_z \rho') \dots))$ is a run in \mathcal{R}_w^c as required.

Thus $(\mathcal{F}_w^c, q_w^c, \mathcal{R}_w^c, \triangleleft_w^c)$ is indeed a block with w as root. \square

5. CONCLUSIONS

It has been shown that there is a quasimodel for φ iff there exists a finite set S of finite blocks. The cardinality of S and the size of blocks are bounded by φ . From these blocks we are able to construct the quasimodel we need. This means that we are able to build up a finite construction for any formula φ of logic PDL×PDL. The number of this kind of finite constructions is finite for any formulae, hence this logic is decidable.

REFERENCES

- [1] László Aszalós and Philippe Balbiani. Logical aspects of user authentication protocols. In *Proc. 7th Seminar RelMiCS, 2nd Workshop Kleene Algebra*, pages 277–287, Bad Malente, may 2003.

- [2] László Aszalós and Philippe Balbiani. Some decidability result for logic constructed for checking user authentication protocols. *Journal of Computer Science and Control Systems*, 2008.
- [3] Michael Burrows, Martín Abadi, and Roger Needham. A logic for authentication. In *Proceedings of the Royal Society of London*, volume 426, pages 233–271, 1989.
- [4] John Clark and Jeremy Jacob. On the security of recent protocols. *Information Processing Letters*, 56:151–155, 1995.
- [5] Dov M. Gabbay, Ágnes Kurucz, Frank Wolter, and Michael Zakharyashev. *Many-Dimensional Modal Logics: Theory and Applications*. Studies in Logic and the Foundations of Mathematics. Elsevier, 2003.
- [6] Dov M. Gabbay and Valentin B. Shehtman. Products of modal logics. I. *Logic Journal of the IGPL. Interest Group in Pure and Applied Logics*, 6(1):73–146, 1998.
- [7] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. MIT Press, 2000.
- [8] Ágnes Kurucz. $S5 \times S5 \times S5$ lacks the finite model property. In Frank Wolter, Heinrich Wansing, Maarten de Rijke, and Michael Zakharyashev, editors, *Advances in Modal Logic*, pages 321–327. World Scientific, 2000.
- [9] A. D. Rubin and P. Honeyman. Formal methods for the analysis of authentication protocols. Technical report, 1993.

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF DEBRECEN; INSTITUTE DE RECHERCHE EN INFORMATIQUE DE TOULOUSE

E-mail address: aszalos@inf.unideb.hu, balbiani@irit.fr

USING SPREADSHEETS FOR SOLVING LOGIC PUZZLES

MÁRIA BAKÓ AND LÁSZLÓ ASZALÓS

ABSTRACT. The consequence relation and perfect usage of derivation is a necessary knowledge for lawyers, economists, engineers and for hundreds of other different professionals. Students in the low and mid level education system are able to solve simple logical puzzles without any special training, but some of the puzzles in Smullyan's books present a challenge even for university students. Some of the logic and artificial intelligence courses contain methods for examining deductions and the students might even use special tools and software to facilitate the process. In this article we would like to show you that the well-known spreadsheet software is a great tool to solve puzzles that can be expressed in sentence (or in zero-order) logic. Many students therefore will be able to solve even complicated puzzles without learning any special soft wares. This can be very helpful for teachers teaching deduction, but it can be also helpful for the teachers teaching spreadsheet software usage, because their students would be able to solve challenging problems by learning to use new functions. The knowledge of these functions could be useful later solving other type of problems, too.

1. INTRODUCTION

The curriculum of the elementary and the secondary schools requires the teaching of information technology and training in computer skills.

At many schools students are preparing for the European Computer Driving License exam because it is an internationally recognized qualification, and helps the students to boost their productivity, later on it helps them to be competitive on the labor market and successful at the entrance exams. The students favorite topics are the word processing and the spreadsheets softwares because these are the most commonly used in the everyday life. Spreadsheets

Received by the editors: December 6, 2008.

2000 *Mathematics Subject Classification.* 03B35, 68T20.

1998 *CR Categories and Descriptors.* I.2.3 [**Topic**]: Deduction and Theorem Proving – *case-based deduction*; F.4.1 [**Topic**]: Mathematical logic – *mechanical theorem proving* .

Key words and phrases. Logic puzzles, Case-based deduction, Mechanical theorem proving, Spreadsheets.

This paper has been presented at the 7th Joint Conference on Mathematics and Computer Science (7th MaCS), Cluj-Napoca, Romania, July 3-6, 2008.

TABLE 1. Truth-table of $((A \supset B) \supset A) \supset A$

A	B	$A \supset B$	$(A \supset B) \supset A$	$((A \supset B) \supset A) \supset A$
t	t	t	t	t
t	f	f	t	t
f	t	t	f	t
f	f	t	f	t

are good for budgeting or accounting planning, for calculations and analysis, and further more it is a good tool to practice mathematics or different mathematical concepts/ideas or sometimes just to play with. The journal Spreadsheets in Education [17] contains lots of interesting examples. In this article we will show how we can utilize spreadsheets in the teaching of mathematics, especially how to solve logic puzzles in [14, 15, 16].

These puzzles are suitable tools to show the elements of the deduction [8, 20]. Many students at university level use only true-tables to check the validity of a logical consequence, and they survive with this knowledge while they work with toy problems. But one of the puzzles solved in the paper is so complicate that the corresponding truth-table has more than 4,500 rows. The construction of the spreadsheet which solves this puzzle is not a complicated task and use only simple functions. Hence this method can be used at elementary schools level, too. Moreover the solution process of logic puzzles is a challenging problem, and differs from the usual spreadsheet problems. This kind of diversity helps us to hold up the interest of students, which is the key point of the education. We remark that the mathematical contests at elementary school level in Hungary often contain these kind truth-teller and liar logic puzzles.

2. TRUTH-TABLES

We have several methods to check logical consequences. In the case of sound and complete calculus we can replace the consequence relation with decidability, and we can use automated theorem provers. In the case of sentence logic we can use the truth-tables. In the rows of this table at first we list all the logical combinations of the logical variables, and next we determine the values of the formulae in these cases. The Table 1 shows such a table, where the columns belong to the subformulae of the original formula. In our truth-tables t denote the true, f denotes the false logical value. Our formula has two different logical variables, so the table has 2^2 rows. In practice we use the variant in the Table 2. If we use computers to fill the table, our table on Table 3 became more compact.

TABLE 2. Compact truth-table of $((A \supset B) \supset A) \supset A$

$((A \supset B) \supset A) \supset A$						
t	t	t	t	t	t	t
t	f	f	t	t	t	t
f	t	t	f	f	t	f
f	t	f	f	f	t	f

In the case of the Table 1 the last column determines the property of the formula. If all the values in this column are true, then the formula is a logical law. If all the values in the column are false, then the formula is a contradiction; but if there is at least one true value, the formula is satisfiable.

The formula B is the logical consequence of formulae A_1, \dots, A_n , if the formula $A_1 \wedge \dots \wedge A_n \supset B$ is a logical law. This theorem is the reason why we are interested in true-tables.

TABLE 3. Very compact truth-table of $((A \supset B) \supset A) \supset A$

A	B	$((A \supset B) \supset A) \supset A$
t	t	t
t	f	t
f	t	t
f	f	t

3. SPREADSHEETS AND LOGIC

The true and false logical values can be replaced with the numbers 1 and 0, and we can write arithmetical expressions instead of logical expressions. For example instead of negation of x , conjunction of x and y and disjunction of x and y we can write $1 - x$, $\min\{x, y\}$ and $\max\{x, y\}$, respectively.

At the most recent spreadsheets we do not need to use such tricks. The Excel, which is the most taught and used spreadsheet, contains the following logical functions: **TRUE**, **FALSE**, **NOT**, **AND**, **OR** and **IF**. With these functions we can write any logical function. The implication is not listed before, we need to use the well-know $\neg A \vee B$ transformation of $A \supset B$. The usage of Excel syntax gives shorter formulae than the arithmetic notation, we will use it in the following.

Let us see a simple problem: *the formula $A \vee C$ is the consequence of formulae $A \vee B$ and $B \vee C$?* We have 3 logical variables, so the truth-table has 8 rows. The Table 4 lists all the cases. We can give these data by typing, but as we have more and more variables it became a lengthy procedure. Let us

TABLE 4. Truth-table for checking logical consequence

A	B	C	$A \supset B$	$A \supset C$	$B \supset C$
t	t	t	.	.	.
t	t	f	.	.	.
t	f	t	.	.	.
t	f	f	.	.	.
f	t	t	.	.	.
f	t	f	.	.	.
f	f	t	.	.	.
f	f	f	.	.	.

allow Excel to work! Let us write the numbers from 0 to 7 in the first column of the worksheet! To simplify the description we use the following notation: $A1 \leftarrow 0$, and $A2 \leftarrow =A1+1$. This means that we write 0 and =A1+1 in A1 and A2, respectively. Finally copy the A2 to A3 : A8! Now list the different combinations of logical variables: $B1 \leftarrow =\text{ISEVEN}(A1)$, $C1 \leftarrow =\text{ISEVEN}(A1/2)$ and $D1 \leftarrow =\text{ISEVEN}(A1/4)$, by using the automated conversion of ISEVEN. Next we can copy B1 : D1 to B2 : D8, and we are ready to write down our logical formulae. We suggest to save this and similar files as a template to speed up the work on lessons. In Excel we can write down the hypotheses $A \vee B$, $B \vee C$ and consequence $A \vee C$ as $E1 \leftarrow =\text{OR}(B1, C1)$, $F1 \leftarrow =\text{OR}(B1, D1)$ and $G1 \leftarrow =\text{OR}(B1, D1)$, respectively. Of course we need to copy E1 : G1 to E2 : G8 to fill the truth-table.

According to the definition of logical consequence the remaining question is the following: is there any row where in the columns E and F the values are true, and in the column G the value is false. In this small table we can find the suitable row easily, but in big truth tables we suggest to set up the Excel's automated filter for columns E – G, and choose there the right logical values.

4. SMULLYAN'S PUZZLES

In his book “What is the name of this book?” Raymond M. Smullyan used a lot of puzzles to illustrate the background of the Gödel incompleteness theorem. These puzzles became popular and nowadays are being published in amusement magazines, too. In each section of the book different conditions are met. In the best known type of puzzles we have only two types of people, knights and knaves. Knights always tell the truth and knaves always lie.

In the puzzles the inhabitants make statements about themselves and about the others, for example “*A* is a knave.” or “*B* said that she was not a knave.” Usually, to solve a puzzle we must determine the type of persons.

In these knight-knave puzzles we can use the function `ISEVEN` again, for example we could use the formula `=IF(ISEVEN(A1/4),"knight","knave")` when we list all the combination of types of inhabitants.

In some Smullyan puzzle we have three kinds of persons: truth-teller, liar and story-teller. The truth-teller always tell the truth, the liar always lies and the story-tellers sometimes tell the true and sometimes lie. The problem of Zrínyi Ilona [5] contest, 1992/28 for K7 students is the following:

One of *A*, *B* and *C* is the guilty. The guilty is a truth-teller and the the others are not truth-tellers. The inhabitants said

A : I'm innocent.

B : It is true.

C : *B* is not a story-teller.

Who is the guilty?

Considering these kind of puzzles the function `ISEVEN` is not enough to generate all the combinations of types of persons. We can use the functions `MOD` and `CEILING` instead of it. The complete solution of the puzzle can be found in [4].

5. LADY AND THE TIGER

The previous puzzle could be solved easily without a spreadsheet, of course. Let us see a puzzle [15] which is more complicate:

The prisoner is informed that there is one room with a lady in it; all the others either have a tiger in them or are empty. The sign on the door of the room with the lady in it is true, the signs on all the doors with tigers in them are false, and the signs on the doors of empty rooms can be either true or false. The signs of the rooms are the following:

I The lady is an odd-numbered room.

II This room is empty.

III Either sign on Room V is right or sign on Room VII is wrong.

IV Sign on Room I is wrong.

V Either sign on Room II or sign on Room IV is right.

VI Sign on Room III is wrong.

VII The lady is not in Room I.

VIII This room has a tiger and Room IX is empty.

IX This room has a tiger and sign on Room VI is wrong.

Where is the lady?

For the simpler formalization we say that we have nine rooms, in any room can be a tiger and the lady can be in any room. By this we have $2^9 \times 9 = 4608$ cases, so the traditional truth-table method does not work, hence we use spreadsheet software again. The complete solution of the puzzle can be found again in [4].

We can use the autofilter to find all the different solutions, and it turns out that the lady could be in several room. The text of the original puzzle contains a sentence that I not mentioned until now. If we determine that the room VIII is empty or not, we could solve the puzzle. So apply the filter to the column T (status of room VIII), and we can realize, if this room is empty, the lady can be in many rooms, but if a tiger is in room VIII, then the lady can be only in the room VII. So this is the solution.

After solving this puzzle with our students, we can show them the original solution of Smullyan, which is one page long solution. It is a good thing, if the students realize, that using spreadsheets is only one of the possible solving methods, and maybe not the fastest or simplest one.

6. DISCUSSION

The knight-knave logic puzzles are very popular; the `Knights_and_knaves` Wikipedia page gives a long list of computer games, movies, cartoons and comics where we can meet with such puzzles. You can meet regularly on Hungarian mathematics contests[1, 5]. These kind of puzzles are interesting for researchers, too. In the last thirty years since [14] published, several methods used to solve these kind of methods. Here we give a not complete list of different solving methods:

- an extension of the classical propositional logic [9],
- first-order logic, and automated theorem prover [11],
- propositional logic and tableaux method [13],
- modal logic and tableaux method [3] to solve all the puzzles in [14],
- rewriting graphs [10],
- CLIPS programming language [18],
- Prolog programming language [2, 19],
- SHQL meta-queries [7]
- Smodel system [12],
- SAT solver [6].

From this list we can realize, if our students would like to apply some of the methods mentioned before, then they need to learn some special logic, method or some new programming language. Most of these logics, methods and languages so specific, that only the best students at the university level

are able to understand. Our alternative is the application of a well-known tool. Our the students at primary school are able to use, yet. They do not need to learn a new software, just some functions of the familiar spreadsheet software, and they are able to solve very hard puzzles, too.

The solving of puzzles with and without any tools are different. When somebody solve a puzzle without any tools, then he needs to analyse the problem, needs to consider all the aspects of the problem, and needs to explorer relationships and rules to construct the solution. If somebody use a suitable tool, then usually he only needs to formulate the problem respecting the rules of the tool. Then the tool gives all the solutions of the problem.

Obviously it is better if a student can solve such puzzles without any tool. But most of these puzzles are hard, and an ordinary student cannot solve it alone. At solving puzzles with spreadsheets, the teacher can show interesting relationships by filtering out the uninteresting part of the table. With this the teacher can show/teach the elements of the deduction. Moreover he can eliminate the students misunderstandings showing counterexamples. We think, that the successful solution of hard puzzles generate a positive attitude for students, and this can help to raise the interest of the students for logic.

7. CONCLUSION

In the article we have shown some interesting puzzles, that are very much liked and gladly solved by students. Moreover we have presented their solution method using truth tables. This process can be speeded up using spreadsheets. These puzzles can be refreshing exceptions between statistical and economical exercises. Furthermore the process of solving these kinds of puzzles is a good preparation for understanding and learning the logical consequence and deduction.

REFERENCES

- [1] Abacus Journal of Mathematics <http://www.mategye.hu/abacus/>
- [2] L. Aszalós. *Smullyan's logical puzzles and its automated solving*. Tech. Report of Institute of Mathematics and Informatics, University of Debrecen, 2000/14, (in Hungarian)
- [3] L. Aszalós *Automated puzzle solving*. Journal of Applied Non-Classical Logics, 12(1):99-116, 2002
- [4] M. Bakó and L. Aszalós *Solution of the puzzles of this article* <http://www.inf.unideb.hu/~aszalos/puzzle.xls>, 2008
- [5] Mihály Csordás, Éva Háriné Kun, Zsuzsa Janics, Tibor Nagy, Mária Dr. Palotainé Böröczki, and István Szabó. *Zrínyi Ilona Matematikaverseny feladatai 1992-2000, 7. osztály*. MATEGYE Alapítvány, 2008.
- [6] Elizabeth Cassell *Knights and Knaves translator into SAT matrices* <https://twiki.soe.ucsc.edu/twiki/bin/view/SoeClasses/CMPS240>

- [7] P. Doherty, J. Kachniarz and A. Szalas *Meta-Queries on Deductive Databases* Fundamenta Informaticae, 40, 1, 17-30, 1999
- [8] H. James Hoover and Piotr Rudnicki. Teaching freshman logic with Mizar-MSE. In *DI-MACS Symposium on Teaching Logic and Reasoning in an Illogical World*, Rutgers University, Piscataway, New Jersey, 1996.
- [9] A. Kolany. *A general method of solving Smullyan's puzzles*. Logic. Log. Philos., No. 4. 97–103., 1996.
- [10] B. Nagy *SW-type puzzles and their graphs*, Acta Cybernetica 16, 67-82., 2003
- [11] H. J. Ohlbach *Predicate Logic Hacker Tricks*, Journal of Automated Reasoning 1, 435–440, 1984
- [12] D. W. Orr *Ecological Literacy: Education and the Transition to a Postmodern World* SUNY Press, 1992
- [13] J. F. Pelletier *Using Semantic Tableaux to Solve Knight and Knave problems* <http://www.sfu.ca/jeffpell/Cogs300/KnKTableaux.pdf>
- [14] R. M. Smullyan. *What is the name of this book? (The riddle of Dracula and other logical puzzles)*. Prentice Hall, Inc., 1978.
- [15] R. M. Smullyan. *The Lady or The Tiger? and Other Logical Puzzles*. Alfred A. Knopf, Inc., 1982.
- [16] R. M. Smullyan. *The riddle of Scheherazade, and other amazing puzzles, ancient and modern*. Alfred A. Knopf, Inc., 1996.
- [17] Spreadsheets in Education (ISSN 1448-6156) <http://epublications.bond.edu.au/ejsie/>
- [18] Student group 1. *Programming language CLIPS* <http://www.ref.net.ua/work/det-26349.html> (in Russian)
- [19] P. Szeredi *Teaching Constraints through Logic Puzzles* in Recent Advances in Constraints, Springer LNCS 3010, 196–222, 2004.
- [20] A. Zalewska. An application of Mizar MSE in a course in logic. In J. Szrednicki, editor, *Initiatives in Logic*, pages 224–230. Martinus Nijhoff Publishers, 1987.

UNIVERSITY OF DEBRECEN, FACULTY OF PEDAGOGY, FACULTY OF INFORMATICS
E-mail address: bakom@hwpf.hu, aszalos@inf.unideb.hu

A METHOD WITH RANDOM MODIFICATION OF GRADIENT COMPONENTS FOR CONVEX MODELS

PAVEL BĂLAN

ABSTRACT. A stochastic method is proposed and analyzed, that is a probabilistic generalization of gradient method, for solving convex models with restrictions. A random change of "old" partial derivatives with "new" ones is performed from one iteration to another. Convergence aspects of this method are analyzed for the case when the step is adjusted programmatically. Certain conditions are indicated, that ensure its convergence to the optimal solution with probability 1.

1. INTRODUCTION

Current method has an iterative approach of "Connection-Disconnection" type. Using two random variables a series of indices is generated for next iteration. One is meant to be used for target function and the other one - for function that describes the restriction. Distribution laws that can be used can be a priori set or can be modified within iterations. Usage of different distribution laws may increase convergence speed of the method. Generated indices suggest what components of movement vector should be modified. "Old" components are replaced with corresponding partial derivatives with the same indices from the above mentioned series. The idea of "Connection-Disconnection" mechanism is following: if the restriction inequality is satisfied

Received by the editors: December 19, 2008.

2000 *Mathematics Subject Classification*. 65K05, 90C15.

1998 *CR Categories and Descriptors*. G.1.6 [**Optimization**]: Subtopic - *Convex programming*; Subtopic - *Nonlinear programming*; Subtopic - *Stochastic programming*.

Key words and phrases. stochastic method, differentiable optimization, probabilistic generalization of gradient method, convex model, partial derivatives, movement direction, convergence, with probability 1, almost sure, connect-disconnect type.

at the current iteration, then the vector that determines the movement direction is built using partial derivatives of target function; otherwise this vector contains only partial derivatives of the function from restriction.

2. A METHOD WITH RANDOM MODIFICATION OF GRADIENT COMPONENTS FOR CONVEX MODELS

The following problem is considered:

$$(1) \quad \begin{cases} F(x) \longrightarrow \min \\ \varphi(x) \leq 0 \\ x \in X \end{cases}$$

where X represents a compact and convex set in Euclidian space E^n .

Suppose that problem (1) is solvable.

Let us define $V(X, \varepsilon) = \bigcup_{x \in X} V(x, \varepsilon)$ - vicinity with ε radius of X set. By $V(x, \varepsilon)$ is marked the vicinity with radius $\varepsilon > 0$ of the point $x \in E^n$, or, formally:

$$V(x, \varepsilon) = \{y \in E^n : \|x - y\| < \varepsilon\}$$

Let's admit for some $\varepsilon > 0$ that $F(x)$ and $\varphi(x)$ are convex and differentiable functions (with continuous gradients) on $V(X, \varepsilon)$. Therefore, for $\forall x \in X$ are defined the following vectors:

$$\begin{aligned} (g_{F,1}(x), \dots, g_{F,n}(x)) &= g_F(x) = \text{grad } F(x) = \left(\frac{dF(x)}{dx_1}, \dots, \frac{dF(x)}{dx_n} \right) \\ (g_{\varphi,1}(x), \dots, g_{\varphi,n}(x)) &= g_\varphi(x) = \text{grad } \varphi(x) = \left(\frac{d\varphi(x)}{dx_1}, \dots, \frac{d\varphi(x)}{dx_n} \right) \end{aligned}$$

Obviously, the norms $\|g_F(x)\| = \sqrt{\sum_{i=1}^n g_{F,i}^2(x)}$, $\|g_\varphi(x)\| = \sqrt{\sum_{i=1}^n g_{\varphi,i}^2(x)}$ are continuous functions on X compact and, consequently, a constant C exists, so that $\|g_F(x)\| \leq C$, $\|g_\varphi(x)\| \leq C$ for $\forall x$. Hence $\|g_{F,i}(x)\| \leq C$, $\|g_{\varphi,i}(x)\| \leq C$, $\forall i = \overline{1, n}$, $\forall x \in X$.

The numerical method that is proposed to solve the problem (1) has an iterative approach. Assuming that we are positioned on k^{th} iteration the schema is following:

Step 1. Two random variables ξ^k, ψ^k are simulated in series $m_k \geq 1, l_k \geq 1$ of independent probes with discrete distribution laws a priori defined:

ξ^k	1	2	...	n	ψ^k	1	2	...	n
P	$P_{\xi,1}^k$	$P_{\xi,2}^k$...	$P_{\xi,n}^k$	P	$P_{\psi,1}^k$	$P_{\psi,2}^k$...	$P_{\psi,n}^k$

That is, on every iteration k the sets $I_k = \{i_1, i_2, \dots, i_{m_k}\}, J_k = \{j_1, j_2, \dots, j_{l_k}\}$ of elements that are independent realizations of ξ^k, ψ^k variables with distribution laws defined above are generated, where

$$(2) \quad \begin{aligned} P_{\xi,i}^k &\geq \underline{P}_\xi > 0, \forall i = \overline{1, n}, \forall k = 0, 1, \dots \\ P_{\psi,i}^k &\geq \underline{P}_\psi > 0, \forall i = \overline{1, n}, \forall k = 0, 1, \dots \end{aligned}$$

Particularly, we can take $m_k = l_k = 1$, that is a single simulation is accomplished for every random variable ξ^k, ψ^k on every iteration.

Step 2. $g_F^k(x^k), g_\varphi^k(x^k)$ vectors are calculated according to the rule:

$$(3) \quad \begin{aligned} g_F^k(x^k) &= (g_{F,1}^k, \dots, g_{F,i}^k, \dots, g_{F,n}^k), \quad g_{F,i}^k = \begin{cases} g_{F,i}^{k-1}, & \text{if } i \notin I_k \\ \frac{dF(x)}{dx_i}, & \text{if } i \in I_k \end{cases} \\ g_\varphi^k(x^k) &= (g_{\varphi,1}^k, \dots, g_{\varphi,i}^k, \dots, g_{\varphi,n}^k), \quad g_{\varphi,i}^k = \begin{cases} g_{\varphi,i}^{k-1}, & \text{if } i \notin J_k \\ \frac{d\varphi(x)}{dx_i}, & \text{if } i \in J_k \end{cases} \\ \forall i &= \overline{1, n} \end{aligned}$$

Step 3. The element x^{k+1} is determined according to relation:

$$(4) \quad x^{k+1} = \prod_X (\tilde{x}^{k+1}), \quad \text{where } \tilde{x}^{k+1} = x^k - \rho_k \eta^k$$

$\prod_X (\tilde{x}^{k+1})$ represents the projection of element \tilde{x}^{k+1} on the set X .

Starting point x^0 is arbitrary taken from X (it can be indicated under certain considerations for some concrete situations).

Step 4. The numerical sequence $\{\eta^k\}$ is defined in following way:

$$(5) \quad \eta^k = \begin{cases} \frac{g^k}{\|g^k\|}, & \text{if } g^k \neq \bar{0}, \forall k = 1, 2, \dots \\ \bar{0}, & \text{for } g^k = \bar{0} \end{cases}$$

g^0 is considered to be an arbitrary, but bounded vector.

Necessarily, classical requirements are imposed on sequence $\{\rho_k\}$ to ensure the convergence from probabilistic point of view of the iterative process (4)

which have the form:

$$(6) \quad \rho_k \geq 0; \rho_k \xrightarrow[k \rightarrow \infty]{} 0; \sum_{k=0}^{\infty} \rho_k = \infty$$

Additionally we will require existence of such a number $\bar{\varepsilon} > 0$, that for $\forall r \in (0, \bar{\varepsilon}]$ and $\forall \tau \in (0, 1)$ the convergence of the series occurs [1]:

$$(7) \quad \sum_{k=0}^{\infty} \tau^{L(k,r)} < \infty, \quad L(k,r) = \begin{cases} 0, & \text{if } \rho_k \geq r \text{ or } k = 0 \\ s_k, & \text{if } \sum_{l=k-s_k}^k \rho_l < r \text{ and } \sum_{l=k-s_k-1}^k \rho_l \geq r \end{cases}$$

In other words s_k is the biggest integer number among all numbers $j \geq 0$ that satisfies the relation $\sum_{l=k-j}^k \rho_l$.

Remark 1. *Particularly, it is easy to show that numerical sequence $\rho_k = \frac{R}{(k+1)^\alpha}$, $R > 0$, $\alpha \in (0, 1]$ satisfies the conditions (6)-(7).*

B.T. Polyak proposes a schema to solve general convex models [4]. This is a deterministic "connect-disconnect" schema where the vector g^k defines movement direction and is formulated in following way:

$$g^k = g^k(x^k) = \begin{cases} \text{grad } F(x^k), & \text{if } \varphi(x^k) \leq 0 \\ \text{grad } \varphi(x^k), & \text{if } \varphi(x^k) > 0 \end{cases}$$

But actual method proposes another representation of movement vector g^k :

$$(8) \quad g^k = g^k(x^k) = \begin{cases} g_F^k(x^k), & \text{if } \varphi(x^k) \leq \tau_k \\ g_\varphi^k(x^k), & \text{if } \varphi(x^k) > \tau_k \end{cases}, \quad \tau_k > 0$$

Remark 2. *The iterative process can be modified in following way: different distribution laws can be taken for random variables ξ^k, ψ^k , from one iteration to another, with the condition that relation (2) holds. This can favour the increase of convergence speed of the sequence $\{x^k\}$.*

Applicability of described method can be confirmed, first of all, by establishing convergence, in probabilistic terms, of sequence $\{x^k\}$ towards optimal domain of solutions X^* . A special interest represents the convergence with probability 1 (only this type of convergence can be accepted with confidence from applied point of view).

Theorem 1. For fixed $\forall \varepsilon > 0$, **all** elements of random sequence $\{x^k\}_{k \geq 0}$, obtained as a result of application of described method, are localized almost certain (with probability 1) in vicinity $V(X^*, 2\varepsilon)$, but excepting a finite number of elements. Formally this can be represented in the following way:

$$P \left\{ \theta : \lim_{k \rightarrow \infty} \min_{x^* \in X^*} \|x^k - x^*\| = 0 \right\} = 1,$$

where $x^k = x^k(\theta)$ and $\theta = (\theta^0, \theta^1, \dots, \theta^k, \dots)$,
 $\theta^k = (i^0, i^1, \dots, i^k) \in B^k$ - σ - algebra generated by
 Cartesian product $((I_0 \times J_0) \times (I_1 \times J_1) \times \dots \times (I_k \times J_k))$.

Proof. If $X \subset V(X^*, 2\varepsilon)$ then the statement is obvious.

Let's admit $X \setminus V(X^*, 2\varepsilon) \neq \emptyset$. A problem of the form:

$$(9) \quad \begin{cases} F(x) \longrightarrow \min \\ \varphi(x) \leq \tau_k, \tau_k > 0, \tau_k \rightarrow 0, \sum_{k=0}^{\infty} \rho_k \tau_k = \infty, \frac{\tau_k}{\rho_k} \rightarrow \infty \\ x \in X \end{cases}$$

is associated to initial model (1) on every iteration k .

Two stages for proof development will be accentuated.

Stage 1. Firstly the existence of a subsequence $\{x^{k_l}\} \subset \{x^k\}_{k \geq 0}$, that almost certain is contained in $V_X(X^*, \varepsilon)$ will be demonstrated, i.e.

$$P \left\{ \exists \{x^{k_l}\} \subset \{x^k\}_{k \geq 0} : x^{k_l} \in V_X(X^*, \varepsilon) \right\} = 1.$$

Let's suppose the contrary. In this case for some $q \in (0, 1)$ a natural number K_q can be indicated such that the following event is produced

$$(10) \quad A_1 = \left\{ \exists K_q : \forall k \geq K_q, \left\| x^k - x^* \right\| \geq \varepsilon, \text{ or } x^k \notin V_X(X^*, \varepsilon), \forall x^* \in X^* \right\}$$

with probability $P(A_1) \geq q$.

Let's denote $X_\varepsilon = X \setminus V(X^*, \varepsilon)$.

Since $F(x)$, $\varphi(x)$ are convex and differentiable, the following inequalities are valid:

$$F(x^*) - F(x^k) \geq \left(\frac{dF(x^k)}{dx}, x^* - x^k \right),$$

$$\varphi(x^*) - \varphi(x^k) \geq \left(\frac{d\varphi(x^k)}{dx}, x^* - x^k \right)$$

for $\forall x^* \in X^*, \forall x^k \in X$.

Let us denote $\Delta_F = \min_{x \in X_\varepsilon, x^* \in X^*} [F(x) - F(x^*)]$. Evidently, $\Delta_F > 0$, if $\varepsilon > 0$.

Taking into consideration all properties enumerated above, two constants $C_1 > 0$, $C_2 > 0$ may be chosen, such that $\|x' - x''\| \leq C_1$, $\forall x', x'' \in X$ and $\left\| \frac{dF(x)}{dx} \right\| \leq C_2$, $\forall x \in X$.

If $\varphi(x^k) \leq \tau_k$ and $x^k \in X_\varepsilon$, then $F(x^k) - F(x^*) \geq \Delta_F$, or

$$(11) \quad \left(\frac{dF(x^k)}{dx}, x^k - x^* \right) \geq \Delta_F$$

$$\frac{\left(\frac{dF(x^k)}{dx}, x^k - x^* \right)}{\left\| \frac{dF(x^k)}{dx} \right\| \cdot \|x^k - x^*\|} \geq \frac{\left(\frac{dF(x^k)}{dx}, x^k - x^* \right)}{C_2 C_1} \geq \frac{\Delta_F}{C_1 C_2}$$

Also, if $\varphi(x^k) > \tau_k$ for $x^k \in X_\varepsilon$, then:

$$(12) \quad \left(\frac{d\varphi(x^k)}{dx}, x^k - x^* \right) \geq \tau_k$$

$$\frac{\left(\frac{d\varphi(x^k)}{dx}, x^k - x^* \right)}{\left\| \frac{d\varphi(x^k)}{dx} \right\| \cdot \|x^k - x^*\|} \geq \frac{\left(\frac{d\varphi(x^k)}{dx}, x^k - x^* \right)}{C_2 C_1} \geq \frac{\tau_k}{C_1 C_2}$$

Let's consider some numbers $\delta_F, \delta_\varphi^k$ from intervals $\left(0, \frac{\Delta_F}{C_1 C_2}\right), \left(0, \frac{\tau_k}{C_1 C_2}\right)$ and label $\bar{\delta}_k = \min\{\delta_F, \delta_\varphi^k\}$. As a result the following inequalities may be obtained:

$$(13) \quad \begin{cases} \left(\frac{dF(x^k)}{dx}, x^k - x^* \right) \geq 2\bar{\delta}_k \left\| \frac{dF(x^k)}{dx} \right\| \cdot \|x^k - x^*\|, & \text{if } \varphi(x^k) \leq \tau_k \\ \left(\frac{d\varphi(x^k)}{dx}, x^k - x^* \right) \geq 2\bar{\delta}_k \left\| \frac{d\varphi(x^k)}{dx} \right\| \cdot \|x^k - x^*\|, & \text{if } \varphi(x^k) > \tau_k \end{cases}$$

Particularly, $\delta_F, \delta_\varphi^k$ may be chosen as centres of intervals $\left(0, \frac{\Delta_F}{C_1 C_2}\right), \left(0, \frac{\tau_k}{C_1 C_2}\right)$:

$$(14) \quad \delta_F = \frac{\Delta_F}{2(C_1 C_2)}, \delta_\varphi^k = \frac{\tau_k}{2(C_1 C_2)}$$

The following events are being considered

$A_1^k = \{(\eta^k, x^k - x^*) \geq \bar{\delta}_k \|x^k - x^*\|, \forall x^* \in X^*\}$. Obviously, the opposite event with regards to A_1^k has the following form

$$\overline{A_1^k} = \{\exists x^* \in X^* : (\eta^k, x^k - x^*) < \bar{\delta}_k \|x^k - x^*\|\}.$$

Then $D_1 = \left\{ \bigcup_{k=K_\delta}^\infty \bigcap_{i=k}^\infty A_1^i \right\}$, or, in other words, occurs all $A_1^k (k \geq K_\delta)$, without, perhaps, a finite number. It is obvious that $\overline{D_1} = \left\{ \bigcap_{k=K_\delta}^\infty \bigcup_{i=k}^\infty \overline{A_1^i} \right\}$, or, in other words, an infinite number of events $\overline{A_1^k}$ are produced.

Let us evaluate $P(A_1)$. In order to do this let's represent

$$P(A_1) = P(A_1 \cap D_1) + P(A_1 \cap \overline{D_1})$$

Both terms from last expression will be estimated.

From the realization of event $A_1 \cap D_1$ follows the existence of such a natural number $K_\delta < \infty$ that for all $k \geq K_\delta$ and $\forall x^* \in X^*$ following inequality occurs

$$(15) \quad (\eta^k, x^k - x^*) \geq \bar{\delta}_k \|x^k - x^*\|$$

Taking into consideration (15), for $k \geq K_\delta$ we have the following sequence of relations:

$$\begin{aligned} \|x^{k+1} - x^*\|^2 &\leq \|x^k - \rho_k \eta^k - x^*\|^2 = \\ &= \|x^k - x^*\|^2 - 2\rho_k (x^k - x^*, \eta^k) + \rho_k^2 \|\eta^k\|^2 \leq \\ &\leq \|x^k - x^*\|^2 - 2\rho_k \bar{\delta}_k \|x^k - x^*\| + \rho_k^2 \leq \\ &\leq \|x^k - x^*\|^2 - 2\rho_k \bar{\delta}_k \varepsilon + \rho_k^2 = \\ &= \|x^k - x^*\|^2 - \rho_k (2\bar{\delta}_k \varepsilon - \rho_k) \end{aligned}$$

Because $\rho_k \xrightarrow[k \rightarrow \infty]{} 0$, for some K_φ : $\delta_F > \delta_\varphi^k$ or $\bar{\delta}_k = \delta_\varphi^k$, as soon as $k \geq K_\varphi$. According to (9), (14) for some $K_\varepsilon \geq K_\varphi$: $\rho_k \leq \bar{\delta}_k \varepsilon$, as soon as $k \geq K_\varepsilon$. Evidently, for $k \geq \hat{k} = \max\{K_\delta, K_\varepsilon\}$:

$$\|x^{k+1} - x^*\|^2 \leq \|x^k - x^*\|^2 - \rho_k \bar{\delta}_k \varepsilon,$$

$$\begin{aligned}
\|x^k - x^*\|^2 &\leq \|x^{k-1} - x^*\|^2 - \rho_{k-1} \bar{\delta}_{k-1} \varepsilon \leq \\
&\leq \|x^{k-2} - x^*\|^2 - \varepsilon (\rho_{k-2} \bar{\delta}_{k-2} + \rho_{k-1} \bar{\delta}_{k-1}), \dots \\
\|x^{k+1} - x^*\|^2 &\leq \|x^{\hat{k}} - x^*\|^2 - \varepsilon \sum_{i=\hat{k}}^k \rho_i \bar{\delta}_i \text{ or } \|x^{k+1} - x^*\|^2 \leq \\
&\leq \|x^{\hat{k}} - x^*\|^2 - \varepsilon \sum_{i=\hat{k}}^k \rho_i \delta_\varphi^i
\end{aligned}$$

Due to imposed conditions on τ_k in (9), based on relation (14), we get:

$$(16) \quad \|x^{k+1} - x^*\|^2 \leq \|x^{\hat{k}} - x^*\|^2 - \frac{\varepsilon}{2(C_1 C_2)} \sum_{i=\hat{k}}^k \rho_i \tau_i \rightarrow \infty, \text{ for } k \rightarrow \infty$$

We obtain a contradiction because the norm of any vector, moreover its square value, cannot be negative. Therefore, the realization of event $A_1 \cap D_1$ implies realization of an event, that is practically unrealizable, $F_1 = \left\{ \|x^{k+1} - x^*\|^2 < 0, k \rightarrow \infty \right\}$. That is $P(A_1 \cap D_1) \leq P(F_1) = 0$. It means that $P(A_1) = P(A_1 \cap \overline{D_1})$.

Let us evaluate $P(A_1 \cap \overline{D_1})$. The following event B_1^k is defined:

$B_F^k = \{ \text{at least one time among iterations of the form } j = \overline{k - s_k}, k \text{ is generated every possible value of the discrete random variable } \xi^k \}$.

$B_\varphi^k = \{ \text{at least one time among iterations of the form } j = \overline{k - s_k}, k \text{ is generated every possible value of the discrete random variable } \psi^k \}$.

$$B_1^k = B_F^k \cap B_\varphi^k.$$

Simulation of variables ξ^k and ψ^k is produced in parallel and independently. Thanks to the fact that B_F^k, B_φ^k events are independent, follows that $P(B_1^k) = P(B_F^k) \cdot P(B_\varphi^k)$.

Let us prove that $P(B_1^k) \xrightarrow{k \rightarrow \infty} 1$. Contrary is supposed: $P(B_1^k) \leq p < 1$, for $\forall k$. We have $P(\overline{B_F^k}) = 1 - P(B_F^k)$ and $P(\overline{B_\varphi^k}) = 1 - P(B_\varphi^k)$.

It is absolutely clear that $P\left(\overline{B_F^k}\right) \leq \sum_{i=1}^n \left(1 - P_{\xi,i}^k\right)^{s_k} \xrightarrow[k \rightarrow \infty]{} 0$, $P\left(\overline{B_\varphi^k}\right) \leq \sum_{i=1}^n \left(1 - P_{\psi,i}^k\right)^{s_k} \xrightarrow[k \rightarrow \infty]{} 0$. Indeed, if we label $\underline{P} = \min\left\{\underline{P}_\xi, \underline{P}_\psi\right\}$, the following sequence of relations takes place:

$$\begin{aligned}
 (17) \quad P\left(\overline{B_F^k}\right) &\leq \sum_{i=1}^n \left(1 - P_{\xi,i}^k\right)^{s_k} \leq n \cdot \max_{1 \leq i \leq n} \left(1 - P_{\xi,i}^k\right)^{s_k} \leq \\
 &\leq \left({}^s\sqrt{n} \left(1 - P_\xi\right)\right)^{s_k} \leq \left({}^s\sqrt{n} (1 - \underline{P})\right)^{s_k}, \\
 P\left(\overline{B_\varphi^k}\right) &\leq \sum_{i=1}^n \left(1 - P_{\psi,i}^k\right)^{s_k} \leq n \cdot \max_{1 \leq i \leq n} \left(1 - P_{\psi,i}^k\right)^{s_k} \leq \\
 &\leq \left({}^s\sqrt{n} \left(1 - P_\psi\right)\right)^{s_k} \leq \left({}^s\sqrt{n} (1 - \underline{P})\right)^{s_k}
 \end{aligned}$$

Because $k \rightarrow \infty$, then $s_k \rightarrow \infty$ and ${}^s\sqrt{n} \rightarrow 1 + 0$. For an arbitrary, but fixed value $\tau \in (1 - \underline{P}, 1)$: $\exists K_\tau \in \mathbb{N}$, so that for $k \geq K_\tau$ takes place inequality ${}^s\sqrt{n} (1 - \underline{P}) \leq \tau < 1$. Thus, $\left({}^s\sqrt{n} (1 - \underline{P})\right)^{s_k} \leq \tau^{s_k} \rightarrow 0$. Or $P\left(\overline{B_F^k}\right) \leq \tau^{s_k}$ and $P\left(\overline{B_\varphi^k}\right) \leq \tau^{s_k}$. It means that $P\left(\overline{B_F^k}\right) \rightarrow 0$ and $P\left(\overline{B_\varphi^k}\right) \rightarrow 0$, or, $P\left(B_F^k\right) \rightarrow 1$ and $P\left(B_\varphi^k\right) \rightarrow 1$ for $k \rightarrow \infty$. These considerations conclude to the fact that $P\left(B_1^k\right) \xrightarrow[k \rightarrow \infty]{} 1$.

The realization of event B_1^k means following: "renovation" of all components of the vectors $g_F^{k-s_k}$ and $g_\varphi^{k-s_k}$ is performed during s_k iterations starting from $k - s_k$ till k (inclusively). In other words, movement vector g^k contains as its components all partial derivatives, all evaluated after iteration $k - s_k$.

The realization of event B_1^k and the fact that partial derivatives of functions $F(x)$, $\varphi(x)$ are continuous, conclude to realization of following event:

$$(18) \quad \left\{ \forall \tilde{\varepsilon} > 0 : \left\| \begin{array}{l} g^i - \frac{\partial F(x^k)}{\partial x} \\ g^i - \frac{\partial \varphi(x^k)}{\partial x} \end{array} \right\| \leq \tilde{\varepsilon}, \text{ if } \varphi(x^k) \leq \tau_k, \forall i = \overline{k - s_k, k} \right\}$$

Taking into consideration (18), continuity of dot product and satisfaction of conditions (10), (13), we can draw a conclusion that event A_1^k is realized starting with some $k = \hat{k} = \max\left\{K_\tau, \hat{k}\right\}$.

Thus, $B_1^k \subset A_1^k$. In this case $P(B_1^k) \leq P(A_1^k)$ and, therefore $P(\overline{A_1^k}) \leq P(\overline{B_1^k})$. But, according to (2), (7) and (17) follows:

$$P(\overline{B_1^k}) \leq \tau^{L(k,r)}, \text{ that is } \sum_{k=0}^{\infty} P(\overline{A_1^k}) \leq \sum_{k=0}^{\infty} P(\overline{B_1^k}) \leq \sum_{k=0}^{\infty} \tau^{L(k,r)} < \infty$$

We are in such situation that the conditions of the Borel-Cantelli lemma [3] are satisfied. It means that $P(\overline{D_1}) = 0$. Therefore, $q \leq P(A_1) = P(A_1 \cap \overline{D_1}) \leq P(\overline{D_1}) = 0$. Thus, $q = 0$.

A contradiction has been obtained, because we have supposed that $q > 0$. Thus, it exists a subsequence $\{x^{k_l}\} \subset \{x^k\}_{k \geq 0}$ that almost certainly is contained in $V_X(X^*, \varepsilon)$.

Stage 2. Further will be proved that all elements of sequence $\{x^k\}$, without just a finite number, belong to set $V_X(X^*, 2\varepsilon)$ with probability 1.

Following events are defined:

$$(19) \quad \begin{aligned} A_2 &= \{\exists \{x^{k_l}\} \subset \{x^k\} : \{x^{k_l}\} \subset V_X(X^*, \varepsilon)\} \\ B_2 &= \{\exists \{z^{k_m}\} \subset \{x^k\} : \{z^{k_m}\} \not\subset V_X(X^*, 2\varepsilon)\} \end{aligned}$$

Next, $P(B_2)$ will be appreciated. We will find out that $P(B_2) = P(B_2 \cap A_2)$. Indeed,

$$P(B_2) = P((B_2 \cap A_2) \cup (B_2 \cap \overline{A_2})) = P(B_2 \cap A_2) + P(B_2 \cap \overline{A_2}) = P(B_2 \cap A_2), \text{ because } P(B_2 \cap \overline{A_2}) \leq P(\overline{A_2}) = 0.$$

Further, following event will be considered: $D_2 = A_2 \cap B_2$. Suppose that $P(D_2) > 0$. Realization of event D_2 means that the transfer from $V_X(X^*, \varepsilon)$ to $X \setminus V_X(X^*, 2\varepsilon)$ and vice versa takes place infinitely.

Let us denote by:

- K_1 - the first iteration the event $\{x^{K_1} \in V_X(X^*, \varepsilon)\}$ is produced,
- K_2 - the first iteration the event $\{x^{K_2} \in V_X(X^*, \frac{3}{2}\varepsilon)\}$ is produced,
- K_3 - the first iteration the inequality $\rho_{K_3} \leq 2\varepsilon\bar{\delta}_k$ is satisfied,
- $\bar{K} = \max\{K_1, K_2, K_3\}$.

In case that for some $k \geq \bar{K}$ and $x^k \notin V_X(X^*, \frac{3}{2}\varepsilon)$ is satisfied inequality that defines event A_1^k , then following sequence of inequalities occurs:

$$\|x^{k+1} - x^*\|^2 \leq \|x^k - x^*\|^2 - \rho_k(2\varepsilon\bar{\delta}_k - \rho_k) < \|x^k - x^*\|^2, \text{ because } \|x^k - x^*\| > \varepsilon.$$

That is, as soon as $k \geq \bar{K}$ and $x^k \notin V_X(X^*, \frac{3}{2}\varepsilon)$ means that:

$$(20) \quad \left\| x^{k+1} - x^* \right\| < \left\| x^k - x^* \right\|$$

Since $\rho_k \xrightarrow[k \rightarrow \infty]{} 0$, will appear $K^* \geq \bar{K}$ with property that $x^{K^*} \in V_X(X^*, 2\varepsilon) \setminus V_X(X^*, \frac{3}{2}\varepsilon)$. This will happen certainly. Particularly, for $\rho_k < \frac{\varepsilon}{2}$:

$$\left\| x^{k+1} - x^k \right\| \leq \left\| x^k - \rho_k \eta^k - x^k \right\| \leq \rho_k < \frac{\varepsilon}{2}$$

Therefore, there exists a value k that satisfies $x^k \in V_X(X^*, 2\varepsilon) \setminus V_X(X^*, \frac{3}{2}\varepsilon)$.

According to (20), $\|x^{K^*+1} - x^*\| < \|x^{K^*} - x^*\|$. In the case that $x^{K^*+1} \notin V_X(X^*, \frac{3}{2}\varepsilon)$, then we have $\|x^{K^*+2} - x^*\| < \|x^{K^*+1} - x^*\| < \|x^{K^*} - x^*\|$, and so forth, for all $j \geq 0$ that satisfy $x^{K^*+j} \notin V_X(X^*, \frac{3}{2}\varepsilon)$, takes place

$$(21) \quad \min_{x^* \in X^*} \left\| x^{K^*+j} - x^* \right\| < \min_{x^* \in X^*} \left\| x^{K^*} - x^* \right\| < 2\varepsilon$$

Let us denote $\{x^{k^l}\}_{l \geq 1}$ - sequence of all elements $\{x^k\}$ with the property that $k^l \geq K^*$, $x^{k^l} \in V_X(X^*, 2\varepsilon) \setminus V_X(X^*, \frac{3}{2}\varepsilon)$ and $x^{k^l-1} \in V_X(X^*, \frac{3}{2}\varepsilon)$. Then for $l \geq 1$, $k^l < j < k^{l+1}$ and $x^j \notin V_X(X^*, \frac{3}{2}\varepsilon)$ the following inequality occurs:

$$(22) \quad \min_{x^* \in X^*} \left\| x^j - x^* \right\| < \min_{x^* \in X^*} \left\| x^{k^l} - x^* \right\| < 2\varepsilon$$

Thus, in other words, admitting that for some K elements of type $x^k \notin V_X(X^*, \frac{3}{2}\varepsilon)$, $k < \infty$, $k \geq K$ satisfy inequality from event A_1^k , then event B_2 cannot occur with positive probability. Supposition that D_2 is realized means that beyond layer $V_X(X^*, \frac{3}{2}\varepsilon)$ penetration of layer $X \setminus V_X(X^*, 2\varepsilon)$ takes place only when infinitely is produced event $\overline{A_1^k}$ considered previously. But $P(\overline{D_1}) = 0$. So, the conclusion that can be drawn is that the transfer from layer $V_X(X^*, 2\varepsilon) \setminus V_X(X^*, \frac{3}{2}\varepsilon)$ into layer $X \setminus V_X(X^*, 2\varepsilon)$ occurs only a finite number of times. That is, $P(D_2) = 0$, and implies $P(B_2) = 0$.

Theorem is proved. \square

3. CONCLUSIONS

Elaborated method is especially practical for models where modification of gradients is “relatively slow”. Such models are often encountered in economical, technical problems etc. It represents a significant generalization of methods meant to solve extremum problems. It can be classified as a direct method of optimization and does not use penalty functions or Lagrange function – common toolkit used to solve such kind of problems.

REFERENCES

- [1] Ermolyev Y.M., Gaivoronski A.A., *Stochastic Methods for Solving Minimax Problems*, Kibernetika, Naukova Dumka, Kiev, 1983
- [2] Moiseev N.N, Ivanilov Y.P., Stolyarova E.M., *Methods of optimization*, Nauka, Moskow, 1978
- [3] Shiryaev A.V., *Probability*, Nauka, Moskow, 1980
- [4] Polyak B.T., *A general method for solving extreme problems*, Dokl. Akad. Nauk SSSR. 174, No.1, p.33-36, 1967

MOLDOVA STATE UNIVERSITY, CHISINAU, REPUBLIC OF MOLDOVA
E-mail address: nalab.levap@gmail.com

DIAGNOSIS OF IMAGES OF HUMAN BEHAVIOUR

AMR A. BADR

ABSTRACT. An analysis of single photon emission computerized tomography (SPECT) images was carried out. The purpose is to determine the activities of the different lobes of the brain to classify human behaviour. Pulse-Coupled Neural networks (PCNNs) were used to get the signature of activity in each lobe. This was then used by a k -NN classifier and fuzzy-like rules for diagnosis. Genetic programming was used to evolve a diagnostic grammar which is used for syntactic and semantic diagnosis. Several test cases were used to verify the proposed scheme giving high diagnosis accuracy.

1. INTRODUCTION

SPECT is an acronym for Single Photon Emission Computerized Tomography. It is a sophisticated nuclear medicine study that looks directly at cerebral blood flow and indirectly at brain activity (or metabolism). In this study, a radioactive isotope is bound to a substance that is readily taken up by the cells in the brain. A small amount of this compound is injected into the patient's vein where it runs throughout the blood stream and is taken up by certain receptors sites in the brain. The patient then lies for 14-16 minutes while a SPECT gamma camera rotates slowly around his head. The camera has special crystals that detects where the compound has gone. A supercomputer then reconstructs 3-D images of brain activity levels. The elegant brain snapshots that result offer a sophisticated blood flow/ metabolism brain map. With these maps, physicians are able to identify certain patterns of brain activity that correlate with psychiatric and neurological illnesses. [1]

SPECT studies can be displayed in different kinds. One kind is a 3D 'surface images', looking at the blood flow of the brains cortical surface. These

Received by the editors: March 12, 2009.

2000 *Mathematics Subject Classification.* 68T10.

1998 *CR Categories and Descriptors.* I.5.2 [**PATTERN RECOGNITION**]: Design Methodology – *Pattern analysis*; I.4.9 [**IMAGE PROCESSING AND COMPUTER VISION**]: Applications.

Key words and phrases. SPECT Imaging, Pulse-Coupled Neural Networks, k -NN classifier, Genetic Programming, Grammar.

images are helpful for picking up cortical surface area of good activity as well as low-active areas. They are helpful to look at strokes, brain trauma, effect of drug abuse, etc. A normal 3D surface scan shows good, full, symmetrical activity across the brain's cortical surface. The other kind is a 3D 'active brain image'. These images are helpful for picking up areas of HIGH activity, as seen in active seizures, obsessive compulsive disorder, anxiety problems, certain forms of depression, etc. A normal 3D active scan shows increased activity in the back of the brain (cerebellum or occipital cortex) and average activity everywhere else. Physicians are usually altered that smoothing is wrong in one of three ways:

- a. HIGH activity in certain area
- b. LOW activity in certain area
- c. Asymmetrical area of activity which ought to be symmetrical.

In this paper, intelligent techniques for classification, diagnosis, imaging and understanding were used for the purpose of diagnosis and analysis of images of human behaviour. Similar decision support systems for medical image diagnosis can be found in [2, 4, 5, 6, 10, 15, 16].

2. A PROLOGUE TO SPECT IMAGING

2.1. **Anatomy of Brain.** The human brain is anatomically divided into four main lobes namely: the Frontal lobe, the Parietal lobe, the Occipital lobe, and the Temporal lobe and the cerebellum (Fig.1). The SPECT brain radiologist looks for levels of activity in these lobes mainly. [3]

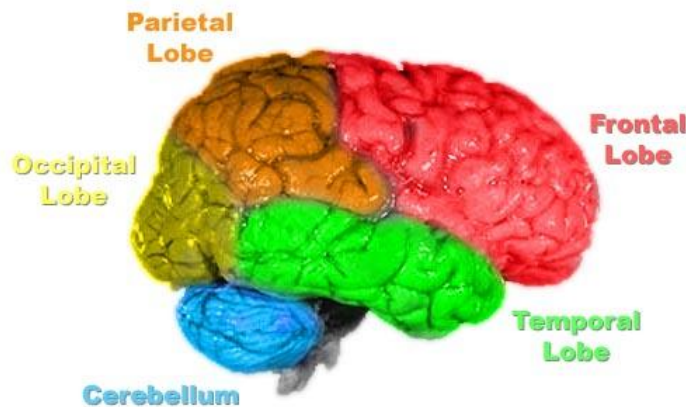


FIGURE 1. Brain Anatomy

2.2. Brain SPECT Views. There are three main views for SPECT imaging: [3]

a. Horizontal View (transaxial)

The brain is viewed in horizontal slices, cut from top to bottom. It is as if you are looking down from a bird's eye view. Fig 2(a)

b. Coronal View (front on view)

The brain is viewed in vertical slices, cut from front to back. It is as if you are looking face on or front on to the brain. Fig 2(b)

c. Sagittal View (side to side)

The brain is viewed in vertical slices, cut from side to side. It is as if you are looking at brain from the side. Fig 2(c)

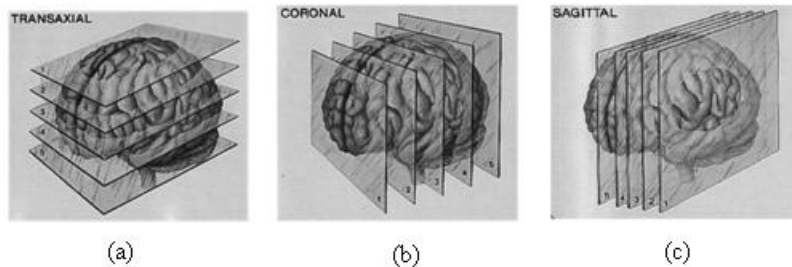


FIGURE 2. (a) horizontal view (Transaxial). (b) Coronal View (Front on View). (c) Sagittal View (Side to Side)

2.3. Normal 3D Brain Images. As stated above, there are two main brain SPECT views: the surface view and the active view. For a normal brain, the surface view shows smooth distribution of dye in the top-down, front-on, underside and side surface views. Fig 3

The active view shows the regions of activity in anatomical brain lobe with a 3D reconstruction of the images. Fig 4

3. PROPOSED SPECT DIAGNOSIS SCHEMES

Two schemes are adopted in this paper. The first uses fuzzy rules to correctly classify the symptom. The second evolves a diagnostic grammar.

The first scheme goes through several phases of image processing and pattern recognition to correctly diagnosis the disorder.

First, the input image whether surface or active view is resized to a particular size. Then the 3D resized coloured image is converted into a 2D grayscale

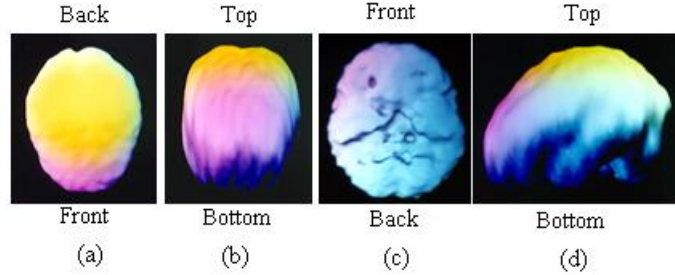


FIGURE 3. (a) Top - Down Surface View. (b) Front - On Surface Views. (c) Underside Surface View. (d) Side Surface View.

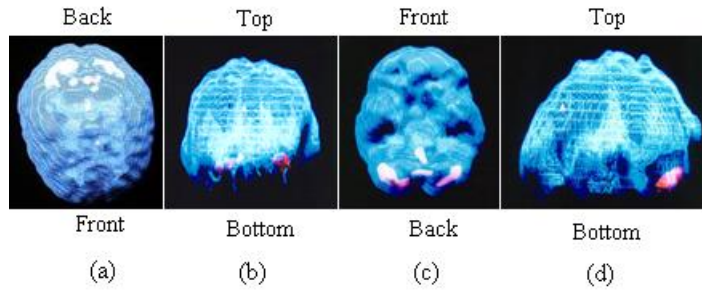


FIGURE 4. (a) Top - down active view. (b) Front - on Active View. (c) Underside Active View. (d) Side Active View

image. The features of the image are determined using the freeman chain code to clarify the brain orientation (horizontal, coronal, or Sagittal). These features are passed to a k -NN classifier [13] for exact determination of the orientation. A 'mask' image is then 'ANDed' to the original image to separate the particular brain lobe. Next, a matrix for each lobe's image and run through a pulse-coupled Neural Network [11] for 100 iterations, so as to get a unique 'signature' that is then used by the k -NN classifier again to identify the state of the brain lobe, whether low, high or normal activity. A set of diagnostic fuzzy-like rules are induced for classification of the brain disorder. To summarize the scheme:

1. Surface or active view input image is resized.
2. Convert 3D color image to 2D grayscale image.
3. Apply freeman- chain - code for feature extraction.
4. Classify features using k -NN classifier.
5. 'AND' a mask image to input image for lobes' separation.
6. Apply PCNN to get signature of each lobe.
7. Classify signature using k -NN again (Low, High, Normal).

8. Build fuzzy - like rules for disorder diagnosis.
9. If image disorder set is finished then stop else goto 1.

The second scheme adopted genetic programming (GP) [8, 9, 12] to evolve an image-understanding diagnostic grammar [12, 14]. It goes through the same steps as above, but replaces step 8 by the application of GP as will be explained.

3.1. Feature Extraction of SPECT Views.

3.1.1. *Freeman Chain Code.* To determine the brain orientation whether top-down, side, underside or front-on view, contour information is needed. This is performed by Freeman Chain Code. It is an ordered sequence of n links C_k , where C_k is a vector connecting neighbouring contour pixels. The directions of C_k are coded with integer values $k = 0, 1, \dots, k-1$ in a contour clockwise sense. A simple example is depicted in Fig 5. Fig 5(a) shows the directions of the eight connected chain code ($K = 8$). Fig 5 (b) is a sample object with chain code presentation in Fig 5 (c).

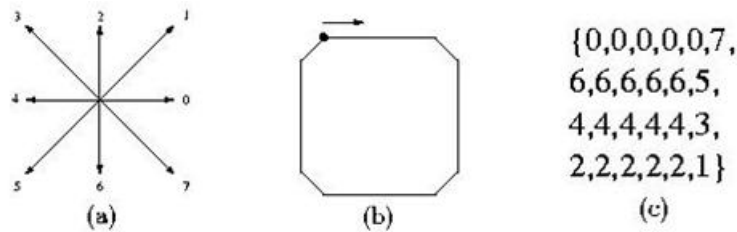


FIGURE 5. (a) The directions of the eight Connected chain code ($K=8$). (b) A sample object, a square. (c) Chain code presentation of the square

The pseudocode for the Freeman Chain Code is as follows:

1. For $i = \text{Starting_row_index} + 1$ till image's matrix rows no.
2. For $j = \text{Starting_column_index} + 1$ till image's matrix columns no.
3. Get the coordinates (i, j) of the first pixel that is not black.
4. End For
5. End For
6. Get the contour from the resulting coordinates.
7. Initialize 8 counters for 8 directions by 0.
8. For $k = 1$ till the number of rows of the contour
9. Get the direction of the pixel by comparing its coordinates to the coordinates of its following pixel
10. Increment the contour of the resulting direction by 1
11. End For

12. Normalize contours
13. Return a vector of normalized contours.

3.1.2. *k-NN Classifier*. The *k*-NN (K-Nearest Neighbour Algorithm) is a method for classifying objects based on closest training examples in the feature space [13]. In our case, the feature space is 8-dimensional since the number of features from chain code is 8. The training examples are vectors in multi-dimensional feature space. The space is partitioned into regions by locations and labels of the training samples. The algorithm pseudocode is as follows:

1. Load the training samples in vectors
2. Get feature vector of image
3. MinDist = 100 000
4. For i = 1 till number of rows of sample vector
5. For j = 1 till number of columns of sample vector
6. Get a training feature vector
7. End For
8. Get the Euclidean distance between image's feature vector and training vector
9. if distance < MinDist
10. MinDist = distance
11. Class = i
12. End For
13. According to the ordering of training samples, class will indicate corresponding view

3.2. **Pulse-Coupled Neural Network (PCNN)**. The Eckhorn model of PCNNs is used. It is based on the cat visual cortex. The model can be abstracted into the following five time dependant equations: [11]

1. $F_{xy}(t) = F_{xy}(t - \Delta t)e^{-\alpha_F \Delta t} + S_{xy} + \sum_{x'y'} W_{x'y' \rightarrow xy}^F Y_{x'y'}(t - \Delta t)$
2. $L_{xy}(t) = L_{xy}(t - \Delta t)e^{-\alpha_L} + \sum_{x'y'} W_{x'y' \rightarrow xy}^L Y_{x'y'}(t - \Delta t)$
3. $U_{xy}(t) = F_{xy}(t) (1 + \beta L_{xy}(t))$
4. $Y_{xy}(t) = \begin{cases} 1 & \text{if } U_{xy}(t) > \theta_{xy}(t - \Delta t) \\ 0 & \text{otherwise} \end{cases}$
5. $\theta_{xy}(t) = \theta_{xy}(t - \Delta t)e^{-\alpha_\theta} + W^\theta Y_{xy}(t)$

The F , L , U , Y and θ are the activations of the five different neuron - types in the model. The α_F , α_L and α_θ are the decaying constants of the respective neuron - types; the W^F , W^L and W^θ are the feedback weights from the output Y to the feeding neurons (F), linking neurons (L) and dynamic threshold neurons (θ) respectively.

For the feeding and linking neurons the weights are four-dimensional matrices and finally the S is the stimuli or image input. The signal-flow between the processing elements (neurons) is illustrated in Fig 6.

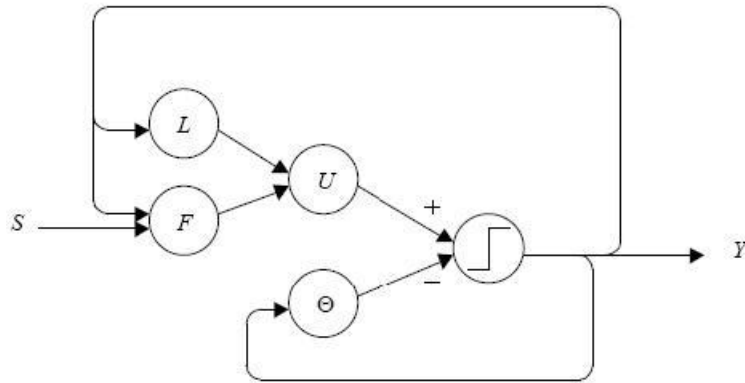


FIGURE 6. Pulse-Coupled Neural Network

Normally, one feeding neuron per pixel is used. Five matrices of the same dimensions of the image store intermediate results for discrete time steps $t = 0, 1, \dots, K$. the output of the PCNN is a series of binary images. When an output neuron goes high or 'fires', its dynamic threshold θ is significantly increased, then decays exponentially and after a couple of iterations the neuron can fire again. This pulsating behaviour is characteristic of the PCNN. The signature of a lobe image is a series of 100 elements each is the number of ones in binary output images for K iterations.

3.3. Genetic Programming Evolution of Diagnostic Grammar. The problem of grammar induction involves recognizing patterns existing in a sequence of data.

In summary, the genetic programming paradigm breeds grammars to solve problems by executing the following three steps:

1. Generate an initial population of random compositions of grammars.
2. Iteratively perform the following substeps until termination criterion has been satisfied.

- (a) Execute each grammar in the population and assign it a fitness value
 - (b) Create a new population of grammars by applying the following two primary operations:
 - (i) Copy existing grammars to new population
 - (ii) Create new grammars by genetically recombining randomly chosen parts of two existing grammars.
3. The best grammar that appeared in any generation (i.e. the best-so-far individual) is designated as the result of genetic programming.

4. ANALYSIS AND RESULTS

4.1. **Fuzzy Rules for Diagnosis.** The following fuzzy rules are deduced and used for diagnostic purpose:

1. IF Frontal _ lobe _ activity = H AND
 Parietal _ lobe _ activity = N AND
 Temporal _ lobe _ activity = N AND
 Occipital _ lobe _ activity = N AND THEN Diagnosis = Stroke.
2. IF Frontal _ lobe _ activity = L AND
 Parietal _ lobe _ activity = L AND
 Temporal _ lobe _ activity = L AND
 Occipital _ lobe _ activity = N AND THEN Diagnosis = Trauma_Induced_ADD.
3. IF Frontal _ lobe _ activity = H AND
 Parietal _ lobe _ activity = N AND
 Temporal _ lobe _ activity = H AND
 Occipital _ lobe _ activity = N AND
 Gyrus activity = H THEN Diagnosis = Ring_of_Fire_ADD.
4. IF Frontal_lobe_activity = L AND
 Parietal_lobe_activity = N AND
 Temporal_lobe_activity = L AND
 Occipital_lobe_activity = N AND THEN Diagnosis = Marijuana_Abuse.
5. IF Frontal _ lobe _ activity = H AND
 Parietal _ lobe _ activity = H AND
 Temporal _ lobe _ activity = H AND

Occipital_lobe_activity = H AND THEN Diagnosis = Heroin_Abuse.

6. IF Frontal_lobe_activity = L AND
 Parietal_lobe_activity = L AND
 Temporal_lobe_activity = L AND
 Occipital_lobe_activity = L AND THEN Diagnosis = Alco-
 hol_Abuse.
7. IF Frontal_lobe_activity = L AND
 Parietal_lobe_activity = L AND
 Temporal_lobe_activity = N AND
 Occipital_lobe_activity = L AND THEN Diagnosis = Schizophre-
 nia.
8. IF Frontal_lobe_activity = N AND
 Parietal_lobe_activity = N AND
 Temporal_lobe_activity = H AND
 Occipital_lobe_activity = N AND THEN Diagnosis = Anxiety.

4.2. **Grammars for Diagnosis.** The following grammars were evolved using Genetic Programming:

1. SYMPTOM \rightarrow STROKE
 STROKE \rightarrow FRONTAL_LOBE_ACTIVITY
 and PARITAL_LOBE_ACTIVITY
 and TEMPORAL_LOBE_ACTIVITY
 and OCCIPITAL_LOBE_ACTIVITY

 FRONTAL_LOBE_ACTIVITY \rightarrow H
 PARITAL_LOBE_ACTIVITY \rightarrow N
 TEMPORAL_LOBE_ACTIVITY \rightarrow N
 OCCIPITAL_LOBE_ACTIVITY \rightarrow N
2. SYMPTOM \rightarrow TRAUMA_INDUCED_ADD
 TRAUMA_INDUCED_ADD \rightarrow FRONTAL_LOBE_ACTIVITY
 and PARITAL_LOBE_ACTIVITY
 and TEMPORAL_LOBE_ACTIVITY
 and OCCIPITAL_LOBE_ACTIVITY

 FRONTAL_LOBE_ACTIVITY \rightarrow L
 PARITAL_LOBE_ACTIVITY \rightarrow L

TEMPORAL_LOBE_ACTIVITY → L
 OCCIPITAL_LOBE_ACTIVITY → N

3. SYMPTOM → RING_OF_FIRE_ADD
 RING_OF_FIRE_ADD → FRONTAL_LOBE_ACTIVITY
 and PARITAL_LOBE_ACTIVITY
 and TEMPORAL_LOBE_ACTIVITY
 and OCCIPITAL_LOBE_ACTIVITY

FRONTAL_LOBE_ACTIVITY → H
 PARITAL_LOBE_ACTIVITY → H
 TEMPORAL_LOBE_ACTIVITY → H
 OCCIPITAL_LOBE_ACTIVITY → N
 GYRUS_ACTIVITY → H

4. SYMPTOM → MARIJUANA_ABUSE
 MARIJUANA_ABUSE → FRONTAL_LOBE_ACTIVITY
 and PARITAL_LOBE_ACTIVITY
 and TEMPORAL_LOBE_ACTIVITY
 and OCCIPITAL_LOBE_ACTIVITY

FRONTAL_LOBE_ACTIVITY → L
 PARITAL_LOBE_ACTIVITY → L
 TEMPORAL_LOBE_ACTIVITY → N
 OCCIPITAL_LOBE_ACTIVITY → N

5. SYMPTOM → HEROIN_ABUSE
 HEROIN_ABUSE → FRONTAL_LOBE_ACTIVITY
 and PARITAL_LOBE_ACTIVITY
 and TEMPORAL_LOBE_ACTIVITY
 and OCCIPITAL_LOBE_ACTIVITY

FRONTAL_LOBE_ACTIVITY → H
 PARITAL_LOBE_ACTIVITY → H
 TEMPORAL_LOBE_ACTIVITY → H
 OCCIPITAL_LOBE_ACTIVITY → H

6. SYMPTOM → ALCOHOL_ABUSE
 ALCOHOL_ABUSE → FRONTAL_LOBE_ACTIVITY
 and PARITAL_LOBE_ACTIVITY
 and TEMPORAL_LOBE_ACTIVITY

and OCCIPITAL_LOBE_ACTIVITY

FRONTAL_LOBE_ACTIVITY → L
 PARITAL_LOBE_ACTIVITY → L
 TEMPORAL_LOBE_ACTIVITY → L
 OCCIPITAL_LOBE_ACTIVITY → L

7. SYMPTOM → SCHIZOPHRENIA

SCHIZOPHRENIA → FRONTAL_LOBE_ACTIVITY
 and PARITAL_LOBE_ACTIVITY
 and TEMPORAL_LOBE_ACTIVITY
 and OCCIPITAL_LOBE_ACTIVITY

FRONTAL_LOBE_ACTIVITY → L
 PARITAL_LOBE_ACTIVITY → L
 TEMPORAL_LOBE_ACTIVITY → N
 OCCIPITAL_LOBE_ACTIVITY → L

8. SYMPTOM → ANXIETY

ANXIETY → FRONTAL_LOBE_ACTIVITY
 and PARITAL_LOBE_ACTIVITY
 and TEMPORAL_LOBE_ACTIVITY
 and OCCIPITAL_LOBE_ACTIVITY

FRONTAL_LOBE_ACTIVITY → N
 PARITAL_LOBE_ACTIVITY → N
 TEMPORAL_LOBE_ACTIVITY → H
 OCCIPITAL_LOBE_ACTIVITY → N

4.3. Test Cases.

4.3.1. Test case 1.

1. Inputs SPECT image in Fig 7(a)
2. Processing
 - (a) Converting the 3D image into gray scale 2D image Fig 7 (b)
 - (b) Determining the brains orientation → Result: Top-Down
 - (c) According to given view and orientation, extract the appropriate lobes of the brain
 Result: Extracted lobes in fig 7 (c), (d), (e)
 - (d) Get the time signature for each lobe

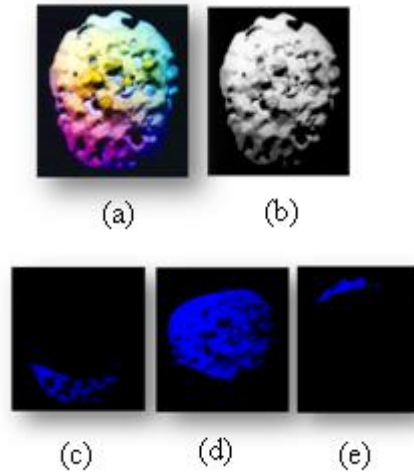


FIGURE 7. (a) SPECT image (View Type: Surface). (b) Gray Scale Image. (c) Occipital lobe. (d) Parietal lobe (e) Frontal lobe

Result for Occipital lobe:

130000 0 1315 1471 165 2308 617 1867 943 2028 851 2263 880
 2206 889 2098 1200 1839 1312 1666 1702 1609 1999 1650 1738
 1690 1953 1637 1937 1529 2079 1890 2364 2250 2507 2472 2584
 2621 2947 2620 2757 3034 3161 3150 3106 3222 3133 3161 106275
 4461 1785 1740 1851 1575 1851 1596 1900 1857 1683 1773 1743
 1908 2056 2013 2145 2218 2205 2442 2297 2352 2449 2481 2539
 2514 2674 2502 2631 2623 2837 2904 2745 2913 2904 2866 2888
 2879 2952 2883 2962 3222 3125 3195 3353 3149 3258 3184 92807
 4452 2182 2273

Result for Parietal lobe:

130000 0 11398 21342 6811 22759 14532 19795 15426 24249 12567
 26529 12643 26270 17750 20624 20820 21350 19881 19640 22214
 22081 21409 21695 18515 24196 24141 18772 22918 20247 24311
 22484 22349 23206 24939 21533 25598 22042 26207 24252 21279
 25309 27040 21616 26374 24257 22876 27154 61187 23650 24696
 20439 23147 20789 21632 23095 22984 22115 23057 22514 21124
 21429 23608 23769 22153 23133 24627 23510 22108 23333 23341
 24636 23338 22436 26678 23127 22314 23732 24311 24486 24017
 24280 24082 24234 23143 23242 24736 24755 22467 24881 24527
 24928 26135 22689 24459 25034 42342 24864 24090 23546

Result for Frontal lobe:

130000 0 56 3315 2533 1491 3394 2098 2677 3402 2867 3591 3663
 3809 3349 2735 3137 4214 2345 4141 2765 3430 4155 3669 3091
 4128 3562 3621 3147 3923 3920 4276 4584 4348 4692 5112 4740
 5137 5971 4698 4489 5835 5587 5212 5344 5787 5209 5681 89031
 6127 3926 3779 3473 3604 3340 3263 4319 4114 3580 3727 3072
 4079 4792 3800 3869 4459 4695 4991 4317 4446 4147 5178 4490
 5184 5321 4288 4422 4453 5320 5005 4968 5208 5065 5355 4718
 5130 5596 5037 4721 5777 5257 5859 6132 5409 5127 5267 73489
 6173 4694 4282

- (e) Compare time signatures' values of existing data to determine their abnormality level

Result:

Occipital : Low
 Parietal : Low
 Frontal : Low

- (f) Apply the rules that identify the disorders.

Result: Schizophrenia

3. The result of the test case

Success in detecting the abnormalities then the disorder accordingly

4. Parse Tree of test case 1

As shown in Fig (8)

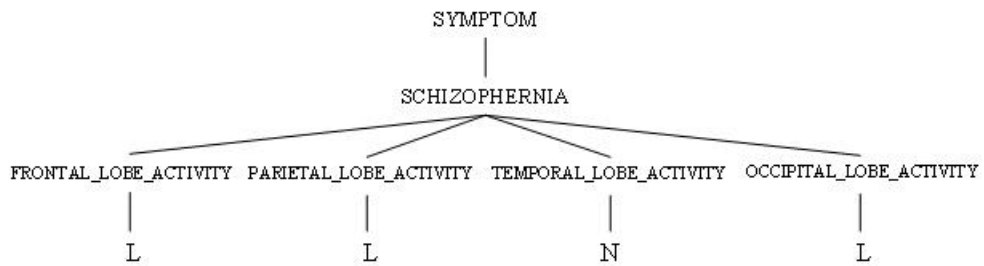


FIGURE 8. Parse Tree of Test Case 1

4.3.2. Test case 2.

1. Inputs

SPECT image 1 in Fig 9(a)

SPECT image 2 in Fig 9(b)

2. Processing

- (a) Converting the 3D image into gray scale 2D image Fig 9 (c), (d)
- (b) Determining the brain's orientation
Result: For the first image fig 9 (c): Side
For the second image fig 9 (d): Top-Down

For the first image 9(a)

According to given view and orientation, extract the appropriate lobes of the brain.

Result: Extracted lobes in Fig 9 (e), (f), (g), (h)

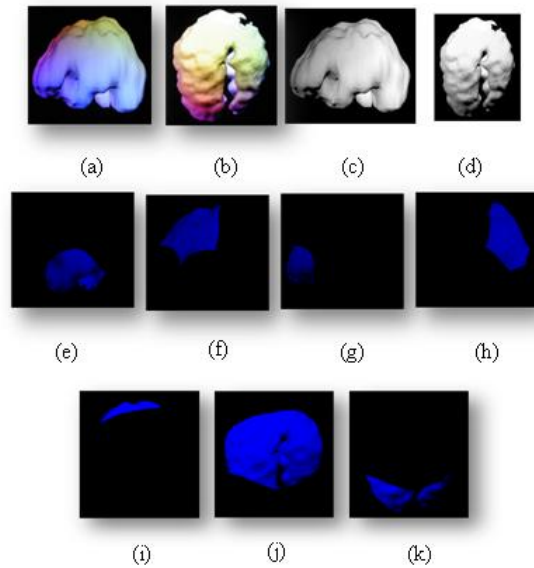


FIGURE 9. (a) first input image. (b) Second Input Image. (c) Grayscale image for first input image. (d) Grayscale image for second input image. (e) Temporal lobe for first image. (f) Parietal lobe for first image. (g) Frontal lobe for first image. (h) Occipital Lobe for first image. (i) Frontal lobe for second image. (j) Parietal Lobe for second image. (k) Occipital Lobe for second image.

- (c) Compare time signatures' values of existing data to determine their abnormality level

Result:

Temporal :Low

Occipital :Normal

Parietal :Normal
 Frontal :Low

For the second image 9(b)

According to given view and orientation, extract the appropriate lobes of the brain.

Result: Extracted lobes in fig 9 (i), (j), (k)

- (d) Compare time signatures' values of existing data to determine their abnormality level

Result:

Occipital: Normal
 Parietal : Low
 Frontal : Low

- (e) Apply the rules that identify the disorders.

*Result:*Trauma Induced ADD

3. The result of the test case

Success in detecting the abnormalities then the disorder accordingly

4. Parse Tree of test case 2

As shown in Fig (10)

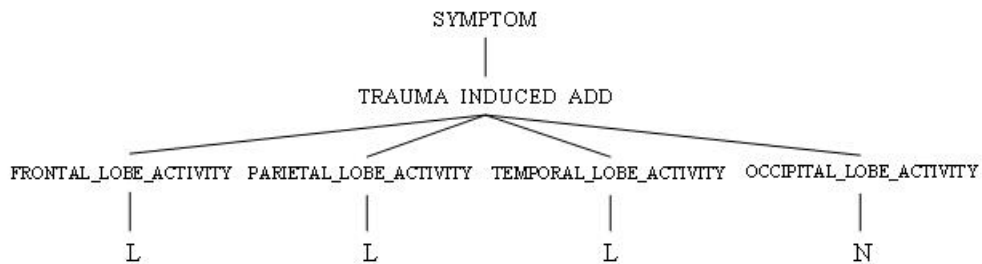


FIGURE 10. Parse Tree of Test Case 2

4.3.3. Test case 3.

1. Inputs SPECT image in Fig 11 (a)

2. Processing

(a) Converting the 3D image into gray scale 2D image Fig 11 (b)

(b) Determining the brains orientation → Result: Underside

- (c) According to given view and orientation, extract the appropriate lobes of the brain

Result: Extracted lobes in fig 11 (c), (d), (e)

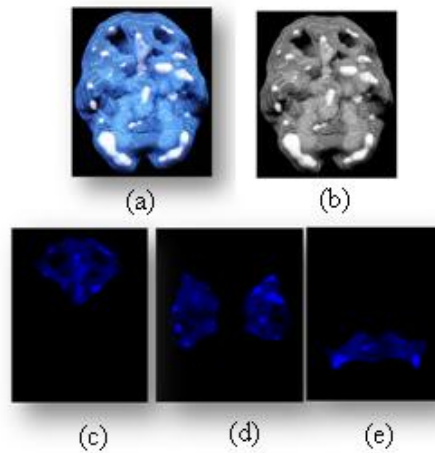


FIGURE 11. (a) SPECT Image (View Type: Active). (b)Grayscale Image. (c)Frontal lobe. (d)Temporal lobe. (e)Occipital lobe

- (d) Compare time signatures' values of existing data to determine their abnormality level

Result:

Temporal : High

Frontal : Normal

Occipital : Normal

- (e) Apply the rules that identify the disorders.

Result: Anxiety

3. The result of the test case

Success in detecting the abnormalities then the disorder accordingly

4. Parse Tree of test case 3
As shown in Fig (12)

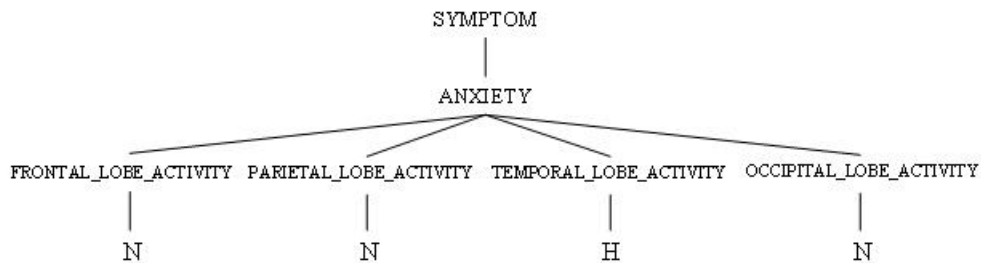


FIGURE 12. Parse Tree of Test Case 3

5. CONCLUSION

A classification and analysis scheme for human behaviour images is proposed. The analysis is dependant on the determination of activities in different brain lobes fuzzy-like rules and grammars were used for diagnosis. Pulse-Coupled Neural Networks were adopted to generate signatures of lobes' activities. Genetic Programming was used to evolve diagnostic grammars. The proposed scheme proved to be accurate and efficient in diagnosis human behaviour.

REFERENCES

- [1] Daniel G. Amen, *Images of human behaviour: A Brain SPECT Atlas*, Brain Imaging Division, Amen Clinics, Inc. for Behavioural Medicine, 2008.
- [2] Ayache N, *Medical image analysis: A Challenge for Computer Vision Research*. In: proc 14th Int Conf Pattern Recognition 2: 1255-1256, 1998.
- [3] J. Douglas Bremner, *Brain Imaging Handbook*, W. W. Norton & Company, 2005.
- [4] Chabat F, Hansel DM, and Yang GZ: *CT:Lung Image Classification with Correction for Perfusion Gradient*. In: Proc 7th Int Conf image Processing and its Applications (IPA'99), pp. 402-407, 1999.
- [5] Doi K, Giger ML, Nishikawa RM, Hoffmann KR, and MacMahon H: *Potential Usefulness of Digital Imaging in Clinical Diagnostic Radiology: Computer-aided diagnosis*, J digital Imag 8: 2-7, 1995.

- [6] Katsuragawa S, Doi K, MacMahon H, Monnier-Cholly L, Ishida T, and Kobayashi T: *Classification of Normal and Abnormal Lungs with Interstitial Diseases by Rule-Based Method and Artificial Neural Networks*. J digital Imag 108-114, 1997.
- [7] Sunil K. Kopparapu, Uday B. Desai, *Bayesian Approach to Image Interpretation*, Kluwer Academic Publishers, 2001.
- [8] John R. Koza, *Genetic Programming: on The Programming of Computers by Means of Natural Selection*, MIT Press, 1992.
- [9] John R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, 1994.
- [10] Krupinski EA, Nodine CF, and Kundel HL: *A Perceptually Based Method for Enhancing Pulmonary Nodules Recognition*. Investigative Radiol 28: 289-294, 1993.
- [11] T. Lindblad, J. M. Kinser, *Image Processing Using Pulse-Coupled Neural Networks*, Springer, 2005.
- [12] Michael O'Neill, Conor Ryan, *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*, Kluwer Academic Publisher, 2003.
- [13] Robert Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches*, Wiley, 1992.
- [14] Ryszard Tadeusiewicz, Marek R. Ogiela, *Medical Image Understanding Technology: Artificial Intelligence and Soft Computing for Image Understanding*, Studies in Fuzziness and Soft Computing, Springer, 2004.
- [15] Taylor P, Fox J, and Todd-Pokropek A: *Computer aids for Decision-Making in Radiology* in: proc IEEE Colloquium on Artificial Intelligence in Medicine, pp. 511-513, 1996.
- [16] Tu S, Shanar Y, Dawes J, Winkles J, Puerta A, and Musen N: *A Problem-Solving Model for Episodic Skeletal-Plan Refinement*. Knowledge Acquisition 4: 197-216, 1992.

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF COMPUTERS AND INFORMATION,
CAIRO UNIVERSITY, EGYPT

E-mail address: ruaab@rusys.eg.net

A HYBRID TECHNIQUE FOR AUTOMATIC MRI BRAIN IMAGES CLASSIFICATION

EL-SAYED A. EL-DAHSHAN, ABDEL-BADEEH M. SALEM, AND TAMER H. YOUNIS

ABSTRACT. This paper presents two hybrid techniques for the classification of the magnetic resonance human brain images. The proposed hybrid technique consists of three stages, namely, feature extraction, dimensionality reduction, and classification. In the first stage, we have obtained the features related with MRI images using discrete wavelet transformation (DWT). In the second stage, the features of magnetic resonance images (MRI) have been reduced using principles component analysis (PCA) to the more essential features. In the classification stage, two classifiers based on supervised machine learning have been developed. The first classifier based on feed forward back-propagation artificial neural network (FP-ANN) and the second classifier based on k-nearest neighbor (k-NN). The classifiers have been used to classify subjects as normal or abnormal MRI human images. A classification with a success of 95.6% and 98.6% has been obtained by the two proposed classifiers FP-ANN and k-NN respectively. This result shows that the proposed hybrid techniques are robust and effective compared with other recently work.

1. INTRODUCTION

Magnetic resonance imaging (MRI) is often the medical imaging method of choice when soft tissue delineation is necessary. This is especially true for any attempt to classify brain tissues [1]. The most important advantage of MR imaging is that it is non-invasive technique [2]. The use of computer technology in medical decision support is now widespread and pervasive across a wide range of medical area, such as cancer research, gastroenterology, hart diseases, brain tumors etc. [3, 4]. Fully automatic normal and diseased human brain

Received by the editors: March 1, 2009.

2000 *Mathematics Subject Classification.* 68T10, 62H30.

1998 *CR Categories and Descriptors.* I.5.2 [**Pattern Recognition**]: Design Methodology – *Pattern analysis*; I.4.9 [**Image Processing and Computer Vision**]: Applications – *General*.

Key words and phrases. MRI human brain Images; Wavelet Transformation (WT); Principle Components Analysis (PCA); Feedforawd-Backpropagation Neural Network (FP-ANN); k-Nearest Neighbors, Classification.

classification from magnetic resonance images (MRI) is of great importance for research and clinical studies.

Recent work [2, 5] has shown that classification of human brain in magnetic resonance (MR) images is possible via supervised techniques such as artificial neural networks and support vector machine (SVM) [2], and unsupervised classification techniques unsupervised such as self organization map (SOM) [2] and fuzzy c -means combined with feature extraction techniques [5]. Other supervised classification techniques, such as k -nearest neighbors (k -NN) also group pixels based on their similarities in each feature image [1, 6, 7, 8] can be used to classify the normal/pathological T2-weighted MRI images. We used supervised machine learning algorithms (ANN and k -NN) to obtain the classification of images under two categories, either normal or abnormal.

Wavelet transform is an effective tool for feature extraction, because they allow analysis of images at various levels of resolution. This technique requires large storage and is computationally more expensive [4]. Hence an alternative method for dimension reduction scheme is used. In order to reduce the feature vector dimension and increase the discriminative power, the principal component analysis (PCA) has been used. Principal component analysis is appealing since it effectively reduces the dimensionality of the data and therefore reduces the computational cost of analyzing new data. To perform the classification on the input data the k -NN and artificial network classifier have been used.

The contribution of this paper is the integration of an efficient feature extraction tool and a robust classifier to perform a more robust and accurate automated MRI normal/abnormal brain images classification. Also, this paper focuses on a comparison of our results with a similar study (by others) using supervised and unsupervised methods [2, 5].

This paper is organized as follows: A short description on the input dataset of images is presented in Sections 2 and methods for feature extraction and reduction as well for classification are presented in Sections 3. Section 4 contains results and discussion while conclusions and future work are presented in Section 5.

2. THE PROPOSED HYBRID TECHNIQUES

The proposed hybrid techniques based on the following techniques, discrete wavelet transforms DWT, the principle components analysis PCA, FP-ANN, and k -NN. It consists of three stages: (1) feature extraction stage, (2) feature reduction stage, and (3) classification stage. The proposed hybrid technique for MRI image classification is illustrated in Fig. 1. In the following sections, a review of basic fundamental of k -NN, principal component analysis, and wavelet decomposition are introduced.

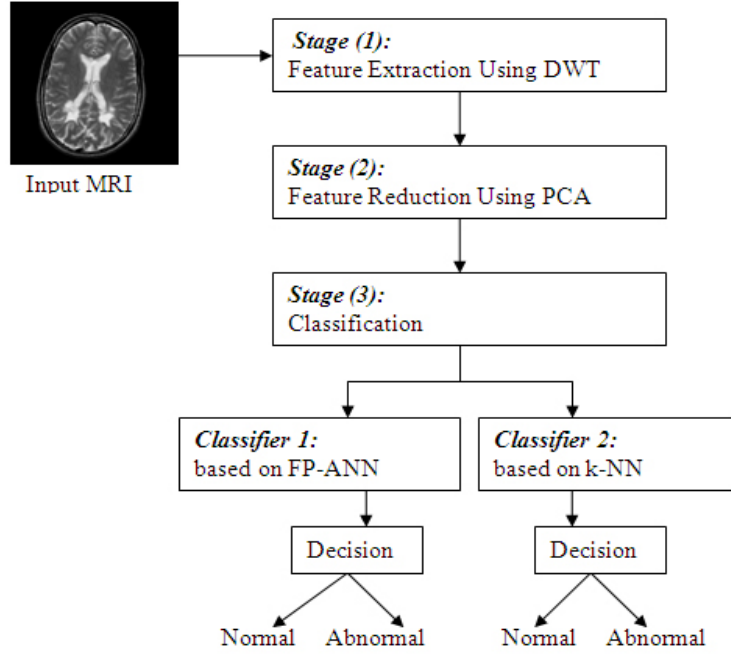


FIGURE 1. The methodology of the proposed technique

2.1. Feature extraction scheme using DWT. The proposed system uses the Discrete Wavelet Transform (DWT) coefficients as feature vector. The wavelet is a powerful mathematical tool for feature extraction, and has been used to extract the wavelet coefficient from MR images. Wavelets are localized basis functions, which are scaled and shifted versions of some fixed mother wavelets. The main advantage of wavelets is that they provide localized frequency information about a function of a signal, which is particularly beneficial for classification [9]. A review of basic fundamental of Wavelet Decomposition is introduced as follows:

The continuous wavelet transform of a signal $x(t)$, square-integrable function, relative to a real-valued wavelet, $\Psi(t)$ is defined as [10]:

$$(1) \quad W_{\Psi}(a, b) = \int_{-\infty}^{\infty} f(x) * \Psi_{a,b}(t) dx$$

where

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \Psi((t-a)/b)$$

and the wavelet $\Psi_{a,b}$ is computed from the mother wavelet Ψ by translation and dilation, wavelet, a the dilation factor and b the translation parameter (both being real positive numbers). Under some mild assumptions, the mother wavelet Ψ satisfies the constraint of having zero mean [11, 12]. The eq. (1) can be discretized by restraining a and b to a discrete lattice ($a = 2^b, a \in R_+, b \in R$) to give the discrete wavelet transform (DWT).

The discrete wavelet transform (DWT) is a linear transformation that operates on a data vector whose length is an integer power of two, transforming it into a numerically different vector of the same length. It is a tool that separates data into different frequency components, and then studies each component with resolution matched to its scale. DWT can be expressed as [13].

$$(2) \quad DWT_{x(n)} = \begin{cases} d_{j,k} = \sum(x(n)h * j(n - 2jk)) \\ d_{j,k} = \sum(x(n)g * j(n - 2jk)) \end{cases}$$

The coefficients $d_{j,k}$ refer to the detail components in signal $x(n)$ and correspond to the wavelet function, whereas $a_{j,k}$ refer to the approximation components in the signal. The functions $h(n)$ and $g(n)$ in the equation represent the coefficients of the high-pass and low-pass filters, respectively, whilst parameters j and k refer to wavelet scale and translation factors. The main feature of DWT is multiscale representation of function. By using the wavelets, given function can be analyzed at various levels of resolution [14]. Fig. 2 illustrates DWT schematically. The original image is process along the x and y direction by $h(n)$ and $g(n)$ filters which, is the row representation of the original image. As a result of this transform there are 4 subband (LL, LH, HH, HL) images at each scale. (Fig.2). Subband image LL is used only for DWT calculation at the next scale. To compute the wavelet features in the first stage, the wavelet coefficients are calculated for the LL subband using Harr wavelet function.

2.2. Feature reduction scheme using PCA. One of the most common forms of dimensionality reduction is principal components analysis. Given a set of data, PCA finds the linear lower-dimensional representation of the data such that the variance of the reconstructed data is preserved [12, 15]. Using a system of feature reduction based on a combined principle component analysis on the feature vectors that calculated from the wavelets limiting the feature vectors to the component selected by the PCA should lead to an efficient classification algorithm utilizing supervised approach. So, the main idea behind using PCA in our approach is to reduce the dimensionality of the wavelet coefficients. This leads to more efficient and accurate classifier.

The following algorithm is used to find out the principal components of the input matrix to the neural network. Now the input matrix consists of

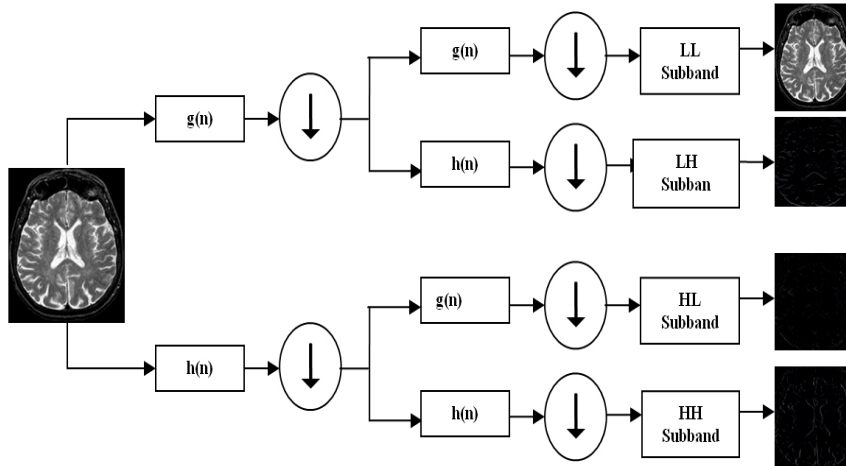


FIGURE 2. DWT schematically

only these principal components. The size of the input matrix is reduced from (1024) to (7). Algorithm 1 shows the involved steps for extracting principal components of the input vector to the two classifiers.

Therefore, the feature extraction process was carried out through two steps: firstly the wavelet coefficients were extracted by the DWT and then the essential coefficients have been selected by the PCA (see Fig.3.)

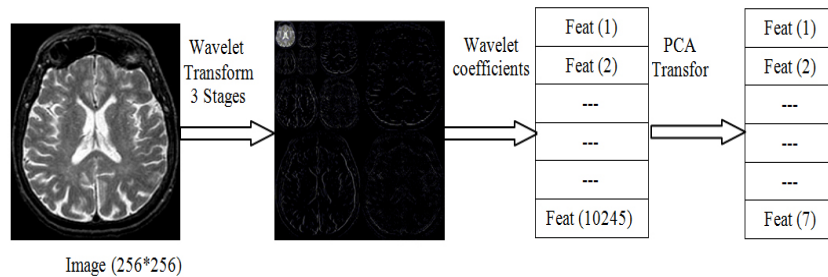


FIGURE 3. Schematic diagram for the used feature extraction and reduction scheme

3. DEVELOPING THE SUPERVISED LEARNING CLASSIFICATION

3.1. k-Nearest Neighbors based Classifier. One of the simplest classification techniques is the k- Nearest Neighbour classifier. Classification of an

Algorithm 1 PCA algorithm

Let X be an input data set (X : matrix of dimensions $M \times N$).

Perform the following steps:

- 1: Calculate the empirical mean $u[m] = (1/N) \sum_{i=1}^N X[m, n]$.
 - 2: Calculate the deviations from the mean and store the data in the matrix $B[M \times N]$; $B = X - u \cdot h$, where h is a $1 \times N$ row vector of all 1's: $h[n] = 1$ for $n = 1 \dots N$.
 - 3: Find the covariance matrix C : $C = (1/N) B \cdot B^*$.
 - 4: Find the eigenvectors and eigenvalues of the covariance matrix $V^{-1} C V = D$, where V : the eigenvectors matrix, D : the diagonal matrix of eigenvalues of C , $D[p, q] = \lambda_m$ for $p = q = m$ is the m th eigenvalues of the covariance matrix C .
 - 5: Rearrange the eigenvectors and eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_N$.
 - 6: Choosing components and forming a feature vector.
Save the first L columns of V as the $M \times L$ matrix W : $W[p, q] = V[p, q]$ for $n = 1 \dots M$, $q = 1 \dots L$ where $1 \leq L \leq M$.
 - 7: Deriving the new data set The eigenvectors with the highest eigenvalues are projected into space, this projection results in a vector represented by fewer dimension ($L < M$) containing the essential coefficients.
-

input feature vector X is done by determining the k closest training vectors according to a suitable distance metric. The vector X is then assigned to that class to which the majority of those k nearest neighbours belong to [12, 16]. The k -NN algorithm is based on a distance function and a voting function in k nearest neighbors, the metric employed is the Euclidean distance. The k -nearest neighbor classifier is a conventional nonparametric supervised classifier that is said to yield good performance for optimal values of k [15]. Like most guided learning algorithms, k -NN algorithm consists of a training phase and a testing phase. In the training phase, data points are given in a n -dimensional space. These training data points have labels associated with them that designate their class. In the testing phase, unlabeled data are given and the algorithm generates the list of the k nearest (already classified) data points to the unlabeled point. The algorithm then returns the class of the majority of that list [15, 17]. Algorithm 2 describes the k -NN steps.

3.2. Artificial Neural Network based Classifier. An ANN is a mathematical model consisting of a number of highly interconnected processing elements organized into layers, geometry and functionality of which have been

Algorithm 2 k-NN algorithm

- 1: Determine a suitable distance metric.
- 2: In the training phase: Stores all the training data set P in pairs(according to the selected features) $P = (y_i, c_i), i=1. . n$, where y_i is a training pattern in the training data set, c_i is its corresponding class and n is the amount of training patterns.
- 3: During the test phase: Computes the distances between the new feature vector and all the stored features (training data).
- 4: The k nearest neighbors are chosen and asked to vote for the class of the new example. The correct classification given in the test phase is used to assess the correctness of the algorithm. If this is not satisfactory, the k value can be tuned until a reasonable level of correctness is achieved.

resembled to that of the human brain. The ANN may be regarded as possessing learning capabilities in as much as it has a natural propensity for storing experimental knowledge and making it available for later use [18].

The neural network which was employed as the classifier required in this study had three layers, as shown in Fig (4). The first layer consisted of 7 input elements in accordance with the 7 feature vectors that selected from the wavelet coefficients by the PCA. The number of neurons in the hidden layer was four. The single neuron in the output layer was used to represent normal and abnormal human brain (see Fig. 4).

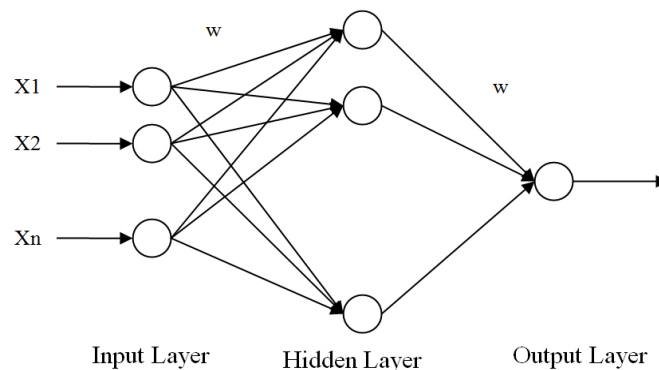


FIGURE 4. The architecture of used ANN

The most frequently used training algorithm in classification problems is the back-propagation (BP) algorithm, which is used in this work also. The details of back-propagation (BP) algorithm are well documented in the literature [18].

The neural network has been trained to adjust the connection weights and biases in order to produce the desired mapping. At the training stage, the feature vectors are applied as input to the network and the network adjusts its variable parameters, the weights and biases, to capture the relationship between the input patterns and outputs [18].

4. CASE STUDY

In this section, the proposed hybrid techniques have been implemented on a real human brain MRI dataset. All the input dataset (total images is 70: 60 images are abnormal and 10 normal) used for classification consists of axial, T2-weighted, 256 -256 pixel MR brain images, 60. These images were collected from the Harvard Medical School website ([http:// med.harvard.edu/AANLIB/](http://med.harvard.edu/AANLIB/)) [19]. Fig.5 shows some samples from the used data for normal and pathological brain: a- normal, b- Glioma, c- Metastatic bronchogenic carcinoma, d- Alzheimer's disease, visual agnosia.

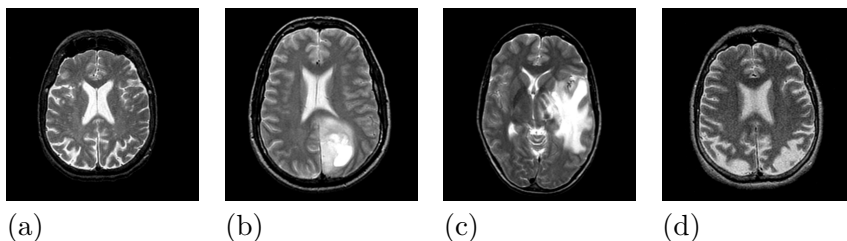


FIGURE 5. Samples from the used data

The algorithm described in this paper is developed locally and successfully trained in MATLAB version 7.1 using a combination of the Image Processing Toolbox, Wavelet toolbox (The MathWorks) for MATLAB. We performed all the computations of DWT+PCA+FP-ANN and DWT+PCA+k-NN classification on a personal computer with 1.5 MHz Pentium IV processor and 384 MB of memory, running under Windows-2000TM operating system. The programs can be run/tested on many different computer platforms where MATLAB is available. Algorithm 3 depicts the steps of the proposed two classifiers.

Algorithm 3 Pseudocode of the used hybrid techniques

Input: 256x256 brain images

Parameters: N is number of images

Stage (1): Features Extraction using DWT

FOR Loop on $i=1$ to N

Read the input images, Resize the images, and apply the DWT for the 3rd level using "Haar" family to extract the wavelet coefficients, Put the wavelet coefficients in a matrix X [MxN]

End Loop of i

ENDFOR

Repeat the above loop for the test image to extract its wavelet coefficients.

Concatenate the feature coefficients of the training images and the test image in one matrix.

Stage (2): PCA Features reduction

Loop on $J =1$ to N

Apply PCA transformation (according to algorithm 1) on the obtained wavelet coefficients. Put the new dataset in a matrix Y

End Loop on j

Stage (3): Classification Using two supervised techniques:

Classifier 1: based on ANN

Create the design of neural network with feed forward back-propagation algorithm. Create target vector. Train the net with the selected dataset and the desired target. Input the Features of test image on. Trained the neural network. Classify it.

Output: Normal or abnormal brain.

Classifier 2: based on k-NN

Loop for $g =1$ to 5, where 5= k-nn level.For $j=1$ to N

Apply the k-NN algorithm PCA.

END Loop j

END Loop on g Classify test image.

Output: Normal or abnormal brain

5. RESULTS AND DISCUSSIONS

In this section, we present the performance evaluation methods used to evaluate the proposed approaches. Finally, we will show the experimental results and examine the performance of the proposed classifiers for the

MRI dataset mentioned above. We evaluate the performance of the proposed method in terms of confusion matrix, sensitivity, specificity and accuracy. The three terms are defined as follows [20]:

Sensitivity (true positive fraction) is the probability that a diagnostic test is positive, given that the person has the disease.

$$(3) \quad \text{Sensitivity} = \frac{TP}{TP + FN}$$

Specificity (true negative fraction) is the probability that a diagnostic test is negative, given that the person does not have the disease.

$$(4) \quad \text{Specificity} = \frac{TN}{TN + FP}$$

Accuracy is the probability that a diagnostic test is correctly performed.

$$(5) \quad \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

TP (True Positives) - Correctly classified positive cases,
 TN (True Negative)- Correctly classified negative cases,
 FP (False Positives) -Incorrectly classified negative cases, and
 FN (False Negative) - Incorrectly classified positive cases.

Table 1 shows the classification rates for performing the proposed hybrid approach. In this experiment two classifiers based on supervised machine learning are presented for MRI normal/abnormal human brain classification. In the proposed methods using DWT, first three levels coefficients of decomposition of MR images with Harr as mother wavelet are computed. These coefficients are used for feature extraction. PCA is used for feature selection and NN and k-NN classifiers for MRI normal/abnormal human brain classification are used in methods 1 and 2, respectively.

For reducing the complexity of the system, PCA was used for feature reduction which was described in Section 3. The dimension of the feature vector was reduced from 1024 to 7 with the PCA algorithm. Limiting the feature vectors to the component selected by the PCA leads to an increase in accuracy rates and decreases the time complexity. In this experimental, MRI dataset that have healthy and diseased brain are classified by the proposed classifiers. The experimental results of the proposed classifiers are compared in the Table 1, which shows the percentage classification for the two different image classes. The analysis of the experimental results shows that classification accuracy 97% is achieved with the FP-ANN classifier and classification accuracy 98

To evaluate the effectiveness of our methods we compare our results with recently results [2, 5] for the same MRI datasets. Table 2 gives the classification accuracies of our method and the recently results. This comparison shows that our system has high classification accuracy and less computation due to the feature reduction based on the PCA.

6. CONCLUSIONS AND FUTURE WORKS

In this study, we have developed a medical decision support system with normal and abnormal classes. The medical decision making system designed by the wavelet transform, principal component analysis (PCA), and supervised learning methods (FP-ANN and k-NN) that we have built gave very promising results in classifying the healthy and patient brain. The benefit of the system is to assist the physician to make the final decision without hesitation.

According to the experimental results, the proposed method is efficient for classification of the human brain, normal or abnormal. Our proposal produced 95.9% sensitivity rate and 96% specificity rate for FP-ANN classifier and 96% sensitivity rate and 97% specificity rate for k-NN classifier. SOM and SVM [2, 5] produced the similar results. ANN method gained the worst sensitivity and specificity rate.

Our results have compared to the results reported very recently based on the same T2-wieghted MRI database. Our method can be employed for all types of MR images T1-wieghted, T2-wieghted, and proton density (T1- T2-PD). This research developed two hybrid techniques, DWT+PCA+FP-ANN and DWT+PCA+k-NN to classify the human brain MR images. The stated results show that the proposed method can make an accurate and robust classifier. The classification performances of this study show the advantages of this technique: it is easy to operate, noninvasive and inexpensive. The limitation of this work is that it requires fresh training each time whenever there is an increase in image database. The extension of the developed techniques for processing the pathological brain tissues (e.g. lesions, tumors)is the topic of future research.

TABLE 1. Classification rates for the used classifiers

Hybrid technique	TP	TN	FP	FN	Sensi- tivity (%)	Speci- ficity (%)	Accu- racy (%)
DWT+PCA+ANN	58	9	2	1	98.3	81.8	95.7
DWT+PCA+k-NN	60	9	1	0	98.4	100	98.6

TABLE 2. Classification performance (P) comparisons for the proposed technique and the recently works for the same MR images datasets.

Technique	P(%)
Our hybrid technique DWT+PCA+ANN	95.7
Our hybrid technique DWT + PCA + k-NN	98.6
DWT+SOM [2]	94
DWT+SVM with linear kernel [2]	96.15
DWT+SVM with radial basis function based kernel [2]	98

REFERENCES

- [1] L. M. Fletcher-Heath, L. O. Hall, D. B. Goldgof, F. R. Murtagh; Automatic segmentation of non-enhancing brain tumors in magnetic resonance images; *Artificial Intelligence in Medicine* 21 (2001), pp. 43-63.
- [2] Sandeep Chaplot, L.M. Patnaik, N.R. Jagannathan; Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network; *Biomedical Signal Processing and Control* 1 (2006), pp. 86-92.
- [3] F. Gorunescu; *Data Mining Techniques in Computer-Aided Diagnosis: Non-Invasive Cancer Detection*; PWASET Volume 25 November 2007 ISSN 1307-6884, PP. 427-430.
- [4] S. Kara , F. Dirgenali; A system to diagnose atherosclerosis via wavelet transforms, principal component analysis and artificial neural networks; *Expert Systems with Applications* 32 (2007), pp. 632-640.
- [5] M . Maitra, A. Chatterjee; Hybrid multiresolution Slantlet transform and fuzzy c-means clustering approach for normal-pathological brain MR image segregation, *Med Eng Phys* (2007), doi:10.1016/j.medengphy.2007.06.009.
- [6] P. Abdolmaleki, Futoshi Mihara, Kouji Masuda, Lawrence Danso Buadu; Neural networks analysis of astrocytic gliomas from MRI appearances' *Cancer Letters* 118 (1997), pp. 69-78.
- [7] T. Rosenbaum, Volkher Engelbrecht, Wilfried Kro?lls, Ferdinand A. van Dorstenc, Mathias Hoehn-Berlagec, Hans-Gerd Lenard; MRI abnormalities in neurofibromatosis type 1 (NF1): a study of men and mice; *Brain & Development* 21 (1999), pp. 268-273.
- [8] C. Cocosco , Alex P. Zijdenbos, Alan C. Evans; A fully automatic and robust brain MRI tissue classification method; *Medical Image Analysis* 7 (2003), pp. 513-527.
- [9] K. Karibasappa, S. Patnaik, "Face Recognition by ANN using Wavelet Transform Coefficients", *IE(I) Journal-CP*, 85,(2004), pp. 17-23, .
- [10] P.S.Hiremath, S. Shivashankar, and Jagadeesh Pujari; Wavelet Based Features for Color Texture Classification with Application to CBIR; *IJCSNS International Journal of Computer Science and Network Security*, 6 (2006), pp. 124- 133.
- [11] I. Daubechies, " Ten Lectures on Wavelets," *Regional Conference Series in Applied Mathematics*, SIAM, Philadelphia, 1992.
- [12] A. Sengur, An expert system based on principal component analysis, artificial immune system and fuzzy k-NN for diagnosis of valvular heart diseases *Comp. Biol. Med.* (2007), doi: 10.1016/j.compbiomed.2007.11.004.

- [13] D. Bouchaffra , J. Tan, Structural hidden Markov models for biometrics: Fusion of face and Fingerprint; Pattern Recognition 41 (2008), pp. 852 - 867.
- [14] M. Kocionek, A. Materka, M. Strzelecki P. Szczypinski Discrete wavelet transform - derived features for digital image texture analysis, Proc. of Interational Conference on Signals and Electronic Systems, 18-21 September 2001, Lodz, Poland, pp. 163-168
- [15] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, New York: John Wiley and Sons, Inc., 2001.
- [16] F. Latifoglu, K. Polat, S. Kara, S. Gunes; Medical diagnosis of atherosclerosis from Carotid Artery Doppler Signals using principal component analysis (PCA), k-NN based weighting pre-processing and Artificial Immune Recognition System (AIRS); Journal of Biomedical Informatics 41 (2008), pp. 15-23.
- [17] O'Farrell, M., E. Lewis, C. Flanagan, N. Jackman "Comparison of K-NN and Neural Network methods in the classification of Spectral Data from an Optical Fibre-Based Sensor System used for Quality Control in the Food Industry" Sensors and Actuators B: Chemical, Vol 111-112C (2005), pp. 354-362.
- [18] S. Haykin, Neural Networks: A comprehensive Foundation, Prentice Hall, 1999.
- [19] Harvard Medical School, Web: data available at <http://med.harvard.edu/AANLIB/>.
- [20] Kemal Polat, Bayram Akdemir, Salih Gnes; Computer aided diagnosis of ECG data on the least square support vector machine; Digital Signal Processing 18 (2008), pp. 25-32.

PHYSICS DEPARTMENT, FACULTY OF SCIENCE, AIN SHAMS UNIVERSITY, ABBASSIA, CAIRO 11566, EGYPT

E-mail address: e_eldahshan@yahoo.com

FACULTY OF COMPUTER AND INFORMATION SCIENCE, AIN SHAMS UNIVERSITY, ABBASSIA, CAIRO, EGYPT

E-mail address: absalem@asunet.shams.edu.eg and abmsalem@yahoo.com

FACULTY OF ENGINEERING, MISR UNIVERSITY FOR SCIENCE AND TECHNOLOGY, 6TH OCTOBER CITY, CAIRO, EGYPT

E-mail address: tyounis@must.edu.eg

OVERVIEW OF FUZZY METHODS FOR EFFORT ESTIMATION BY ANALOGY

MILITON FRENȚIU AND HORIA F. POP

ABSTRACT. Estimation of cost and time to develop a software product P is a well known problem. Among the possible ways to do it is analogy with old finished projects. We describe an experiment using different versions of the Fuzzy c -Means algorithm, to find projects similar to P . Then the information about these projects is used to estimate the effort needed to build P .

1. INTRODUCTION

Estimation of cost and optimal time to develop a software product is a well known problem. There are various methods to solve this problem:

- expert opinion, i.e. guessing on personal experience;
- analogy with already finished projects;
- decomposition of the system into its smallest components and estimating the effort to build each component;
- models using mathematical formulas to compute the cost.

There are some models for cost estimation. Here we mention COCOMO I [Boe81], Albrecht's Function Points [Alb83], and COCOMO II [Boe85]. The COCOMO I is based on the size of the product, which is unknown at the beginning of the software process. Albrecht's model is based on the requirements and uses the information about the input data, the needed results, the files used in the system and the inquiries made by the user. COCOMO II effort estimation equations are adjusted basic equation using 17 cost drivers and five scale factors. These cost drivers are personnel attributes, process attributes, product attributes, and computer attributes.

Received by the editors: April 5, 2009.

2000 *Mathematics Subject Classification.* 68N30.

1998 *CR Categories and Descriptors.* D.2.3 [**Software**] : Software Engineering – *Coding Tools and Techniques*; I.5.1 [**Computing Methodologies**] : Pattern Recognition – *Models – Fuzzy set*; G.3 [**Mathematics of Computing**] : Probability and Statistics – *Data analysis*.

Key words and phrases. Software metrics, Cost estimation, Fuzzy clustering.

To predict by analogy the cost (effort) and time to develop a software system P means to find the most similar finished system S and to use its known characteristics for prediction [She96]. The most similar system is that one which is closest to P in the Euclidean space.

First problem that arise is to define and compute the similarity of two projects S and P . We need some important characteristics of these projects, and the possibility to obtain these characteristics from the requirements of the projects. They appear in Albrecht's model, or in COCOMO2, and may be quantifications for the complexity of the system, the required reliability, the number of inputs, the number of outputs, the number of files, the number of screens and so forth. If we use n characteristics to describe a project, it is represented as a point in the n -dimensional Euclidean space. The similarity of two projects may be defined as the distance between the corresponding points.

Estimating the cost of P by analogy is reduced to finding the closest previously completed similar project S . Then the actual data from the project S is extrapolated to estimate the cost of P .

An improvement to the above described method is to select some projects most similar to P and to deduce the cost for the proposed project from data of these most similar completed projects. Here we suggest using both the well-established Fuzzy Hierarchical Clustering procedure and as well a restricted approach thereof, to find the class C of the most similar projects to P .

The cost of P is computed as a weighted average of $Cost(S)$ for $S \in C$, i.e.

$$Cost(P) = \sum w(S) \times Cost(S), S \in C$$

where the costs $Cost(S)$ are estimations of $Cost(P)$ based on information available for project S , the weights $w(S)$ are determined based on the membership degrees of all items in the class and the sum of all $w(S)$ for all $S \in C$ is 1. Alternate estimations of the cost will be studied in a future paper.

2. RESTRICTED FUZZY CLUSTERING

Let us consider a set of classified objects, $X = \{x^1, \dots, x^p\} \in \mathbb{R}^s$ and the fuzzy partition $P = \{A_1, \dots, A_n\}$ corresponding to the cluster substructure of the set X . Let $x^0 \in \mathbb{R}^d$ be an object that needs to be classified with respect to the fuzzy partition P .

The algorithm we are presenting here computes the optimal fuzzy partition \tilde{P} corresponding to the set $\tilde{X} = X \cup \{x^0\}$, by using a mechanism similar to Fuzzy n -means, with the difference that the membership degrees of the objects in X to the classes A_i , $i = 1, \dots, n$ may not be modified.

In what follows we consider a metric d in the Euclidean space \mathbb{R}^s . We will suppose that d is norm induced, so $d(x, y) = (x - y)^T M(x - y)$, $x, y \in \mathbb{R}^s$, where M is a symmetrical and positively defined matrix.

The objective function we have in mind for our problem is similar to that for the Fuzzy n -Means Algorithm:

$$\tilde{J}(\tilde{P}, L) = \sum_{i=1}^n \sum_{j=0}^p (A_i(x^j))^2 d^2(x^j, L^i),$$

with the mention that $A_i(x^j)$ are kept constant for each i and for $j = 1, \dots, p$.

The main result with respect to determining the fuzzy partition \tilde{P} and its representation L minimizing the function \tilde{J} is the following

Theorem. (i) The fuzzy partition $\tilde{P} = \{A_1, \dots, A_n\}$ has the minimum value of the function $J(\cdot, L)$ if and only if

$$(1) \quad A_i(x^0) = \frac{1}{\sum_{k=1}^n \frac{d^2(x^0, L^i)}{d^2(x^0, L^k)}}.$$

(ii) The set of prototypes $L = \{L^1, \dots, L^n\}$ has the minimum value of the function $J(\tilde{P}, \cdot)$ if and only if

$$(2) \quad L^i = \frac{\sum_{j=0}^p (A_i(x^j))^2 x^j}{\sum_{j=0}^p (A_i(x^j))^2}.$$

With this result, the optimal membership degrees for x^0 to the classes A_i will be determined using an iterative method in which \tilde{J} is successively minimized with respect to \tilde{P} and L . The process will start with the initialization of prototypes L^i to the values that correspond to the fuzzy membership degrees of the original fuzzy partition P .

The resulted algorithm, **Restrictive Fuzzy n -Means Clustering Algorithm**, follows:

- S1 Let us have X and P as given variables.
- S2 Determine the initial positions of the prototypes L^i according to value of P .
- S3 Determine the membership degrees $A_i(x^0)$, $i = 1, \dots, n$, using relation (1).
- S4 Determine the new positions of prototypes L^i , $i = 1, \dots, n$, using relation (2).

S5 If the new positions of the prototypes L^i are close enough to the former positions, then **stop**, else return to step S3.

3. HIERARCHIC FUZZY CLUSTERING WITH INCOMPLETE DATA

The problem of incomplete data is extremely important [Hat01]. If a cluster substructure of a data set with incomplete data is required, the option of ignoring the incomplete data items altogether is not realistic, because that would assume ignoring useful, available data. Hathaway and Bezdek have proposed in 2001 a few strategies to cope with the incomplete data problem [Hat01]. Their approach has been to extend the Fuzzy c -means algorithm in such a way as to accept incomplete data as well. We have used the approach of Hathaway and Bezdek and produced hierarchic clustering versions thereof.

The main issue is that, in computing the fuzzy membership degrees, the square distances $d^2(x^j, L^i)$ from the data item x^j to the class prototype L^i cannot be computed for incomplete data. Assuming that we use the Euclidean metric, we would need to compute

$$(3) \quad D_{ij} = \sum_{k=1}^s (x_k^j - v_k^i)^2 .$$

However, for some values of j and k , x_k^j may be missing. In our paper we are going to use the *Partial Distance Strategy* of Hathaway and Bezdek [Hat01]. Namely, we are going to compute this sum considering only the available values and we are going to scale the result to take into account the missing dimensions. As such, we are actually going to compute

$$(4) \quad \tilde{D}_{ij} = \frac{s}{\sum_{k=1}^s I_{jk}} \sum_{k=1}^s I_{jk} (x_k^j - v_k^i)^2 ,$$

where $I_{jk} = 1$ if the value x_k^j is available, and $I_{jk} = 0$ if the value is unavailable, in which case the difference will not be computed.

Similarly, the new prototypes will have to be computed by taking into account only the available data:

$$(5) \quad L_k^i = \frac{\sum_{j=1}^p I_{jk} (A_i(x^j))^2 x_k^j}{\sum_{j=0}^p I_{jk} (A_i(x^j))^2},$$

where I_{jk} have the same meaning as above.

The resulted algorithm, **Hierarchic Fuzzy Clustering with Incomplete Data**, follows:

- S1 Let us consider the data set X and the initial fuzzy partition P .
- S2 Determine the positions of prototypes L^i , $i = 1, \dots, n$, using relation (5).
- S3 Determine the membership degrees $A_i(x^0)$, $i = 1, \dots, n$, using dissimilarities from (4).
- S4 If the new fuzzy partition is close enough to the former fuzzy partition, then **stop**, else return to step S2.

4. NON-BINARY DIVISIVE HIERARCHIC FUZZY CLUSTERING

One of the key problems of divisive clustering is the choice for binary split at every level of the clustering hierarchy. This is a very simple, intuitive and effective approach. While it leads to desired results in most of the cases, there are situations where the data does not show a natural binary split-up. Here a different approach must be used.

Our approach is to generalize the binary divisive approach introduced in [Dum88].

The central point of the Fuzzy Hierarchic Divisive method is the binary polarization index. Thus, considering a binary fuzzy partition $P = \{C_1, C_2\}$, of the fuzzy set C , the partition separation index is defined as

$$R(P) = \frac{\sum_{i=1}^2 \sum_{k=1}^n C_{i,1/2}(x_k)}{\sum_{k=1}^n C(x_k)}$$

where

$$C_{i,t}(x) = \begin{cases} C_i(x) & C_i(x) > t \\ 0 & \text{otherwise} \end{cases}$$

Now, instead of considering a binary partition, let us consider one with p arity, $P = \{C_1, \dots, C_p\}$. We define the generalized polarization index quite naturally as

$$R(P) = \frac{\sum_{i=1}^p \sum_{k=1}^n C_{i,1/2}(x_k)}{\sum_{k=1}^n C(x_k)}.$$

Our key problem is then to obtain the best partition arity at every node of the classification tree. Fortunately, this is a very simple issue from a constructive point of view. Our assumption has been that the data does not display a binary structure. So, at every node of the classification tree we are going to construct both a 2-partition and a 3-partition. The decision of which is best to adopt is based on the higher partition polarisation index. By working recursively, in this way, we are going to obtain a classification tree where each node has either two or three children, as appropriate.

5. EXPERIMENT

We have used the information we had about 19 student software projects to estimate the cost of a new project (numbered 20). This information refers to the complexity, reliability, considered difficulty (all three metrics have values between 1 and 5, denoting very low, low, normal, high and very high), number of inputs, number of outputs, number of files and number of screens). Also, the computed function points and the cost of these projects are known.

The final partition produced using this 20×8 data matrix contains three classes. All the three classes are well separated. The core elements of the classes have quite high fuzzy membership degrees, while only a few of elements per class, not members of the class, have fuzzy membership degrees reasonably large.

We are estimating the cost of project 20 using the projects member of the very same class project 20 is a member of. Very similar results are obtained using the other suggested approach, i.e. Restricted Fuzzy Clustering, where we first find the cluster substructure of projects 1-19 and then embed the 20-th project as an extra data item in the former fuzzy partition.

6. CONCLUSIONS AND FUTURE RESEARCH

We have used fuzzy classification as a way to find the projects similar to our project P . There are usually more than one very similar projects, and these are the projects found in the same class with P . Then, to predict the cost/effort needed to built P , we have used the information about all these projects similar to P .

It is considered that estimating by analogy is a superior technique to estimation via algorithmic model in at least some circumstances [She96]. However, it requires to maintain a database with information about the finished systems. Oftenly, the information about finished projects is incomplete, and we cannot find and add the missing information about these old projects. It would be useful to obtain the classification in the absence of some information. Factorial analysis [Wat67] may also be used to predict the effort to build a new project from the information about finished projects.

7. ACKNOWLEDGEMENT

This material is based upon work supported by the Romanian National University Research Council under award PN-II no. ID_550/2007.

REFERENCES

- [Alb83] Albrecht A. J., Gaffney Jr J. E. (1983). Software function, source lines of code, and development effort prediction: A software science validation. *IEEE Transactions on Software Engineering* 9 (6), 639-648.
- [Boe81] Boehm B. W. (1981). *Software engineering economics*, Prentice-Hall: Englewood Cliffs, N.J.
- [Boe85] Boehm B. W., Clark B., et al. (1995). Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering* 1, 57-94.
- [Dum88] D. Dumitrescu, Hierarchical pattern classification, *Fuzzy Sets and Systems* 28 (1988), 145-162.
- [Hat01] Hathaway R. J., Bezdek J. C., Fuzzy C-Means Clustering of Incomplete Data, *IEEE Trans. Systems, Man, Cybernetics, Part B: Cybernetics*, 31, 5 (2001), 1062-1071.
- [She96] Shepperd M., Schofield C., Kitchenham B., Effort Estimation Using Analogy, *Proceedings of the 18th International Conference on Software Engineering*, Berlin, 170-178.
- [Pop95] Pop H. F., Supervised fuzzy classifiers. *Studia Universitatis Babeş-Bolyai, Series Mathematica* 40, 3 (1995), 89-100.
- [Wat67] Watanabe S., Haven H., Karhunen-Loeve expansion and Factor analysis. Theoretical remarks and applications, in *Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, Prague 1967.

DEPARTMENT OF COMPUTER SCIENCE, BABEȘ-BOLYAI UNIVERSITY, 1 M. KOGĂLNICEANU ST., 400084 CLUJ-NAPOCA, ROMANIA
E-mail address: mfrentiu@cs.ubbcluj.ro

DEPARTMENT OF COMPUTER SCIENCE, BABEȘ-BOLYAI UNIVERSITY, 1 M. KOGĂLNICEANU ST., 400084 CLUJ-NAPOCA, ROMANIA
E-mail address: hfpop@cs.ubbcluj.ro

A COMPARATIVE ANALYSIS OF CLUSTERING ALGORITHMS IN ASPECT MINING

GRIGORETA SOFIA COJOCAR, GABRIELA CZIBULA AND ISTVAN GERGELY
CZIBULA

ABSTRACT. *Aspect mining* is a research direction that tries to identify crosscutting concerns in already developed software systems, without using aspect oriented programming. The goal is to identify them and then to refactor them to aspects, to achieve a system that can be easily understood, maintained and modified. In this paper we aim at comparatively analyzing four clustering algorithms in aspect mining. The comparison is made using a set of quality measures previously introduced in aspect mining literature.

1. INTRODUCTION

Separation of concerns [12] is a very important principle of software engineering that, in its most general form, refers to the ability to identify, encapsulate and manipulate those parts of software that are relevant to a particular concept, goal, or purpose.

Crosscutting concerns [8] are parts of a program which affect or crosscut other concerns. Usually these concerns cannot be cleanly decomposed from the rest of the system, and they are mixed with many core concerns from the system leading to code scattering and code tangling, and, also, to systems that are hard to explore and understand. Identifying crosscutting concerns automatically improves both the maintainability and the evolution of the software system. Crosscutting concerns are a relevant source of problems to program comprehension and software maintenance. Examples of crosscutting concerns are persistence, synchronization, exception handling, error management and logging.

The aspect oriented paradigm (AOP) is one of the approaches proposed, so far, for designing and implementing crosscutting concerns [8]. Aspect oriented

Received by the editors: April 15, 2009.

2000 *Mathematics Subject Classification*. 62H30, 68N99.

1998 *CR Categories and Descriptors*. I.5.3 [**Computing Methodologies**]: Pattern Recognition – *Clustering*; D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement – *Restructuring, reverse engineering, and reengineering*.

Key words and phrases. Aspect mining, Clustering, Crosscutting concerns.

techniques allow crosscutting concerns to be implemented in a new kind of module called *aspect*, by introducing new language constructs like pointcuts and advices.

Aspect mining is a research direction that tries to identify crosscutting concerns in already developed software systems, without using AOP. The goal is to identify them and then to refactor them to aspects, to achieve a system that can be easily understood, maintained and modified. There exist many reasons for migrating a legacy system to an aspect oriented based system. An inadequate solution for crosscutting concerns implementation has a negative impact on the final system with consequences like duplicated code, *scattering* of concerns throughout the entire system and *tangling* of concern-specific code with that of other concerns. The *code scattering* symptom means that the code that implements a crosscutting concern is spread across the system and the *code tangling* symptom indicates that the code that implements some concern is mixed with code from other (crosscutting) concerns. These consequences lead to software systems that are hard to maintain and to evolve. When aspect oriented techniques are used, the crosscutting concerns are cleanly separated from the core concerns, the latter becoming oblivious of them.

Clustering [6] has been already applied in aspect mining, as clustering aims at identifying groups of similar objects, and crosscutting concerns in legacy systems can be viewed as such groups (of methods, statements, etc.). Several partitional and hierarchical clustering algorithms were developed in [1, 2, 13, 14] for crosscutting concerns identification.

In this paper we are focusing on comparing the results obtained by four clustering algorithms (*kAM*, *HAM*, *PACO* and *HACO*) that were previously introduced in the aspect mining literature.

The paper is structured as follows. Section 2 presents the clustering based approach that we have used for the evaluation that we aim at performing. The clustering algorithms used in our evaluation for identifying crosscutting concerns are described in Section 3. Section 4 provides the comparative analysis of the considered clustering algorithms, and Section 6 gives some conclusions of the paper and further work.

2. CLUSTERING BASED ASPECT MINING

In this section we present the problem of identifying *crosscutting concerns* as a clustering problem [11].

Let $S = \{es_1, es_2, \dots, es_n\}$ be a software system, where $es_i, 1 \leq i \leq n$, is an *element* from the software system. An *element* can be a statement, a method, a class, a module, etc. We denote by n ($|S|$) the number of elements of the system.

In the following, we will consider a crosscutting concern as a set of elements $C \subset S$, $C = \{e_1, e_2, \dots, e_{cn}\}$, elements that implement this concern. The number of elements in the crosscutting concern C is $cn = |C|$. Let $CCC = \{C_1, C_2, \dots, C_q\}$ be the set of all crosscutting concerns that exist in the system S . The number of crosscutting concerns in the system S is $q = |CCC|$. We suppose that two different crosscutting concerns do not have elements in common, meaning that $C_i \cap C_j = \emptyset, \forall i, j, 1 \leq i, j \leq q, i \neq j$.

Definition 1. [11] *Partition of a system S .*

The set $\mathcal{K} = \{K_1, K_2, \dots, K_p\}$ is called a **partition** of the system S iff:

- (1) $1 \leq p \leq n$;
- (2) $K_i \subseteq S, K_i \neq \emptyset, \forall i, 1 \leq i \leq p$;
- (3) $S = \bigcup_{i=1}^p K_i$;
- (4) $K_i \cap K_j = \emptyset, \forall i, j, 1 \leq i, j \leq p, i \neq j$.

We will refer to \mathcal{K} as a *set of clusters* and to K_i as the *i -th cluster* of \mathcal{K} .

In fact, the problem of aspect mining can be viewed as the problem of finding a partition \mathcal{K} of the system S .

A partition of a software system S can be obtained by different kinds of algorithms (i.e., a clustering algorithm).

Definition 2. [11] *Optimal partition of a system S .*

Being given a partition $\mathcal{K} = \{K_1, K_2, \dots, K_p\}$ of the system S , \mathcal{K} is called an **optimal partition** of system S with respect to the set $CCC = \{C_1, C_2, \dots, C_q\}$ of all crosscutting concerns, iff:

- (1) $p \geq q$;
- (2) $\forall C \in CCC, \exists K_C \in \mathcal{K}$ such that $C = K_C$.

Intuitively, \mathcal{K} is an optimal partition of the system S if all the elements implementing a crosscutting concern C_i ($1 \leq i \leq q$) are in the same cluster K_{j_i} ($1 \leq j_i \leq p$) and they are the only elements in K_{j_i} .

2.1. Identification of Crosscutting Concerns. In order to discover the crosscutting concerns from the system, we analyze the source code of the software system to be mined. All classes, methods and relations between them are computed. Afterwards, a clustering algorithm is used to identify a partition of a software system S in which the methods belonging to a crosscutting concern should be grouped together. The final step is to manually analyzed the obtained results.

Let us consider that the software system to be mined consists of a set of classes $\mathcal{C} = \{c_1, c_2, \dots, c_s\}$, each class containing one or more methods. In our clustering approach, the objects to be clustered are the methods from the

software system S , i.e., $\mathcal{M} = \{m_1, m_2, \dots, m_n\}$. Our focus is to group the methods such that the ones belonging to the same crosscutting concern to be placed in the same cluster.

3. CLUSTERING ALGORITHMS FOR CROSSCUTTING CONCERNS IDENTIFICATION

In order to group the methods in clusters we have used four clustering algorithms especially defined for aspect mining: *kAM* introduced in [14], *HAM* introduced in [13], *PACO* introduced in [1], and *HACO* introduced in [2]. In the following we briefly describe the algorithms used for our evaluation.

3.1. kAM. This is based on *k-means* clustering techniques, but it tries to avoid the two main disadvantages of *k-means* algorithm: the dependence on the number of clusters given as input and the dependence on the initial choice of the centroids. In order to accomplish this, *kAM* uses a heuristic for choosing the optimal number p of clusters, and the initial centroids. This heuristic provides a good enough choice of the initial centroids and is particular to aspect mining. The main idea of *kAM*'s heuristic is the following:

- (i) The initial number p of clusters is n (the number of methods from the system).
- (ii) The method chosen as the first centroid is the most "distant" method from the set of all methods (i.e., the method that maximizes the sum of distances from all other methods).
- (iii) For each remaining methods (that were not chosen as centroids), the minimum distance ($dmin$) from the method and the already chosen centroids is computed. The next centroid is chosen as the method m that maximizes $dmin$ and this distance is greater than a positive given threshold ($distMin$). If such a method does not exist it means that m is very close to its nearest centroid nc and should not be chosen as a new centroid (m and nc should belong to the same cluster). In this case, the number p of clusters will be decreased.
- (iv) The step (iii) will be repeatedly performed, until the number p of clusters and the number of centroids are equal.

3.2. HAM. This algorithm is based on the idea of hierarchical agglomerative clustering, but uses a heuristic for merging two clusters, heuristic that is particular for aspect mining. In this algorithm, the distance between two clusters K_i and K_j is considered to be the largest distance between the objects from the clusters, i.e., it uses the complete linkage metric [6].

The heuristic used in *HAM* is that, at a given step, the most two similar clusters (the pair of clusters that have the smallest distance between them) are

merged only if the distance between them is less or equal to a given threshold, *distClusMin*.

Both, *kAM* and *HAM* algorithms, use a vector space model in order to compute the dissimilarity between two methods. The following vector space model is used: a method m is characterized by an $(s + 1)$ -dimensional vector $\{FIV, B_1, B_2, \dots, B_s\}$, where s is the number of classes from the software system S (called application classes), FIV is the value of the *fan-in* metric and B_i is the value of the attribute corresponding to the application class c_i ($1 \leq i \leq s$), as follows:

$$B_i = \begin{cases} 1 & \text{if } m \text{ is called from at least one method belonging to} \\ & \text{application class } c_i \\ 0 & \text{otherwise} \end{cases} .$$

For both *kAM* and *HAM* algorithms, the distance between two methods is computed using the *Euclidian* distance.

3.3. PACO. This algorithm is based on *k-medoids* or *PAM* (*Partitioning around medoids*) algorithm [7], that finds representative objects, called *medoids*, in clusters. The algorithm starts with p initial representative objects for the clusters (medoids), then iteratively recalculates the clusters (each object is assigned to the closest cluster - medoid), and their medoids until convergence is achieved. At a given step, a medoid of a cluster is replaced by a non-medoid if it improves the total distance of the resulting clustering [7]. *k-medoids* algorithms have the same disadvantages as *k-means* algorithm: the dependence on the number of clusters given as input and the dependence on the initial choice of the medoids.

In order to avoid these disadvantages, *PACO* algorithm uses a heuristic for choosing the number of medoids (clusters) and the initial medoids. The heuristic is similar to the one used for *kAM*.

After selecting the initial medoids, *PACO* behaves like the classical *k-medoids* algorithm.

In order to compute the dissimilarities between methods three distance metrics were used:

- *Scattering* distance that captures the *scattering* symptom of crosscutting concerns.

$$(1) \quad d_S^P(m_i, m_j) = \begin{cases} 1 - \frac{|in(m_i) \cap in(m_j)|}{|in(m_i) \cup in(m_j)|} & \text{if } in(m_i) \cap in(m_j) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

where, for a given method $m \in \mathcal{M}$, $in(m)$ defines the set of methods and classes that invoke m , as expressed in Equation (2).

$$(2) \quad in(m) = \{m\} \cup \{m' \in \mathcal{M} \cup \mathcal{C} \mid m' \text{ invoke } m\}.$$

- *Tangling* distance that captures the *tangling* symptom of crosscutting concerns:

$$(3) \quad d_T^P(m_i, m_j) = \begin{cases} 1 - \frac{|r(m_i) \cap r(m_j)|}{|r(m_i) \cup r(m_j)|} & \text{if } r(m_i) \cap r(m_j) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

where, for a given method $m \in \mathcal{M}$, $in(m)$ is defined as in Equation (2), and $r(m)$ denotes the set of relevant properties for each invocation context $inv \in in(m)$.

- *Scattering-Tangling* distance that tries to capture both *scattering* and *tangling* symptoms:

$$(4) \quad d_{ST}^P(m_i, m_j) = \min\{d_S(m_i, m_j), d_T(m_i, m_j)\}.$$

3.4. HACO. This algorithm is based on the idea of hierarchical agglomerative clustering, and uses a heuristic for determining the number of clusters. In order to determine the number p of clusters, the focus is on determining p representative methods from the software system S . The method chosen as the first representative method is the most “distant” method from the set of all methods (the method that maximizes the sum of distances from all the other methods). At each step we select from the remaining methods the most distant method relative to the already chosen methods. If the selected method is close enough to the already chosen representative methods, then the process is stopped, otherwise the selected method is considered as a new representative method.

The distance metric used for computing the similarity between two methods is defined as follows:

$$(5) \quad d_{ST}^H(m_i, m_j) = \begin{cases} 0 & i = j \\ 1 - \frac{|Col(m_i) \cap Col(m_j)|}{|Col(m_i)| + |Col(m_j)|} & \text{if } Col(m_i) \cap Col(m_j) \neq \emptyset, \\ \infty & \text{otherwise} \end{cases}$$

where $Col(m)$ is a collection consisting of: the method itself, the class in which the method is defined, the classes and methods that invoke m and the classes in which the classes and methods that invoke m are contained. This distance function considers both the *scattering* and *tangling* symptoms.

4. COMPARATIVE ANALYSIS

In this section we comparatively analyze the results obtained by the above presented clustering algorithms. The comparison is made considering how well did a clustering algorithm succeed in identifying clusters corresponding to the crosscutting concerns from the software system to be mined.

For this evaluation we have used two quality measures introduced in [11]:

- **DISP.** The dispersion degree of crosscutting concerns in clusters.
- **DIV.** The degree to which each cluster contains elements from different crosscutting concerns or elements from other concerns.

DISP measure defines the dispersion degree of crosscutting concerns in clusters, considering, for each crosscutting concern, the number of clusters that contain elements belonging to the concern. *DIV* measure defines the degree to which each cluster contains elements from different crosscutting concerns or elements from other concerns.

For each measure the values are in the interval $[0, 1]$, the ideal value being 1 for both of them. Larger values for *DISP* and *DIV* indicate better partitions with respect to set of the crosscutting concerns to be discovered, meaning that both measures have to be maximized. Theorem 3 that gives the necessary and sufficient conditions for a partition to be an optimal partition was proven in [11]:

Theorem 3. *If \mathcal{K} is a partition of the software system S and CCC is the set of crosscutting concerns in S , then \mathcal{K} is an **optimal partition** iff $DISP(CCC, \mathcal{K}) = 1$ and $DIV(CCC, \mathcal{K}) = 1$.*

We will use these properties of the above presented quality measures in analyzing the results obtained by the different clustering algorithms used for this comparison.

4.1. Case study. We have considered a medium size software application, the open source JHotDraw version 5.4b1 case study [3]. It is a Java GUI framework for technical and structured graphics, developed by Erich Gamma and Thomas Eggenchwiler, as a design exercise for using design patterns. It consists of **396** classes and **3359** methods.

The set of crosscutting concerns used for the evaluation is: *Adapter, Command, Composite, Consistent behavior, Contract enforcement, Decorator, Exception handling, Observer, Persistence, and Undo*. The set of crosscutting concerns and their implementing methods was constructed using the results reported by Marin et al. and publicly available at [10].

We mention that the value of the threshold is 1 for all clustering algorithms. Analyzing the results from Table 1 we can conclude the following:

Algorithm	Distance function	DISP	DIV
kAM	Euclidian	0.4005	0.9972
HAM	Euclidian	0.4005	0.9973
PACO	d_S^P	0.4444	0.9753
PACO	d_T^P	0.4433	0.8732
PACO	d_{ST}^P	0.4207	0.8798
HACO	d_{ST}^H	0.457	1

TABLE 1. Results for JHotDraw case study.

- From the analyzed clustering algorithms, the algorithm that provides the best results is *HACO*, as *DISP* and *DIV* have the larger values. This means that in the partition obtained by *HACO* algorithm, the methods from the crosscutting concerns were better grouped than in the partitions obtained by the other algorithms. *HACO* also provides the maximum value for *DIV* measure, i.e. 1, meaning that for each crosscutting concern its elements are not mixed with elements from other (crosscutting) concerns.
- The elements of crosscutting concerns are spread in two or more clusters for all algorithms, as the values of the *DISP* measure are less than 0.5 for all algorithms.
- For *PACO* algorithm, the *scattering* distance has obtained the best results, for both *DISP* and *DIV* measures.
- The vector space model approach has obtained better results for *DIV* measure than *PACO*, but it has obtained the worst results for *DISP* measure.
- The hierarchical clustering approach seems to be more appropriate in aspect mining than partitional clustering.
- None of the clustering algorithms used has succeeded in obtaining an optimal partition of the software systems. The value of the *DISP* measure is less than 1 for all algorithms, and the value of *DIV* measure is 1 only for *HACO* algorithm.

As a conclusion, *DISP* measure can be improved for *HACO* algorithm by improving the distance semi-metric d_{ST}^H used for discriminating the methods from the software system in the clustering process.

5. RELATED WORK

In this section we briefly present other existing clustering based approaches used for crosscutting concerns identification.

Shepherd and Pollock [15] use clustering to find methods with similar name as an indication of crosscuttingness. They perform agglomerative hierarchical

clustering in order to group methods. The objects to be clustered are the names of methods from the software system under analysis. The distance function between two methods m_1 and m_2 is proportional with the common substring length. The authors have developed a tool, called **AMAV**, that helps users navigate and analyze the obtained clusters. The rest of the approach is just manual analysis of the obtained results using the tool.

He and Bai [4] have proposed an aspect mining technique based on dynamic analysis and clustering that also uses association rules. They first use clustering to obtain crosscutting concern candidates and then use association rules to determine the position of the source code belonging to a crosscutting concern in order to ease refactoring.

We did not provide a comparison of the clustering algorithms considered in this study with the two other existing clustering based aspect mining approaches for the following reasons:

- Shepherd and Pollock have proposed in [15] an aspect mining tool based on clustering, but it does not automatically identify the crosscutting concerns. The user of the tool has to manually analyze the obtained clusters in order to discover crosscutting concerns.
- The technique proposed by He and Bai cannot be reproduced, as they do not report neither the clustering algorithm used, nor the distance metric between the objects to be clustered. Also, the results obtained for the case study used by the authors for evaluation are not available. For these reasons, we cannot provide a comparison with this technique.

6. CONCLUSIONS AND FURTHER WORK

In this paper we have provided a comparative analysis of four clustering algorithms that are used for crosscutting concerns identification: *kAM*, *HAM*, *PACO*, and *HACO*. For the evaluation we have used two quality measures that were previously introduced in the aspect mining literature.

In the future we plan to apply the clustering algorithms considered in this paper to other larger software systems. We will also consider to use other *unsupervised learning* techniques (such as self-organizing maps [9], Hebbian learning [5]) in order to identify crosscutting concerns in existing software systems.

ACKNOWLEDGEMENT. This work was supported by research project ID_2268/2009, sponsored by the Romanian National University Research Council (CNCSIS).

REFERENCES

- [1] Gabriela Czibula, Grigoreta Sofia Cojocar, and Istvan Gergely Czibula. A Partitioned Clustering Algorithm for Crosscutting Concerns Identification. In *Proceedings of the International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS '09)*, pages 111–116, Cambridge, UK, February, 21-23 2009.
- [2] Istvan Gergely Czibula, Gabriela Czibula, and Grigoreta Sofia Cojocar. Hierarchical Clustering for Identifying Crosscutting Concerns in Object Oriented Software Systems. In *Proceedings of the 4th Balkan Conference in Informatics (BCI'09)*, Thessaloniki, Greece, September, 17-19 2009, submitted.
- [3] E. Gamma. JHotDraw Project. <http://sourceforge.net/projects/jhotdraw>.
- [4] Lili He and Hongtao Bai. Aspect Mining using Clustering and Association Rule Method. *International Journal of Computer Science and Network Security*, 6(2):247–251, February 2006.
- [5] John Hertz, Richard G. Palmer, and Anders S. Krogh. *Introduction to the Theory of Neural Computation*. Perseus Publishing, 1991.
- [6] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey, 1998.
- [7] L. Kaufman and P. J. Rousseeuw. Clustering by means of Medoids. *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pages 405–416, 1987.
- [8] Gregor Kiczales, John Lamping, Anurag Menhdhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier, and John Irwin. Aspect-Oriented Programming. In *Proceedings European Conference on Object-Oriented Programming*, volume LNCS 1241, pages 220–242. Springer-Verlag, 1997.
- [9] Teuvo Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, Heidelberg, New York, 2001.
- [10] Marius Marin. Fan-in results. <http://swerl.tudelft.nl/bin/view/AMR/FanInAnalysisResults>.
- [11] Grigoreta Sofia Moldovan and Gabriela Serban. Clustering Based Aspect Mining Formalized. *WSEAS Transactions on Computers*, 6(2):199–206, 2007.
- [12] David L. Parnas. On the Criteria To Be Used in Decomposing Systems into Modules. *Communications of the ACM*, 15(12):1053–1058, December 1972.
- [13] Gabriela Serban and Grigoreta Sofia Cojocar. A New Hierarchical Agglomerative Clustering Algorithm in Aspect Mining. In *Proceedings of 3rd Balkan Conference in Informatics (BCI'2007)*, pages 143–152, Sofia, Bulgaria, September, 27-29 2007.
- [14] Gabriela Serban and Grigoreta Sofia Moldovan. A New k-means Based Clustering Algorithm in Aspect Mining. In *Proceedings of 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'06)*, pages 69–74, Timisoara, Romania, September, 26-29 2006. IEEE Computer Society.
- [15] David Shepherd and Lori Pollock. Interfaces, Aspects, and Views. In *Proceedings of Linking Aspect Technology and Evolution Workshop(LATE 2005)*, March 2005.

BABEȘ-BOLYAI UNIVERSITY, DEPARTMENT OF COMPUTER SCIENCE, 1, M. KOGALNICEANU STREET, CLUJ-NAPOCA, ROMANIA

E-mail address: {grigo, gabis, istvanc}@cs.ubbcluj.ro

UNCONVENTIONAL COMPUTING: A SHORT INTRODUCTION

MIHAI OLTEAN

ABSTRACT. Physics imposes some limits to the computations that we can perform. Because of these limits we cannot solve problems in almost no time and with almost zero energy consumption. The main problem is that we are asking too much from a single device: we want computers that play chess, solve equations, navigate on the internet etc. Because of this great generality we are paying some big prices (in time and energy consumption). Some problems can be efficiently solved by using other concepts. Some materials have intrinsic properties which makes them very suitable for solving some kind of problems. In this paper we provide a short survey to the field of Unconventional Computing. We review and compare the most popular problems and methods belonging to this field.

1. INTRODUCTION

The quest for problem solvers has started since the humans have appeared on Earth. It seems that there is something in the human nature which push us to create machines for replacing our work.

Maybe the most prominent effort in this direction are the computers - which try to replace to human intellectual effort. Standard computers are able to perform faster computations than the human brain (for some problems). Take for instance the multiplication of 2 numbers. A computer can do this operation with several orders of magnitude faster and better than humans.

However, standard computer have a lot of limitations too. If you show them a picture they are not able to tell you which person is an old-friend of yours. They also cannot speak very well. They have difficulties in recognizing spoken words. These are just some of the limitations that a computer has.

Fortunately standard (digital) computers are just a part of the set of the possible computation devices. Recent years have shown a growing interest in

Received by the editors: April 2, 2009.

2000 *Mathematics Subject Classification*. 68Q05, 68Q10, 68Q17, 68Q99, 68W10, 81-xx, 78Axx.

1998 *CR Categories and Descriptors*. F [**Theory of Computation**]: General; F.1.2 [**Modes of Computation**]: Parallelism and concurrency.

Key words and phrases. unconventional computing, natural computing, survey.

Unconventional Computing (UC) devices. These types of computers are called unconventional mainly because they have been only recently invented, operate with some exotic principles, and because they have not been yet introduced on the market.

In this paper we make a short survey of the unconventional computation field. We have mainly focused our attention on NP-complete [13] problems only because these are the most difficult problems for standard computer. There is no digital computer which can solve NP-complete problems in polynomial time. Some researchers hope that unconventional computers are capable of solving NP-complete problems in polynomial time. There are already some examples which support this claim.

Another direction is given by some problems for which unconventional devices can perform much better (in both running time and energy consumption) than digital computers. Recently [29] some basic problems (such as sorting) have been shown to be solved in $O(1)$ time on an unconventional device. With such improvements over more basic problems we can get important benefits (especially in energy consumption).

The paper is structured as follows: We start in section 2 by defining what computing (both conventional and unconventional) is. The relationship between Natural Computing, Artificial Intelligence and Unconventional Computing is described in sections 2.1 and 2.2. The motivation for studying and researching unconventional computing is given in section 3. Problems that we want to solve are briefly described in section 4 and 5. Section 6 investigates if UC can solve NP-complete problems in polynomial time. The question if analog devices are better in all cases than their digital counterparts is negatively answered in section 7. Whether physical implementation is possible is discussed in 8. A table showing a comparison between UC devices is displayed in section 9. Advantages and weaknesses of the unconventional methods are discussed in 10. The main journals and conferences in the UC community are given in section 11. Finally, section 12 concludes our paper.

2. CONVENTIONAL AND UNCONVENTIONAL COMPUTING

Before starting our talk about unconventional computing we have to see what computing (either conventional or unconventional) means. Most people believe that computing is performed only by standard computers such as desktops or laptops. This is completely incorrect.

All objects around us perform some kind of computing. Imagine a box full with sand. One might say that it is not a computing device because it does not have a standard configuration like we know it: it does not have processor, memory, screen, keyboard etc. But, this is wrong, because even without all these components one might still perform some great computations. Another

one might say that a glass of water is unable to perform computations. This is again completely wrong.

Definition. *A computer is any physical object that can be reconfigured to solve multiple problems, that is, to answer many different questions.* [21]

If the device cannot be reconfigured it is usually called experiment [21].

Having this said we can define unconventional computing as being computing without standard digital computers. We use DNA, quantum properties, light, water, chemical substances, mechanical devices, nano-technologies, etc for performing computations, but not standard computers.

2.1. Natural computing vs. unconventional computing. There is a strong relationship between Natural Computing (NC) and Unconventional Computing. Natural computing is a large field contain all possible techniques and devices using principles inspired from nature or using nature's materials (exception being made for standard computers which - even if are made from nature's materials - are not considered unconventional).

The following facts differentiate UC from NC:

- Unconventional computing is a subset of natural computing.
- Unconventional computing is mainly focused on physical devices whereas NC includes algorithms too (such as Genetic Algorithms [17], Particle Swarm Optimization, etc),
- Unconventional computing also includes algorithms, but these algorithms are specially designed for unconventional machines (for instance algorithms for solving Hamiltonian Path Problem on a DNA computer [2], Integer factorization on a Quantum computer [31], Steiner tree on a machine with soap bubbles [6] etc) .

2.2. Artificial Intelligence and Unconventional Computing. There is also a strong relationship between Unconventional Computing and Artificial Intelligence (AI). This connection is given by one of the research direction in AI. Actually, the research in the field of AI is divided into 2 main directions:

- Weak AI - whose purpose is to develop some intelligent algorithms for solving some particular problems. Weak AI can run on standard computers. Weak AI is everywhere. Neural networks [36], Evolutionary Algorithms [17], Fuzzy Systems [35] are all techniques belonging to weak AI field. These methods help us to solve some problems, but they cannot operate without human control.
- Strong AI - whose purpose is to develop universal intelligence capable to match human performances. Currently there is no computer capable of supporting strong AI. The computer's architecture is too different from the requirements of strong AI paradigm. This is why is widely accepted that a new, unconventional architecture is required

for achieving this level of intelligence. (For instance human brains are unconventional devices running strong AI algorithms.)

3. MOTIVATION

Currently no unconventional device can replace standard computers because the technology is at beginning. More work is needed on all directions. However, in the near future we could expect to see special devices operated by principles totally different from those of standard computers.

There are a large number of reasons for which UC field requires a special attention. Here are some of them:

- Standard computers have a limit. It will not be long until we cannot fit more transistors on a square unit because we cannot decrease the size of the components forever. At that moment of time we have to search for alternate modes for performing computations. Some of these methods could be the unconventional methods of today.
- Standard computers are too slow for some problems. For instance sorting is a critical operation inside most of the programs, but no general algorithms operating in less than $O(n \log_2(n))$ is know for this problem. However, Rainbow sort [29] can do the sorting in $O(1)$, which is unimaginable fast.
- Standard computers consume too much energy. In California, the second consumer of electrical energy are the computers (first one being electrical bulbs) [7]. Something must be done here too because in the near future the number of computers will increase several times worldwide. Note that with one liter of DNA we can perform much more computations that all computers in the world have performed so far. The reduction in energy usage is significant in this case.
- The discovery of a new algorithm requires a huge work. In some cases it can take years to develop a good algorithm. The nature has solved a lot of problems with its own algorithms. (for instance problems with requires sorting or the computation of the shortest path). Taking these natural algorithms as the source of inspiration could lead to a faster development of practical algorithms for today's problems.

4. PROBLEMS TO SOLVE

There are 2 main categories of problems that we want to solve with unconventional devices.

- Those which are difficult for standard computers. I am refereeing here mainly to NP-complete problems [13] because most of these problems

are very practical. More discussion over the NP-complete problems is given in section 5.

- We are also interested in some devices offer direct advantages over digital computers for some particular problems. This aspect is only interesting if the benefits are huge (see for instance Rainbow sort [29] which does sorting in $O(1)$).

5. NP-COMPLETE PROBLEMS

The main category of problems that we want to solve with UC devices are NP-complete problems [13]. Unfortunately no polynomial-time is known for them. Nor the exponential algorithms has been proved to be optimal. This field lives in a great uncertainty.

This is why the Clay Mathematics Institute offers a 1 million prize for a solution to the $P=?NP$ problem. Even if the answer is negative we still have some great benefits: a big number of intelligent people will focus their attention on other problems.

The NP field was initiated by Stephen Cook which has made one of the most important contribution to computer science [8]. Since then the field has grown exponentially.

5.1. Formal definition. A problem C is said to be NP-complete if:

- Any given solution to the problem can be verified quickly (in polynomial time). The set of problems with this property is called NP.
- Every problem in NP is reducible to C in polynomial time.

Another well-know category is composed of NP-hard problems for which the second condition holds, but not necessarily the first one.

Some examples of NP-complete problems together with some practical applications are:

- Travelling Salesman Problem - with applications to path planning,
- Scheduling - with applications to machines and tasks,
- Subset sum - with applications to cutting and packing.

5.2. How do we solve NP-complete problems ? There are 3 major ways in which NP-complete problems can be solved:

- Brute force - which takes an exponential time because the size of the search space increases exponentially with the problem size.
- Heuristics - which are not optimal for all cases and even worse : require human intelligence.
- Unconventional devices - this is the aspect that we investigate in this paper.

5.3. What benefits we get if we solve NP-complete problems? Eloquent for this question is the answer given in 1956 by Gödel to John von Neumann. We reproduce here this well-known phrase:

*If there actually were a machine with [running time] $K * n$ (or even only with $K * n^2$), this would have consequences of the greatest magnitude. That is to say, it would clearly indicate that, despite the unsolvability of the Entscheidungsproblem, the mental effort of the mathematician could be completely (apart from the postulation of axioms) replaced by machines.*

5.4. Do we have to solve all NP-Complete problems? The answer is simple: NO. One problem is enough because there is a polynomial time reduction between them [13]. Thus is enough to find a solution to TSP and all other problems all solved instantly.

6. CAN UNCONVENTIONAL COMPUTERS SOLVE NP-COMPLETE PROBLEMS IN POLYNOMIAL TIME ?

This is a critical question. We already have evidence on small instances (see the Steiner tree with Soap Bubbles [6]). When solving larger instances of the Steiner tree problem (with soap bubbles) we get a lot of errors. Advocates of this method say that errors are due to imperfect experimental conditions.

Anyway, it is difficult to derive general statements because it is difficult to analyze the complexity of UC devices.

Some say that this is the perpetuum mobile of the modern times and what we should expect is to get solutions for particular problems.

7. CAN ANALOG COMPUTERS DO BETTER THAN DIGITAL COMPUTERS IN ALL CASES?

Unfortunately the answer to this question is negative. One major problem is related to the digital to analog conversion [33]. All unconventional devices are analog, thus this conversion is unavoidable if we want to make a fair comparison between conventional and unconventional computers.

Take for instance a simple problem: *Compare 2 numbers A_1, A_2 represented over n bits.*

One possible way to solve this problem in an analog is to create 2 analog objects having masses A_1 and A_2 . Because we need to represent any mass between 0 and $2^n - 1$ we will need an exponential (in n) quantity of matter .

For these kinds of problems the standard (digital) computers will always perform better than their analog counterparts.

8. PHYSICAL IMPLEMENTATION

A physical implementation is required for a better understanding of the mechanisms behind unconventional devices. For a computer scientist this could be very difficult because most of the phenomena not visible at macro-scale (take for instance a tube with DNA strings !).

Most of the UC methods requires a lot of equipments whereas mostly are expensive (for optical, DNA or quantum computers). Very few methods can be implemented with home-made tools (soap bubbles, linear programming machine etc).

Most of the work is done at theoretical level plus some simulations. Complicated experiments are rarely repeated (take as an example the Adleman experiment).

9. COMPARING VARIOUS UC DEVICES

Comparing some poorly understood methods is a difficult task. In this section (see Table 1) we make a comparison based on the following aspects:

- Speed - how fast the solution of the problem is obtained.
- Size - the quantity of materials involved in the experiments. Note that the size of most devices is exponential (see section 7 and paper [33]).
- Price - the total cost of the materials involved in the experiments. Since prices can vary depending on the producer we have used some general labels: low - for few dollars, and high for more than hundreds or thousands dollars. The price includes the equipment for reading the output which can be sometimes the most expensive part (see for instance Rainbow sort [29] where sorting is very cheap, but reading the output is very expensive).
- Know how - how much knowledge is required for building such device.
- Number of problems - the number of problems that can be solved by the device.
- Approx. solution - if the device generates an approximate solution (e.g. has a heuristic behaviors) or an exact solution.

10. ADVANTAGES AND WEAKNESSES

Unconventional methods have a huge number of advantages over the standard methods, but also have some weaknesses which is the main reason for which they cannot be seen yet in real-world applications.

Some of the advantages are:

- Parallel Computing - can solve problems in parallel, exploring multiple solutions in the same time.

TABLE 1. A comparison of the major unconventional computing paradigms.

Method	Speed	Size	Price	Know how	Number of problems	Approx solution
DNA computing [2, 27, 34]	Constant	Exponential (see [20])	High	High	Many	NO
Quantum computing [10, 31, 32]	sub-exponential (hopefully)	Polynomial	High	High	Many	NO
Optical devices [19, 18, 24, 25, 28, 30, 37, 38]	Exponential (see HPP)	Exponential (see HPP)	High	High	Many	NO
Bubble soap [1, 6]	No known	Polynomial	Low	Low	1 (Steiner tree)	YES
Rainbow sort [29]	Very fast	Exponential	High	Low	1 (sorting)	NO
Spaghetti sort [14]	Polynomial	Exponential	Low	Low	1 (sorting)	NO
Bead sort [3, 12]	Polynomial	Exponential	Low	Low	1 (sorting)	NO
Protein folding machine [5, 15, 21, 22, 26]	Not known	Polynomial	High	High	1 (protein folding)	Not known
Smart glass [23]	Polynomial	Exponential	Medium	Medium	Multiple	NO
Time travel computing [1]	Instant	Polynomial	Not known	Not known	Multiple	NO
Linear programming machine [33]	Exponential	Exponential	Low	Low	1 (linear programming)	NO

- Light weight - in some cases the UC methods operate at nanoscale level.
- Low power - working on nanoscale means a very low power consumption.
- Solves Complex Problems quickly - in some cases the solution is found in an extremely small time (see Rainbow sort [29]).

Major weaknesses are:

10.1. Weaknesses.

- Preprocessing time - A lot of time is spent for preparing the input for the unconventional device.
- Sometimes slower - Simple problems are solved much faster on electronic computers.
- Reading the output - It can take longer to read the answer to a problem than it takes to solve the problem itself. Thus the advantage of speed is virtually eliminated.
- Reliability - can have errors (see DNA computing). These errors require extra corrections or too advanced tools which makes the computation impractical.

11. UNCONVENTIONAL COMPUTING COMMUNITY

The UC community is growing every year. More and more papers are published on UC topics. The technology is improving, thus allowing us to perform larger and larger experiments with an increasing precision. In the next sections we give a short (but significant) list of journals and conferences focused on UC topics.

11.1. Journals. Main journals publishing papers in the field of unconventional computing are given in Table 2. Other journals focused on theoretical computing or on physics, chemistry, biology can also publish papers about UC topics.

11.2. Conferences. There is a huge number of conferences and workshops on Natural Computing topics. All of them accept papers about UC paradigms. Also, most of the Artificial Intelligence conferences accept papers about unconventional computing machines due to the strong connection between AI and UC (see sectionaiuc). It is impossible to list here all these conferences.

However, the leading conference in this field is Unconventional Computation which was started in 1998.

TABLE 2. A short list of journals publishing papers on Unconventional Computing.

Journal	Publisher	Starting year	Issues / year
Natural Computing	Springer	2002	4
New Generation Computing	Springer	1982	4
International Journal of Unconventional Computing	Old City Publishers	2005	4
Theoretical Computer Science	Elsevier	1975	61 (in 2008)
Fundamenta Informaticae	IOS Press / EATCS	1977	20
Journal of Universal Computer Science	Graz University of Technology and Universiti Malaysia Sarawak	1994	22 (in 2008)
International Journal of Foundations of Computer Science	World Scientific	1990	6
New Mathematics and Natural Computing	World Scientific	2005	3

12. CONCLUSIONS AND FURTHER WORK

In this paper we have investigated some of the most important analog devices capable of performing computations. Future work will be focused on:

- Comparing the proposed paradigms from a practical perspective.
- Checking the current limits of the compared methods. These limits can be advanced once the technology improves.
- Finding killer applications [9] for each of the paradigms. A killer application proves the value of the technology and pushes for more development of the corresponding technique. An example of killer application is *VisiCalc* for Apple II, which has generated huge sales for those platforms. Currently, for quantum computers we have a killer application: integer factorization [31] which runs in sub-exponential time. This is why big players (such as IBM and other private companies) have tried to implement quantum computers. For DNA computing a potential killer application has been proposed in [34]: a nano-scale robot which can fix diseases inside of a cell. For other paradigms more investigation should be performed in order to discover killer applications.

ACKNOWLEDGMENT

This work was supported from grant IDEI-543 from CNCSIS, Romania.

REFERENCES

- [1] S. Aaronson, *NP-complete Problems and Physical Reality*, arXiv : quant - ph0502072v2, 2005
- [2] L. Adleman, *Molecular computation of solutions to combinatorial problems*, Science, 266, (1994), 1021-1024
- [3] J. J. Arulanandham, C. S. Calude, M. J. Dinneen, *Bead-Sort: A natural sorting algorithm*, EATCS Bull, 76 (2002), 153-162
- [4] C. H. Bennett, *Logical reversibility of computation*, IBM Journal of Research and Development, 17:6 (1973), pp. 525-532.
- [5] B. Berger, T. Leighton, *Protein folding in the hydrophobic-hydrophilic (HP) is NP-complete*, Proceedings of the second annual international conference on Computational molecular biology, (1998) pp. 30-39.
- [6] S. Bringsjord, J. Taylor, *P=NP*, arXiv:cs.CC/0406056 v1, 2004
- [7] H. J. Caulfield, *Optics Goes Where No Electronics Can Go: Zero-Energy-Dissipation Logic*, in Optical SuperComputing Workshop, S. Dolev, T. Haist, M. Oltean (Eds.), LNCS 5172 (2008) pp. 1-8.
- [8] S.A. Cook, *The complexity of theorem proving procedures*, Proceedings, Third Annual ACM Symposium on the Theory of Computing, ACM, New York, (1971) pp. 151-158,
- [9] P. Dasgupta, *Killer Applications (overview)*, Arizona State University in Tempe, webpage: <http://cactus.eas.asu.edu/partha/Columns/2002/07-01-killer-app.htm>, last accessed on April 16, 2009.
- [10] D. David, *Quantum Computation*, Science 270 (1995), pp. 255-261.
- [11] Iwamura, H., Akazawa, M., Amemiya, Y. (1998), *Single-electron majority logic circuits*, IEICE Trans. Electronics E81-C(1), (1998) pp. 42-48.
- [12] A. J. Jeyasoorian, *Ball Sort - A natural algorithm for sorting*, Sysreader (1995), pp. 13-16.
- [13] M. Garey, D. Johnson, *Computers and Intractability*, W.H. Freeman, New York, NY, 1979.
- [14] A. K. Dewdney, *On the spaghetti computer and other analog gadgets for problem solving*, Scientific American 250:6 (1984), pp. 19-26
- [15] A.S., Fraenkel, *Protein folding, spin glass and computational complexity*, Theoretical Computer Science, 48 (1999), pp. 101-121.
- [16] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* New York: W.H. Freeman, 1979.
- [17] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [18] J.W. Goodman, *Architectural development of optical data processing systems*, Aust. J. Electr. Electron. Eng. 2 (1982), pp. 139-149
- [19] T. Haist, W. Osten, *An Optical Solution For The Traveling Salesman Problem*, Optics Express, 15, (2007), pp. 10473-10482
- [20] J. Hartmanis, *On the weight of computations*, Bulletin of the EATCS 55 (1995), pp. 136-138
- [21] J. W. Mills, *Introduction to the EAC with Protein Folding Example*, PPT presentation, accessed 4 April 2009

- [22] J. T. Ngo, J. Marks, *Computational complexity of a problem in molecular-structure prediction*, Protein Engineering, 5:4 (1992), pp. 313-321.
- [23] M. Oltean, *Switchable Glass: A possible medium for Evolvable Hardware*, NASA conference on Adaptive Hardware Systems, Stoica A., et al. (Eds), IEEE CS Press (2006), pp. 81-87.
- [24] M. Oltean, *A light-based device for solving the Hamiltonian path problem*, Unconventional Computing, Calude C. (et al.) (Eds), LNCS 4135, Springer-Verlag (2006), pp. 217-227
- [25] M. Oltean, *Solving the Hamiltonian path problem with a light-based computer*, Natural Computing, Springer-Verlag, 8:1 (2008), pp. 57-70.
- [26] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, M. Yannakakis, *On the complexity of protein folding*, Journal of Computational Biology, 5:3 (1998), pp. 423-65.
- [27] R. S. Braich, N. Chelyapov, C. Johnson, P. W. K. Rothemund, L. Adleman, *Solution of a 20-Variable 3-SAT Problem on a DNA Computer*, Science, 296 (2002), 499-502.
- [28] J.H. Reif, A. Tyagi, *Efficient parallel algorithms for optical computing with the discrete Fourier transform primitive* Applied optics 36(29) (1997), pp. 7327-7340.
- [29] D. Schultes, *Rainbow Sort: Sorting at the speed of light*, Natural Computing, Springer-Verlag, 5(1) (2005), pp. 67-82.
- [30] N.T. Shaked, S. Messika, S. Dolev, J Rosen, *Optical solution for bounded NP-complete problems*, Applied Optics, 46 (2007), pp. 711-724.
- [31] P. Shor, *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, IEEE Symposium on Foundations of Computer Science, IEEE Press (1994), pp. 124-134.
- [32] M. Steffen, M.K. Lieven, I.L. Vandersypen Chuang, *Toward quantum computation: a five-qubit quantum processor* IEEE Micro, 21 (2001) pp. 24-34.
- [33] A. Vergis, K. Steiglitz, B. Dickinson, *The complexity of analog computation* Mathematics and Computers in Simulation, 28 (1986), pp. 91-113.
- [34] Y. Benenson, B. Gi, U. Ben-Dor, R. Adar, E. Shapiro, *An autonomous molecular computer for logical control of gene expression*, Nature 429 2004, pp. 423-429.
- [35] L. Zadeh, *Fuzzy sets and systems* In: Fox J, editor. System Theory. Brooklyn, NY: Polytechnic Press, (1965), pp. 29-39.
- [36] J. Zurada, *Introduction to artificial neural systems*, West Publishing Co. St. Paul, MN, USA, 1992.
- [37] D. Woods, T.J. Naughton, *An optical model of computation* Theoretical Computer Science, 334(1-3) (2005), pp. 227-258
- [38] *Thoughts on the Subset Sum Problem (P vs. NP)* [http : //idrone.net/2006/06/11/thoughts-on-the-subset-sum-problem-p-vs-np/](http://idrone.net/2006/06/11/thoughts-on-the-subset-sum-problem-p-vs-np/) (accessed 2006)

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF MATHEMATICS AND COMPUTER SCIENCE, BABEȘ-BOLYAI UNIVERSITY, KOGĂLNICEANU 1, CLUJ-NAPOCA, 400084, ROMANIA

E-mail address: moltean@cs.ubbcluj.ro

A MULTIOBJECTIVE METAHEURISTIC FOR JOB-SHOP SCHEDULING

CRINA GROȘAN

ABSTRACT. In this paper, we introduce a nature inspired meta-heuristic for scheduling jobs on computational grids. Our approach is to dynamically generate an optimal schedule so as to complete the tasks in a minimum period of time as well as utilizing the resources in an efficient way. The approach proposed is a variant of particle swarm optimization which uses mutation operator. The mutation operator can affect both particle's personal best and the swarm's global best. The experiments performed show the efficiency of the proposed approach over the standard PSO and other metaheuristics considered (namely genetic algorithms and simulated annealing).

1. INTRODUCTION

Grid Computing (GC) is the ultimate framework to meet the growing computational demands in the new millennium [2], [3], [4], [5], [6], [7]. To meet the growing needs of the computational power, geographically distributed resources need to be logically coupled together to make them work as a unified resource. Computing resources are geographically distributed under different ownerships each having their own access policy, cost and various constraints. Every resource owners will have a unique way of managing and scheduling resources and the grid schedulers are to ensure that they do not conflict with resource owner's policies.

In a grid environment knowing the processing speeds of the available resources and the job length of user applications is a tedious task. Usually it is easy to get information about the speed of the available resources but quite complicated to know the computational processing time requirements from the user. When the computing power demand is much greater than the available resources only dynamic scheduling will be useful. To conceptualize the problem as an algorithm, we need to dynamically estimate the job lengths from user application specifications or historical data. Soft computing techniques like fuzzy logic, evolutionary computation and artificial neural networks

Received by the editors: April 3, 2009.

might be of useful aid in the parameters estimation process especially in times of uncertainty and vague data.

The paper is organized as follows. Section 2 deals with some theoretical foundations related to job scheduling. A variation of the particle swarm optimization heuristic is introduced in Section 3. In Section 4, experiment results and discussions are provided. Finally, we conclude our work.

2. SCHEDULING PROBLEM FORMULATION

To formulate the problem, we consider J_n independent user jobs $n=1,2,\dots,N$ on R_m heterogeneous resources $m=1,2,\dots,M$ with an objective of minimizing the completion time and utilizing the resources effectively. The speed of each resource is expressed in number of cycles per unit time, and the length of each job in number of cycles. Each job J_n has processing requirement P_j cycles and resource R_m has speed of S_i cycles/second. Any job J_n has to be processed in resource R_m , until completion.

To formulate our objective, define C_j as the completion time the last job j finishes processing.

Define:

$$C_{max} = \max\{C_j, j = 1, \dots, N\},$$

the makespan and $\sum C_j$ as the flowtime.

An optimal schedule will be the one that optimizes the flowtime and makespan [8]. The conceptually obvious rule to minimize $\sum C_j$ is to schedule Shortest Job on the Fastest Resource (SJFR). The simplest rule to minimize C_{max} is to schedule the Longest Job on the Fastest Resource (LJFR). Minimizing $\sum C_j$ asks the average job finishes quickly, at the expense of the largest job taking a long time, whereas minimizing C_{max} , asks that no job takes too long, at the expense of most jobs taking a long time. In summary, minimization of C_{max} will result in maximization of $\sum C_j$.

Several optimization criteria can be considered for this problem, certainly the problem is multiobjective. The fundamental criterion is that of minimizing the makespan, that is, the time when finishes the latest task. A secondary criterion is to minimize the flowtime of the grid system that is, minimizing the sum of finalization times of all the tasks:

- minimization of makespan;
- minimization of flowtime.

The most common approaches of a multiobjective optimization problem use the concept of Pareto dominance as defined below:

Definition (Pareto dominance) Consider a maximization problem. Let x, y be two decision vectors (solutions) from the definition domain. Solution x dominate y if and only if the following conditions are fulfilled:

- (i) $f_i(x) \geq f_i(y); i = 1, 2, \dots, n;$
- (ii) $\exists j \in 1, 2, \dots, n : f_j(x) > f_j(y).$

That is, a feasible vector x is Pareto optimal if no feasible vector y can increase some criterion without causing a simultaneous decrease in at least one other criterion.

3. PROPOSED PARTICLE SWARM MODEL FOR JOB SCHEDULING (PSJS)

The classical particle swarm model consists of a swarm of particles, which are initialized with a population of random candidate solutions. They move iteratively through the d -dimension problem space to search the new solutions, where the fitness, f , can be calculated as the certain qualities measure.

For a d -dimensional search space the position of the i -th particle is represented as: $X_i = (x_{i1}, \dots, x_{id}).$

Each particle maintains a memory of its previous best position $P_{best.i} = (p_{i1}, p_{i2}, \dots, p_{id}).$

The best one among all the particles in the population is represented as

$$P_{gbest} = (p_{g1}, p_{g2}, \dots, p_{gd}).$$

The velocity of each particle is represented as: $V_i = (v_{i1}, v_{i2}, \dots, v_{id}).$

In each iteration, the P vector of the particle with best fitness in the local neighborhood, designated g , and the P vector of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity. Equations of velocity vector and position vector given by:

$$(1) \quad v_{id} = w \cdot v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})$$

$$(2) \quad x_{id} = x_{id} + v_{id}.$$

First part of equation 1 represents the inertia of the previous velocity, second part is the cognition part, third part represents the cooperation among particles and is called social component. Acceleration constants c_1, c_2 and inertia weight w are the predefined by the user. r_1, r_2 are the uniformly generated random numbers between 0 and 1.

Even tough PSO is known as a good optimization technique, the method still lacks in several aspects and require further improvements in order to performed a better convergence. In this paper we propose the usage of mutation

which can affect both the global best particle as well as the best known position of each particle.

The personal best position of each particle is mutated at the end of each iteration while the global best position is mutated only after a couple of iterations (this is a parameter of the algorithm).

4. EXPERIMENTS

We performed two experiments. Results obtained by PSJS are compared to the ones obtained by Genetic Algorithm (GA), Simulated Annealing (SA) and the standard multiobjective Particle Swarm Optimization (PSO). We should mention that in the case of SA and GA the objectives are aggregated using weighted sum method while in the case of both particle swarm approaches we used Pareto dominance.

Specific parameter settings of all the considered algorithms are described in Table 1.

Each experiment (for each algorithm except for the MOEA) was repeated 10 times with different random seeds. Each trial (except for MOEA) had a fixed number of $50 \cdot m \cdot n$ iterations (m is the number of the grid nodes, n is the number of the jobs). The makespan values of the best solutions throughout the optimization run were recorded. In a grid environment, the main emphasis was to generate the schedules as fast as possible. So the completion time for 10 trials was used as one of the criteria to improve their performance.

First we tested a small scale job scheduling problem involving 3 nodes and 13 jobs represented as (3,13). The node speeds of the 3 nodes are 4, 3, 2 CPUT, and the job length of 13 jobs are 6,12,16,20,24,28,30,36,40,42,48,52,60 cycles, respectively.

The results (makespan) for 10 runs were as follows:

- GA: 47, 46, 47, 47.3333, 46, 47, 47, 47, 47.3333, 49, with an average value of 47.1167.
- SA: 46.5, 46.5, 46, 46, 46, 46.6667, 47, 47.3333, 47, 47 with an average value of 46.6.
- PSO : 46, 46, 46, 46, 46.5, 46.5, 46.5, 46, 46.5, 46.6667, with an average value of 46.2667.
- PSJS: 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, with an average value of 46.

Further, we tested the algorithms for the case (10, 50). All the jobs and the nodes were submitted at one time. The average makespan values for 10

TABLE 1. Parameters used by the algorithms considered in experiments

Algorithm	Parameter	Value
GA	Population size	20
	Crossover probability	0.8
	Mutation probability	0.02
	Scale for mutations	0.1
SA	Number operations before temperature adjustment	20
	Number of cycles	10
	SA Temperature reduction factor	0.85
	Vector for control step of length adjustment	2
	Initial temperature	50
PSO	Swarm size	20
	PSO Self-recognition coefficient c_1	1.49
	Social coefficient c_2	1.49
	Inertia weight w	0.9 \rightarrow 0.1
PSJS	Swarm size	20
	PSO Self-recognition coefficient c_1	1.49
	Social coefficient c_2	1.49
	Inertia weight w	0.9 \rightarrow 0.1
	Number of iterations for a global best mutation	10

trials are illustrated in Table 2. Although the average makespan value of SA was better than that of GA for (3,13), the case was reversed for this second case.

Results obtained by GA, SA and PSO are taken from [1]. As evident from the data obtained above, MOEA results clearly outperform results obtained by the other techniques considering a single objective approach.

TABLE 2. Performance comparison for the case (10, 50).

Algorithm	Average makespan
GA	38.04
SA	41.78
PSO	37.66
PSJS	36.7

5. CONCLUSIONS

In this paper a new variant of multiobjective PSO which uses mutation operator affection both personal best of each particle and the swarm's global

best is proposed for the scheduling problem in grid computing. Based on the experimental results presented we can conclude that PSJS performs better than standard PSO and it also obtains better results than the approaches which aggregate the objective functions and consider the problem as single objective instead of using Pareto dominance.

ACKNOWLEDGEMENT

The author acknowledges the support from the CNCSIS Grant IDEI 2412/2008.

REFERENCES

- [1] Abraham A, Liu H, Zhang W, Chang TG, Scheduling Jobs on Computational Grids Using Fuzzy Particle Swarm Algorithm, Proceedings of 10th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, England, pp. 500-507, 2006.
- [2] Abraham A, Liu H, Grosan C, Khafa F, Nature Inspired Metaheuristics for Grid Scheduling: Single and Multiobjective Optimization Approaches, Metaheuristics for Scheduling: Distributed Computing Environments, Studies in Computational Intelligence, Springer Verlag, Germany, ISBN: 978-3-540-69260-7, pp. 247-272, 2008.
- [3] Baker M, Buyya R, Laforenza D, The Grid: International Efforts in Global Computing, International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, Rome, Italy, 2000.
- [4] Buyya R, Abramson D, Giddy J, Grid Resource Management, Scheduling, and Computational Economy, International Workshop on Global and Cluster Computing, Japan, 2000.
- [5] Foster I, Kesselmann C, (Eds.), The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, USA, 1999.
- [6] Liu H, Abraham A, An Hybrid Fuzzy Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems, Journal of Universal Computer Science, Volume 13, No. 7, pp. 1032-1054, 2007.
- [7] Pant M, Thangaraj R, Abraham A, Particle Swarm Optimization Using Adaptive Mutation, 2nd International Workshop on Evolutionary Techniques in Data-processing (DEXA'08/ETID '08), IEEE Computer Society Press, USA, ISBN 978-0-7695-3299-8, pp. 519-523, 2008.

DEPARTMENT OF COMPUTER SCIENCE, BABES-BOLYAI UNIVERSITY, KOGALNICEANU
1, CLUJ-NAPOCA 400084, ROMANIA

E-mail address: `cgrosan@cs.ubbcluj.ro`

FINGERPRINT MATCHING ON SMALL IMAGES

FLORINA MUNTENESCU

ABSTRACT. The uniqueness of the ridges' flow pattern is the basis of forensic applications on *fingerprints*, the accuracy of matching two or more fingerprints being its most important issue. A big problem arises when the fingerprint image used is small. In this paper we aim at introducing a method of analyzing a small image so that valid minutiae points can be extracted. For this purpose a constraint satisfaction based technique for finding a match between a candidate fingerprint with one from a database of fingerprints is introduced. We also provide experimental results and a comparison with other similar existing approaches.

1. INTRODUCTION

Fingerprints were used as a method of identifying a person for the first time in the 19th century when the classification methods based on patterns appeared and rapidly developed [2]. By 1946 FBI had processed 100 million manually maintained fingerprint cards and by 1971, 200 million cards, the necessity of an automated fingerprint identification system became obvious [2]. Nowadays fingerprints are not used only in forensic science but also as a biometric system of security in everything that requires a high accuracy rate: from devices that allow users to log in to Windows, to enter restricted areas or to authorize transactions [3]. The images used by FBI have 8 bits of grayscale resolution and 768 x 768 pixels, turning a single fingerprint card in 10 Mb of data.

In the approach from this study we are focusing on small images of 90 x 125 pixels, in grayscale. Small images are a solution when we don't have access to large images of a fingerprint and when large images, that are harder to process and also require more disk space for their storage, are not really needed. When using small images in extracting minutiae points [5] it is almost

Received by the editors: April 17, 2009.

2000 *Mathematics Subject Classification.* 68T10, 68U10.

1998 *CR Categories and Descriptors.* I.2.8 [**Computing methodologies**]: Artificial Intelligence – *Problem Solving, Control Methods, and Search*; I.5.0 [**Computing Methodologies**]: Pattern Recognition – *General*.

Key words and phrases. Pattern recognition, Fingerprint matching.

impossible to extract core and delta points, that is why we have focused on introducing elements that, to our knowledge, were not used in fingerprint matching, so far. More precisely the division of the image in quadrants, adding to the features of a minutiae point the quadrant to which it belongs was considered. Another challenging problem that we managed to overcome is to reliably extract minutiae in these small images eliminating many fake points.

The aim of this paper is to introduce small images as a trustworthy source for extracting minutiae points and also to propose a new technique in fingerprint matching based on a constraint satisfaction approach. The rest of the paper is structured as follows. Section 2 describes minutiae points, the way they are extracted from an image and also presents a few matching techniques existing in the literature. A new method of fingerprint matching is introduced in Section 3. Some experimental results are presented in Section 4 and some conclusions of the paper and further work are given in Section 5.

2. MINUTIAE POINTS EXTRACTION AND FINGERPRINT MATCHING TECHNIQUES

In this section we will give a definition for minutiae points and will present some fingerprint matching techniques existing in the literature.

When using a minutiae based algorithm [7] the fingerprint image is seen as a set of *minutiae points*, i.e., a set of *ridge endings* and *bifurcations* with uniquely determined orientation and location. Even though there are as many as eighteen types of ridge features defined, they can all be considered as complex combinations of two basic features: ridge end and bifurcation, referred together as minutiae points [6]. A ridge end is a point at which a ridge terminates. Bifurcation is a point where a single ridge splits into two ridges. A fingerprint can be seen as an xOy plane, therefore, a point in this plane can be seen in terms of Ox and Oy coordinates. Also an orientation of the ridge can be defined [1]. But to the usual features, we add a new one: the quadrant to which the point belongs. Consequently, we managed to create Equation (1) to define a minutiae point as a 5 tuple, and to illustrate it in Figure 1:

$$(1) \quad MP = \{(x, y, \theta, t, q) \mid x \in Ox, y \in Oy, \theta \in [0, 360), \\ t \in \{end, bifurcation\}, q \in \{1, 2, 3, 4\}\}$$

where:

- (x, y) are the coordinates of the point;
- (x, y) θ is the orientation angle;
- t is the type of the point: ridge end or bifurcation;
- q is the quadrant.

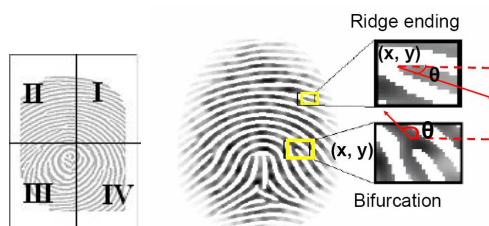


FIGURE 1. Left: A fingerprint divided in quadrants; Right: Minutiae points defined by (x,y) position, θ - orientation and type (ending or bifurcation)

Given two (an input and one from the database) representation, the matching technique determines whether the associated prints are impressions of the same finger. The matching phase typically defines a similarity between two fingerprint representations but finding a similarity function is usually very difficult.

There are various algorithms for fingerprint matching, each with its advantages and disadvantages, and with a more or less possibility of being applied for small images. Next, we are going to briefly describe three of these algorithms.

2.1. Point Based Matching. The minutiae point based techniques [5] typically match the input and the template minutia point by first aligning them and afterwards counting the number of matched minutiae. The alignment can be obtained using the orientation field of the fingerprints, the location of singular points as the core and the delta, inexact graph matching on the minutiae graphs etc. The inapplicability of this technique is given by the difficulty of finding the core and the delta points. These points are used to compute the amount of translation and also to easily compare the Euclidian distance to the first ridges that are in the vicinity of the points.

2.2. String Distance-based Minutiae Matching Algorithm. Each set of extracted minutiae features is converted to polar coordinates with respect to an anchor point [5]. The two-dimensional features are reduced to one-dimensional string by concatenating points in an increasing order of radial angle in polar coordinates. The string matching algorithm is applied to compute the edit distance between two strings. The edit distance is then used to compute a matching score.

2.3. Hough Transform-based Minutiae Matching. First the algorithm estimates the transformation parameters: translation on x and y directions, the rotating angle and a scaling factor. The two sets of minutiae points with

the estimated parameters are aligned and the matching pairs are counted. The previous two steps are repeated for the set of discretized allowed transformations. The transformation with the highest matching score is considered to be the correct one [5].

3. OUR APPROACH IN FINGERPRINT MATCHING

In this section we introduce an approach for fingerprint matching, describing first how we extract the minutiae points and next, how we manage to establish whether two impressions belong to the same finger.

Every image is, in fact, a collection of pixels, each containing three colors in different ratio, range over 0 to 255 RGB values, and where a 0 value of the RGB means that the pixel contains black and 255 white. The image has to be transformed into a matrix having the size of the image that would contain 1 for the pixels where there is a ridge (dark color) and 0 for those where is a valley (light color). So, the pixels that are closer to 0 in RGB are 1 and the rest of them 0. We have noticed that a very important part in this step towards minutiae points' extraction is the value of the threshold, because a value too large could lead to overlapping one or more ridges, loosing in such a way important points: ridges that end could be connected with continuous ones and a ridge that bifurcates in other two can be considered by the computer as just one thicker ridge. But, on the other hand, choosing a very small value for the threshold could have as consequences the appearances of ridge endings in places where the ridge is just thicker or where the finger pressure on the scanner was lighter, and the transformation of a bifurcation point to three ridge endings.

In order to extract the bifurcation points we parse the binary matrix in order to find a 3x3 pixels area whose sum is greater than 5 and the center of the area is a pixel with the value 1. The image being small, the angle of the minutiae can only be 0 or 180. To find the ridge endings we proceed like in the bifurcation points' case but we search for a sum equal with 2. The angles for end points can be: 0, 45, 90, 135, 180, 225, 270 or 315, depending on where the other pixel is placed, with respect to the center of the area.

So far, the algorithm appears to be simple due to the small image but when analyzing the points found, we realize that many of the minutiae points found are *fake*. Our algorithm detects in the area from the edge of the fingerprint a lot of points (both end and bifurcation) that are actually just the border of the fingerprint and those have to be eliminated. Another problem appears at the bifurcation points where the algorithm seems to find consecutive minutiae points around one valid point. So, the solution to these fake points is to delete

the sequences of consecutive points, keeping just the first one, which is the valid point.

When trying to find a match for an input fingerprint in a large database, computational time is an important issue. So, instead of applying a time consuming matching algorithm for each fingerprint, we propose a different approach: we first use an algorithm that would *filter* the fingerprints, keeping only those that have a good chance of being a possible match. The algorithm does not work on partial fingerprints.

Our problem has a number of constraints that are, one by one eliminated: number of points per quadrant, translation and rotation factors.

Based on experimental results we have noticed that the number of points from each quadrant is different: for quadrant one is between 9 and 25, for quadrant two between 10 and 50, for quadrant three: 1 to 15 points and for quadrant four 0 to 15. This means that the first step in matching two fingerprints is searching for possible matches in certain ranges, specific for each quadrant. Moreover, we have noticed that we can find a scope even for end and bifurcation points. Therefore, when we remain with a smaller number of fingerprints, hence number of minutiae points, we can 'afford' applying a matching algorithm.

The second step is finding the translation factors on O_x and O_y [4]. In order to do that, we search again in a certain range, the number of pixels (Euclidian distance) with which the image had translated, finding the most possible position to which the point had relocated. The higher number of points that have the same distance, the higher the probability of the fingerprint of being a match. When searching for the analogous points we search between points in the same quadrant, that have the same type and, in case of the bifurcation minutiae points, the same angle.

Knowing the distance, consequently, the translated corresponding points, the third step is to find between the ridge endings how many of them were rotated with the same angle. If the angles seem to vary a lot and a rotation cannot be found the two impressions surely do not match and the fingerprint from the database is removed from the search space. The previous two steps are repeated for each set of minutiae points and the fingerprint that has the highest score is considered to be the match.

4. EXPERIMENTAL RESULTS

The algorithms described in this paper were tested on fingerprint images that were obtained from notebook scanners alongside with artificially generated ones, resulting a database with almost 22,000 fingerprints and more than 1,300,000 minutiae points. The subjects were taken 2 impressions for each

finger, from which one was inserted in the database and the other one used for testing. Partial fingerprints were removed. At the first step, 10 to even 1 possible fingerprints were found. At the second filtering 3 to 1 possible fingerprints were found and at the last 1 possible fingerprint was found. The very small number of fingerprints found after the filtering step that we have proposed show the effectiveness of this preprocessing step applied before matching.

Errors were noticed to appear when the pressure on the scanner was not constant, resulting, especially in the upper one and two quadrants, a very large number of ridge endings that made the algorithm consider that the fingerprint does not have a match.

5. CONCLUSIONS AND FURTHER WORK

We have presented in this paper a fingerprint matching algorithm for small images that can be applied when time and space consuming techniques cannot be used. The capability of the minutiae extracting and matching was proved in experimental results and the benefits of using these algorithms on small images in comparison with others, were proved. The further work will be revolving around accepting partial fingerprints.

REFERENCES

- [1] Jidnya Shah Arun Ross and Anil K. Jain, *Towards reconstructing fingerprints from minutiae points*, Proc. Of SPIE Conference On Biometric Technology For Human Identification (2005).
- [2] Christopher M. Brislawn, *The fbi fingerprint image compression standard*.
- [3] A.K. Jain S. Prabhakar D. Maltoni D. Maio, *Introduction*, Handbook of fingerprint recognition, 2003, pp. 1–53.
- [4] Venu Govindaraju Praveer Mansukhani Sergey Tulyakov, *Using support vector machines to eliminate false minutiae matches during fingerprint verification*.
- [5] Nalini Ratha and Ruud Bolle, *Fingerprint matching*, Automatic fingerprint recognition systems, 2003, pp. 229–246.
- [6] Nalini K. Ratha Ruud M. Bolle Andrew W. Senior and Sharath Pankanti, *Fingerprint minutiae: A constructive definition*.
- [7] S. M. Zamshed Farhan S. M. Mohsen and M. M. A. Hashem, *Automated fingerprint recognition: Using minutiae matching technique for the large fingerprint database*, 3rd International Conference on Electrical and Computer Engineering (2004).

DEPARTMENT OF COMPUTER SCIENCE, BABEȘ-BOLYAI UNIVERSITY, 1, M. KOGĂLNICEANU STREET, CLUJ-NAPOCA, ROMANIA
E-mail address: mfie0602@scs.ubbcluj.ro