Special Issue:
Advances in Computer Science and Engineering

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 23**, Noviembre, 2006. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Imagen de la portada: Mural "60 años del IPN", Metro Politécnico. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela,* RFC *SOAJ560723*

Research in Computing Science *is published by the Center for Computing Research of IPN.* ***Volume 23****, November, 2006. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, November, 2006, in the IPN Graphic Workshop – Publication Office.*

Volume 23

Volumen 23

Special Issue:
Advances in Computer Science and Engineering

Volume Editors:
Editores del Volumen

Alexander Gelbukh
Sergio Suárez Guerra

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2006

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.ipn.mx
http://www.cic.ipn.mx

Printing: 500
Impresiones: 500

Printed in Mexico
Impreso en México

Table of Contents

Índice

Page/Pág.

Software Engineering

Cryptography and Security

Computer Networks

Educational Software

Control

Computer Architecture

Logic Programming

Preference Logic Programs using Answer Sets

Claudia Zepeda[1] and Mauricio Osorio[2]

[1] Universidad Politécnica de Puebla,
Tercer Carril del Ejido Serrano, San Mateo Cuanala,
Municipio Juan C. Bonilla, Puebla, 72640 México
czepedac@gmail.com,
[2] Universidad de las Américas, CENTIA,
Sta. Catarina Mártir, Cholula, Puebla, 72820 México
osoriomauri@gmail.com

Abstract We introduce preference rules which allow us to specify preferences as an ordering among the possible solutions of a problem. Specifically, we propose the semantics for preference logic programs. The formalism used to develop our work is Answer Set Programming(ASP). Most research on ASP and in particular about preferences in ASP supposes syntactically simple rules. So, our approach permits expressing preferences for general theories.
Keywords: Logic Programming, Answer Set Programming, Preferences.

1 Introduction

Preferences can be used to compare feasible solutions of a given problem, in order to establish if there is an order among these solutions or to establish whether such solutions are equivalents w.r.t. some requirements. Currently there are several approaches in non monotonic reasoning dealing with preferences [5]. In this paper we introduce *preference logic* (PL) programs which permit represent preferences and desires. The formalism used to develop our work is Answer Set Programming (ASP) [6]. ASP is a declarative knowledge representation and logic programming language. ASP represents a new paradigm for logic programming that allows us, using the concept of negation as failure, to handle problems with default knowledge and produce non-monotonic reasoning. Two popular software implementations to compute answer sets are DLV[1] and SMODELS[2]. The efficiency of such programs allowed to increase the list of practical applications in the areas of planning, logical agents and artificial intelligence.

Most research on ASP and in particular about preferences in ASP supposes syntactically simple rules (see for example [2,1,12]). This is justified since, most of the times, those restricted syntaxes are enough to represent a wide class of interesting and relevant problems. It could seem unnecessary to generalize the notion of answer sets to some more complicated formulas. However, a broader

[1] http://www.dbai.tuwien.ac.at/proj/dlv/
[2] http://www.tcs.hut.fi/Software/smodels/

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 3-14

Received 07/07/06
Accepted 07/10/06
Final version 13/10/06

syntax for rules could bring some benefits. For example, the use of nested expressions could simplify the task of writing logic programs and improve their readability. Hence, PL programs is an approach about preferences and desires to general theories.

A PL program is a set of well formed formulas joined to a set of preference rules. The set of well formed formulas of a PL program allows us to obtain the different answer sets representing the solutions of a problem. For instance, $\{ice\_cream \vee (coffee \wedge cake) \leftarrow .\}$ represent two solutions, *ice-cream* and *coffee with cake*. The set of preference rules of a PL program express the preferences and desires of somebody. Preference rules use a new connective, $*$, called *preference operator* to represent an ordering among the preference options. For instance, $ice\_cream * cake \overset{pr}{\leftarrow} .$ indicates preference for ice-cream over cakes. If a preference rule has only one preference option, then it represents a desire. For instance, $coffee \overset{pr}{\leftarrow} .$ indicates the desire of drinking coffee. The complete PL program is:

$$\begin{aligned} ice\_cream \vee (coffee \wedge cake) \leftarrow . \ &\%\text{To obtain the solutions} \\ ice\_cream * cake \overset{pr}{\leftarrow} . \quad &\%\text{a preference rule} \\ coffee \overset{pr}{\leftarrow} . \quad &\%\text{a desire} \end{aligned} \tag{1}$$

Currently, there are some answer set approaches that suggest a broader syntax [8,10,4]. In particular the authors of [4] describe an approach for preferences called Answer Set Optimization (ASO) programs. ASO programs have two parts. The generating program and the preference program. The first one produce answer sets representing the solutions, and the second one expresses user preferences. The body of a rule representing preferences is defined as a conjunction of literals, and its head is defined as an ordering among the preference options. The preference options are particular formulas called boolean combinations. A boolean combination is a formula built of atoms by means of disjunction, conjunction, strong and default negation, with the restriction that strong negation is allowed to appear only in front of atoms, and default negation only in front of literals.

At this point, we could think that PL programs and ASO programs [4] could be similar approaches to represent preferences. These idea could come from the fact that both approaches have two parts (one to get the answer sets and one to express preferences), and both approaches allow us a broader syntax to express preferences. However, ASO programs and PL programs differ considerably in syntax and semantics.

As we mentioned, the first difference is related to syntax. Specifically, the parts of both approaches to get the answer sets and their parts to express preferences have different syntax. ASO programs allow us to get the answer sets from any type of logic program (for instance, normal, extended, disjunctive, etc). PL programs allows us to get the answer sets from a set of well formed formulas. Additionally, the part of ASO programs to express preferences uses boolean combinations which have a restricted syntax. The part of PL programs to express preferences uses well formed formulas. So, PL programs allow us a

broader syntax than ASO programs. The idea of permitting a broader syntax of PL programs comes from [10]. In [10] the authors propose a broader syntax for logic programs with ordered disjunction (LPOD) and its semantics. LPOD's are introduced in [2].

The second difference between ASO programs and PL programs is their semantics. For instance, we shall see in Section 3 that $\{ice\_cream\}$ and $\{coffee, cake\}$ are the preferred answer sets of the PL program (1). However, if we represent the two parts of the PL program (1) as an ASO program then, we can verify that $\{ice\_cream\}$ is its only preferred answer set. Additionally, there is only one criterion to get the preferred answer sets from an ASO program; and there are three different criteria to get the preferred answer sets from a PL program. Finally, in [4] is not defined whether ASO programs can have preferences with only one option or not. PL programs allow us to have preferences with only one option and they are called desires. We want to mention that, the semantics of PL programs is inspired by the semantics of LPOD's introduced in [2], the extended semantics for LPOD's proposed in [10], and the work about preferences in [13]. It is worth mentioning that the authors of [13] indicate that their work arose from the necessity to model a real problem. The real problem is related to represent preferences about the evacuation plans in a risk zone.

Our paper is structured as follows. In Section 2 we introduce the general syntax of the logic programs used in this paper. We also provide the definition of answer sets in terms of logic G_3. In Section 3 we present the semantics for preference logic programs. In Section 4 we present how our approach is related to extended LPOD's. Finally in Section 5 we present some conclusions.

2 Background

In this section we review some fundamental concepts and definitions that will be used along this work. We introduce first the syntax of formulas and programs based on the language of propositional logic. We also present the definition of answer sets in terms of logic G_3.

2.1 Propositional Logic

In this paper, logic programs are understood as propositional theories. We shall use the language of propositional logic in the usual way, using propositional symbols: $p, q, \ldots$, propositional connectives $\wedge, \vee, \rightarrow, \bot$ and auxiliary symbols: $(,)$. The well formed propositional formula $f \leftarrow g$ is just another way of writing $g \rightarrow f$. We assume that for any well formed propositional formula f, $\neg f$ is just an abbreviation of $f \rightarrow \bot$ and $\top$ is an abbreviation of $\bot \rightarrow \bot$. We point out that $\neg$ is the only negation used in this work. An *atom* is a propositional symbol. A *literal* is either an atom a (a positive literal) or the negation of an atom $\neg a$ (a negative literal). A *negated literal* is the negation sign $\neg$ followed by any literal, i.e. $\neg a$ or $\neg\neg a$. In particular, $f \rightarrow \bot$ is called *constraint* and it is also denoted as $\leftarrow f$. Given a set of well formed formulas F, we define $\neg F = \{\neg f \mid f \in F\}$.

Sometimes we may use *not* instead of $\neg$ and a, b instead of $a \wedge b$, following the traditional notation of logic programming. A *regular theory* or *logic program* is just a finite set of well formed formulas or rules, it can be called just *theory* or *program* where no ambiguity arises. We shall define as a *rule* any well formed formula of the form: $f \leftarrow g$. The parts on the left and on the right of " $\leftarrow$ " are called the *head* and the *body* of the rule, respectively. The signature of a logic program P, denoted as $\mathcal{L}_P$, is the set of atoms that occur in P. We want to stress the fact that in our approach, a logic program is interpreted as a propositional theory. For readers not familiar with this approach, we recommend [11,9] for further reading. We will restrict our discussion to propositional programs.

2.2 The logic G_3

Some logics can be defined in terms of truth values and evaluation functions. Gödel defined the multivalued logics G_i, with i truth values. In particular, G_2 coincides with Classical Logic C. We briefly describe in the following lines the 3-valued logic G_3 since our work uses the logical characterization of answer sets based on this logic presented in [9]. Gödel defined the logic G_3, with 3 values, with the following evaluation function I:

- $I(B \leftarrow A) = 2$ if $I(A) \leq I(B)$ and $I(B)$ otherwise.
- $I(A \vee B) = \max(I(A), I(B))$.
- $I(A \wedge B) = \min(I(A), I(B))$.
- $I(\bot) = 0$.

An interpretation is a function $I\colon \mathcal{L} \to \{0, 1, 2\}$ that assigns a truth value to each atom in the language. The interpretation of an arbitrary formula is obtained by propagating the evaluation of each connective as defined above. Recall that $\neg$ and $\top$ were introduced as abbreviations of other connectives. For a given interpretation I and a formula F we say that I is a *model* of F if $I(F) = 2$. Similarly I is a *model* of a program P if it is a model of each formula contained in P. If F is modeled by every possible interpretation we say that F is a *tautology*. For instance, we can verify that $\neg\neg a \rightarrow a$ is not a tautology in G_3, and $a \rightarrow \neg\neg a$ is a tautology in G_3. For a given set of atoms M and a program P we will write $P \vdash_{G_3} M$ to abbreviate $P \vdash_{G_3} a$ for all $a \in M$, and $P \Vdash_{G_3} M$ to denote the fact that $P \vdash_{G_3} M$ and P is consistent w.r.t. logic G_3 (i.e. there is no formula A such that $P \vdash_{G_3} A$ and $P \vdash_{G_3} \neg A$).

2.3 Answer sets

As usual in ASP, we take for granted that programs with predicate symbols are only an abbreviation of the ground program. We shall define answer sets of logic programs. The answer sets semantics was first defined in terms of the so called *Gelfond-Lifschitz reduction* [6] and it is usually studied in the context of syntax dependent transformations on programs[3]. We follow an alternative approach

[3] Currently, there are several answer set solvers, such as: DLV (http://www.dbai.tuwien.ac.at/proj/dlv/) and SMODELS(http://www.tcs.hut.fi/Software/smodels/)

started by Pearce [11] and also studied by Osorio et.al. [9]. This approach characterizes the answer sets for a propositional theory in terms of logic G_3 and it is presented in the following definition. There are several nice reasons to follow this approach. One of these reasons is that it is possible to use logic G_3 to provide a definition of ASP for arbitrary propositional theories, and at the same time to use the logic framework in an explicit way [9]. Moreover, this approach provides a natural way to extend the notion of answer sets in other logics [9]. The notation is based on [9]. We point out that $\neg$ denotes *default negation* and it is the only type of negation considered in this paper. However, it is worth mentioning that we always can handle the other negation called classical or even strong negation, denoted by $-$, by transforming the atoms with classical negation [7]. Each atom with classical negation , $-a$, that occurs in a formula of a logic program should be replaced by a new atom, a', and the rule $\neg(a \wedge a')$ should be added to the logic program. Rule $\neg(a \wedge a')$ can also be written as $(a \wedge a') \rightarrow \bot$.

Definition 1. *[9] Let P be a program and M a set of atoms. M is an* answer set *of P iff $P \cup \neg(\mathcal{L}_P \setminus M) \cup \neg\neg M \Vdash_{G_3} M$.*

For instance, the answer sets of

$$ice\_cream \vee (coffee \wedge cake) \leftarrow .$$

are $\{ice\_cream\}$ and $\{coffee, cake\}$ because,

$$P \cup \{\neg coffee, \neg cake\} \cup \{\neg\neg ice\_cream\} \Vdash_{G_3} \{ice\_cream\}, \text{ and}$$
$$P \cup \{\neg ice\_cream\} \cup \{\neg\neg coffee, \neg\neg cake\} \Vdash_{G_3} \{coffee, cake\}.$$

3 Syntax and semantics for preferences

In order to specify preferences we introduce a new connective, $*$, called *preference operator*. A preference rule specifies the preferences for something. Its head corresponds to an ordered list of well formed formulas connected using the operator $*$, where each well formed formula represents a preference option.

Definition 2. *A* preference rule *is a formula of the form: $f_1 * \cdots * f_n \overset{pr}{\leftarrow} g$ where $f_1, \ldots, f_n, g$ are well formed propositional formulas. A* preference logic (PL) program *is a finite set of preference rules and an arbitrary set of well formed formulas. If $n = 1$ the preference rule is called* desire.

If $g = \top$ the preference rule can be written as $f_1 * \cdots * f_n \overset{pr}{\leftarrow}$. The formulas $f_1 \ldots f_n$ are called the *options* of a preference rule.

Example 1. A restaurant has two options for dessert, *ice-cream* or *coffee with cake*. Peter wants ice_cream rather than cake and if possible he desires coffee. Hence, restaurant's options, and Peter's preferences and desires can be simply represented as the PL program (1), i.e.,

$$ice\_cream \vee (coffee \wedge cake) \leftarrow .$$
$$ice\_cream * cake \overset{pr}{\leftarrow} .$$
$$coffee \overset{pr}{\leftarrow} .$$

Let r_1 and r_2 be the preference rule $ice\_cream * cake \overset{pr}{\leftarrow}$. and the desire $coffee \overset{pr}{\leftarrow}$. respectively.

The answer sets of a PL program are the answer sets of the logic program obtained by remove the preference rules from the original PL program.

Definition 3. *Let $Pref$ be the set of preference rules of a PL program P. Let M be a set of atoms. M is an answer set of P iff M is an answer set*[4] *of $P \setminus Pref$.*

So, the answer sets of the PL program of Example 1 are $\{ice\_cream\}$ and $\{coffee, cake\}$.

Part of the semantics of PL programs was inspired by the semantics of LPOD's [2]. Due to lack of space we do not present the semantics of LPOD's, but readers not familiar with this approach can review [2]. The semantics of PL programs is based on a function called *satisfaction degree*. The satisfaction degree with respect of an answer set of a preference rule indicates how well is this preference rule satisfied by the answer set. A satisfaction degree equal to 1 is better than all others. A satisfaction degree lower than other is better than the second one. Our definition of satisfaction degree is in terms of logic G_3, however since logic programs (or theories) used in this work are complete (i.e. for any formula A of a program P, either $P \vdash_{G_3} A$ or $P \vdash_{G_3} \neg A$), we could use classic logic too [5].

Definition 4. *Let M be an answer set of a PL program P. Let $r := f_1 * \cdots * f_n \overset{pr}{\leftarrow} g$ be a preference rule of P. Let m be the $max\{n \mid f_1 * \cdots * f_n \overset{pr}{\leftarrow} g$ is a preference rule of $P\}$. We define the* satisfaction degree of r *in M, denoted by $deg_M(r)$, as a correspondence rule that defines the following function:*

1. 1 *if* $M \cup \neg(\mathcal{L}_P \setminus M) \nvdash_{G_3} g$.
2. $min\{i \mid M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{G_3} f_i\}$ *if* $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{G_3} g$.
3. $m + 1$ *if* $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{G_3} g$ *and there is not* $1 \leq i \leq n$ *such that* $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{G_3} f_i$.

The part (1) of Definition 4 indicates that rule r has a satisfaction degree equal to 1 in the answer set because, the rule r does not apply and this fact makes rule r irrelevant to prefer the answer set. The part (2) of Definition 4 indicates that rule r is satisfied in the answer set to some degree. Finally, the part (3) of Definition 4 indicates that the rule r has the largest value of satisfaction degree in the answer set because non of the options of the preference rule r holds. According to our intuition, this part (3) of Definition 4 will be useful in case

[4] Note that since we are not considering strong negation, there is no possibility of having inconsistent answer sets.

[5] For complete theories, logic G_3 is equivalent to classic logic [9].

that non of the options of each preference rule in P holds in all the answer sets of P. So, in this case all the answer sets of the PL program should be preferred. For instance, let us suppose that I have a preference for fruit over cookies and orange juice over milk for breakfast. Additionally, I have only two options for breakfast, salad or eggs. In this case both options are incomparable with respect to my preferences and both of them should be preferred.

Example 2. Let P be the PL program of Example 1. By Definition 3 we know that program P has two answer sets: $M_1 = \{ice\_cream\}$ and $M_2 = \{coffee, cake\}$. According to Definition 4 we can verify that $m = 2$ and that,

$$\begin{array}{ll} deg_{M_1}(r_1) = 1, & deg_{M_2}(r_1) = 2, \\ deg_{M_1}(r_2) = 3, & deg_{M_2}(r_2) = 1. \end{array}$$

It is interesting to mention that $deg_{M_1}(r_2)$ is equal to 3 because, non of the options of r_2 holds in M_1 (see part (3) of Definition 4). So, $deg_{M_1}(r_2) = m + 1$.

The following theorems and definitions are about the preferred answer sets of a PL program. All of them are similar to the definitions given in [2]. However we do not have to forget that they are defined for general theories (see Definition 2) and are based on our own definition of satisfaction degree.

Theorem 1. *Let $Pref$ be the set of preference rules of a PL program P. If M is an answer set of P then M satisfies all the rules in $Pref$ to some degree.*

The satisfaction degree of each preference rule of a PL program allows us to define the set of preference rules with the same satisfaction degree. We use these sets to obtain the preferred answer sets of the PL program according to some criterion.

Definition 5. *Let P be a PL program and let $Pref$ be the set of preference rules of P. Let M an answer set of P. We define $S^i_M(P) = \{r \in Pref \mid deg_M(r) = i\}$.*

Example 3. Let P be the PL program of Example 1. Let us consider the satisfaction degree of rules r_1 and r_2 in Example 2. Then we can verify that,

$$\begin{array}{lll} S^1_{M_1}(P) = \{r_1\}, & S^2_{M_1}(P) = \{\}, & S^3_{M_1}(P) = \{r_2\}, \\ S^1_{M_2}(P) = \{r_2\}, & S^2_{M_2}(P) = \{r_1\}, & S^3_{M_2}(P) = \{\}. \end{array}$$

The following definitions indicate how to apply different criteria to the sets of preference rules $S^i_M(P)$ of a PL program, in order to know if one answer set is preferred to another answer set and to obtain the most preferred answer sets. The criteria used are *set inclusion*, *set cardinality* or *pareto criterion*. We start describing how to apply the set inclusion criterion.

Definition 6. *Let M and N be answer sets of a PL program P. M is* inclusion preferred *to N, denoted as $M >_i N$, iff there is an i such that $S^i_N(P) \subset S^i_M(P)$ and for all $j < i$, $S^j_M(P) = S^j_N(P)$.*

Definition 7. *A set of atoms M is an* inclusion-preferred *answer set of a PL program P, if M is an answer set of P and there is not answer set M' of P, $M \neq M'$, such that $M' >_i M$.*

Example 4. Let P be the PL program of Example 1. If we consider the results of Example 3 then, we can verify that M_1 is not inclusion-preferred to M_2 or vice versa since $S^1_{M_1}(P)$ is not a subset of $S^1_{M_2}(P)$ or vice versa. We also can see that there is not M answer set of P, $M \neq M_1$ and $M \neq M_2$, such that $M >_i M_1$ or $M >_i M_2$. Hence M_1 and M_2 are both the inclusion-preferred answer sets of P.

From Example 4 we can see that if M_1 and M_2 are inclusion-preferred answer sets of P then, it means that Peter could have *ice-cream* or *coffee with cakes* for breakfast. This result agrees with our intuition because according to Peter's preferences and desires: M_1 includes *ice-cream* that is one of the options with the highest preference, and M_2 includes *coffee* that is a desire with the highest preference too.

The following two definitions indicate how to use set cardinality criterion to prefer an answer set to another answer set, and to obtain the most preferred answer sets. Before presenting these definitions, we have to mention that this criterion was particularly useful to specify preferences for evacuation plans using ASP approaches in [13]. One of the criteria used to prefer the evacuation paths in [13] was the number of segments of roads in each evacuation path. In this case, the paths with the minimum number of segments of road were the preferred paths. So, the idea of using the cardinality set criterion resulted very natural and easy in this case.

Definition 8. *Let M and N be answer sets of a PL program P. M is* cardinality preferred *to N, denoted as $M >_c N$, iff there is an i such that $|S^i_M(P)| > |S^i_N(P)|$ and for all $j < i$, $|S^j_M(P)| = |S^j_N(P)|$.*

Definition 9. *A set of atoms M is a* cardinality-preferred *answer set of a PL program P, if M is an answer set of P and there is not answer set M' of P, $M \neq M'$, such that $M' >_c M$.*

Example 5. Let P be the PL program of Example 1. If we consider the results of Example 3 then, we can verify that M_2 is cardinality-preferred to M_1 since $|S^2_{M_2}(P)| > |S^2_{M_1}(P)|$ and $|S^1_{M_2}(P)| = |S^1_{M_1}(P)|$. We also can see that there is not answer set M' of P, $M' \neq M_2$, such that $M' >_c M_2$. Hence M_2 is the cardinality-preferred answer set of P.

From the Example 5 we can see that if the criterion to prefer one of the menu options for breakfast is the number of things that the option includes then, we agree that $M_2 := \{coffee, cakes\}$ is a better option than $M_1 := \{ice\_cream\}$ because M_2 has two things to eat and M_1 has only one.

Finally, we describe the pareto criterion. As it is described in [2], in some cases the result of adding not achievable options to preference rules does not agrees with what we could expect. For instance, let us extend the problem specification of Example 1 as follows: the restaurant also can offer gelatine in the mornings, Peter prefers gelatine to ice_cream, and Peter can't have gelatine because he always arrives at the restaurant at night. So, restaurant's options and Peter's preferences can be represented as the following PL program,

$$gelatine \vee ice\_cream \vee (coffee \wedge cake) \leftarrow .$$
$$\leftarrow gelatine.$$
$$gelatine * ice\_cream * cake \overset{pr}{\leftarrow} .$$
$$coffee \overset{pr}{\leftarrow} .$$

In spite of our intuition indicates that this program should have the same two inclusion-preferred answer sets of Example 4, we can verify that $\{coffee, cake\}$ is the only inclusion preferred-answer set of this PL program. In order to avoid effects of this kind we can use the pareto criterion:

Definition 10. *Let M and N be answer sets of a PL program P. Let $Pref$ be the set of preference rules of P. M is* pareto preferred *to N, denoted as $M >_p N$, iff there is an $r \in Pref$, such that $deg_M(r) < deg_N(r)$, and for no $r' \in Pref$ $deg_N(r') < deg_M(r')$.*

Definition 11. *A set of atoms M is a* pareto-preferred *answer set of a PL program P, if M is an answer set of P and there is not answer set M' of P, $M \neq M'$, such that $M' >_p M$.*

Example 6. Let P be the PL program of Example 1. Let us consider the results of Example 2. We can verify that $deg_{M_1}(r_1) < deg_{M_2}(r_1)$ and $deg_{M_2}(r_2) < deg_{M_1}(r_1)$. So, M_1 is not pareto-preferred to M_2. In a similar way we can verify that M_2 is not pareto-preferred to M_1. We also can see that there is not M answer set of P, $M \neq M_1$ and $M \neq M_2$, such that $M >_p M_1$ or $M >_p M_2$. Hence M_1 and M_2 are both the pareto-preferred answer sets of P.

4 Related work

Currently there are several approaches in non monotonic reasoning dealing with preferences [5]. Balduccini et al. relate ASP with preferences introducing CR-programs with preferences in [1]. Work that relates Answer Set Planning with preferences using language $\mathcal{PP}$ can be found in [12]. Work that relates ASP with preferences as an LPOD can be found in [2]. LPOD's are a useful extension of ASP, providing more natural modeling and easier solutions to many problems. Currently it is possible to compute the preferred answer sets under the ordered disjunction semantics using *Psmodels* [3]. *Psmodels* is a modification of SMODELS to compute the preferred answer sets of LPOD's. In [10] a broader syntax for LPOD's and its semantics is proposed. This approach was called extended logic programs with ordered disjunction (ELPOD).

The approach presented in this paper follows the approaches given in [2,10]. In spite of ELPOD's [10] and PL programs have a similar broader syntax and use the three criteria (set inclusion, set cardinality and pareto criterion) to get the preferred answer sets, both approaches are different. Specifically, PL programs differ from ELPOD's in two main features.

The first difference between PL programs and ELPOD's is their semantics. ELPOD's represent a particular prioritized form of disjunction over the preference options and PL programs represent preference over the preference options. For

instance, if we represent the two parts of the PL program of Example 1 as an ELPOD then, we can verify that $\{ice\_cream, coffee\}$ and $\{ice\_cream, coffee, cake\}$ are the preferred answer sets using the three criteria: *set inclusion*, *set cardinality* and *pareto criterion*. Clearly, this result does not agree with our intuition, results and discussions presented in Example 4, Example 5, and Example 6.

The second difference between PL programs and ELPOD's is the form to represent a problem with preferences. A PL program represents this kind of problems as a set of well formed formulas joined to a set of preference rules. So a PL program has two parts, one to generate the answer sets and one to express the preferences. In ELPOD's this does not occur. An ELPOD represents a problem with preferences as a set of extended ordered disjunction rules.

On the other hand, in [14] the authors define the concept of maximal answer sets of a program w.r.t. a set of atoms. For instance, if the answer sets of a program P are $\{b, c, e\}$, $\{b, c, d\}$, and $\{e, a, c\}$ then, $\{b, c, d\}$ is the maximal answer set with respect the set of atoms $A := \{b, d, f\}$ because it has the maximum number of elements in the set A. Moreover, in [14] is described how this concept could be useful in a real application related to argumentation in the domain of organ transplantation. The idea in [14] is to get the maximal answer sets from a particular program such that, these maximal answer sets correspond to the preferred extensions of an argument framework.

It is easy to verify that PL programs can be used to obtain the maximal answer sets of a logic program P w.r.t. a set of atoms A. The PL program used is obtained by adding to P a set of *desires* (see Definition 2). The set of desires is defined as follows: for each atom a in A, the desire $a \overset{pr}{\leftarrow} .$ is added. So, in the previous example the PL program used to obtain the maximal answer sets of P w.r.t. A is: $P \cup \{b \overset{pr}{\leftarrow} .,\ \ d \overset{pr}{\leftarrow} .\ \ f \overset{pr}{\leftarrow} .\}$. The intuition behind each desire is to indicate that we want that each atom a in A is in the answer sets of the original program P.

Finally, we want to mention that if we use a PL program P' to compute the maximal answer sets of a program P w.r.t. a set of atoms A then, it is possible to translate P' to a particular ELPOD P'' and P'' to a LPOD P'''. So, it is possible to use *Psmodels* [3] with the inclusion set criterion to compute the maximal answer sets of P from P'''.

The translation of the PL program P' to an ELPOD P'' is as follows: each desire $a \overset{pr}{\leftarrow}$ in P' has to be replaced by the extended ordered rule $\neg\neg a \times a^{\bullet}$ where $a^{\bullet}$ is an atom that neither occurs in P' nor occurs in A. Since $\neg\neg a$ is equivalent to the restriction $\leftarrow \neg a$, the intuition behind $\neg\neg a$ is to indicate that a should be in the model of a program. We can find the details about why to use double default negation in [14].

The translation of the ELPOD P'' to a standard LPOD P''' is due to the fact that neither running *Psmodels* [3] nor following the definition given by Brewka [2] for LPOD's we can obtain the inclusion preferred answer sets for ELPOD's. The reason is that the definition given by Brewka for ordered disjunction has syntactical restrictions. So, in [14] it is also indicated how to translate the par-

ticular ELPOD P'' to a standard LPOD P'''. It is worth to mention that this translation is very simple.

5 Conclusions

In this work we provide two semantics for PL programs. We propose the only ASP approach about preferences and desires that allows us to express preference rules for general theories. We discuss why a broader syntax for rules could bring some benefits. We also present how our approach is related to ELPOD's and how it differs from ELPOD's.

In future work we plan to research about the properties of the semantics of preference logic programs.

References

1. M. Balduccini and V. S. Mellarkod. A-prolog with cr-rules and ordered disjunction. In *International Conference on Intelligent Sensing and Information Processing*, pages 1–6, 2004.
2. G. Brewka. Logic Programming with Ordered Disjunction. In *Proceedings of the 18th National Conference on Artificial Intelligence, AAAI-2002.* Morgan Kaufmann, 2002.
3. G. Brewka, I. Niemelä, and T. Syrjänen. Implementing Ordered Disjunction Using Answer Set Solvers for Normal Programs. In *Proceedings of the 8th European Workshop Logic in Artificial Inteligence JELIA 2002.* Springer, 2002.
4. G. Brewka, I. Niemela, and M. Truszczynski. AnswerSet Optimization. In *IJCAI-03*, pages 867–872, 2003.
5. J. Delgrande, T. Schaub, H. Tompits, and K. Wang. A classification and survey of preference handling approaches in nonmonotonic reasoning. *Computational Intelligence*, 2004.
6. M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. Bowen, editors, *5th Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.
7. M. Gelfond and V. Lifschitz. Logic Program with Classical Negation. In D. H. D. Warren and P. Szeredi, editors, *Proceedings of the 7th Int. Conf. on Logic Programming*, pages 579–597, Jerusalem, Israel, June 1990. MIT.
8. N. Leone and S. Perri. Parametric Connectives in Disjunctive Logic Programming. In *ASP03 Answer Set Programming: Advances in Theory and Implementation*, Messina, Sicily, Sept. 2003.
9. M. Osorio, J. A. Navarro, and J. Arrazola. Applications of Intuitionistic Logic in Answer Set Programming. *Theory and Practice of Logic Programming (TPLP)*, 4:325–354, May 2004.
10. M. Osorio, M. Ortiz, and C. Zepeda. Using CR-rules for evacuation planning. In G. D. I. Luna, O. F. Chaves, and M. O. Galindo, editors, *IX Ibero-american Workshops on Artificial Inteligence*, pages 56–63, 2004.
11. D. Pearce. Stable Inference as Intuitionistic Validity. *Logic Programming*, 38:79–91, 1999.
12. T. C. Son and E. Pontelli. Planning with preferences using logic programming. In *LPNMR*, pages 247–260, 2004.

13. C. Zepeda. *Evacuation Planning using Answer Sets.* PhD thesis, Universidad de las Americas, Puebla and Institut National des Sciences Appliquées de Lyon, 2005.
14. C. Zepeda, M. Osorio, J. C. Nieves, C. Solnon, and D. Sol. Applications of preferences using answer set programming. In *Submmited to Answer Set Programming: Advances in Theory and Implementation (ASP 2005).*

Optimization and Classification

A Genetic Algorithm for Reassigning Work on the Assembly Line: A Real Scenario

Sergio Ramirez-Campos[1] and Luis Torres-Trevino[2] and Gaston Cedillo-Campos[2] and Jorge Villegas-Leza[3] and Pedro Perez Villanueva[2]

[1] Instituto Tecnologico de Saltillo, Blvd. V. Carranza 2400, Saltillo Coahuila, Mexico
sramirez@its.mx
[2] Corporacion Mexicana de Investigacion en Materiales, S. A. de C. V. Blvd. Oceania 190, Saltillo Coahuila, Mexico
ltorres@comimsa.com.mx, mgaston@comimsa.com.mx, pperez@comimsa.com.mx
[3] Daimler Chrysler Planta Derramadero Saltillo Coahuila, Mexico
jv@dcx.mx

Abstract. The problem at hand corresponds to a real situation with characteristics that make it a flow shop NP-hard type of combinatorial optimization problem. This is a unidirectional line flow with serial workstations in a sequential order for all products. Additionally, restrictions of precedence, different workstations, number of operators and jobs greater than one per workstation, and zoning restrictions for certain equipment are considered. This scenario makes it highly complex, under an increasing number of work elements, to reassign within the workstations. A simple genetic algorithm to readjust the assembly line subject to perturbations that force the reduction of cycle time was designed. To meet this demand, work elements are reassigned in order to find the minimum number of workstations without exceeding cycle time. Finally, experimental results in a real-life automobile assembly plant indicate the effectiveness and applicability of the proposed approach in practice.

1 Introduction

The term production line is applied to a flow oriented production system, which is still typical in the automotive industry. According to Carnahan *et al.* [1], a production line of this type, is defined as a series of manual or automated assembly workstations through which one or multiple product(s) are sequentially assembled. The workers, equipment, tools and other devices are located in each workstation according to the tasks that are to be carried out in each one of them.

All the components flow toward each workstation at the last possible moment and in the required quantity to avoid unnecessary inventories and consequently, an unnecessary use of space.

Ideally, the demand (a variable of independent flow) produces work in process ending up with minimum inventory of finished product (a variable of dependent

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 17-29

Received 10/07/06
Accepted 12/10/06
Final version 16/10/06

state). However, these dynamic systems are subject to interferences like changes in the mixture or volume of products, delays, and production speed. Consequently, plants must adjust their production due to unexpected behavior in a market with growing uncertainty. Companies must analyze frequently and have to react to competitive pressure. In other words, appropriate analytical tools are required in order to offer effective solutions with quick implementation. If this is a common practice, the company will be ahead of its competitors having more possibilities to satisfy demand.

2 Background

Several authors have carried out studies in relation to the treatment of disturbances in manufacturing. Matson *et al.* [4], [5] define some important concepts in relation to the nature of the disturbances, analyze how to evaluate their impact on production goals, and determine a procedure for identifying the most important disturbances, although they recognizes that such a procedure does not provide enough information for a detailed study of the response mechanisms. Other investigations have been limited to the process of decision-making, like Meskill *et al.* [6], who explore the way the person who makes the decisions recognizes and identify the interferences. This can be useful for purposes of selection and training.

On the other hand, Kritchanchai and MacCarthy [3] outline the necessity of more investigation, specifically in regard to the process of satisfying production orders. Said authors clarify the definitions and descriptions of the related components through the analysis of the corresponding state of the art, and the study of several industrial sectors, in order to find similarities and differences.

The work that is presented in this article was inspired by the study carried out by Hui Chi and Ng San [2] who outline a mathematical model used to evaluate the impact of variations in the time standards on the balance of an assembly line.

3 The Problem under Study

In order to adequately identify the problem presented in this paper, it is necessary to differentiate between the conventional flow shop problem and an assembly line. A flow shop is characterized by a flow that is almost continuous using multiple serial machines. The flow is unidirectional, and all jobs follow the same route designed by the machine. Although this description of a flow shop is similar to the operation of an assembly line, there are various differences.

First: a flow shop is equipped to handle a variety of jobs, whereas an assembly line handles a standard product. Second: the work in a flow shop does not have to be processed on all of the machines. Some jobs omit operations according to technological requirements. However, in an assembly line, all work must follow all stations without skipping any. Third: in a flow shop, every machine is independent from the others and can carry itself as such. In an assembly line,

every workstation depends on the preceding one. Finally, in a flow shop every job has its own process time on each machine, yet on an assembly line, all the operations in the production of a product have the same standard time at each station.

It is also important to mention that there are a great number of articles that study the balance of a flow shop, such as the single assembly line balancing problem (SALBP). A study that analyzes a variant of SALBP is given by Nicosia *et al.* [14] who formulate a solution based on a dynamic programming algorithm related to the problem of operation assignment in an ordered sequence of different workstations, observing the restrictions of precedence and cycle time, and minimizing the cost. In this scenario the unidirectional flow is not contemplated, since the work does not have to follow the same route of the machines. Another approach is described by Ponnabalam *et al.* [15] who analyze the problem of a mixed model assembly line (MMAL) in which different workstations, standard times and methods, different equipment, and components and raw material are all considered. Ponnabalam separates the problem in two aspects: (a) the assignment of jobs to workstations and (b) the sequence of models in a line. Said author only addresses the second case.

A scenario that is closer to the problem we are discussing here is shown by Dimitradis [16] who proposes a different design in the assembly line. Dimiatradis proposes that instead of workstations with a single operator, more operators without surpassing the "maximum concentration of workers per product" should be considered established according to the structure and size of the product so that the operators are not in each other's way. The use of the right number of tools reduces the wait times along with an adequate distribution of workstations. One omission is that the time for each operator to go from one job to another is not considered.

For the description of a certain type of programming problem, the nomenclature used by Pinedo [17] $(\alpha|\beta|\gamma)$ can be used, in which α represents the area where the machines are consistent with a single entry, β gives details of characteristics and restrictions and it is possible for it to contain either no entry or multiple entries, and finally γ describes the objective to be minimized and usually contains a single entry.

In our study, the problem is in the reassignment of tasks to the workstations and it can be defined as $Fm|block, prec|Ct$ since it is a discrete system of unidirectional in line flow with: m workstations in series with zero capacity buffers, blocks in between stations, and restrictions of precedence. The goal is to minimize the cycle time. Other important considerations are:

- The number of operators in a single workstation is not limited to one.
- The number of operations assigned to a workstation is not limited to one.
- If equipment or tools are not to be relocated in other stations, the corresponding restriction is respected.
- Interruptions are not allowed in operations once initiated.
- The order, in which the jobs are processed, is the same for all.

We have not found an analysis of the defined problem in any literature. Furthermore, Pinedo [17] determines that the problem $Fm||Cj$ is NP-hard for $m \geq 2$ and our problem stated is generalized by incorporating the restrictions of precedence, and therefore also is NP-hard. The solution proposed represents a contribution to this area.

4 Modeling the System

The Critical Path Method (CPM) provides a simple form of capturing the most representative facts of the process when there is a great number of tasks and resources along with the interaction among the activities. Mathur and Solow [7] explain this technique. The necessary minimum information consists of the description of each activity, its standard time, and its predecessors and successors. With this information available, the corresponding network of the production line is designed. In this way, Ramirez-Campos and Salais-Fierro [8] illustrate an example of the use of this technique in the case of a process of on-line flow.

It is also possible to incorporate another class of restrictions referring to position, like the fact that some work elements must be carried out particularly in a certain station or, on the contrary, should not be carried out in one or several specific stations.

Network is a group of precedent restrictions that allows the identification of feasible solutions, i.e., the solutions that satisfy this group of restrictions and the restrictions referring to position.

5 The Simple Genetic Algorithm

The simple genetic algorithm (SGA) belongs to an evolutionary computation paradigm which emulates the processes proposed in the theory of natural selection of Darwin [9]. Genetic algorithms have been used to solve problems of optimization through the generation of solutions called "population by analogy to natural process". The process of evaluation, selection and reproduction is applied to the population of solutions in a cyclical way.

According to Goldberg [10], the simple genetic algorithm uses non-complex mechanisms to carry out selection and reproduction, however, as in any evolutionary algorithm, a key point is the representation of the possible solutions to the problem and the corresponding evaluation.

The first step is to design an initial network that is revised and modified until the representation of the process that should be carried out in the production line is reached. After obtaining the definitive network containing m work elements, the following step is to represent each feasible solution using a chain or chromosome. This chain represents a group of stations and each station represents a gene of the chromosome. Each station contains a certain number of assigned work elements.

To generate a feasible solution (or chromosome), the desired minimum speed is chosen, for example 47 vehicles per hour (47 vph), and the current speed of the

production line that is shown here. This speed implies a maximum cycle (t_{cm}) of 1.277 minutes per vehicle (1.277 mpv), i.e., 60/47. The following algorithm is applied in order to obtain a feasible initial solution:

1. $k = 0$ (station number)
2. $k = k + 1$
3. Generate a random integer r_i so that $1 \leq r_i \leq m$ (i^{th} random integer)
4. If the precedent restrictions and positions are satisfied, we continue on step 5. if not, we continue on step 3.
5. If t_{cm} is not exceeded, the r_i^{th} work element is assigned to station k and we continue on step 6. If not, we continue on step 2.
6. If the all work-elements were already assigned, the algorithm is stopped. If not, we continue on step 3.

After the quantity of desired solutions is generated, an initial population is available with p individuals or p feasible solutions. Since the aptitude value depends on the cycle time of the slowest station, the population is sorted according to the cycle times to identify the best solution reached to the moment. The following step is reproduction. A rate is established, t_R recommended between 0.5 and 0.8 according to Jahangirian and Conroy [11]. If this rate is established at 0.7 it implies that 70 percent of the population will change one aptitude value to another.

Starting from the sorted population, there are two selection methods for carrying out reproduction:

1. A selection of mates according to $(1, 2), (3, 4), (5, 6), \ldots$
2. A random selection of mates

After selecting a couple (s_i, s_j), the next step is to proceed to reproduction by generating a random whole number R_k so that $1 \leq R_k \leq m$. R_k represents the work element and is in the position to carry out the exchange between s_i and s_j, in order to obtain two new solutions. It is important to verify the feasibility of the new solutions, h_1 and h_2. If not feasible, one or both are discarded and reproduction continues with another couple until completing the t_R rate. Once this rate has been reached, insertion by choosing a t_r rate that can be up to a 100 percent according to Badibaru and Cheung [12] is the next step. A t_r value of 0.5 divides the current sorted population in two parts with the worst group being the second part which will be replaced by the first half (the best group) of the new sorted generation.

According to Badibaru and Cheung [12], there are two methods for conforming the new population:

1. The proportion used in the substitution is compounded exclusively for new solutions.
2. The proportion used in the substitution is composed of new solutions, the part of the current population's solutions that did not reproduce and the solutions that result from mutation.

For mutation, a rate (t_m) is chosen with a value around 0.2 according to Jahangirian and Conroy [11]. If the value of t_m is 0.2, it implies that 20 percent of the current population will change. A mutation is obtained by the following algorithm:

1. A random integer is generated so that $0 < r_j < p$ selecting the next solution to be submitted to mutation. A new feasible solution is generated in order to partially replace the r_j solution.
2. A random number r_i is generating so that $0 \leq r_i \leq 1$ (i^{th} random number)
3. The probability of mutation is calculated using:

$$P_m = \frac{1}{2}\left(\frac{1}{p} + \frac{1}{l}\right) \tag{1}$$

   where p = initial population, l = length of chromosome or number of stations in the particular solution
4. If $r_i < P_m$, the mutation is applied, otherwise the mutation does not proceed and the algorithm ends.
5. If the mutacion proceeds, another random integer is generated so that $0 < r_k < l$ for defining the position where the mutation starts. This is from the gene in the r_k position.
6. Finally, the mutation is executed and its feasibility is verified. In case of an unacceptable solution, step 5 is repeated until a feasible solution is achieved.

As a result of each iteration, the process described creates new generations and is stopped when it reaches an established goal or satisfies a detention rule. Three measures can be selected for the aptitude function F:

1. Smaller standard deviation
2. Smaller cycle time
3. Greater average efficiency

The smallest standard deviation is calculated using the cycle times of each station, t_{ci}. For instance, a certain solution with 14 stations has 14 cycle-times to calculate their standard deviation. In this way, for a number (n) of stations, the smallest cycle time is obtained using the group of t_{ci} (or genes) of a given solution. The efficiency average (E_a) is the average of the efficiencies of all stations (E_i):

$$E_a = \frac{\sum_{i=1}^{n} E_i}{n} \tag{2}$$

The efficiency of a station i (E_i) is the percentual relationship of the cycle time of the station (t_{ci}) and the allowed maximal cycle time (t_{cm}) that it is determined by the user:

$$E_i = \left(\frac{t_{ci}}{t_{cm}}\right)100 \tag{3}$$

6 The COMSOAL Heuristic Approach

COMSOAL was originally a solution approach for the assembly-line balancing problem. In fact, according to Johnson and Montgomery [13] the COMSOAL approach finds solutions as follows:

1. Select the tasks without precedences and open the first station ($k = 1$).
2. Assign the tasks of the step 1 to the station k without exceeding the cycle time (t_{mc}).
3. In the event it is necessary to continue with step 2, make $k = k+1$; otherwise, go to step 4.
4. Given the tasks already assigned, identify the possible inmediate tasks (m).
5. Given the tasks already assigned, generate the possible sequences (permutations) of m tasks.
6. Select the task(s) with the smallest number of possible different positions and, if there is tie, randomly choose one of them.
7. Assign task s to station k without exceeding the t_{mc} and go to step 9; otherwise go to step 8.
8. Open a new station ($k = k + 1$).
9. If all the tasks of the network were already assigned, end the procedure; otherwise go to step 4.

7 Experimental results

In Section 7.1 steps are shown for applying the response mechanism to readjustments in the assembly due to changes in the configuration of the product. Results are obtained. In Section 7.2 the comparative results of said algorithms are explained.

7.1 The Response Mechanism

For application the response mechanism described was implemented in a section of the process of an assembly plant. This section is the door line that at the moment consists of 14 work stations and runs to a speed of 47 vehicles per hour (vph). The desired objective was a new configuration of this line to reach a minimum speed of 48 vph. The steps followed are shown.

a. The standard times of each work element were upgraded and the corresponding network was designed, and verified by the analysis of the department of Industrial Engineering so that the structure was attached to the process. Once the network was concluded, it was captured on the interface. Table 5 shows partial information corresponding to this network.
b. Since work elements 1 and 6 must be carried out only in station 1, two additional restrictions were considered.

c. In order to conclude correctly, it was decided to make 5 runs at each one of the speeds: 45, 46, 47, 48 and 49 vph. Tables 1, 2, and 3, where t_{cpu} is cpu seconds, show the corresponding results. We took into account the following parameters:

$$m = 71 \qquad p = 15 \qquad t_R = 0.7 \qquad t_m = 0.2 \qquad t_r = 0.5$$

The fitness function (F) is the smallest standard deviation (minutes). Also, the detention rule is activated when 15 consecutive intents are accumulated and in each case the difference between the cycle times of one solution and another is smaller than 0.0001.

Table 1. Results for 45 and 46 vph

vph	F	n	t_{cpu}	E_a	vph	F	n	t_{cpu}	E_a
45.079	0.089	13	112.344	92.898	46.620	0.087	14	114.734	88.181
45.249	0.074	13	89.734	92.898	46.404	0.089	14	103.094	88.181
45.079	0.071	13	128.563	92.898	46.440	0.092	14	128.641	88.181
45.249	0.109	13	93.969	92.898	46.048	0.062	13	90.203	94.064
45.249	0.076	13	107.688	92.898	46.765	0.075	14	133.563	88.181

d. When analyzing the results of the runs, we can observe that the answer mechanism is consistent, i.e., for a lower speed, a lower number of stations (n) and vice versa. Also, the same efficiency average was always obtained

Table 2. Results for 47 and 48 vph

vph	F	n	t_{cpu}	E_a	vph	F	n	t_{cpu}	E_a
48.154	0.068	14	163.313	90.094	48.154	0.070	14	112.063	92.011
47.170	0.082	14	133.219	90.094	48.077	0.085	14	103.625	92.011
47.356	0.076	14	109.281	90.094	48.232	0.096	14	116.188	92.011
47.133	0.081	14	125.438	90.094	48.820	0.080	14	107.609	92.011
48.309	0.088	14	119.781	90.094	48.154	0.071	14	103.672	92.011

(E_a) for the speeds of 47 and 48 vph.
For 46 vph, four runs give 14 stations and one gives 13 stations, but the efficiency average is low in the first case. Considering that the network is a constant, we can see that, seemingly, for this particular network the speed of 49 vph offers the highest efficiency average, with n = 14. In other words, with a higher or lower speed than 49 vph, E_a diminishes or increases or another number of stations is required. In the case of 49 vph, if the first

Table 3. Results for 49 vph

vph	F	n	t_{cpu}	E_a
49.342	0.068	14	163.313	93.928
49.140	0.118	15	125.266	87.666
49.669	0.088	15	108.047	87.666
50.000	0.082	15	128.031	87.666
49.020	0.109	15	115.234	87.666

run had not been better than the other runs, the low efficiency of the other four runs would indicate the possibility of a better solution. It is advisable to make other runs by increasing the number of intents in the stop rule, and the number of iterations, or by making other types of changes.
To illustrate how to make a more thorough search, table 4 shows the results of 5 runs, applying the following changes:

$$p = 60 \qquad tries = 100$$

The results of this search revealed that four of them reached the same efficiency average of 93.928 percent with only 14 stations, showing that these are the best solutions. The cost of this is the time invested, which went from 1.5 to 3 minutes in the first part of the search, changing to an approximate range of between 4 and 28 minutes in an intensive search (table 4).

Table 4. Results for a thorough search

vph	F	n	t_{cpu}	E_a
49.221	0.064	14	250.328	93.928
49.261	0.075	14	255.438	93.928
49.342	0.078	14	243.641	93.928
50.761	0.073	15	1682.328	87.666
49.180	0.066	14	115.234	93.928

e. In conclusion, the best solution for a desirable minimum speed of 48 vph is the run with the smallest value of F shown in line 1 of table 4.
Corresponding configuration is partially shown in table 5, because there are 71 work elements and 14 stations and, therefore, the description of the work element is omitted. In this table, column 1 is the number of work elements, column 2 is the station number i, columns 3 and 4 the nodes (a, b) of work element k, column 5 the element time, column 6 the cycle time of the station i, and column 7 the efficiency of station i. Columns 8 to 14 have the same meaning.

Table 5. Final configuration for 48 vph

		nodes							nodes				
k	i	a	b	t_k	t_{ci}	E_i	k	i	a	b	t_k	t_{ci}	E_i
1	1	1	5	0.158			15	2	1	500	0.254		
2	1	5	15	0.305			16	2	1	185	0.266		
3	1	5	25	0.119			17	2	1	195	0.164		
4	1	5	20	0.115			18	2	1	200	0.152		
5	1	20	25	0			19	2	185	500	0		
6	1	15	30	0.260			20	2	195	500	0		
7	1	25	30	0			21	2	200	500	0	1.214	99.14
8	1	1	180	0.240			.	.	.	.	.	.	.
9	1	180	500	0	1.197	97.75	.	.	.	.	.	.	.
10	2	5	10	0.034			.	.	.	.	.	.	.
11	2	10	30	0			68	13	148	160	0.758		
12	2	5	38	0.063			69	13	160	165	0.229	0.987	80.60
13	2	25	35	0.281			70	14	155	500	0.502		
14	2	35	38	0			71	14	165	500	0.634	1.136	92.77

Table 6. The best results

	Number of stations	Cycle time (minutes)
SGA	14	1.222
COMSOAL	15	1.178

Table 7. Average performance data shown in Figure 1

Iteration	1	2	3	4	5	6	7	8
SGA	18.015	17.887	17.825	18.015	17.985	17.658	17.786	17.477
COMSOAL	19.436	19.350	19.342	19.301	19.223	19.464	19.389	19.375
Iteration	9	10	11	12	13	14	15	16
SGA	17.820	17.761	17.748	17.741	17.706	17.765	17.902	17.770
COMSOAL	19.375	19.194	19.315	19.241	19.398	19.352	19.241	19.191
Iteration	17	18	19	20	21	22	23	24
SGA	17.815	17.759	17.932	17.932	17.773	17.985	17.713	17.825
COMSOAL	19.443	19.445	19.362	19.422	19.289	19.410	19.345	19.304
Iteration	25							
SGA	17.797							
COMSOAL	19.259							

7.2 Performance

The comparison between the SGA and the COMSOAL was based on specifc runs for each technique. In the case of SGA, 100 runs with 25 iterations each were done. In the COMSOAL case, 2500 runs were completed. This comparison process hads two main objectives. The first one was to find the smallest number of work stations. The second one was to get the smallest cycle time not exceeding 1.2245 minutes. This time corresponds to a speed of 49 vph. For the SGA case, 100 values of each respective iteration were averaged and in the COMSOAL case, sets of 25 serial values were taken and averaged.

As a result, this analysis found that the SGA obtained solutions with a smaller number of stations than the COMSOAL approach. COMSOAL achieved a feasible solution of 15 work stations with a cycle time of 1.178 minutes. On the other hand, SGA was able to find a better solution with 14 work stations that perform in a cycle time of 1.222 minutes (see Tables 6 and 7 and Figure 1).

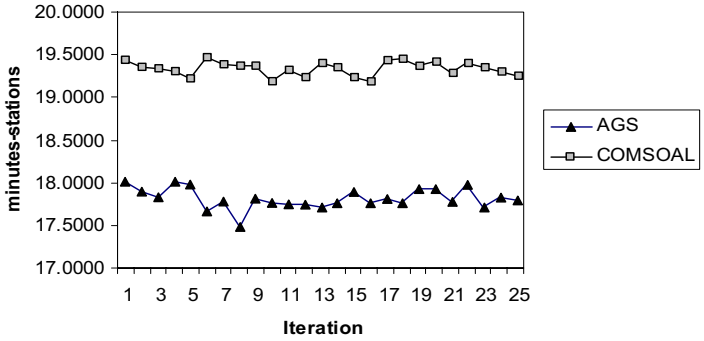


Fig. 1. Comparative performance

8 Conclusions

The door line at the moment is running at 47 vph with 14 work stations, however, a solution has been found with the same 14 work stations reaching a speed of 49 vph. This is a very high increase, considering that in the automotive industry the cost of relocating materials, tools and equipments, is very low in comparison to the benefit of a substantial increase in speed. This mechanism can be applied to any section of the automotive line for similar analysis, subject to an attractive cost-benefit relationship. It is therefore necessary to stress the importance of the estimate of the costs of adjusting the production line according to configuration as a result of applying the response mechanism. These costs are derived from locating work elements in different stations or incorporating new work elements into one or more work stations. It is therefore necessary to relocate tools, diverse devices, material containers and equipment.

If the cost of adjustment is very near the benefit, then a careful additional analysis must be made of the convenience of carrying out the involved changes.

In this light, the mechanism provides valuable information that allows us to avoid making inconvenient decisions about adjustments in a production line. On the other hand, this mechanism can be applied when the production line has not been put into operation yet. In this case, the benefit is greater than in a line in operation, since relocation costs do not exist.

More exhaustive comparison is needed to determine the efficiency of this genetic algorithm, so we needed further heuristics from literature to apply to the stated scenario. Although it is important to note that there is a large amount of literature with heuristics in relation to certain types of flow shop problems, they apply to a multitude of different scenarios, so it is not easy to find a heuristic that adapts to a specific scenario such as the one stated in this paper. However, the genetic algorithm can be improved by taking into account other focuses such as: a multi-objective function of fitness, a small initial population that can grow from one generation to the next, the generation of only feasible solutions, and a more efficient representation of a chromosome to reduce epistasis in the code.

References

1. Carnahan, Brian J., Norman Bryan A., and Redfern Mark S.: Incorporating physical demand criteria into assembly line balancing. IIE Transactions, **33** (2001) 875–887
2. Hui Chi L. P. and Ng Sau F. F.: A study of the effect of time variations for assembly line balancing in the clothing industry. International Journal of Clothing Science and Technology, **11** (1999) 181–188
3. Kritchanchai Duangpun and MacCarthy, B. L.: Responsiveness of the order fulfilment process. International Journal of Operations & Production Management, **19** (1999) 812–833
4. Matson, Jeremy B. and McFarlane, Duncan C.: Tools for Assesing the responsiveness of existing production operations. In Proceedings of IEE Workhshop, Responsiveness in Manufacturing, London, February (1998)
5. Matson, Jeremy B. and McFarlane, Duncan C.: Assesing the responsiveness of existing production operations. International Journal of Operations & Production Management, **19** (1999) 776–784
6. Meskill Margaret, Mouly Souchitra and Dakin, Stephen: Managerial disturbance handling: a case study approach. Managerial Psychology, **14** (1999) 443–454
7. Mathur, Kamlesh and Solow, Daniels: Investigacion de Operaciones. Prentice-Hall Hispanoamericana, S. A. first edition, (1996) 492–541
8. Ramirez-Campos, Sergio and Salais-Fierro, Tomas.: Flexibility and Time Optimization in an Automotive Assembly Line: A Neural Network Approach. In proceedings of Business & Industry Symposium: Advanced Simulation Technology Conference, Arlington, VA, USA, April, (2004)
9. Darwin, Charles: El origen de las especies UNAM, (1997)
10. Goldberg, David E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Pub Co. (1989)
11. Jahangirian, M. and Conroy, G. V.: Intelligent dynamic scheduling system. Integrated Manufacturing Systems, **11** (2000) 247–257

12. Badibaru, A. B. and Cheung, J. Y.: Fuzzy Engineering Expert Systems with Neural Network Applications. John Wiley & Sons, Inc. (2002)
13. Johnson, Lynwood A. and Montgomery Douglas C.: Operations Research in Production Planning, Scheduling, and Inventory Control. John Wiley & Sons, Inc, first edition, (1974) 373–374
14. Nicosia, G. and Pacciarelli, D. and Pacifici, A.: Optimally balancing assembly lines with different workstations. Discrete Applied Mathematics, **118** (2002) 434–456
15. Ponnabalam, S. G. and Aravindan, P. and Rao, M. Subba: Genetic algorithms for sequencing problems in mixed model assembly lines. Discrete Applied Mathematics, **45** (2003) 669–690
16. Dimitriadis, S. G.: Assembly line balancing and group working: A heuristic procedure for workers'groups operating on the same product and workstation. Computers & Operations Research, **33** (2005) 2757–2774
17. Pinedo, M.: Scheduling, Theory, Algorithms and Systems. Prentice Hall, second edition, (2002)

Scheduling Jobs in Flexible Manufacturing Cells with Genetic Algorithms

António Ferrolho[1] and Manuel Crisóstomo[2]

[1] Superior School of Technology of Viseu, Polytechnic Institute of Viseu
Campus Politécnico de Repeses, 3504-510 Viseu, Portugal
antferrolho@elect.estv.ipv.pt
[2] Institute of Systems and Robotics, University of Coimbra
Polo II, 3030-290 Coimbra, Portugal
mcris@isr.uc.pt

Abstract. In this paper, we studied scheduling problems with Genetic Algorithms (GA) in Flexible Manufacturing Cells (FMC). We used a GA for solving the optimization scheduling problem. First, we developed an FMC with industrial characteristics, with the objective of studying scheduling problems in these types of manufacturing systems. Then, we developed a software tool, called HybFlexGA, to study scheduling problems, with GA, in the FMC. Finally, we applied the HybFlexGA to solve scheduling problems in the FMC. The practical results obtained from the FMC for the various scheduling problems show how efficient HybFlexGA is in solving these problems.

Keywords: Flexible manufacturing cells, scheduling and genetic algorithms.

1 Introduction

Scheduling problems in Flexible Manufacturing Cells (FMC) are studied in this paper. We also use Genetic Algorithms (GA) to optimize this type of problems. We have developed an FMC with industrial characteristics and a software tool, called HybFlexGA, with the objective of studying scheduling problems in this type of manufacturing systems. The FMC and the HybFlexGA were used to study single machine total weighted tardiness (SMTWT) problems.

In SMTWT problems each job i has an associated processing time p_i, a weight w_i, and a due date d_i, and the job becomes available for processing at time zero. The tardiness of a job i is defined as T_i=max{0, C_i-d_i}, where C_i is the completion time of job i in the current job sequence. The objective is to find a job sequence which minimizes the sum of the weighted tardiness given by $\sum_{i=1}^{n} w_i.T_i$. Because the SMTWT problem is NP-hard, optimal solutions for this problem would require a computational time that increases exponentially with the problem size [1] and [2]. In recent years, several heuristics, such as Simulated Annealing, Tabu Search, Genetic Algorithms and Ant Colony [1] and [3], have been proposed to solve the SMTWT problem.

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 31-40

Received 04/07/06
Accepted 03/10/06
Final version 13/10/06

2 Developed Flexible Manufacturing Cell

An FMC with industrial characteristics was developed with the objective of studying scheduling problems in these types of manufacturing systems. The hierarchical structure implemented in the FMC is shown in Fig. 1. This FMC is comprised of four sectors, which are controlled by PCs and different software [5] and [6]. The four sectors are:

- The manufacturing sector, made up of two CNC machines (mill and lathe), one ABB IRB140 robot and one buffer (see Fig. 1);
- The assembly sector, made up of one Scorbot ER VII robot, one small conveyor and an assembly table (see Fig. 1);
- The handling sector, made up of one big conveyor (see Fig. 1);
- The storage sector, made up of five warehouses and one robot ABB IRB1400 (see Fig. 1).

Control of existing equipment in each sector is carried out by four computers: PC1 – manufacturing sector, PC2 – assembly sector, PC3 – handling sector and PC4 – storage sector. Coordination, synchronization and integration of the four sectors is carried out by the of FMC Manager computer.

The first layer contains the engineering and design functions where the product is designed and developed. The outputs of this activity are the drawings and the bill of materials.

The second layer is process planning. Here the process plans for manufacturing, assembling and testing are carried out.

The third layer is scheduling. The process plans together with the drawing, the bill of materials and the customer orders are the input to scheduling. The output of scheduling is the release of the order to the manufacturing floor. The PCM – FMC Manager is responsible for the engineering, planning and scheduling activities.

The fourth layer is control. Manufacturing is controlled by a hierarchically structured real time computer system (PC1, PC2, PC3 and PC4). Theirs set points are the operating parameters used to start and control the activities on the production floor.

The fifth layer is data acquisition. The operations of the machine tools and material movement equipment are monitored by a data acquisition system. The collected data represents the state of the manufacturing system and is the feedback information used for control.

The central computer (FMC Manager) controls all of the production of the FMC, calling several computers and using local nets to exchange data which allow control and supervision of the operations in real time, picking up and processing information flows from the various resources. Concisely, the FMC manager PC is responsible for:

- Developing and designing new products to manufacture – the engineering layer;
- Production plans, assemblies and product tests – the planning layer;
- Finding the optimum processing sequence so as to optimize CNC machine use – the scheduling layer;
- Coordination, integration and control of all the sectors in the FMC;

- Maintaining a database of jobs to manufacture, including the respective NC programmes;
- Synchronizing the various sectors so as to produce variable lots of different types of parts depending on the customer's orders;
- Monitoring the current state of production;
- Guaranteeing tolerance of failures, safety and coherence of data.

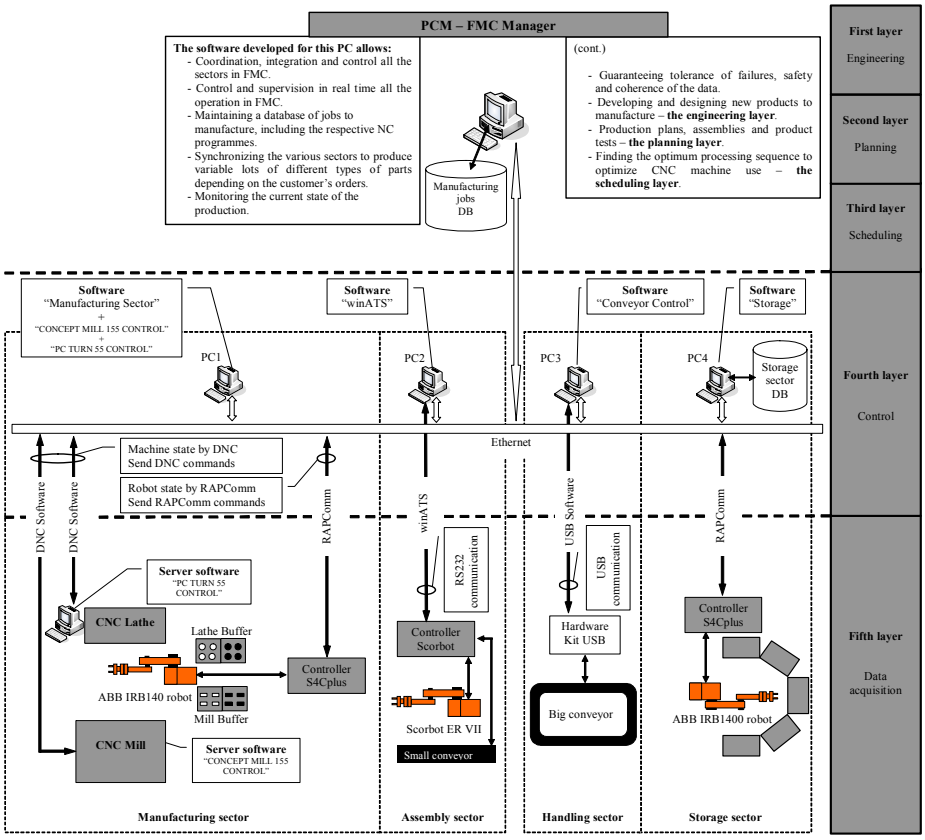


Fig. 1. FMC hierarchical structure.

3 Hybrid and Flexible Genetic Algorithm

We developed a software tool, called HybFlexGA (Hybrid and Flexible Genetic Algorithm), to solve scheduling problems in developed FMC (see the third layer in Fig. 1). The HybFlexGA was coded in C++ language and its architecture is composed of three modules: interface, pre-processing and scheduling module.

3.1 Interface Module

The interface module with the user is very important for the scheduling system's success. Thus, this interface should be user-friendly and dynamic so as to allow easy manipulation of the scheduling plan, jobs, and so forth. This interface allows the connection between the user and the scheduling module, facilitating data entry (for example, parameter definition and problem definition) and the visualization of the solutions for the scheduling module. Fig. 2 shows the interface window.

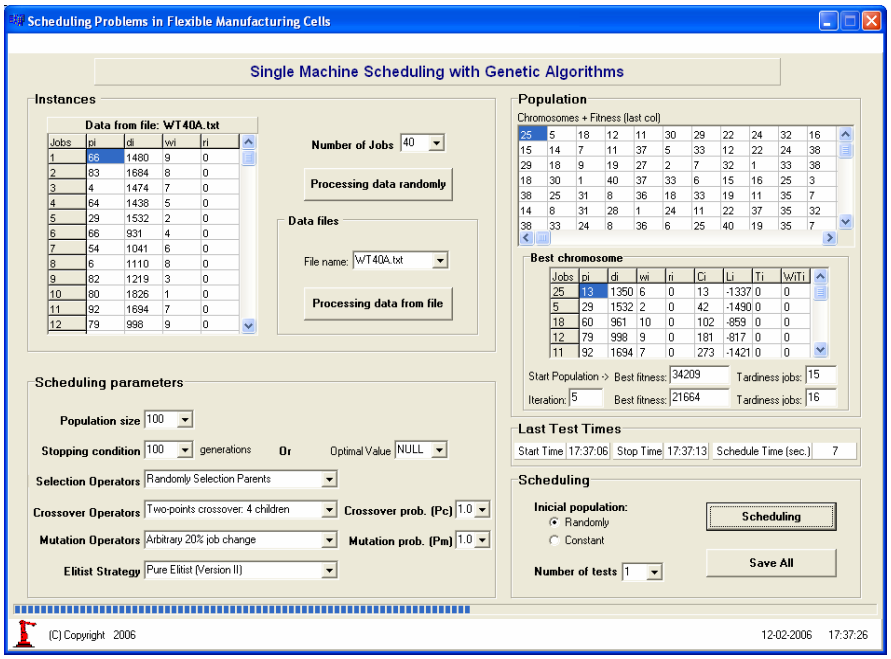


Fig. 2. Interface window.

3.2 Pre-processing Module

The inputs of the pre-processing module are the problem type and the scheduling parameters. The instance of the scheduling problem can be randomly generated or generated by PC file, as shown in Fig. 2. This module pre-processes the input information and then sends the data to the next module – the scheduling module.

3.3 Scheduling Module

The objective of the scheduling module is to give the optimal solution for any scheduling problem. If the optimal solution is not found, the GA gives the best solution found (near-optimal solution). In this module, we implemented the GA shown in Fig. 3.

Step 1 - Initialization
Let t=0, where t is the generation index, and generate an initial population randomly Ψ_0 including N_{pop} solutions (N_{pop} is the number of solutions in each population, i.e., N_{pop} is the population size). The number of solutions (chromosomes) in the t generation is given by $\Psi_t = \left\{x_t^1, x_t^2, ..., x_t^{N_{pop}}\right\}$.

Step 2 - Selection
Select pairs of solutions (parents' chromosomes) from the current population Ψ_t. Each chromosome x_t^i is selected according to the selected operator chosen in the interface module.

Step 3 - Crossover
Apply a crossover operator, selected in the interface module, to each of the selected pairs in step 2. This way, new chromosomes will be generated according to the selected crossover probability (P_c).

Step 4 - Mutation
Apply a mutation operator, selected in the interface module, to the generated chromosomes in step 3, according to the selected mutation probability (P_m).

Step 5 – Elitism
Select the best N_{pop} chromosomes to generate the next population Ψ_{t+1} and the other chromosomes are eliminated. Thus, the best chromosomes, i.e. solutions, will survive into the next generation. However, duplicated solutions may occur in Ψ_{t+1}. To minimize this, new chromosomes are generated for all duplicated chromosomes.

Step 6 – Termination test
Stops the algorithm if the stopping condition, previously specified in the interface module, is satisfied. Otherwise, update t for t:=t+1 and return to step 2.

Fig. 3. GA implemented in the scheduling module.

4 Genetic Operators

In this section, we propose a new concept of genetic operators for scheduling problems. We evaluate each of various genetic operators with the objective of selecting the best performance crossover and mutation operators.

4.1 Crossover Operators

Crossover is an operation to generate a new sequence from two sequences. We examine the following crossover operators:

- One-point crossover: 1 child (OPC1C) in Fig. 4 a);
- Two-point crossover: 1 child (Version I) (TPC1CV1) in Fig. 4 b);
- Two-point crossover: 1 child (Version II) (TPC1CV2) in Fig. 4 c).

We also developed crossover operators with 2, 3 and 4 children. The crossover operators with 2 children are:

- One-point crossover: 2 children (OPC2C). This crossover is similar to the OPC1C, but it generates two children;

- Two-point crossover: 2 children (Vers. I) (TPC2CV1). This crossover is similar to the TPC1CV1, but it generates two children;
- Two-point crossover: 2 children (Vers. II) (TPC2CV2). This crossover is similar to the TPC1CV2, but it generates two children.

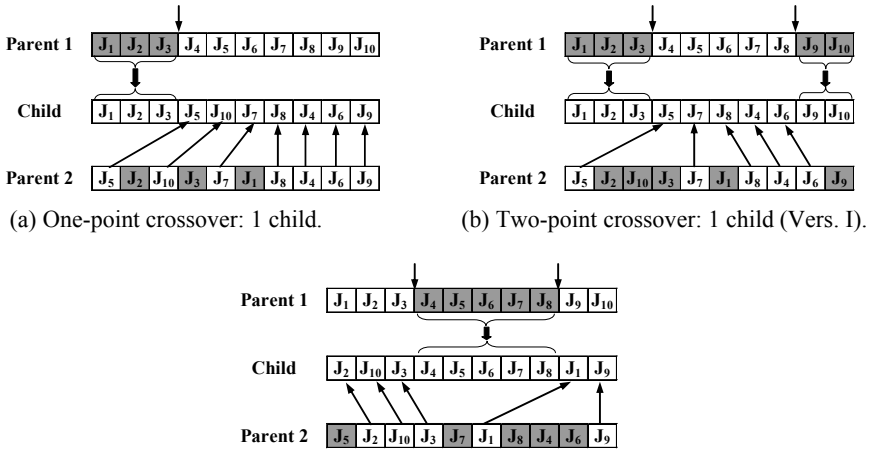
The crossover operators with 3 children are:

- Two-point crossover: 3 children (Version I) (TPC3CV1). This crossover is a mix of TPC1CV1 plus TPC2CV1;
- Two-point crossover: 3 children (Version II) (TPC3CV2). This crossover is a mix of TPC1CV2 plus TPC2CV2.

The crossover operator with 4 children is called a two-point crossover: 4 children (TPC4C). This operator is a mix of TPC2CV1 plus TPC2CV2.

The following three crossover operators are also examined in this paper for FMC scheduling problems:

- Order crossover (OX) in Goldberg [4];
- Cycle crossover (CX) in Oliver [7];
- Position based crossover (PBX) in Syswerda [8].



(a) One-point crossover: 1 child.

(b) Two-point crossover: 1 child (Vers. I).

(c) Two-point crossover: 1 child (Vers. II).

Fig. 4. Illustration of crossover operators.

4.2 Mutation Operators

We examined the following four mutations used by Murata in [9]: adjacent two-job change (Adj2JC), arbitrary two-job change (Arb2JC), arbitrary three-job change (Arb3JC) and shift change (SC).

We developed a new mutation operator called the arbitrary 20%-job change (Arb20%JC), as we can see in Fig. 5. This mutation selects 20% of the jobs in the child chromosome. The 20% of the jobs to be changed are arbitrarily and randomly selected, and the order of the selected jobs after the mutation is randomly specified.

The percentage in this mutation operator gives the operator some flexibility, i.e. the number of jobs to be changed depends on the size of the chromosome.

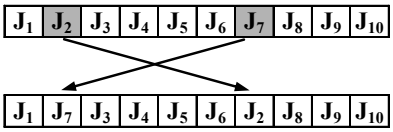


Fig. 5. Arbitrary 20%-job change.

4.3 Examination of Crossover and Mutation Operators

The aim of this subsection is to examine the twelve crossover operators and the five mutation operators, presented in the last two subsections.

When the crossover operators were examined the mutation operator was not used and when the mutation operators were examined the crossover operator was not used. Each crossover operator was examined by using the following conditions: number of tests, 20; initial population Ψ_t, constant; number of jobs, 40; instance used, constant; population size N_{pop}, 20, 40, 60, 80 and 100; stopping condition, 1000 generations; crossover probabilities P_c, 0.2, 0.4, 0.6, 0.8 and 1.0; mutation probabilities P_m, 0.2, 0.4, 0.6, 0.8 and 1.0; mutation operators and mutation probabilities, not used in the examination of crossover operators; crossover operators and crossover probabilities, not used in the examination of mutation operators.

We used the following performance measure with the aim of evaluating each genetic operator:

$$Performance = f(\bar{x}_{initial}) - f(\bar{x}_{end}) \quad \textbf{(1)}$$

where $x_{initial}$ is the best chromosome in the initial population and x_{end} is the best chromosome in the last population. That is, $f(\bar{x}_{initial})$ is the fitness average (of the 20 computational tests) of the best chromosomes in the initial population and $f(\bar{x}_{end})$ is the fitness average of the best chromosomes at the end of the 1000 generations. The performance measure in (1) gives the total improvement in fitness during the execution of the genetic algorithm.

We used 20 computational tests to examine each crossover and mutation operator. The average value of the performance measure in (1) was calculated for each crossover and mutation operator with each crossover probability (P_c), each mutation probability (P_m) and each population size (N_{pop}). Table 1 and Table 2 show the best average value of the performance measure obtained by each crossover operator and by each mutation operator with its best crossover probability, best mutation probability and best population size.

Table 1. Classification of the crossover operators.

Position	Crossover	P_c	N_{pop}	Performance
1st	TPC4C	1.0	100	3834.1
2nd	TPC3CV2	1.0	100	3822.9
3rd	TPC2CV2	1.0	100	3821.8
4th	PBX	1.0	80	3805.8
5th	TPC1CV2	0.8	100	3789.3
6th	CX	0.8	80	3788.7
7th	TPC3CV1	0.8	80	3680.2
8th	TPC2CV1	1.0	80	3662.1
9th	OPC2C	0.6	100	3647.8
10th	OX	1.0	100	3635.4
11th	TPC1CV1	1.0	100	3624.7
12th	OPC1C	0.6	100	3570.5

Table 2. Classification of the mutation operators.

Position	Mutation	P_m	N_{pop}	Performance
1st	Arb20%JC	1.0	100	3833.9
2nd	Arb2JC	0.8	100	3826.4
3rd	Arb3JC	1.0	60	3814.9
4th	SC	0.8	60	3673.5
5th	Adj2JC	0.4	100	3250.4

5 Computational Tests

This section presents the computational results obtained with 40, 50 and 100 jobs. From the OR-Library [10] we randomly selected some instances of SMTWT problems with 40, 50 and 100 jobs. We used 20 computational tests for each instance of the SMTWT problem. We used the six best crossover operators (see Table 1) and the best mutation operator (see Table 2) in the HybFlexGA. Each instance of the SMTWT problem was examined using the following conditions:

- Number of tests: 20;
- Initial population Ψ_t : randomly generated;
- Number of jobs: 40, 50 and 100;
- Instance used: from the OR-Library [10];
- Population size N_{pop}: 80 and 100 (see Table 1 and Table 2);
- Stopping condition: 1000 generations for the instances with 40 and 50 jobs or the optimal solution, and 5000 generations for the instances with 100 jobs or the optimal solution;
- Crossover operators: the six best crossover operators in Table 1;
- Crossover probabilities P_c: 0.8 and 1.0 (see Table 1);
- Mutation operators: the best mutation operator in Table 2;
- Mutation probabilities P_m: 1.0 (see Table 2).

Table 3 shows the computational results obtained for the SMTWT problems with 40, 50 and 100 jobs. In this table we have the number of tests with optimal solution, the CPU time average (in seconds) and the generation average for each instance problem. For example, we chose the TPC4C with P_c=1.0, Arb20%JC with P_m=1.0 and instance 40A (SMTWT problem with 40 jobs, from the OR-Library [10]) in the HybFlexGA. We used 20 computational tests for this instance. In these tests we obtained the optimal solutions in 16 tests. In these 16 tests, the CPU time average was 362.4 seconds and the generation average was 593.

As shown in Table 3, we obtained good results with the TPC4C+Arb20%JC, TPC3CV2+Arb20%JC and TPC2CV2+Arb20%JC combination, for all the instances with 40, 50 and 100 jobs. However, this table also shows the best results are obtained for the TPC4C+Arb20%JC combination.

When we used the TPC4C+Arb20%JC combination, the HybFlexGA is very efficient. For example, in the six instances with 40, 50 and 100 jobs (see Table 3) the HybFlexGA found 20 tests with an optimal solution in four instances (40B, 50A, 50B and 100B), and 16 tests with optimal solutions in two instances (40A and 100A).

Table 3. Computational results obtained for the SMTWT problems with 40, 50 and 100 jobs.

	Instance	**40A**	**40B**	**50A**	**50B**	**100A**	**100B**
	Optimal solution	**6575**	**1225**	**2134**	**22**	**5988**	**8**
TPC4C + Arb20%JC	Tests with optimal solution	16	20	20	20	16	20
	CPU time average (sec.)	362.4	190.0	88.3	45.5	2405.1	523.9
	Generations average	593	284	107	54	1611	323
TPC3CV2 + Arb20%JC	Tests with optimal solution	13	15	18	20	15	20
	CPU time average (sec.)	382.9	231.3	112.3	50.4	3851.0	1012.4
	Generations average	725	402	158	70	3042	727
TPC2CV2 + Arb20%JC	Tests with optimal solution	8	16	17	20	9	20
	CPU time average (sec.)	369.1	216.8	146.2	92.0	3921.3	1339.8
	Generations average	380	449	246	152	3685	1142
PBX + Arb20%JC	Tests with optimal solution	0	8	18	20	1	20
	CPU time average (sec.)	----	236.1	144.6	86.3	2067.0	1413.6
	Generations average	----	747	371	218	2957	1828
TPC1CV2 + Arb20%JC	Tests with optimal solution	0	3	13	20	1	19
	CPU time average (sec.)	----	230.0	224.7	118.2	4104.0	2190.7
	Generations average	----	609	487	252	4948	2405
CX + Arb20%JC	Tests with optimal solution	0	4	17	20	3	20
	CPU time average (sec.)	----	165.3	107.8	51.0	2594.0	736.4
	Generations average	----	551	292	136	3912	1011

6 Conclusion

In this paper we developed an FMC and a software tool called HybFlexGA to study the scheduling of jobs in FMC with GA. We propose a new concept of genetic operators for scheduling of jobs in FMC. With the software tool HybFlexGA, we

examine the performance of various crossover and mutation operators by computing simulations on job scheduling problems. The HybFlexGA obtained good computational results in the instances of SMTWT problems with 40, 50 and 100 jobs (see Table 3). As we demonstrated, the HybFlexGA is very efficient with the TPC4C+Arb20%JC combination. With this combination, the HybFlexGA always found more optimal solutions than with the other combinations: TPC3CV2+Arb20%JC, TPC2CV2+Arb20%JC, and so on. When we used this combination (TPC4C+Arb20%JC) in the HybFlexGA, the genetic algorithm required fewer generations and less CPU time to find the optimal solutions. The results obtained in the scheduling problems of the FMC show how efficient HybFlexGA is in solving these problems.

7 References

1. Morton, E. Thomas and David W. Pentico: Heuristic Scheduling Systems. John Wiley & Sons, (1993)
2. Potts, C.N. and L.N. Van Wassenhove: Single Machine Tardiness Sequencing Heuristics. IIE Transactions, vol. 23, no 4 (1991) pp. 346-354
3. Peter A. Huegler and Francis J. Vasko: A performance comparison of heuristics for the total weighted tardiness problem. Computers & Industrial Engineering, vol. 32, no 4 (1997) pp. 753–767
4. D.E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
5. Ferrolho A. and Crisóstomo M.: Scheduling and Control of Flexible Manufacturing Cells Using Genetic Algorithms. WSEAS Transactions on Computers, vol. 4 (2005) pp. 502–510
6. Ferrolho A. and Crisóstomo M.: Flexible Manufacturing Cell: Development, Coordination, Integration and Control. Proc. of the IEEE 5th Int. Conference on Control and Automation, Hungary (2005) pp. 1050-1055
7. J. Oliver, D. Smith and J. Holland: A study of permutation crossover operators on the traveling salesman problem. Proc. of the Second ICGA (1987) pp. 224–230
8. Syswerda: Scheduling optimization using genetic algorithms. L. Davis (Ed.) Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York (1991) pp. 332–349
9. T. Murata and H. Ishibuchi: Positive and Negative Combination Effects of Crossover and Mutation Operators in Sequencing Problems. Proc. of the IEEE Int. Conf. Evol. Computation, Piscataway (1996) pp. 170–175
10. http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html

Decision Support Systems Intelligent to Predict and Optimize the Assembly Time for New Product Design Using a Fuzzy-Evolutionary Multimodal Approach

Pérez V. Pedro [1], Torres-Trevino Luis [1], Carrum S. Elias [1] and Ramirez C. Sergio [2]
[(1)]Corporación Mexicana de Investigación en Materiales
Blvd. Oceanía 190, Saltillo Coah. México
01 52 (844) 411 32 00
pperez,ltorres,ecarrun@comimsa.com.mx

[(2)]Instituto Tecnológico de Saltillo
Blvd. V. Carranza 2400 Saltillo Coahuila México
sramirez@fenix.its.mx

Abstract. In this paper three case studies are presented applying a Fuzzy-Evolutionary Multimodal approach to predict and optimize the prediction assembly time, this methodology can be a good complement of the design for assembly methodology of Boothroy-Dewhurst. This hybrid method analyzes different situations and conditions, identifies sources of variability that should be avoided during the process design because it increases the assembly time. The evolutionary algorithm searches the set of inputs or conditions in the fuzzy model and uses the predicted time to evaluate every set of inputs generated, the best tree conditions with and without constrains design are found by this approach. The research show the advantages of use a Fuzzy-Evolutionary Multimodal approach in order to analyze the variables behavior to predict, control and optimize the assembly time in the new product design.

Key words: Fuzzy Logic, Evolutionary Strategies Multimodal, Design for Assembly

1 Introduction

The Product design involves a complex group of activities which are affected by diverse sources of variability that should be detect and analyzed carefully to avoid them in the design. A bad product design impact on the functionality, fitness and the customer satisfaction. The assembly time is a critical activity and is very important to select the appropriate assembly method to reduce cost and time. Historically, motion time study has been used successfully in time measurement of product, the problem is predict the assembly time of new products because many assembly parts have not been built yet. It causes the need to research and use new methodologies and new technologies to improve the new product design.

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 41-51

Received 10/08/06
Accepted 03/10/06
Final version 13/10/06

The researchers Boothroyd and Dewhurst [1] have developed a successful method to help designers to redesign their products, to reduce costs, improve performance and reduce assembly time; this methodology is called Design for Assembly (DFA). In this methodology the assembly time is divided into two categories, one generated by the insertion time and the second one generated by the handling time or transfer of every one of the components until achieving the total assembly.

The DFA Boothroy methodology has had great success in a good number of leadings design companies around the world [2, 5]; unfortunately it does not have all the characteristics required in the industry [4, 2]. Others inconveniences of the methodologies are that it is not flexible to different applications and is subject to find the assembly time values in charts (See **Table.1**), it does not have tolerance to lack of entrance data, and it does not have self-learning capacity based on data and previous events.

Table 1. Information to predict manual handling assembly time with one hand without the aid of grasping tools – estimated time (seconds) [1].

		No handling difficulties			Part nests or tangles		
		Thickness > 2 mm		< 2 mm	Thickness > 2 mm		< 2 mm
sym(deg)= (alpha + beta)		size > 15 mm	6 mm≤ size ≤15 mm	size > 6 mm	size > 15 mm	6 mm≤ size ≤15 mm	size > 6 mm
		0	1	2	3	4	5
sym<360	0	1.13	1.43	1.69	1.84	2.17	2.45
360<=sym<540	1	1.5	1.8	2.06	2.25	2.57	3.0
540<=sym<720	2	1.8	2.1	2.36	2.57	2.9	3.18
sym=720	3	1.95	2.25	2.51	2.73	3.06	3.34

To aid and solve this problem an intelligent system is proposed, it is a Fuzzy-Evolutionary Multimodal (FEM) approach to predict and optimize the assembly time in a new product. This intelligent system will serve as a decision support system; the goal is to predict the assembly time using fuzzy logic and evolutionary strategies to optimize the assembly time of new products giving the system the best tree options and choose the better for the design, before send the drawings to production.

The object of this research is contribute to the decision support process predicting and optimizing the assembly time improving the design and innovation of products.

This work is organized in the following way: On section 1 a brief introduction will be given, the section 2 has a short summary of previous research using intelligent systems in the manufacturing and assembly process. The research tools foundations, Fuzzy Logic and Evolutionary Strategies, are shown in section 3, experimentation and results are shown in section 4, the research conclusions and future work are given in sections 5 and 6 respectively.

2 Literature Review

In the last two decades, the application of intelligent systems in manufacturing has been subject of extensive research. Different intelligent systems are being proposed to improve the new product design product optimize the manufacturing and the process quality [6,10]. This technology has become in a important computing tool to solve engineering problems [6, 10, 11]. It has led to increasing research on a wide variety of industrial applications, such as product manufacturability control, process planning, etc. The application of this technology on the assembly and disassembly areas is actually in research and it is possible to find intelligent systems using fuzzy logic and neural networks to improve the new product design, [9, 12].

Great utility has been demonstrated the use of Fuzzy Logic, evolutionary computation and neural networks in the parametric design. In these designs a great number of variables and interrelations are involved which generate a great quantity of completely unknown parameters [11]. Fuzzy Logic technology has the ability to handle lexical uncertainties. These systems have been proposed in order to reach a better design process and to improve product design quality [10, 9].

The application of intelligent systems to predict the assembly time and optimize the assembly sequence is on research, this area have many variety opportunities to make new developments and new applications. This research proposes a approach to predict and optimize the assembly time in new products. This hybrid method use Evolutionary Strategies Multimodal to optimize the estimated time by the fuzzy system. (See Flow Diagram **Fig. 1**).

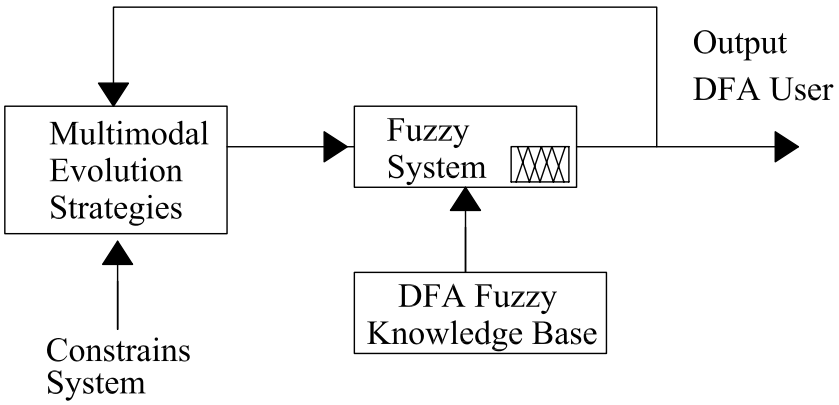


Fig. 1 Fuzzy-Evolutionary approach flow diagram, the output of DFA system is optimized by evolution strategies multimodal.

3 Research tools

3.1 Fuzzy Logic

The Fuzzy logic was developed by Lofti Zadeh a professor from Berkeley University in California who did not agree with the unique existence of classical sets (crisp sets) [4, 5]. Fuzzy logic technology has the ability to handle lexical uncertainty with imprecision, it is very common in most human words used to evaluate concepts and to get conclusion compared to the conventional logic which works with well defined information; this is a methodology that gets a conclusion from poor, ambiguous or not precise information. In general the fuzzy takes decisions based on information with previously mentioned characteristics.

A basic principle of the fuzzy set is called membership relations it is something that cannot be determined easily. The fuzzy logic theory is based on the principle that the group is not always defined clearly, a classic example is the person height, and it can be classified as tall, medium or short.

Definition 3.1 (Fuzzy Set): A fuzzy set A on universe X is a set defined by the membership function μA(x) mapping from the universe X into the unit interval:

$$\mu A(x) : X \rightarrow [0,1] \quad \textbf{(1)}$$

$f(x)$ Denotes the all fuzzy sets on X. If the value of the membership function, ca lled the membership degree $(\mu A(x))$, equals one, "x" belongs completely to the fuzzy set. If it equals zero, "x" does not belong to the set. If the membership degree is between 0 and 1," x" is a partial member of the fuzzy set:

$$\mu A(x) = 1 \quad x \text{ is a full member of } A$$
$$\mu A(x) = (0,1) \quad x \text{ is a partial member of } A$$
$$\mu A(x) = 0 \quad x \text{ is not member of } A$$

A fuzzy inference system has the capacity of processing numeric and linguistic variables, giving a numeric variable or linguistics as a result. (See **Fig. 2**).

The proceeding to build a fuzzy system is.

1. Inputs and outputs definition
2. Linguistic values definition associated to the input and output required
3. Membership function definition
4. Logical operators definition
5. Rules development and evaluation
6. Defuzzification of the system
7. System output

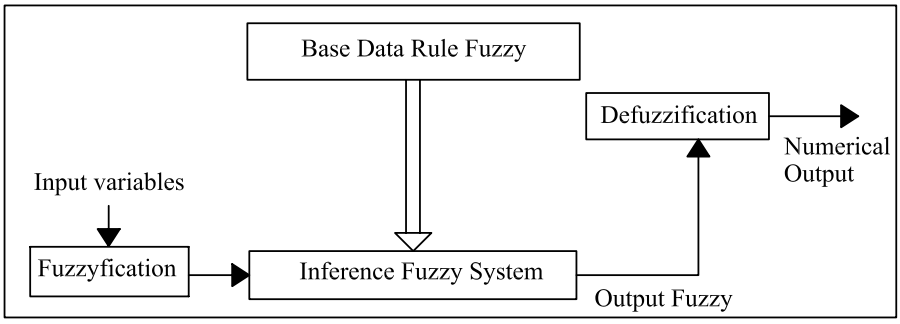


Fig. 2 Mandany Type Fuzzy evaluation process

The key point is to transform all the linguistic variables into numeric variables and process them by logical formulas or rules. The rule has the following structure type:

If (The weight piece is heavy)
And (the piece size is big) **and** (the handling method is two hands with tools)
Then (the assembly time is high).

3.2 Evolution Strategies

Evolution strategies (ES) were developed by Bienert, Rochenberg and Schwefell from the Technical University of Berlin, Germany in the 1960[14], it was inspired by the evolution that an individual can suffer and how he can be affected during the mutation and recombination processes of the ancestors genes. The evolution strategies are similar to the genetic algorithms (AG), but they have two main differences: In ES it is not necessary to code solutions, and the main operator is the mutation, while in AG is necessary to use codification and the main operator is the crossover.

There are several ES algorithms, some use recombination, another consider correlations between variables [2]. In the methodology it is used the most simple version of the algorithm, the ES without recombination. The algorithm is the following.

1) Generation of a population of μ parents
2) Generation of λ off springs from μ parents
3) Evaluation of off springs
4) Selection of the best μ off springs to be new parents.
5) Repeat (2), (3), and (4) until N generations reached

All the individuals (parents and off springs) have a mutation parameter that changes through time (self-adaptation). The generation of offspring is only by mutation according to:

$$O\,i \le O\;\; i + N(0,k) \tag{2}$$

Where i indicates the parameter involved and $N\,(a,b)$ is a random number generator with normal distribution with mean and standard deviation b. Every mutation parameter k changes according to:

$$k\quad i \le k\;\; i * \exp(\tau * Nt\; i + \tau p * N(0,1) \tag{3}$$

Where i indicate the parameter involved, $\tau = 1/sqrt(2 * NT\,\mathrm{Pr})$,
$\tau p = 1/sqrt(2 * sqrt(NT\,\mathrm{Pr}))$ and
$NT = N(0,1)$ are used without change for all the parameters.

The evolutionary strategies algorithm with multimodal property uses a selection procedure where only similar individuals compete. The distance between the solutions vectors is calculated by the Euclidean distance. The solution promotes the population diversity.

4 Experimentation and Results

Three study cases were developed to analyze the performance of the Fuzzy-Evolutionary multimodal approach to predict and optimize the assembly time in a new product. In the first case the assembly time was predicted using fuzzy logic, in the second case using the same data base used in case 1 and the resultant assembly time as the output variable it was optimized by evolution strategies multimodal to find the conditions to get the best tree minimum assembly time. In the third case was added two restrictions design and the goal was detect the best three assembly time prediction.

4.1 Study Case 1: Assembly Time Calculations by Fuzzy Logic

The assembly time is classified in two categories, insertion time and handling time [1]. The fuzzy logic method is used for manipulating the characteristic fuzziness of insertion and handling difficulty problems. In this research we will focus on time handling estimation,

Using the assembly people's experience of a local company, it was possible to develop a database. Different ranges were established for each variable, according to the assembly features of a product. (See **Table 2**). The codification used for the "symmetry variable", "assembly methods" and accessibility were in a range between 0 through 1. The variable that has the highest value between zero and one could correspond to the one having the most critics, the heaviest or the most difficult accessibil-

ity. In the assembly time the ranges were: Lowest time 10-25 min, normal time 15-35 min and the highest time upper 35 minutes.

Table 2. Data base, showing ranges for each variable used in this research, according to assembly features of a product.

Definition of linguistic variables

Weight	Normal	0-2 kgs
	Heavy	0.75-3.25 kgs
	Very heavy	2-4 kgs
Part size	Small	0-7 cms
	Medium	2-12 cms
	Large	7-17 cms
	Very Large	17-25 cms
Part symmetry	Complete	0 - 0.3
	Symmetric	0.1 - 0.5
	Symmetry acceptable	0.3 – 0.7
	Asymmetric	0.5 - 1
Handling_methods	One_hand/without_tools	0 - 0.3
	Two_hands / without_tools	0.1 - 0.5
	One_hand/with_tools	0.3 – 0.8
	Two_hands / without_tools	0.5 - 1
Easy to reach	Far	0 - 0.5
	Medium	0.2 - 0.8
	near	0.5 - 1
Time	Low time	10 – 25 min
	Medium time	15 – 35 min
	High time	over 35 min

It was very important to decide the appropriate membership functions for each linguistic variable that should be used in the research. Several researches have suggested many advanced types of memberships functions for linguistic variables [3, 4], in this paper, we decide to use the following types of membership functions: Trapezoidal-type and Triangular-type, the defuzzification method used in this research was the center of maximum (CoM). See **Fig. 3**.

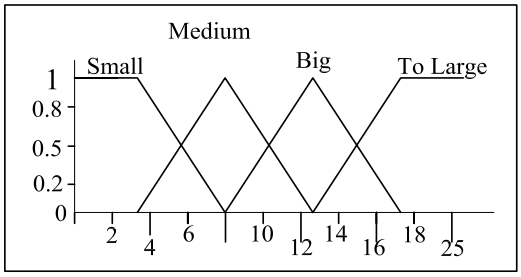


Fig. 3 Belonging degrees of dimension variable Inference rule based on experience

Based on this information, different simulations with different conditions were done for all variables. In the Fuzzy Logic system was possible to calculate the assembling time in a fast and precise way without looking up information from tables /charts as it is suggested in the methodology proposed by Boothroy [1].(See results **table 3**).

Table 3. Results of study case No.1 to estimate the assembly time.

Weight	2 Kgs.
Dimensions	25 cms
Part symmetry	acceptable
Handling Methods	One hand /with tools
Easy to reach	Medium
Easy to insert	Medium
Total assembly time	25.4 seg.

By fuzzy logic is possible predict the assembly time and can be usefully adapted to different companies.

4.2 Study Case 2: Optimizing the Assembly Time by Fuzzy-Evolutionary Multimodal approach

To get a minimum assembly time, suitable conditions are required. The ES algorithm searches the best input value that maximizes or minimizes the assembly estimated time. To develop the second study case was used the same fuzzy data base, the target now was to find the three different inputs combinations that will produce the best minimum assembly time. The output of the fuzzy logic system was optimize by evolutionary multimodal getting the tree best options that produce the best minimum assembly time, was used the Euclidian distance to find the near neighbor. The results are shown at **Table 4.**

Table 4 Results of study case No.2 shows the minimums assembly times

Weight	3.87 Kgs.	3.84 Kgs.	2.51 Kgs.
Dimensions	3.41 cms	7.38 cms	9.36 cms
Part symmetry	asymmetry acceptable	asymmetry	asymmetry acceptable
Handling Methods	Two hand /with tools	Two hand /with tools	One hand with tools
Easy to reach	Near	Medium	Medium
Easy to insert	Easy	Medium	Medium
Total assembly time	7.62 seg.	8.47 seg.	9.28 seg.

With this information is possible have tree alternatives to design the part according our convenience, so with this characteristic we assure a minimum assembly time

4.3 Study Case 3

To develop the third study we continue using the same data base the difference now was that the input variables has two constrains of design, the dimension part must have a size between 8 and 20 centimeters and the weight should be between 2 and 3 Kilograms, the results are shown at **Table 5.**

Table 5 Results of study case No.3. The target was to find the inputs variables that produce the best minimum assembly time, according to the constrains and could have tree possible alternatives to design the part.

Weight	2.95 Kgs.	2.0 Kgs.	2.59 Kgs.
Dimensions	8 cms	17.2 cms	15.3 cms
Part symmetry	asymmetry acceptable	asymmetry	asymmetry acceptable
Handling Methods	Two hands /with tools	Two hands /with tools	One hand with tools
Easy to reach	Medium	Medium	Medium
Easy to insert	Easy	Medium	Medium
Total assembly time	8.42 seg.	9.0 seg.	9.5 seg.

So it is preferable the parts with this characteristic because its variables assure a minimum assembly time. This Fuzzy-Evolutionary Multimodal approach can easily used to optimize the assembly time product and it can be adapted to different industrial situations.

5 Conclusions

1. We have shown how fuzzy logic can help to predict the assembly time and how it could be used as a complementary technique to design for assembly methodology (DFA). The following are the main advantages and disadvantages found when developing the fuzzy logic system.

Advantages of the fuzzy system

- It is easy to incorporate qualitative linguistic information.
- It is a robust system because in absence of a rule, it will be replaced by others.

Disadvantages of the fuzzy system

- If the entrances to the system are increased, the rules are increased.
- Hard to integrated quantitative knowledge to a function.

2. The research showed the advantages of using a Fuzzy-Evolution Multimodal approach to analyze the variables behavior to predict, control and optimize the assembly time in a new product. With this hybrid method is possible to find the input variables that produce the best minimum and worst assembly time. It is very important to find the conditions that a piece must have in order to achieve reduced assembly time. With this methodology these conditions can be easily found.

Finally, It is possible conclude that the Fuzzy-Evolutionary Strategies method is a suitable methodology to predict and optimize the assembly time of new products.

6 Future Work

It is convenient to keep on working in techniques to supplement these methodologies, analyze different evolutionary algorithms. To continue improving results is possible and recommendable to integrate evolutionary multiobjetive algorithms.

To improve the Fuzzy-Evolutionary Multimodal approach, will be increasing the data base rules is necessary to get new rules to improve the performance of the methodology. The use of type II fuzzy systems can get a better performance of the proposed system because it can consider uncertain and tolerances of the assembly components.

References

1. Boothroy Geoffrey, Dewhurst Winston, Product Design for Manufacturing and Assembly New York: Marcel Dekker (1994).
2. Back Thomas and Schwefel Hans, Evolutionary Algorithms in theory and practice. Oxford University Press, New York. (1996)

3. Jin and B. Sendhoff, Knowledge incorporation into neural networks from fuzzy rules. Neural Processing Letters, vol. 10, 231-242. (2001)
4. Yen John, Reza Langari, Fuzzy Logic Intelligence, Control and Information Prentice Hall. (1999)
5. Mascle, Christian. Decision Support in a Design for Assembly and Disassembly, International Symposium on Assembly 0-7803-7770-2/03 IEEE. (2003)
6. Whitney Daniel E. Mechanical Assemblies, Oxford University Press. ISBN:0-19-515782-6 (2004).
7. Meziane, F., Vadera, S. (2000) Intelligent systems in manufacturing: current developments and future Integrated manufacturing Systems 11 218-238. (2000).
8. W. He, Y.F. Zhang, K.S. Lee and T.I. Liu. February. Development of a fuzzy-neuro system for parameter resetting of injection molding. Transactions of the ASME. Vol. 123. (2001).
9. Galantucci L. and Percoco G. (2004) Assembly and Desassembly planning by using Fuzzy Logic and Genetic Algorithms International Journal of Advance Robotic Systems.
10. Geromei, Luis H, Applications of intelligent systems in power transformer design CEFET/PR0-7803- 7278-6/02 IEEE. (2002)
11. Gallo S. and Murino T. Time manufacturing prediction: In Neuro Fuzzy Expert System. 7th European Congress on Intelligent Techniques. (1999)
12. Huang,Hsin-hao and Johnson M.R. Disassembly secuence generation using a neural network approach. Journal of manufacturing systems vol. 19 No.2. (2000).
13. W. He, Y.F. Zhang, K.S. Lee and T.I. Liu February Development of a fuzzy-neuro systemfor parameter resetting of injection molding. Transactions of the ASME. Vol. 123. (2001).

Maximum Entropy Stochastic Tasks Classification

Miguel A. Toledo C. and J. Jesús Medel J.

Instituto Politécnico Nacional, CIC
jjmedelj@yahoo.com.mx, matoledo@uvmnet.edu

Abstract. In this work we introduce a task classification model for soft real-time systems. Task processes are analyzed as probabilistic distribution functions which parameters are considered as stationary random variables over the long time. We present a technique for tasks classification based on the maximum entropy level as element of differentiation among tasks. The classification model is described and evaluated based on probabilistic distribution functions properties.

1 Introduction

In computer systems, the term "tasks" is referred to those processes which are executed in servers whose perform a specific action. Tasks are software entities that aim to respond to the information generated by Real-time systems; peripheral devices or internal processes [1]. In applications with explicit timing constrains and logical correctness, tasks completion algorithms that accomplish these requirements has been purposed in many works. Tasks scheduling involve certain knowledge of tasks characteristics. Since its earliest definitions, tasks have been classified as: periodic, semiperiodic, aperiodic or sporadic for various authors according the analysis context [1],[7], [8], [16]. In most of cases, task classification is established priory. Periodicity in generated events mark a classical division among tasks types.

Scheduling in real time systems concerns the determination of a temporal ordering of tasks within constraints of some specified timing, precedence and resource requirements. Depending on the nature of the application, scheduling may be classified into different kinds: static scheduling, dynamic scheduling and mixed scheduling [8].

Static scheduling assumes prior knowledge of the relevant characteristics of all tasks, which may be taken into account in scheduling. When this information is available priority ordering is assigned to tasks. However, any change in tasks requires the complete rescheduling. On the other hand, Dynamic scheduling algorithms are designed to work with unpredictable arrival times of tasks and possible uncertainties in their execution times or deadlines; priority assignation is done in line; that is, tasks priority is assigned at the task arrival time.

Assign priority ordering only have sense when tasks are preemptable. A task is preemptable by other task with higher priority if it can be interrupted and resumed later meanwhile its overall goal can be achieved. In dynamic scheduling algorithms, tasks assigned priority vary over time from request to request, but in a statistical way the priority tasks ordering remains the same.

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 53-59

Received 08/07/06
Accepted 07/10/06
Final version 10/10/06

Tasks priority ordering depends mostly on real time constraints; such as arriving time or deadline. Scheduling based on priority ordering aims to deal with these constraints optimally. In dynamic priority scheduling algorithms, where tasks characteristics are known at the time they arrives, statistical knowledge of their process is required.

In stationary systems which maintain its characteristics over long time, it is possible to determine some tasks performance information based on their statistics. Tasks process outcomes can be described by density distribution function with defined parameters. Probability does not say much about individual events, but describes a faithfulness level of task predictability. Meanwhile tasks processes repeat during an infinite number of times. Some of these relevant variables are: incoming channel, inter-arrival time or period, relative deadline and, computations or communications streams. Here tasks processes are considered stochastic stationary processes with defined probability distribution function; likewise task statistical behavior can be described. In soft-real-time systems, it is statistically required that task processes be completed in their deadlines. This means that a statistical distribution response time service is acceptable.

While the system evolves in time, tasks knowledge increases. A measure of amount of uncertainty can be represented in probability with the maximum entropy. That is, the information is more informative kind, if it consists of mean values of task variables. Therefore, the analysis of tasks processes estimate a measure of the degree to which a system is schedulable. One such measure is the task classification on maximum entropy level what is the focus of this work.

The purpose of this paper is introducing a scheduling theory based on tasks continuous stochastic processes for soft-real-time systems. Our aim is give an entropy based classification for tasks treatment. As well as task classification is generated while the process evolves. Tasks classification helps us in scheduling algorithms based on tasks characteristics.

2 Preliminaries

In this section, we define task functions as continuous time stochastic processes. This approach let us to handling any kind of task; where periodicity in task time arrival becomes a density distribution function. Based on M/M/1 queueing theory [4], tasks time constraints; arriving and deadline time are independently assigned according to a probability distribution. Next assumptions are made in order to simplify out model, but they can be treated in future works.

1. An task becomes a active an instant after it arrives.
2. All tasks are preemptive and overhead are depreciable.
3. Deadline is result of arrival time and calculus time.
4. Start time, lateness and precedence are considered nulls.

Let the tuple $(\Omega, \mathcal{F}, \Pr)$ be the probability space that describe task process. The Ω set throws outcomes out during the sample time. Here $\mathcal{F}$ is the set field in Ω. Then $\Pr$ is the probability measure function in this space. Tasks constraints are random variables generated by a random variables family whom are none decreasing in time [9], [10].

Definition 1. *(Real-time Task). A real-time task is a stochastic process described by* $\pi\{A(t,\omega), I(t,\omega), D(t,\omega)\} \in \mathcal{F}$*, characterized by the random variables family* A, I *and* D *defined in the probability space* $(\Omega, \mathcal{F}, \Pr)$ *with* $t \epsilon T\ t > 0$*, and* $\omega \subset \Omega$*.*

Definition 2. *(Arrival Time). The arrival time is a stochastic process* $\{A(t,\omega)\} \in \mathcal{F}$ *defined as the required waiting time to observe an arrival occurrence when this outcomes at rate* λ *time units, is described by an exponential density function.*

$$\begin{aligned} A(t,\omega) &= \frac{1}{\lambda} e^{-\frac{1}{\lambda}t} \\ A(t,\omega) &> 0 \end{aligned} \tag{1}$$

Definition 3. *(Task Information). The information contained in tasks that will be executed according to server performance is a stochastic process* $\{I(t,)\} \in \mathcal{F}$ *described by an normal density function.*

$$\begin{aligned} I(t,\omega) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{t^2-\mu^2}{2\sigma^2}} \\ I(t,\omega) &> 0 \end{aligned} \tag{2}$$

Definition 4. *(Deadline). The time when a task must finish its service is a stochastic process* $\{D(t,)\} \in \mathcal{F}$ *described by the sum of* $A(t,\omega)$ *and* $I(t,\omega)$*;* $D(t,\omega) = A(t,\omega) * I(t,\omega)$ *Thus the normalized Deadline distribution function is defined by*

$$\begin{aligned} D(t,\omega) &= \frac{exp(-\frac{1}{2}\frac{\sigma^2+2t\lambda-2\mu\lambda}{\lambda^2})}{\lambda exp(\frac{\sigma^2}{2\lambda^2}+\frac{\mu}{\lambda})} \\ D(t,\omega) &> 0 \end{aligned} \tag{3}$$

Task information is taken as the time unit to determine the calculation time. Based on the calculation time is a product of the task information and a constant of process capability of the system. So that, in following developments theses will be teated as synonyms. As independent random variables, the deadline process is the sum of the arrival time and the task information or calculation time.

3 Maximum Entropy Principle

In 1948 Claude Shannon [14] published his theory of communication; where he derives a measure of uncertainty denoted entropy. Referring to some system with certain physical or conceptual entities, the messages they produce have meaning. Shannon proposed a measure of how much information is "produced" by such a process, or better, at what rate information is produced.

The measure $H(p_1, p_2, ..., p_n)$ of a set of possible events whose probabilities of occurrence are $p_1, p_2, ..., p_n$. The entropy is the measure function of the uncertain of an event outcome. It is reasonable to such a measure to require of it the following properties:

1. H should be continuous in the p_i.

2. If all the p_i are equal, $p_i = 1/n$ then H should be a monotonic increasing function of n. With equally likely events there is more choice, or uncertainty, when there are more possible events.
3. If a choice be broken down into two successive choices, the original H should be the weighted sum of the individual values of H.

Definition 5. *(Entropy). The only function H that satisfies the three above assumptions is of the form:*

$$H(p_1, p_2, ..., p_n) = -K \sum_{i=1}^{n} p_i \log p_i \tag{4}$$

The form $H = \sum p_i \log p_i$ is recognized as entropy for probabilities p_i because of the mathematical similarity with the thermodynamical definition of entropy. Typically, K is taken as unity and the logarithm either in base 2 or Napierian (natural) base. In an analogous manner the entropy of a continuous distribution with the density distribution function $p(x)$ is given by:

$$H(x) = -\int_{-\infty}^{\infty} p(x) \log p(x) dx \tag{5}$$

For task processes with known density distributions; mathematical maximization techniques like Lagrange multipliers are used to determine a measure of certainness for continuous random variables based on the maximum entropy principle.

Theorem 1. *(Maximum task arrive entropy). Lets $A(t, \omega)$, $t > 0$, be the arrival time of a task, subject to the following constraints:*

1. $\int_0^{\infty} A(t, \omega) dt = 1$
2. $\int_0^{\infty} tA(t, \omega) dt = \lambda$ *for* $\lambda > 0$
3. $H(A(t, \omega)) = -\int_0^{\infty} A(t, \omega) \log A(t, \omega) dt$

The maximum entropy for $A(t, \omega)$ variable with pre-specified first moment in $[0, \infty)$ is $H_{i,M}$ given by

$$H(A(t, \omega)) = \log \lambda e \tag{6}$$

Proof. In stochastic sense, the arrival time function $A(t, \omega)$ is defined as an exponential type function, [4] and [6]. Using the method of Lagrangian multipliers, the density distribution that satisfies the constraints 1) and 2), is the one described in DEF 2. The associated maximal entropy is then obtained by $H(A(t, \omega)) = -\int_0^{\infty} \frac{1}{\lambda} e^{-\frac{1}{\lambda} t} \log \frac{1}{\lambda} e^{-\frac{1}{\lambda} t} dt = ln\lambda \int_0^{\infty} \frac{1}{\lambda} e^{-\frac{1}{\lambda} t} dt + \frac{1}{\lambda} \int_0^{\infty} \frac{t}{\lambda} e^{-\frac{1}{\lambda} t} dt$ $H(A(t, \omega)) = ln\lambda + 1$

Theorem 2. *(Maximum task deadline entropy). Lets $D(t, \omega)$, $t > 0$, be the deadline of a task, subject to the following constraints:*

1. $\int_0^{\infty} D(t, \omega) dt = 1$
2. $\int_0^{\infty} tD(t, \omega) dt = \lambda$ *for* $\lambda > 0$
3. $H(D(t, \omega)) = -\int_0^{\infty} D(t, \omega) \log D(t, \omega) dt$

the maximum entropy for $D(t,\omega)$ variable with pre-specified first moment in $[0,\infty)$ is $H_{i,M}$ given by

$$H(D(t,\omega)) = \log \lambda e \tag{7}$$

Proof. Deadline defined by the convolution between task information and arrival time distribution functions where $D(x,\omega) = \int_{-\infty}^{\infty} A(t,\omega)I(x-t,\omega)dt$ normalized is considered. Thus, the density distribution that satisfies the constraints 1) and 2), is the one described in DEF 4. The associated maximal entropy is then obtained by $H(D(x,\omega)) = -\int_0^{\infty} D(x,\omega)\log D(x,\omega)dt$, this yield $H(D(x,\omega)) = ln\lambda + 1$

4 Maximum Entropy Tasks Classification

In previous section we establish a probabilistic criteria for task classification in dynamic algorithms, according with tasks temporal constraints. As in firsts works in the area, periodicity and deadline constraints have been considered bases on task planning algorithms.

A main task property is the period. In Liu and Lyland pioneer analysis, [7], the length of successive tasks is a constant called period T. A periodic task is said to have a regular release time, or it is regular time triggered. If a task does not accomplish with this criteria is called a nonperiodic or sporadic task. Task periodicity plays a relevant role in scheduling algorithms because of the assumptions considered in its analysis; in order to have the expected results [5]. Periodicity is considered as a measure of task order timely speaking. A higher hierarchy task corresponds a higher periodicity task. What is defined by the task arrival distribution function.

Deadline property is considered in dynamic scheduling as the EDF algorithm [15]. By means of the processor utilization factor, task priority is selected by the absolute deadline. Tasks with earlier deadlines will have higher priorities.

Definition 6. *(Hierarchy in Probabilistic Sense). Let $J_M = J(\pi(t,\omega))$ be the task parameter assigned that describes the relative importance among tasks in the system*

$$J(\pi(t,\omega)) = min \begin{cases} H(A(t,\omega)) \\ H(D(t,\omega)) \end{cases} \tag{8}$$

Theorem 3. *(Tasks Hierarchy based on Maximum Entropy). Lets $\pi_i(t,\omega)$ be a task process with arrival time $A_i(t,\omega)$, $I_i(t,\omega)$ and $D_i(t,\omega)$. Task hierarchy of $J_{i,M}$ respect to $J_{j,M}$ according with the maximum entropy principle is established by*

$$J_{i,M} > J_{j,M} \qquad \textit{if and only if} \qquad H_{i,M} < H_{j,M} \tag{9}$$

Proof. Results of THEO 1 and THEO 2 relate maximum entropy measure with the distribution density mean for the task parameters under consideration. Accomplish with inequality $H_{i,M} < H_{j,M}$ give the hierarchy inequality

5 Simulation Results

Assigning hierarchy to task processes is a first step in task planning due to the relevance of classify tasks in dynamic scheduling algorithms. That is, while tasks arrive and we have enough information about their probabilistic parameters, we are able to distinguish those tasks that must be served first from the others.

A simulation algorithm for task classification using the maximum entropy principle is presented. Here we take tasks arriving time samples from whom we observed their statistical parameters during the simulation time. With the first probability moment values we applied the classification based on maximum entropy principle in order to give them a relative hierarchy giving an order of tasks to be processed by the server.

In Figure 1 a simulation outcome is showed. Here, the simulation algorithm is tested for three tasks with different first probability moments which arriving task density distribution are stationary. Moreover, the tasks classification is presented for these task samples giving them a priority value in function of maximum entropy certainness measure.

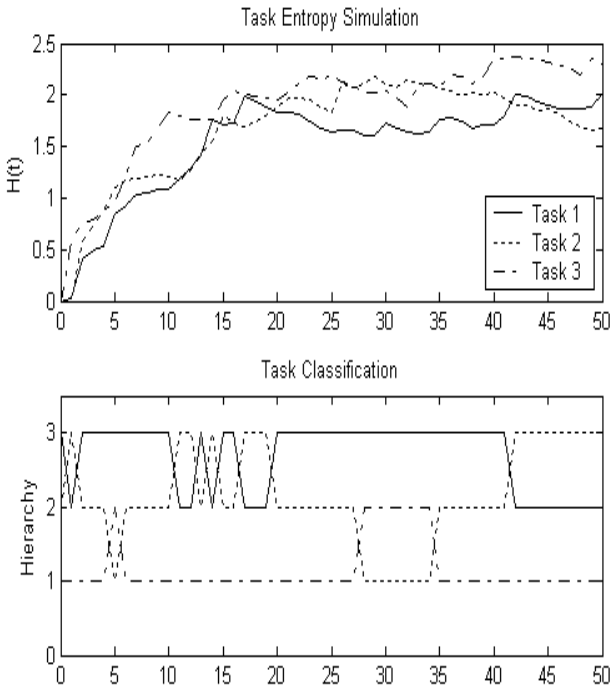


Fig. 1. A tasks classification simulation outcome

We observe from the tasks hierarchy values for figure 1), that while the simulation time evolve; these values remains unchanged because of the information gathered from tasks.

6 Summary and Conclusions

In this paper the priority rule problem for task planning is treated. Assigning priority to tasks based on the maximum entropy principle let us deal with tasks with differences in their statistical parameters. The technique presented here is applied to not only periodic tasks but aperiodic tasks also. A remarkable result of this work is the classification method, in relation with tasks periodicity and deadline constraints is a similar manner.

Based on the maximum entropy level as element of differentiation among tasks, this theory gives a continuous analysis in task planning processes. We have modeled soft real-time systems as tasks processes with stationary distribution functions. This analysis establish the basis for scheduling algorithms future works.

References

1. G. Buttazzo. *Hard Real-time Computing Systems*. Kluwer Academic Publishers, Norwell, MC, USA, 1997.
2. K. Jeffay and S. Goddard. A theory of rate-based execution. *in Proceedings of the 20th IEEE Real-Time Systems Symposium*, 1999.
3. K. Jeffay and D. Stanat. On non-preemptive scheduling of periodic and sporadic tasks. *in Proceedings of the 12th IEEE Real-Time Systems Symposium*, 1991.
4. J. P. Lehocky. Real-time queuing network theory. *in Proceedings of the 18th Real-Time System Symposium*, pages 58–67, December 1997.
5. S. L. Lehoczky, J. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. *in Proceedings of the IEEE Real-Time Systems Symposium*, December 1989.
6. T.-H. Lin and W. Tarng. Scheduling periodic and aperiodic tasks in hard real-time computing systems. *Performance Evaluation Review*, 19(1):31–38, May 1991.
7. C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of ACM*, 20(1), 1973.
8. N. Nissanke. *Real-time Systems*. Prentice Hall, Great Britain, 1997.
9. A. Papoulis. *Probability, Random Variables, and Stochastic processes*. McGraw-Hill, 1965.
10. E. Parzen. *Stochastic Processes*. Holden-day, 1962.
11. F. Reza. *An Introduction to Information Theory*. Dover Publications, NY, USA, 1994.
12. K. Schwan. Dynamic scheduling of hard real-time threads. *IEEE Transactions on Computers*, 18(8), 1992.
13. L. Sha. Real time scheduling theory: A historical perspective. *Journal of Real-Time Systems*, (28), 2004.
14. C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27, July 1948.
15. B. G. Spuri, M. Scheduling aperiodic tasks in dynamimc priority systems. *Journal of Real-Time Systems*, 2(10), 1996.
16. J. Stankovic. Real-time and embedded systems. *Journal of ACM Computing Surveys*, 29(1), March 1996.

Neural Networks and Evolutionary Algorithms

RBF Neural Network for Estimating Locational Marginal Prices in Deregulated Electricity Market

Seema Nirved Pandey[1], Shekhar Verma[1] and Laxmi Srivastava[2]

[1] Information Technology Department, ABV-Indian Institute of Information Technology and Management, Gwalior, India
[2] Electrical Engineering Department, Madhav Institute of Technology and Science, Gwalior, India
{Seema N. Pandey, Shekhar Verma, Laxmi Srivastava se_np@rediffmail.com, sverma@iiitm.ac.in, laxmi@sancharnet.in

Abstract. In the deregulated Power Market scenario, due to liberalized market structure and non-discriminatory open transmission access, the issue of congestion management and hence optimum use of transmission capacity, has become more crucial issue.. The pricing mechanism based on capacity allocation principle, to determine Locational Marginal Prices (LMP) can be proved to be substantial, about efficient utilization of transmission grid and available generation capacity. Regarding Congestion Management the Optimal pricing strategy breaks the Nodal pricing into two components; one is Locational Marginal Price (LMP) and second is Nodal Congestion Price (NCP). Both of these are significant for market participants as system security parameter. In the emerging deregulated environment, the Artificial Intelligent techniques like ANN provide instant and accurate LMPs, which boost up the motive of spot power market. This paper presents Radial Basis Function Neural Network (RBFNN) for estimating LMPs.
Since the test results are very accurate and awfully fast, these instant results can be directly floated to OASIS (open access same time information system) web site. The Market Participants willing to make transactions can access this information for any location of the market. The effectiveness of the proposed ANN has been established by comparing the testing results with those obtained with conventional Interior-Point OPF based method for a 6-bus test system having three generating units.

Keywords: Locational Marginal Price, Nodal Price, Congestion Management, Radial basis function (RBF) neural network.

Nomenclature

C_{S_i} Supply bid of unit i [$ / MWh];
C_{D_j} Demand bid of unit j [$ / MWh];
P_{S_i} Supply bid volume of unit i [MW];

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 63-74

Received 08/07/06
Accepted 04/10/06
Final version 13/10/06

P_{D_j} Demand bid volume of unit *j* [MW];

$P_{S_{\max_i}}$ Upper limit of power bid of unit *i* [MW];

$P_{S_{\min_i}}$ Lower limit of power bid of unit *i* [MW];

$P_{D_{\max_j}}$ Upper limit of demand bid of unit *j* [MW];

$P_{D_{\min_j}}$ Lower limit of demand bid of unit *j* [MW];

Q_{G_i} Reactive power output of unit *i* [MVar];

$Q_{G_{\min_i}}$ Upper limit of Reactive power at unit *i* [MVar];

$Q_{G_{\max_i}}$ Lower limit of Reactive power at unit *i* [MVar];

V_i Voltage magnitude at unit *i*

I Set of indexes of generating units;

J Set of indexes of consumers

B Set of indexes of network buses;

N Set of indexes of transmission lines;

$\mathrm{LMP_k}$ Locational Marginal Prices at bus k;

P_i = real power load at i^{th} bus

Q_i = reactive power load at i^{th} bus

1 Introduction

With the advent of deregulation and policies of open access, allocation of scarce transmission resources has become a key factor for the efficient operation of electricity market. To ensure efficient use of transmission grid and generation resources by providing fast and correct economic signals, a spot price or nodal price theory has been developed [1]. To design a reasonable pricing structure of Power Systems and to provide an effective Congestion Management procedure nodal prices have been decomposed into a variety of parts corresponding to concerned factors such as generations, voltage limitations, transmission congestion and other constraints [2].

The optimal nodal prices [3] and [4] of electricity comprise of two components, one is price for cost of supplying next increment of electric power demand (based on Lagrangian Multiplier), at a specific node or bus, involving generation marginal cost and all power system equality constraints; other is price for congestion (i.e. shadow prices) considering inequality constraints only. The first component is called as Locational Marginal Price (LMP) and other is Nodal Congestion Price (NCP). A multi-objective optimal power flow approach to account for pricing system security through the use of voltage stability constraints is presented in [5]. In the deregulated electricity market, the market dispatch (unconstrained) stage is almost same for different market structures. These auction-based dispatches have no consideration for

transmission situations and the resulting price is the Market Clearing Price (MCP). In economic terms, the market clearing price is the point of intersection of supply and demand bid curves. However LMP reflects the security constrained pricing. The various pricing mechanisms for Congestion Management are discussed in [6]. The method of Nodal Pricing (LMP and NCP) is adopted by PJM, ISO-NE and ISO-NY. Nodal prices are not necessarily capped by the marginal cost of the marginal units and can be higher than the most expensive unit running, and can be negative, in constrained out areas [7]. When there are no congestion in the market, then the LMPs are same at all buses and equal to the marginal cost to serve load in control area. Nodal Pricing of Electricity is highly volatile under constrained transmission condition. Sensitivities of LMPs are explored with respect to power demands in [8]. [9] presents an AC-OPF model for investigating marginal cost pricing for real and reactive power flow. An overview of various congestion management methodologies are presented in [10]. Rather than computing spot prices, the thrust area is forecasting spot prices and Day-Ahead load demand [11] using Artificial Intelligent techniques. [12] presents a method that forecasts next-day electricity prices, based on fuzzy logic and neural networks. The RBFNN is proposed in [13-15] for handling various problems in Power System operation and control.

This paper presents an RBFNN approach [13-15] for predicting LMPs in deregulated power market. Theoretically with enough RBF neurons, the RBFNN can realize zero error to all the training samples. Besides, the number of RBF neurons in the hidden layer can be determined during the parameter optimization process. The optimization process also speeds up the training process of neural network. These features make this ANN very attractive in practical use.

2 Methodology

2.1 The RBFNN Architecture

The architecture of the proposed RBF neural network consists of three layers, the input layer, hidden layer and output layers with the hidden layer composed of RBF neurons. The nodes within each layer are fully connected to the previous layer as shown in Fig. 1. The input variables are assigned to each node in the input layer and are forwarded to the hidden layer directly, without weights. The hidden nodes contain the radial basis functions and are analogous to the sigmoid function commonly used in feed-forward multi-layer perceptron model.

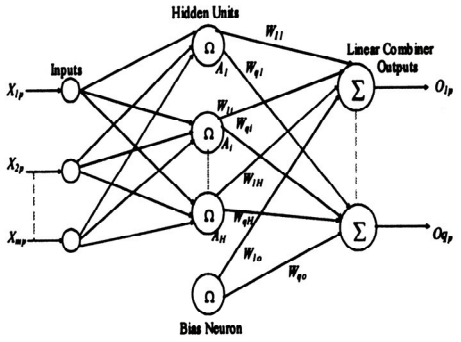


Fig. 1. Radial Basis Function NN Model

Most commonly used MLP models suffer from local minima and over fitting problems. The RBF neural networks have increasingly attracted interest for engineering applications due to their advantages over traditional MLP models, namely faster convergence, smaller extrapolation errors, optimized system complexity, minimized learning and recall times and higher reliability. They are highly promising for multivariable interpolation given irregularly positioned data points and provide good generalization ability with a minimum number of nodes to avoid unnecessarily lengthy calculations. The radial basis function is similar to the Gaussian density function, which is defined by a centre position and a width parameter. The width of the RBF unit controls the rate of decrease of function. The output of the i^{th} unit $a_i(Xp)$ in the hidden layer is given by:

$$a_i(X_p) = \exp\left(-\sum_{j=1}^{r}\left[x_{jp} - \bar{x}_{ji}\right]^2 / \sigma_i^2\right) \quad (1)$$

Where $\bar{x}_{ji}$ is the centre of i^{th} RBF unit for input variable j, σ_i is width of i^{th} RBF unit and x_{jp} is the j^{th} variable of input pattern p. The connection between the hidden units and the output units are weighted sums as shown in Fig. 1. The output value o_{qp} of the q^{th} output node is given as:

$$o_{qp} = \sum_{i=1}^{H} w_{qi}\, a_i(X_p) + w_{qo} \quad (2)$$

Where w_{qi} is weight between i^{th} RBF unit and q^{th} output node and w_{qo} is biasing term at q^{th} output node.

The parameter of the RBF units is determined in three steps of the training process. First of all some form of clustering algorithm explores the unit centres. Then the widths are calculated by a nearest neighbor method. Finally weights connecting the RBF units and the output units are determined using multiple regression techniques. Euclidean distance based clustering technique has been employed in this paper to

select the number of hidden (RBF) units and unit centres. The normalized input data are used for training of the RBF neural network. In this paper the dynamic version of RBFNN is used to make faster training. In this new RBFNN hidden nodes are altered until desired goal is reached.

2.2 Locational Marginal Pricing

The Locational Marginal Prices are the nodal prices, which are typically calculated as Lagrange multipliers associated with equality constraints, from the OPF solution [16]. For calculating these prices, the Pool Operator receives supplier bid and customer bid as bid plots. Fig. 2 shows bid plots for both supply as well as demand. Then Pool Operator determines the market clearing price (MCP) and a set of successful bidders on the basis of some auction mechanism. In the price assessment process, both optimal bidding strategy and auction mechanism play an important role. The intersection of supply and demand plots provides market clearing price and market clearing volume, as shown in Fig. 2 by G_m and H_m in \$/ MWh and in MW respectively. The supply bid plot shows the minimum price at which a generator is willing to produce a certain amount of power, while demand bid plot shows the maximum price, which is accepted by customers to buy a certain amount of power.

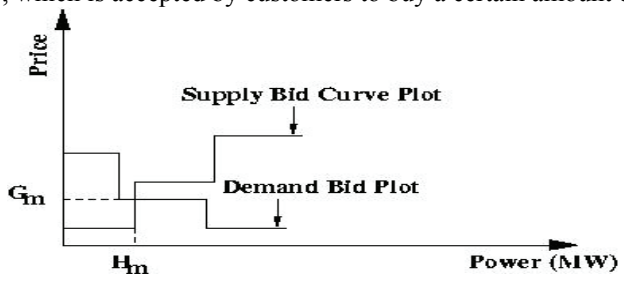


Fig. 2. Demand and Supply bid Plot

For the sake of simplicity it is assumed here that supply and demand bid is a single price not complete plot. In power market security pricing field, OPF-based approach is basically a non-linear constrained Optimization problem. One crucial outcome of this optimization procedure is Locational Marginal Prices. This outcome in pool-market operation is achieved through objective-function as maximization of social benefit *i.e.* maximizing the generator's income for their power production and simultaneously ensuring that consumers pay cheapest price for their power purchase. The OPF objective function is:

$$Min \quad G = \sum_{i \in I} C_{S_i} P_{S_i} - \sum_{j \in J} C_{D_j} P_{D_j} \tag{3}$$

subject to following equalities and inequalities constraints.

Power flow equations:

$$f(P_G, Q_G, V, \theta) = 0 \tag{4}$$

Power balance equations:

$$P_k = \sum_{i \in I} (P_{G_{io}} + P_{S_i}) - \sum_{j \in J} (P_{L_{jo}} + P_{D_j}) \quad \textbf{(5)}$$

$$Q_k = \sum_{i \in I} (Q_{G_{io}}) - \sum_{j \in J} (Q_{L_{jo}} + Q_{D_j}) \tan \phi_{D_j} \text{ where } \forall k \in B \quad \textbf{(6)}$$

Supply Bid Blocks:

$$P_{S_{\min_i}} \le P_{S_i} \le P_{S_{\max_i}} \quad \forall i \in I \quad \textbf{(7)}$$

Demand Bid Blocks:

$$P_{D_{\min_j}} \le P_{Dj} \le P_{D_{\max_j}} \quad \forall j \in J \quad \textbf{(8)}$$

Reactive Power Generation Limit:

$$Q_{G_{\min_i}} \le Q_{G_i} \le Q_{G_{\max_i}} \quad \forall i \in I \quad \textbf{(9)}$$

Voltage Security Limit:

$$V_{\min_k} \le V_k \le V_{\max_k} \quad \forall k \in B \quad \textbf{(10)}$$

Thermal Limit:

$$I_{mk}(\theta, V) \le I_{mk_{\max}} \quad \forall (m,k) \in N \quad \textbf{(11)}$$

The Lagrange function of the optimization problem (3) – (11) can be written as,

$$\begin{aligned} Min \quad \Im = {} & G - \lambda^T f(\delta, V, P_S, P_D, Q_G) \\ & - \mu_{PS_{\max}}^T \left(P_{S\max} - P_S - s_{Ps\max}\right) \\ & - \mu_{PS_{\min}}^T \left(P_S - s_{Ps\min}\right) \\ & - \mu_{PD_{\max}}^T \left(P_{D\max} - P_D - s_{PD\max}\right) \\ & - \mu_{PD_{\min}}^T \left(P_S - s_{PD\min}\right) \\ & - \mu_{\mathrm{Im}k_{\max}}^T \left(I_{\max} - I_{mk} - s_{Imk\max}\right) \\ & - \mu_{Ikm_{\max}}^T \left(I_{\max} - I_{km} - s_{Ikm\max}\right) \end{aligned}$$

$$- \mu_{QG_{\max}}^{T} \left(Q_{G\max} - Q_G - s_{QG\max} \right)$$

$$- \mu_{QG\min}^{T} \left(Q_G - Q_{G\min} - s_{QG\min} \right)$$

$$- \mu_{V_{\max}}^{T} \left(V_{\max} - V - s_{V\max} \right)$$

$$- \mu_s \left(\sum_i \ln s_i \right) \tag{12}$$

where λ and μ are Lagrange multipliers with respect to equality and inequality constraints. Here s is a non negativity slack variable. The Lagrange multiplier λ *reflects* the locational marginal cost price at each node considering all equality constraints. The optimization of (12) is satisfied by the KKT optimality condition:

$$\nabla_z \Im_\mu(z) = 0 \tag{13}$$

Then by applying (13) in (12):

$$\partial \Im / \partial P_{S_i} = 0 = C_{S_i} - \lambda_{P_{S_i}} + \mu_{P_{S\max_i}} - \mu_{P_{S\min_i}} \tag{14}$$

$$\partial \Im / \partial P_{D_i} = 0 = -C_{D_i} + \lambda P_{D_i} + \lambda Q_{D_i} \tan(\phi_{D_i}) + \mu P_{D_{\max_i}} - \mu P_{D_{\min_i}} \tag{15}$$

Thus the LMPs can be defined as,

$$LMP_{S_i} = \lambda_{P_{S_i}} = C_{S_i} + \mu_{P_{S\max_i}} - \mu_{P_{S\min_i}} \tag{16}$$

$$LMP_{D_i} = \lambda_{P_{D_i}} = C_{D_i} + \mu P_{D_{\max_i}} - \mu P_{D_{\min_i}} - \lambda Q_{D_i} \tan(\phi_{D_i}) \tag{17}$$

Thus by this classical approach, the LMPs at both supply and demand node are determined for the 6-bus system [5]. Furthermore, it is established that system congestion do significantly affect market bids and associated costs, hence still there is a need for a precise model for taking in account security constraints. In this way LMPs are always better to take into consideration than MCPs. Equation (19) and (20) provide LMP_k i.e. LMPs at k^{th} node in the given system for both supply and demand bids.

3 Training Algorithm for RBFNN

Neural Network models are the trainable analytic tools that attempt to mimic the information-processing pattern in human brain. It has capabilities of learning generalization and fault tolerance, which make it suitable for on-line application environment among all other artificial intelligent techniques. The Radial Basis Function Neural Network (RBFNN) is proposed for providing LMPs at every node of the given 6-bus test system. The RBFNN can be designed in a fraction of time as compared with other design approaches for training standard feed-forward networks.

The solution algorithm for Locational Marginal Pricing using RBFN is given below.

(i) A large number of load patterns are generated randomly in wide range of load variation at each load bus.

(ii) For each load case Locational Marginal Price at different buses is computed by classical IPM-OPF method.

(iii) Real and Reactive loads at load buses i.e. bus no.4,5 and 6 are selected as input features for the RBF network.

(iv) For training of the RBF network, initialize all the connection weights between hidden nodes and output nodes.

(v) Compute the Gaussian function at the hidden node using equation (1) i.e.

$$a_i(X_p) = \exp\left(-\sum_{j=1}^{r}\left[x_{jp} - \bar{x}_{ji}\right]^2 / \sigma_i^2\right)$$

Where r is dimension of input vector.

(vi) For the given values of loads at bus no 4,5 and 6, calculate the output of the RBFNN, which is locational marginal prices at all the six buses using equation (2) i.e.

$$o_{qp} = \sum_{i=1}^{H} w_{qi}\, a_i(X_p) + w_{qo}$$

(vii) Calculate the Mean Squared Error e_p for the p^{th} input pattern using

$$e_p = \frac{1}{2}\cdot\frac{1}{NO}\sum_{q=1}^{NO}\left(t_{qp} - o_{qp}\right)^2 \quad \textbf{(18)}$$

Here t_{qp} is the target value at q^{th} neuron of output layer for p^{th} pattern and NO is the number of neurons in output layer.

(viii) Repeat steps (v) to (viii) for all the 240 training patterns comprising real and reactive power demand at bus no. 4, 5, 6 and locational marginal prices at all six buses.

(ix) Calculate the error function E_k using equation,

$$E_K = \sum_{p=1}^{p^{\max}} e_p = \frac{1}{2}\sum_{p=1}^{p^{\max}}\frac{1}{NO}\sum_{q=1}^{NO}\left(t_{qp} - o_{qp}\right)^2 \quad \textbf{(19)}$$

Where, $p^{\max}$ is the total number of training patterns which is 240 for this application of estimating locational marginal prices all the nodes/buses of the given system i.e. six-bus test system.

(x) Update the connection weights using equation

$$w_{qi}(K+1) = w_{qi}(K) + \Delta w_{qi}(K) \tag{20}$$

Where

$$\Delta w_{qi}(K) = \eta(K).\sum_{p=1}^{p^{\max}} \delta_{qp}.A_{pi} + \alpha.\Delta w_{qi}(K-1) \tag{21}$$

And $$\delta_q = T_q - O_q \tag{22}$$

where, $\eta(K)$ is learning rate or adaptive size at K^{th} iteration and δ_q is the error signal for unit q and $\propto$is momentum term and T_q , O_q are target and actual output respectively.

(xi) The procedure is continued till the error becomes negligible.

(xii) This trained RBFNN model is tested for previously unseen load patterns for estimating LMPs all the buses of the system.

4 Results and Discussion

Though the trained ANN has provided the accurate results (i.e. LMPs at each node) for all the 60 testing patterns, due to limited space, testing results for only 15 patterns are given in Table 1. This table also shows the percentage error as a performance index for proposed RBFNN. The actual and target output for all 60 load patterns are shown in Fig. 3.

From this figure it is clear that RBFNN has achieved the target within permissible accuracy limits (can be seen as overlapped display). It can be noted from Table 1, that the maximum percentage error for LMP_1, LMP2, LMP3, LMP_4, LMP_5 and LMP_6 are 0.267, 0.304, 0.328, 0.319, 0.315 and 0.340 respectively. For same ANN structure and same learning rate of 0.30 the performance of RBFNN is compared with a BPMLP model. The rms errors for RBFNN and a BPMLP model were obtained as 0.0867 pu and 0.1039 pu respectively, which clearly shows its superiority over other BPMLP models

It is evident from Fig. 3, that LMP at node-1 ranges from 9.0 – 9.85 $/ MWh, LMP at node-2 ranges from 8.85 – 9.85 $/ MWh, LMP at node-3 ranges from 9 – 10 $/ MWh, LMP at node-4 ranges from 9.5 – 10.5 $/ MWh, LMP at node-5 ranges from 9.6 – 10.4 $/ MWh and LMP at node-6 ranges from 9.2 – 10.2 $/ MWh. The different values. of LMPs at nodes show the presence of congestion in the system.

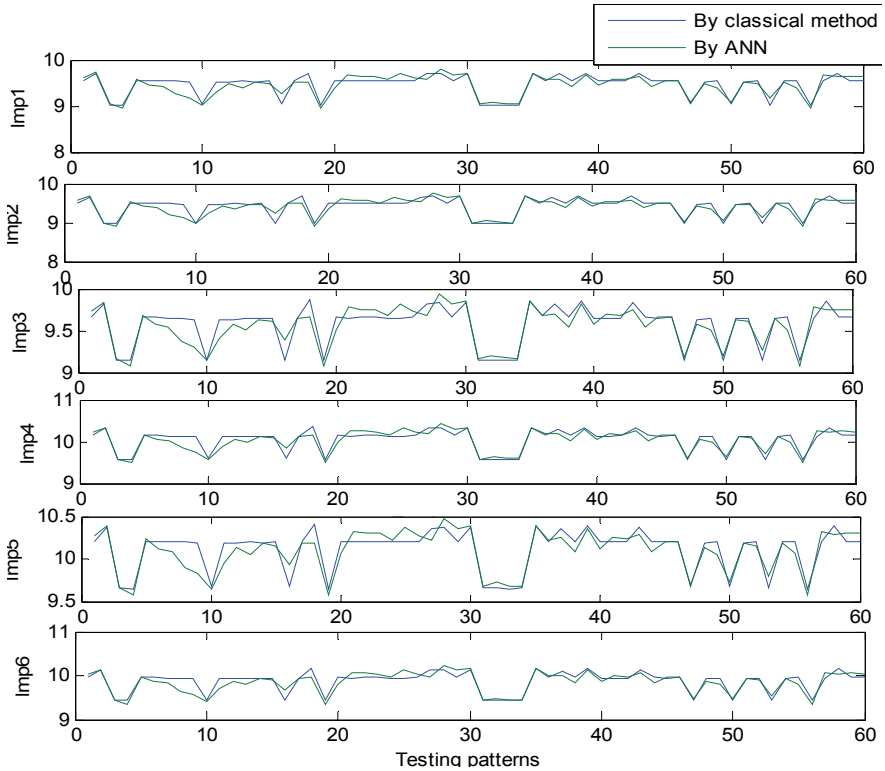


Fig. 3. Testing Performance of RBFNN for all 60 patterns.

5 Conclusion

A novel approach using radial basis function neural network is proposed, for estimating LMPs at both supply and demand nodes in the system. During testing phase the trained neural network furnished results within acceptable accuracy limits for previously unseen load patterns. The floating of this significant information of LMPs on OASIS website, would enable the Market Participants to peruse their transactions.

The MCPs do not have any hold on generation and transmission constraints. On the other hand LMPs are calculated as Lagrangian Multipliers (or dual variables) associated with OPF framework and are substantially affected by changes in demands, supply and transmission constraints that's why constitute essential information in an Electricity Market. The AI techniques are now admired specifically for spot pricing in budding deregulated power market and establishing economic signals among Market Participants.

Table 1. Testing Performance of Trained RBFNN for Locational Marginal Pricing

NP	TPD	TQD	METHOD	LMP1	LMP2	LMP3	LMP4	LMP5	LMP6
1	2.966	2.013	CLASSICAL	9.700	9.660	9.827	10.320	10.366	10.128
			BY ANN	9.726	9.677	9.842	10.335	10.394	10.140
			% Error	0.267	0.181	0.157	0.147	0.270	0.114
2	2.796	1.897	CLASSICAL	9.030	8.987	9.148	9.575	9.664	9.429
			BY ANN	9.033	8.993	9.158	9.576	9.667	9.442
			% Error	0.035	0.070	0.111	0.012	0.032	0.133
3	2.945	1.998	CLASSICAL	9.559	9.509	9.670	10.157	10.205	9.964
			BY ANN	9.574	9.524	9.686	10.169	10.233	9.979
			% Error	0.158	0.155	0.164	0.119	0.270	0.149
4	2.923	1.983	CLASSICAL	9.530	9.478	9.638	10.116	10.194	9.929
			BY ANN	9.529	9.476	9.637	10.115	10.188	9.928
			% Error	0.013	0.016	0.015	0.014	0.055	0.015
5	2.947	2.000	CLASSICAL	9.543	9.492	9.653	10.135	10.199	9.945
			BY ANN	9.531	9.484	9.646	10.126	10.189	9.939
			% Error	0.123	0.089	0.075	0.089	0.100	0.063
6	3.027	2.054	CLASSICAL	9.700	9.670	9.839	10.330	10.378	10.141
			BY ANN	9.706	9.678	9.847	10.339	10.386	10.150
			% Error	0.061	0.080	0.082	0.089	0.079	0.090
7	2.787	1.891	CLASSICAL	9.024	8.983	9.146	9.567	9.657	9.428
			BY ANN	9.039	8.999	9.165	9.583	9.673	9.449
			% Error	0.167	0.182	0.210	0.163	0.168	0.224
8	2.836	1.924	CLASSICAL	9.021	8.981	9.146	9.563	9.654	9.428
			BY ANN	9.043	9.000	9.161	9.590	9.679	9.441
			% Error	0.243	0.210	0.164	0.282	0.259	0.141
9	3.027	2.054	CLASSICAL	9.700	9.676	9.846	10.336	10.386	10.150
			BY ANN	9.706	9.678	9.847	10.339	10.386	10.150
			% Error	0.061	0.018	0.010	0.031	0.001	0.001
10	2.980	2.022	CLASSICAL	9.562	9.512	9.674	10.162	10.206	9.968
			BY ANN	9.574	9.527	9.689	10.173	10.225	9.984
			% Error	0.126	0.160	0.160	0.111	0.190	0.159
11	3.016	2.047	CLASSICAL	9.700	9.676	9.847	10.336	10.386	10.150
			BY ANN	9.680	9.647	9.815	10.303	10.353	10.115
			% Error	0.205	0.304	0.328	0.319	0.315	0.340
12	2.948	2.000	CLASSICAL	9.534	9.482	9.643	10.121	10.195	9.933
			BY ANN	9.551	9.501	9.663	10.145	10.205	9.956
			% Error	0.174	0.197	0.206	0.232	0.094	0.227
13	2.949	2.001	CLASSICAL	9.551	9.500	9.661	10.146	10.202	9.954
			BY ANN	9.551	9.501	9.663	10.145	10.205	9.956
			% Error	0.004	0.007	0.019	0.015	0.026	0.016
14	2.923	1.983	CLASSICAL	9.530	9.478	9.638	10.116	10.194	9.929
			BY ANN	9.529	9.476	9.637	10.115	10.188	9.928
			% Error	0.013	0.016	0.015	0.014	0.055	0.015
15	2.947	2.000	CLASSICAL	9.543	9.492	9.653	10.135	10.199	9.945
			BY ANN	9.531	9.484	9.646	10.126	10.189	9.939
			% Error	0.123	0.089	0.075	0.089	0.100	0.063

References

1. Schweppe, F.C., Caramanis, M.C., Tabors, R.D.: " Spot Pricing of Electricity", (Boston, MA:Kluwer Academic Publishers, 1988)
2. Chen, L., Suzuki, H., Wachi, T., Shimura, Y.: "Components of Nodal Prices for Electric Power Systems", IEEE Transactions on Power Systems, vol. 17, No. 1, February 2002, pp. 41-49.
3. Xie, K., Song, Y.H., Stonfam, J., Yu, E., Liu, G.: " Decomposition Model and Interior Point Methods for Optimal Spot Pricing of Electricity in Deregulated Environment," IEEE Transaction on Power Systems, vol. 15, no. 1, pp. 39-50, Feb. 2000.
4. Tanaka, Mokoto: Optimal Transmission Capacity under Nodal Pricing and Incentive Regulation for Transco, RIETI Discussion Paper Series 05-E-021, University of Tokyo, Japan, May 2005, pp. 1-19.
5. Milano, Federico: Pricing System Security in Electricity Market Models with Inclusion of Voltage Stability, Ph.D. thesis, University of Waterloo, Waterloo, ON, Canada, 2002.
6. Wu, Y.K.: " Comparison of Pricing Schemes of Several Deregulated Electricity Markets in the World,", IEEE/PES, Int. conf. on Security and Sustainable Development under Deregulation, August 14-18, 2005, PL2 - Panel Discussion, Power Market Design and Associated Issues , T&D-F1266, PL2-7, Dalian, China
7. Kirby, B.J., Van Dyke, J.W.: "Congestion Management Requirements, Methods and Performance Indices" Oak Ridge National Laboratory, South California, June 2002. Available at http:// www.osti.gov/bridge
8. Konejo, A.J., Castillo, E., Minguez, R., Milano, F.: "Locational Marginal Price Sensitivities" IEEE Transactions on Power Systems, vol. 20, No. 4, November 2005, pp. 2026-2032.
9. Rotoras, Recardo: "Transmission Pricing of Active and Reactive Power, Congestion Price and Supplement cost Allocation Method : A case of Philippines," Ph.D. Thesis, Available at www.asdu.ait.ac.th
10. Pandey, Seema N., Verma Shekhar, Srivastava., Laxmi: "Congestion Management In Restructured Power System Scenario : An Overview", Presented at International Conference on Challenges and Strategies for Sustainable Energy, Efficiency and Environment, held at UPTU, Lucknow, 10^{th} – 11^{th} June, 2006, pp. 410 – 421.
11. Hong, Ying-Yi., Lee, C.F.: "A Neuro-Fuzzy Price Forecasting Approach in Deregulated Electricity Markets," Electric Power System Research, Vol.73, Issue2, Feb2005, pp. 151-157.
12. Iyer, Vanaja, Fung, Chun Che, Gedeon, Tamas: "A Fuzzy-Neural Approach to Electricity Load and Price Forecasting in a Deregulated electricity Market,"TENCON 2003, Conference on Convergent Technologies for Asia Pacific region, Vol.4,15-17 Oct.2003, pp.1479-1482.
13. Bi, T.S., Ni, Y.X., Shen, C.M., Wu, F.F., Yang, Q.X.: "A Novel Radial Basis Function Neural Network for Fault Estimation in Transmission Networks", Proc. of 5^{th} International Conference in Advances in Power System Control, Operation and Management, APSCOM 2000, Hong Kong, Oct. 2000.
14. Jain, T., Srivastava**,** L, and Singh, S.N.: "Fast voltage contingency screening using Radial Basis Function neural network", *IEEE Transactions on Power Systems*, Vol 18, No. 4, 2003, pp. 1359-1366.
15. Ram, D., Srivastava, L., Pandit, M., Sharma, J.: "Line Overloading Alleviation Using Radial Basis Function Neural Network," *Proceedings International Conference on Power Systems: "Challenges to Electric Utilities in the New Millennium"*, held in Kathmandu, Nepal, Nov. 3-5, 2004, 838-841.
16. Milano, F.: "PSAT - Matlab-based Power System Analysis Toolbox", available at http:// thundebox.uwaterloo.ca / ~fmilano, 2002.

Evonorm: Easy and Effective Implementation of Estimation of Distribution Algorithms

Luis Torres-Trevino

Corporación Méxicana de Investigación en Materiales, Oceanía 190,
Tel. (52) 844-411-32-00, ext 1251,
25290 Saltillo, Coahuila, México
ltorres@comimsa.com.mx
http://www.comimsa.com.mx

Abstract. This paper shows some preliminary results about the performance of an estimation of distribution algorithm of easy implementation but effective to solve optimization problems. Computer simulation compares Evonorm versus Evolution Strategies to optimize complex functions. The results show a better efficiency and easy implementation than Evolution Strategies in the optimization of De Jong functions.

1 Introduction

There is a new tendency to generate simplified versions of evolutionary algorithms where crossover and mutation procedures are replaced. The population is built by a model that represents an estimation of distributions of the best individuals selected. Evonorm is a new evolutionary algorithm where the population is built by random variables with normal distribution. The parameters of these random variables are determined by the calculation of the mean and the standard deviation of selected population of solutions. The evolutionary algorithm replaces the crossover and the mutation procedure with new procedures to calculate parameters of random variables and to generate new individuals from these random variables with normal distribution.

Evonorm is an easy and effective way to apply estimation of distribution algorithms (EDAs). Evonorm can be used to find the decision variables that optimize a given function. There are several approaches to apply EDAs and can be classified in two classes, discrete and continuous. Examples of discrete EDAs are PBIL [1], UMDA [2], and CGA [3]. Examples of continuous EDAs are UMDAc [4] and IDEA [5]. In every class there are two subclasses, EDAS that consider a dependences between decision variables like BOA [6], hBOA[7], and eCGA[8], and EDAs that consider a independencies between variables. In continuous EDAs there is the same sub classification [9].

EDAs replace the use of crossover and mutation mechanism so it is expected a simplification in the implementation of an EDA, but in some of them it implies the use of high cost search mechanism to get an effective model to estimate the distribu-

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 75-83

Received 09/08/06
Accepted 03/10/06
Final version 13/10/06

tion of individuals selected in order to generate a new population. It is like a new low-level optimization problem into high-level optimization problem. Some experiments have shown the efficient of EDAs versus another evolutionary algorithms and the conclusion is the same, the EDAs requires more computational time and do not improve significantly the solutions found [10]. Evonorm can be effective founding good solutions but without the use of complex search mechanism to estimate distributions.

The present section is the introduction. The second section is an introduction of EvoNorm. The third section shows the performance of EvoNorm versus Evolution strategies (μ, λ) to make comparison between them optimizing De Jong Functions. The conclusion and future work is given in section fourth.

2 EVOlutionary Algorithm of Random Variables with NORMal Distributions (Evonorm)

Evonorm uses random variables with normal distribution. The normal distribution function is a random variable and describes many random phenomena that occur in every day life. It is simulated the normal distribution function with two parameters, the first is the mean and it is a numeric measure of the central tendency of the random variable. The second parameter is the standard deviation and it is a measure of the dispersion of a variable around the mean. A normal distribution function can be used to represent a set of possible values of a decision variable, so it is necessary to use a set of parameters (mean and standard deviation) of the normal distribution function per decision variable. Equation 1 shows an easy implementation of a random variable with a normal distribution.

$$N(\mu,\sigma) = \mu + \sigma \sum_{i=1}^{12} U \tag{1}$$

Where: μ is the mean, σ is the standard deviation, and U is a uniform random number generator. The Evonorm procedure has the same philosophy of an evolutionary algorithm because there are an evaluation process, a selection, and a variation procedure where the crossover and mutation are substituted by new procedures, the calculation of the parameters of the normal distribution functions per decision variable and the generation of a new population.

In continuous optimization a vector of real numbers can represent continuous decision variables. It is used a random variable per decision one. EvoNorm evolves the random variables to generate new real vectors of decision variables. These variables will be evaluated and the best of them will be selected to calculate new parameters of the distribution function to generate a new population. The process is repeated again and again (Table I). The calculation of the mean and standard deviation is a common known arithmetic procedure (2, 3)

$$\mu(pr) = \sum_{i=1}^{n} PS(pr,i)/n \tag{2}$$

$$\sigma(pr) = \sqrt{\sum_{i=1}^{n} (PS(pr,i) - \mu(pr))^2} \tag{3}$$

Where pr represents the decision variables involved, and PS represents the selected individuals. n represents the number of individuals selected. PS is a matrix of $NT\,\mathrm{Pr}$ columns and NTR rows both constants represents the total of parameters and individuals respectively.

Table 1. Seudocode of evonorm

```
1) Generation of a uniform random popula-
   tion P of size m.
2) Evaluation of the m individuals.
3) Selection of the best n individuals
   (n<m)
4) Calculation of mean and standard de-
   viation from n selected individuals.
5) Modify standard deviation if intensive
   exploration is active.
6) Generation of a new population of size
   m from random variables with parame-
   ters calculated in (4) and (5)
7) If a criterion satisfied then end else
   go to step (2)
```

EVONORM is similar to UMDAc because uses random variables that generate numbers with Gaussian distribution. The parameters of these random variables are calculated from the population selected too. Evonorm searches the appropriate parameters of the normal distribution to improve the random variables associated.

Evonorm consider an intensive exploration phase because some functions are difficult to optimize. In this phase it is used a constant standard deviation equivalent to the half of the range of a decision variable and it is used on a half of the total generations of Evonorm. Some functions do not require the intensive exploration phase, nevertheless the use of this procedure improve lightly the performance of the search in these functions.

The exploration is implemented by a simple condition: If the generation counter is below of the 50 percent of the total generations then the standard deviation is equal to the half of the range of the decision variable (to get a very intensive exploration) else Evonorm uses the standard deviation calculated from the selected population. The calculated mean is always used.

3 Performance Comparisons between Evonorm and Evolution Strategies

It is analyzed the performance of Evonorm and Evolution Strategies (μ, λ) without recombination [11] on classic test De Jong functions [12, 13]. These functions take care to include continuity, discontinuity, convex, nonconvex, unimodal, multimodal, quadratic, non quadratic, low dimensional, high dimensional, stochastic and deterministic characteristics (4 to 8)
The exploration is applied during the 50 percent of the total generations at the beginning of the run. Table II shows the Evonorm parameters used to optimize every function. Evolution Strategies was adjusted to get the same number of evaluations. The average performance of both algorithms in 100 runs to optimize all De Jong functions are shown from figure 1 to figure 5.

De Jong function 1

$$f(X) = \sum_{i=1}^{3} x_i^2 \tag{4}$$

Where $-5.12 \le x_i \le 5.12$

De Jong function 2

$$f(X) = 100(x_1^2 - x_2) + (1 - x_1)^2 \tag{5}$$

Where $-2.048 \le x_i \le 2.048$

De Jong function 3

$$f(X) = \sum_{i=1}^{5} \text{int}(x_i) \tag{6}$$

Where $-5.12 \le x_i \le 5.12$

De Jong function 4

$$f(X) = \sum_{i=1}^{30} i x_i^4 + gauss(0,1) \tag{7}$$

Where $-1.28 \le x_i \le 1.28$

De Jong function 5

$$f(X) = \frac{1}{1/K + \sum_{j=1}^{25} f_j^{-1}(x1, x2)} \tag{8}$$

Where $f(x1,x2) = c_j + \sum_{i=1}^{2}(x_i - a_{ij})^6$

Where $-65.536 \le x_i \le 65.536$, $K = 500$, $c_j = j$, and

$$[a_{ij}] = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{bmatrix}$$

Table 2. Evonorm parameters used to optimize every De Jong function

	De Jong 1	De Jong 2	De Jong 3	De Jong 4	De Jong 5
Generations	50	100	50	100	150
Total of individuals	25	50	50	50	200
Individuals selected	5	10	10	10	40
Use exploration	Not	Yes	not	not	Yes

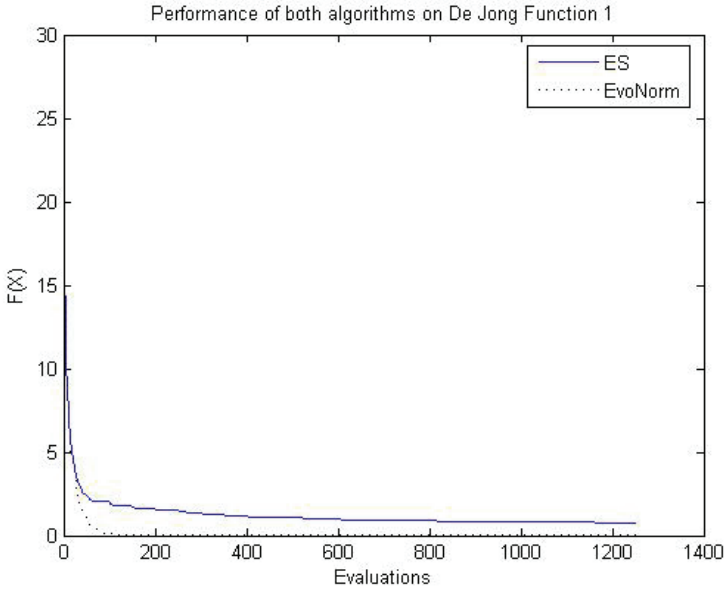


Figure 1. Performance of the algorithms to optimize De Jong function 1.

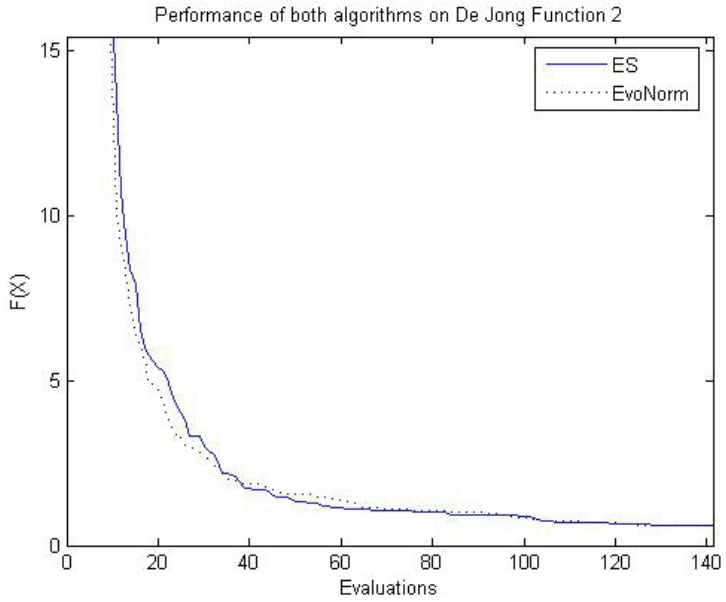


Figure 2. Performance of the algorithms to optimize De Jong function 2.

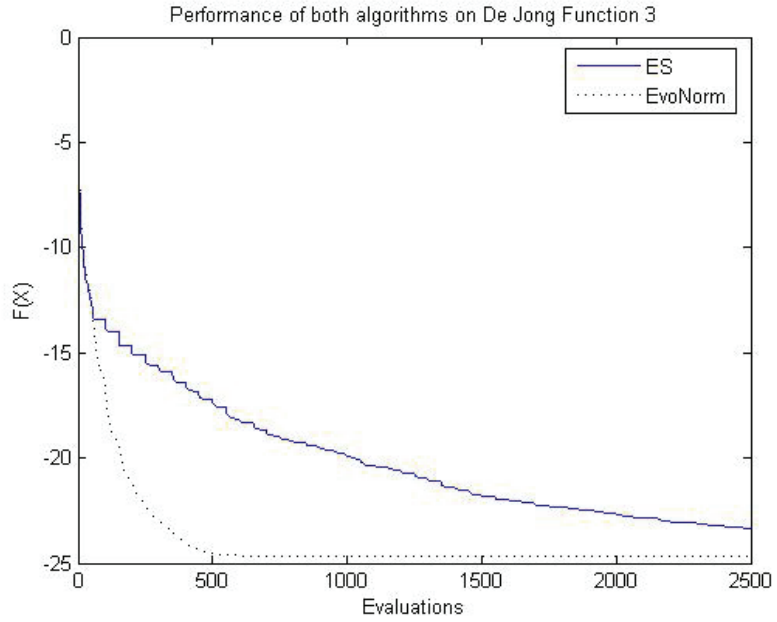


Figure 3. Performance of the algorithms to optimize De Jong function 3.

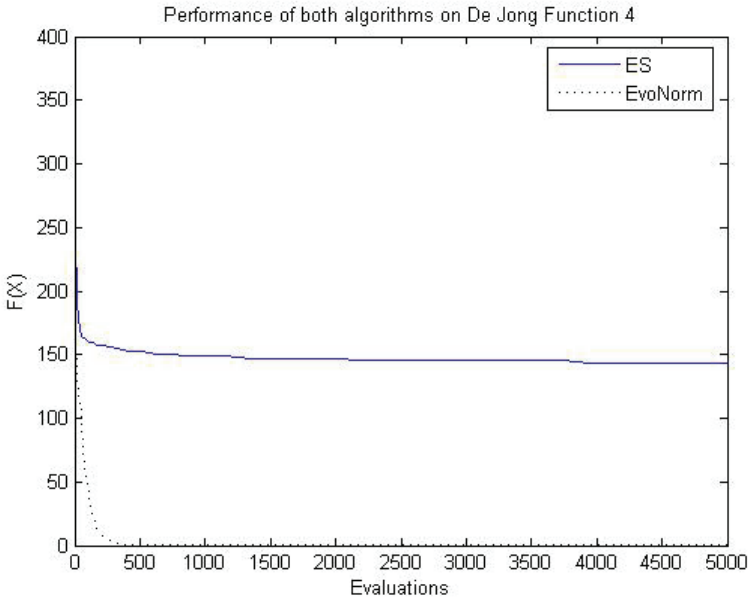


Figure 4. Performance of the algorithms to optimize De Jong function 4.

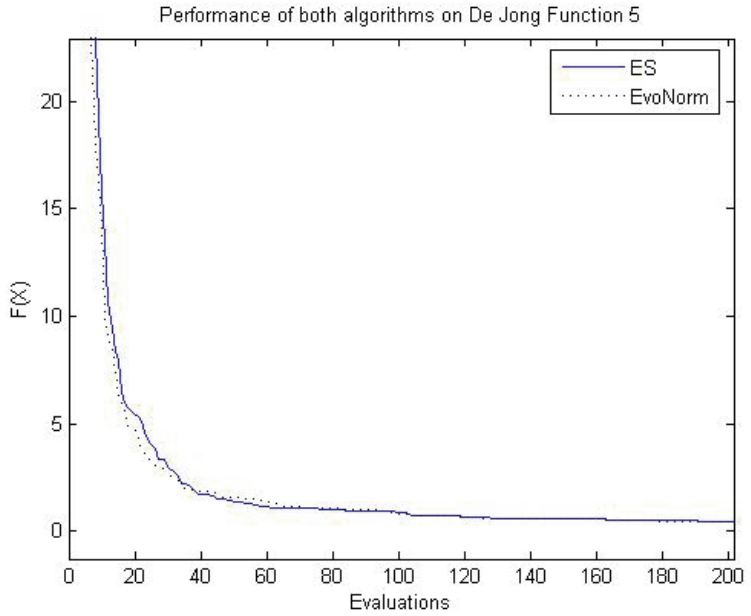


Figure 5. Performance of the algorithms to optimize De Jong function 5.

4 Conclusions

The performance of Evonorm is superior to evolution strategies in De Jong functions 1, 3 and 4. In other functions 2 and 5 the performance is very similar in both algorithms. It is important to mention the use of exploration in these test functions. There is an open opportunity to improve this algorithm because it is necessary to test Evonorm with and without exploration. The Evonorm is effective to optimize De Jong functions and the implementation of the algorithm is easy because the calculation of mean and standard deviation involve common used arithmetic operations. Evonorm is an evolutionary algorithm for continuous optimization based in estimation of parameters of random variables with normal distribution functions. The parameters are calculated from a set of selected individuals. The algorithm shows a good performance with the comparison again Evolution Strategies. The future work includes new test functions and comparisons with similar evolutionary algorithms for continuous optimization and makes the algorithm more independent to the problem. It is supposed an independent interaction between variables so will be important to may use multivariable normal distribution functions and different distribution functions not only the normal one. It is expected to extend the use of Evonorm in multimodal, constrain satisfaction and multi objective optimization.

References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121
2. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438
3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
5. Shumeet Baluja, 1994, “Population-Based Incremental Learning: {A} Method for Integrating Genetic Search Based Function Optimization and Competitive Learning”, Tech. Rep. No. CMU-CS-94-163, Pittsburgh, PA, Carnegie Mellon University. (1994)
6. H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I", In "Parallel Problem Solving from Nature", 178-187, Lecture Notes in Computer Science", September (1996)
7. Georges R. Harik and Fernando G. Lobo and David E. Goldberg, “The compact genetic algorithm", IEEE transactions on evolutionay computation. Volumen 3, NUMBER 4, 287-297, November. (1999)
8. P. Larrañaga, R. Etxeberria, J.A. Lozano, and J.M. Peña, “Optimization in continuous domains by learning and simulation of gaussian networks.”

9. In A.S. Wu, editor, Proceeding of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, (2000) pages 201-204
10. Peter Alexander Nicolaas Bosman, "Design and Application of Iterated Density Estimation Evolutionary Algorithms" Utrecht, Institute of Information and Computing Sciences. Phd. Thesis. (2003)

11. Martin Pelikan and David E. Goldberg and Erick Cantú-Paz, "BOA: The Bayesian Optimization Algorithm", Wolfgang Banzhaf and Jason Daida and Agoston E. Eiben and Max H. Garzon and Vasant Honavar and Mark Jakiela and Robert E. Smith, "Proceedings of the Genetic and Evolutionary Computation Conference {GECCO}-99, 525--532", Orlando, FL, Morgan Kaufmann Publishers, San Francisco, CA. (1999)
12. Martin Pelikan and Kumara Sastry and Martin V. Butz and David E. Goldberg, "Hierarchical BOA on Random Decomposable Problems", Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign., IlliGAL Report No. 2006002, January, (2006)
13. Cláudio F. Lima and Kumara Sastry and David E. Goldberg and Fernando G. Lobo, "Combining Competent Crossover and Mutation Operators: A Probabilistic Model Building Approach", Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign. IlliGAL Report No. 2005002}, February, (2005)
14. P. Larrañaga and J. A. Lozano, "Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation." Kluwer Academic Publishers (2001)
15. Cantú-Paz, E., "Feature subset selection by estimation of distribution algorithms". In W. B. Langdon, E. Cantu-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, N. Jonoska (Eds.), GECCO-2002: Proceedings of the Genetic and Evolutionary Computation Conference. (pp. 754). San Francisco, CA: Morgan Kaufmann. (2002)
16. Thomas Bäck", "Evolutionary Algorithms in Theory and Practice", Oxford University Press, New York, (1996)
17. K. A. DeJong, "An analysis of the behavior of a class of genetic adaptive systems", journal= "Dissertation Abstracts International, 36(19), 5140B, University Microfilms No. 76-9381. (1975)
18. Zbigniew Michalewicz, "Genetic algorithms + data structures = evolution programs", Springer-Verlag", New York. (1999)

Bioinformatics and Medical Applications

Generation of Linguistic Rules on the Genes Mediating the Development of Lung Adenocarcinoma

Rajat K. De[1] and Anupam Ghosh[2]

[1] Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India.
rajat@isical.ac.in
[2] Department of Computer Science and Engineering, Netaji Subhash Engineering College, Kolkata 700152, India.
anupam.ghosh@rediffmail.com

Abstract. In the present article, we propose a method for generation of rules on genes mediating the development of human lung adenocarcinoma. The method involves the application of cyclic loess normalization technique followed by the incorporation of the fuzzy sets *low*, *medium* and *high*. Linguistic rules are generated on the gene expression values. The system has been successfully applied on a microarray gene expression data consisting of expression values of 7129 genes in 10 normal and 86 tumor samples. In our results we have found that nine genes, including *RPLP0*, *ADH1*, *UGB*, *FMO2*, *HBA2*, *SFTPA1*, *SFTPA2*, *HBB* are the most significant mediating the development of lung adenocarcinoma. The results are in accordance with a number of earlier investigations.

1 Introduction

Lung cancer continues to be the most common cause of cancer related mortality in men and women. The treatments of lung cancer are primarily based on the broad classification of tumors into small cell, non-small cell types and histological subtyping. The heterogeneity of lung cancer patients at each disease stage with respect to outcome and treatment response suggests that additional subclassification and substaging remains possible.

Recent studies [12, 6, 18] involving gene expression profiling of clinical specimens have had a profound impact on cancer research. In some examples [12, 6], correlations have been made between the expression levels of a gene or set of genes and clinically relevant subclassifications of specific tumor subtypes. These results have shown that true molecular classification and substaging of multiple tumor types may be possible, leading to taking effective measures in prognosis and patient management. Microarray Technology can be used to correlate the gene-expression patterns with numerous clinical parameters including patient outcome to better predict tumor behavior in individual patients [18]. Analysis of lung cancers using array technologies has identified subgroups of tumors that differ according to tumor types and histological subclasses, and to lesser extent, survival among adenocarcinoma patients.

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 87-98

Received 17/07/06
Accepted 04/10/06
Final version 13/10/06

Fuzzy set theory is capable of handling uncertainty in the gene expression values arising due to incompleteness, imprecision, noise and experimental errors. The notion of fuzzy sets has been used in the domain of gene expression analysis. These include identifying interacting genes that fit a known "fuzzy" model of gene interaction by testing all combinations of gene expression determining profiles [29], a list of differentially expressed genes [7], a knowledge-based model of Zhang et. al. [31] to determine the influence of genes on classification of a sample into a tumor category, transforming gene expression values into qualitative descriptors using a set of linguistic rules involving fuzzy logic [29], fuzzy inferencing for classification of tumor samples [20], application of fuzzy ARTMAP to identify normal and tumor patients [4]. Recently, Sokhansanj et. al. [27] have demonstrated an approach with exhaustive search for fuzzy gene interaction models that best fit transcription measurements by microarray technology.

In this article, we have applied linguistic fuzzy sets on gene expression data of lung adenocarcinoma to identify a set of genes mediating the development of lung adenocarcinoma. The method involves a normalization method of cyclic loess [11] to reduce the variation among the expression levels of the gene over different samples. Then we have represented the whole-normalized data set in form fuzzy linguistic variables [23]. In this way, we have found three different classes corresponding to *low*, *medium* and *high* on gene expression values, for normal and tumor samples separately. In the next step, we have performed the matching operation with normal to tumor samples, which has led to identify 293 genes that have changed significantly from normal samples to tumor samples. Finally, rules are generated using the technique involving confidence factor for these genes. We report here the nine significant rules corresponding to nine genes among these 293 genes that have changed their expression values most significantly from normal samples to tumor samples. The gene expression data we have considered here are oligonucleotide arrays containing gene expression profiles for 10 normal and 86 lung adenocarcinoma including 67 stage I and 19 stage III tumor samples on 7129 genes.

2 Related Work

Zhang has proposed a rule discovery procedure that is based on prior knowledge of the influence of each gene for classification of a sample into a tumor category [31]. Only (gene, expression) descriptors that are consistent with the classification are considered as the antecedent of the rule. For example, if some parameter is positive for gene A, then only (A, $HIGH$) will be retained. On the other hand, a negative value of the parameter indicates that only (A, LOW) would be retained. This is certainly arbitrary and depends on the coding for the class variable. The magnitude of the parameter is ignored in this rule generation process. All possible descriptor sets are considered to compose the rules in the initial step. For example, if parameter for genes A and B are both positive in the logistic regression model, the corresponding rules are (A, $HIGH$), (B, $HIGH$), (A, $HIGH$), and (B, $HIGH$). Every rule thus constructed is considered for

elimination using the impact of its removal in the number of misclassifications. If there is no change or the number of misclassifications goes down, then the rule is eliminated.

A set of heuristic rules in the fuzzy logic framework to transform expression values into qualitative descriptors are evaluated in [29]. This model is used to find triplets of activators, repressors and targets in a gene expression data set. The predictions made by the algorithm agree well with experimental data. The algorithm can also assist in determining the function of uncharacterized proteins and is able to detect a substantially a larger number of transcription factors than that could be found at random. This technique extends current techniques such as clustering to allow the users to generate a connected network of genes using only expression data.

An investigation has been made using information on importance of genes in classification using fuzzy inferencing. This is similar to that of other classifiers, but simpler and easier to interpret [20]. The fuzzy inference system has the theoretical advantage that it does not need to be retrained when using measurements obtained from a different types of microarrays.

Interpretation of classification models derived from gene expression data is usually not simple. Rather, it is an important aspect in the analytical process. The performance of small rule-based classifiers is based on fuzzy sets and distribution of data [32]. The classifiers result in the rules that can be readily examined by biomedical researchers. The fuzzy logic-based classifiers compare favorably with logistic regression in all data sets they have considered.

Sokhansanj demonstrated an approach with exhaustive search for fuzzy gene interaction models that best fit transcription measurements obtained by microarray technology [27]. Applying an efficient, universally applicable data normalization and fuzzification scheme, the search converged to a small number of models that individually predict experimental data within an error tolerance. Although gene transcription levels are only used to develop the models, they include both direct and indirect regulation of genes. Biological relationships in the best-fitting fuzzy gene network models successfully recover direct and indirect interactions predicted from previous knowledge to result in transcriptional correlation. Fuzzy models that fit on data set were used for robust prediction of another experimental data set for the same system. Linear fuzzy gene networks and exhaustive rule search are the first steps towards a framework for an integrated modelling and experiment approach to high-throughput "reverse engineering" of complex biological systems.

Advances in molecular classification of tumors may play a central role in cancer treatment. Using gene expression profiles obtained by cDNA microarrays, a neural network model known as simplified fuzzy ARTMAP has been developed that is able to identify normal and tumor patients [4]. Furthermore, it makes the distinction among patients with molecularly different forms of carcinoma without any previous knowledge of those subtypes.

3 Proposed Method

In this section, we describe the method of generating linguistic rules on genes mediating the development of lung adenocarcinoma. The method has two parts. In the first part, we have used cyclic loess normalization technique [11] for transforming expression values in different normal samples of a gene into one value. Similarly, expression values in different lung adenocarcinoma samples of a gene into one value. The entire method is described in details below.

3.1 Normalization

The need of normalization arises naturally when we deal with experiments involving multiple arrays. There may be two broad characterizations one could use for the type of variation in different arrays: interesting variation and obscuring variation. Interesting variation deals with the biological differences, for example [14], when large differences in the expression level of particular genes between a diseased and a normal source are observed. On the other hand, obscuring variation is introduced during the process of carrying experiment with different samples of either normal or diseased type. The purpose of normalization is to deal with this obscuring variation.

Here we use cyclic loess method [11] to normalize the data set for normal lung samples and as well as tumor samples. This approach is based upon the idea of the M versus A plot, where M is the difference in log expression values and A is the average of the log expression values corresponding to a pair of samples. An M versus A plot for normalized data should show a point cloud scattered about the $M = 0$ axis.

This is due to the fact that the expression values of the pair of samples become closer on application of a normalization method. In particular, for any two arrays i, j with probe intensities x_{ki} and x_{kj} where $k = 1, \ldots, p$ is the probe index, we calculate $M_k = \log_2(x_{ki}/x_{kj})$ and $A_k = 1/2 \log_2(x_{ki}x_{kj})$. A normalization curve is fitted to these M versus A plot using loess. Here we fit a parabolic curve. The fits based on the normalization curve are $\hat{M}_k$ and thus the normalization adjustment is given by $(M_k - \hat{M}_k)$. This adjustment is apportioned equally to x_{ki} and x_{kj}.

To deal with more than two arrays, the method is extended to look at all distinct pair wise combinations. The normalization is carried out in a pair wise manner, recording an adjustment for each of the two arrays in each pair. After looking at all pairs of arrays we have a set of adjustments that can be applied to the set of arrays. Then we repeat the process until the difference in the expression values becomes less than some predefined threshold. Typically only 5 or 6 complete iterations through all pair wise combinations are needed to achieve an acceptable result. After getting the normalized values of the genes, we have taken mean of these normalized values of each gene to represent a gene by a single expression value. The steps of the method [11] are provided below for the sake of clarity.

For each gene, do

STEP 1: Choose pair wise samples.

STEP 2: Compute M_k and A_k for each pair using the above method.

STEP 3: If there are n samples for a gene, there should be $\binom{n}{2}$ pairs, so as to get $\binom{n}{2}$ number of M_k and A_k.

STEP 4: Fit M_k with respect to A_k. Here we use the parabolic curve fitting algorithm. In this algorithm we use the formula $\hat{M}_k = a + bA_k + c{A_k}^2$. So for a set of A_k values, we can get a set of $\hat{M}_k$ values. Finally we can get $\binom{n}{2}$ number of $(M_k - \hat{M}_k)$ values. We call these $(M_k - \hat{M}_k)$ values as an adjustment.

STEP 5: Record these adjustments for each sample and compute the resultant adjustment of each sample.

STEP 6: Update the old log expression value for each sample by the following formula,

$$new \log_2 x_{ik} = old \log_2 x_{ik} + resultant\ adjustment$$

STEP 7: Repeat Step 1 to Step 6 until the differences between the log expressions values are less than some threshold values specified by the analyzer (i.e. repeat those steps until the log expression values of different samples are close enough).

3.2 Grouping (into classes) based on fuzzy sets

In conventional statistical methods, the absolute expression pattern of genes is presented to a system for computations. However, in real life situations, gene expression pattern may be uncertain and/or incomplete. In such cases it may become convenient to use linguistic variables such as *low*, *medium*, *high*, *very high*, or *more or less* to replace numerical feature information [23].

The proposed model is capable of handling absolute expression pattern i.e, numerical and inexact i.e, linguistic forms of the input data. Any input expression value can be described through a combination of membership values in the linguistic property sets *low*, *medium* and *high*.

Each input expression value x_{jk} of jth gene of kth sample in quantitative form can be expressed in terms of membership values to each of the three linguistic properties *low*, *medium* and *high*. Therefore for n samples, we have an n-dimensional gene expression pattern $\mathbf{x}_j = [x_{j1}, x_{j2}, \ldots, x_{jn}]^T$ for jth gene, which may be represented as a $3n$ dimensional vector

$$\mathbf{v}_j = [U_{low}(x_{j1}), U_{medium}(x_{j1}), U_{high}(x_{j1}), \ldots, U_{high}(x_{jn})]^T. \tag{1}$$

Here $U_{low}(x_{jk})$ is membership value of jth gene with expression value x_{jk} in kth sample, to the fuzzy set *low*. Hence in trying to express input $\mathbf{x}_j$ with the linguistic properties, we are effectively dividing the dynamic range of expression value into three overlapping partitions called low, medium, and high for each gene. Note that, for reducing complexity, we have already applied normalization

technique (described in Section 2.1) for transforming expression values in various samples of a gene into one value. This makes the dimension of $\mathbf{x}_j$ to be one and hence the dimension of $\mathbf{v}_j$ to be three. That is, for a jth gene we have only one expression value x_j.

We now describe the formulation of the membership functions corresponding to the fuzzy sets *low*, *medium* and *high*. These three membership functions are termed as U_{low}, U_{med}, U_{high} corresponding to the fuzzy sets *low*, *medium* and *high*. Here we have considered triangular membership functions for modelling the fuzzy sets. Thus, the membership function U_{low} is defined as

$$\begin{aligned} U_{low}(x_j) &= 1, && if\ x_j \le x_{min} \\ &= 1 + (x_j - x_{min})/(x_{min} - c_{med}), && if\ c_{low} \le x_j < c_{med} \\ &= 0, && otherwise \end{aligned} \tag{2}$$

Similarly, U_{med} and U_{high} are defined as

$$\begin{aligned} U_{med}(x_j) &= (x_{min} - x_j)/(x_{min} - c_{med}), && if\ c_{low} \le x_j < c_{med} \\ &= (x_{max} - x_j)/(x_{max} - c_{med}), && if\ c_{med} < x_j \le c_{high} \\ &= 0, && otherwise \end{aligned} \tag{3}$$

$$\begin{aligned} U_{high}(x_j) &= 1, && if\ x_j > c_{high} \\ &= 1 + (x_j - x_{max})/(x_{max} - c_{med}), && if\ c_{med} < x_j < x_{max} \\ &= 0, && otherwise \end{aligned} \tag{4}$$

Here x_{max} and x_{min} denote the upper and lower bounds of the observed range of the gene expression values. The parameters are computed as follows:

$$\begin{aligned} c_{med} &= (x_{min} + x_{max})/2 \\ c_{low} &= (c_{med} - x_{min})/2 + x_{min} \\ c_{high} &= (x_{max} - c_{med})/2 + c_{med} \end{aligned}$$

The basic nature of these membership functions is as follows: (i) Maximum value of each function is 1. (ii) Minimum value of each function is 0. (iii) The membership functions corresponding to *low* and *medium*, cut at the point for which $U_{low} = U_{med} = 0.5$. Similar is the case for U_{med} and U_{high} such that at the point of intersections of the membership functions corresponding to *medium* and *high*, $U_{med} = U_{high} = 0.5$. (iv) The membership value corresponding to a gene expression value to a fuzzy set is maximum at the center of the fuzzy set and decreases as it is away from the center of the fuzzy set. It may be noted that one may use other membership functions for modelling the fuzzy sets *low*, *medium* and *high*, keeping the similar basic nature of the membership functions. The choice of c-values automatically ensures that one of the membership values U_{low}, U_{med} or U_{high} of each gene in the corresponding three dimensional linguistic space should be greater than 0.5, and among the other two one should be zero. This allows a gene to have a strong membership to at least one of the properties *low*, *medium*, *high*. So after representing the genes with three linguistic variables, we group the genes based on their membership values into *low*, *medium* or *high*. That is, a gene with membership value to *low* greater than 0.5 is considered,

as a member of the fuzzy set *low*. Thus we have got three classes of genes in *low*, *medium* and *high*. This process is executed both on normal and tumor samples separately. However, the values of the parameters (i.e, c-values) of the membership functions, for both normal and tumor samples are computed, based on the normal lung samples.

3.3 Rule generation based on linguistic variables

The membership values of various genes in both normal and tumor samples are used for rule generation in *if-then* form in order to justify any decision on some genes reached. These rules describe the extent to which a gene is responsible for causing adenocarcinoma in lung. The rules generated are in the form of *if-then*, where the antecedent is formed by two conjunctive clauses – one corresponding to the linguistic representation of a gene in normal samples and the other corresponding to that in tumor samples. The consequent part of the rule represents whether the tumor sample is adenocarcinomic or not. In order to generate the antecedent ("if") part of a rule, we compute confidence factor $CONF$ as given by

$$CONF = \frac{1}{2}[{v_{max}}^{n_{max}} + (1/(cl-1)) \times (\Sigma_j(v_{max} - v_j))],\ 0 \leq CONF \leq 1 \quad (5)$$

where $j = 1, 2, \ldots, cl$; cl being the number of classes. (Here $cl = 3$, as the classes are *low*, *medium* and *high*.) Here $v_{max} = \max_{j=1}^{cl}\{v_j\}$, v_j is the membership value to jth class and n_{max} indicates the number of occurrences of v_{max} in vector $\mathbf{v}$. Note that $CONF$ takes care of the fact that the difficulty in assigning a particular gene to a fuzzy class depends not only on the highest entry in the output vector v_{max} but also on its differences from the other entities v_j . It is seen that the higher the value of $CONF$, the lower is the difficulty in deciding a fuzzy set to which the gene belongs, and hence greater is the degree of certainty of the output decision. Based on the value of $CONF$ the system makes the following decisions heuristically while generating a rule. Let $v_k = v_{max}$ such that the pattern under consideration belongs to the class C_k. We have: (i) if $(0.8 \leq CONF_k \leq 1.0)$ then very likely fuzzy set C_k. (ii) if $(0.6 \leq CONF_k < 0.8)$ then likely fuzzy set C_k. (iii) if $(0.4 \leq CONF_k < 0.6)$ then more or less likely fuzzy set C_k. (iv) if $(0.1 \leq CONF_k < 0.4)$ then not unlikely fuzzy set C_k. (v) if $(CONF_k < 0.1)$ then unable to recognize fuzzy set C_k.

4 Results

In this section, the effectiveness of the proposed method is demonstrated on lung adenocarcinoma gene expression data [1, 2, 5].

4.1 Description of the data set

The data set is obtained by microarray experiments of Affymetrix Corporation for Ann Arbor tumors and normal lung samples [1, 2, 5]. In this data set, there

are expression values of 7129 genes (more specifically, Affymetrix probe-sets) for 86 lung tumor and 10 normal lung samples[19, 28].

Among these 86 tumor samples 67 samples corresponding to stage I and 19 to stage III tumors. There are also 10 neoplastic lung samples. The data set was trimmed of genes expressed at extremely low level. That is, genes were excluded if the measure of their 75th percentile value was less than 100 [2]. Array to array variation in the overall distribution of gene expression values detected by quantile-quantile plots was removed by applying a quantile normalization using a linear spline as a monotone transformation [5]. The gene expression profile of each tumor was normalized to the median gene expression value among all the samples. Features on the oligonucleotide arrays representing the genes in the individual tumors found as outliers were carefully reviewed to confirm expression levels and exclude artifacts. More details on this data set is found in [1, 2, 5].

4.2 Analysis of the results

The above data contains expression value of 10 normal samples of 7129 genes. So there are $\binom{10}{2}$ pairs, i.e., 45 pairs for each gene.

Table 1. Computation of resultant adjustment. Ad_i stands for the adjustment for ith sample.

Sample1	Sample2	Sample3	Sample4	Sample5	Sample6	Sample7	Sample8	Sample9	Sample10
$+a_{1,2}/2$	$-a_{1,2}/2$	$-a_{2,3}/2$	$+a_{2,4}/2$	$-a_{2,5}/2$	$+a_{2,6}/2$	$-a_{2,7}/2$	$+a_{2,8}/2$	$-a_{2,9}/2$	$+a_{2,10}/2$
$-a_{1,3}/2$	$+a_{2,3}/2$	$+a_{1,3}/2$	$-a_{3,4}/2$	$+a_{4,5}/2$	$-a_{5,6}/2$	$-a_{3,7}/2$	$-a_{5,8}/2$	$+a_{3,9}/2$	$-a_{5,10}/2$
$-a_{1,4}/2$	$-a_{2,4}/2$	$+a_{3,4}/2$	$+a_{1,4}/2$	$+a_{3,5}/2$	$+a_{3,6}/2$	$+a_{4,7}/2$	$+a_{3,8}/2$	$+a_{4,9}/2$	$-a_{6,10}/2$
$+a_{1,5}/2$	$+a_{2,5}/2$	$-a_{3,5}/2$	$-a_{4,5}/2$	$-a_{1,5}/2$	$-a_{4,6}/2$	$+a_{5,7}/2$	$-a_{4,8}/2$	$+a_{5,9}/2$	$+a_{7,10}/2$
$-a_{1,6}/2$	$-a_{2,6}/2$	$-a_{3,6}/2$	$+a_{4,6}/2$	$+a_{5,6}/2$	$+a_{1,6}/2$	$+a_{6,7}/2$	$-a_{6,8}/2$	$+a_{6,9}/2$	$-a_{8,10}/2$
$-a_{1,7}/2$	$+a_{2,7}/2$	$+a_{3,7}/2$	$-a_{4,7}/2$	$-a_{5,7}/2$	$-a_{6,7}/2$	$+a_{1,7}/2$	$+a_{7,8}/2$	$-a_{7,9}/2$	$+a_{9,10}/2$
$-a_{1,8}/2$	$-a_{2,8}/2$	$-a_{3,8}/2$	$+a_{4,8}/2$	$+a_{5,8}/2$	$+a_{6,8}/2$	$-a_{7,8}/2$	$+a_{1,8}/2$	$+a_{8,9}/2$	$-a_{4,10}/2$
$+a_{1,9}/2$	$+a_{2,9}/2$	$-a_{3,9}/2$	$-a_{4,9}/2$	$-a_{5,9}/2$	$-a_{6,9}/2$	$+a_{7,9}/2$	$-a_{8,9}/2$	$-a_{1,9}/2$	$-a_{3,10}/2$
$+a_{1,10}/2$	$-a_{2,10}/2$	$+a_{3,10}/2$	$+a_{4,10}/2$	$+a_{5,10}/2$	$+a_{6,10}/2$	$-a_{7,10}/2$	$+a_{8,10}/2$	$-a_{9,10}/2$	$-a_{1,10}/2$
Ad_1	Ad_2	$-Ad_3$	Ad_4	$-Ad_5$	Ad_6	$-Ad_7$	$-Ad_8$	$-Ad_9$	Ad_{10}

We have first of all, applied normalization algorithm described in Section 2.1. According to the algorithm, we have consider pairwise samples and calculate adjustment iteratively until the expression values of the two samples become very closed. This is depicted in Table 1. In Table 1, $a_{1,2}$ indicates the adjustment value of pair, sample 1 and sample 2. We have distributed this adjustment value to the sample 1 and sample 2. Here the log expression value of sample 1 is less than sample 2. So we divided the adjustment, say $a_{1,2}$, such that sample 1 got $+a_{1,2}/2$ and sample 2 got $-a_{1,2}/2$. In this way each sample has 9 values after distribution. Finally, we calculate the resultant adjustment by adding those values including sign for each sample. Now for each sample, we update the old log expression value of a gene by adding resultant adjustment of the corresponding sample. This completes one iteration. After 5 or 6 iterations we have got the normalized value of each sample for a gene. That is, log expression value of the 10 samples

become close enough. In this way we normalized 7129 genes for normal and tumor samples. After normalization, we have performed mean operation on the values of the genes. So ultimately we represented each gene with a single value. This would help us a lot for further analysis as well as for implementation with respect to the complexity of the problem is concerned.

The membership values of 7129 genes to the classes *low*, *medium* and *high* were then calculated using equations (2)-(4), and grouped into three fuzzy classes based on these membership values. In case of normal samples there are 6288 genes in class *low*, 835 genes in class *medium*, 6 genes in class *high*. Similar steps were followed for the 86 tumor samples using the parameter values already computed for normal samples. In the case of tumor samples, there are 6251 genes in class *low*, 871 genes in class *medium*, 7 genes in class *high*. It is interesting to note that number of genes of corresponding to the classes of normal and tumor samples changed significantly. In order to determine the extent of changes, we have compared these classes, i.e., between (low_{normal}, low_{tumor}), ($medium_{normal}$, $medium_{tumor}$), ($high_{normal}$, $high_{tumor}$). Based on this comparison, we have identified a set of 293 genes, each of which has changed the corresponding class. That is, one of these 293 genes may belong to the class *low* for normal samples, but is included in the class other than *low* for tumor samples.

Finally, we applied the rule generation technique based on the algorithm specified in Section 2.3 on these 293 genes to generate 293 rules. Among these 293 genes, we have reported the rules for 9 genes that have changed significantly from normal to tumor samples. These rules are provided in Table 2. For example, the rule for the gene FMO2 is as follows:

If FMO2 is very likely in class medium for normal samples and very likely in class low for tumor samples then the tumor sample is adenocarcinomic.

The results are validated by some earlier investigations on these genes. That is, these genes were found to be responsible for lung adenocarcinoma by these investigations. These include the references in [3] for RPL0, [10, 24] for ADH1, [22, 26] for UGB, [17] for FMO2, [8, 13] for HBA2, [30, 15, 25] for SFTPA1, [21] for SFTPA2, [16, 9] for HBB.

5 Conclusions

In this article, we have described a rule generation method for identifying a few genes responsible for a specific disease. First of all, the expression values for genes in different samples were normalized to remove sources of variation between the arrays. Here we have used a cyclic loess normalization method [11] on normal and tumor samples. After normalization, we have performed the mean operation that is mainly used to represent a gene by a single log expression value. We have then applied the concept of fuzzy sets to classify the genes into three fuzzy classes, viz., *low*, *medium*, and *high*. Incorporation of fuzzy set theory makes the system capable of handling uncertainty in the gene expression values arising due to incompleteness, imprecision, noise and experimental errors. Now we have

Table 2. Rules corresponding to the nine most significant genes mediating the development of lung adenocarcinoma. The consequent parts of all these rules are "the tumor sample is adenocarcinomic".

Gene Name	Antecedent clauses corresponding to	
	Normal	Tumor
RPLP0	Likely medium	Very likely high
NULL	Very likely medium	Very likely high
ADH1	Likely medium	Very likely low
UGB	Likely medium	Very likely low
FMO2	Very likely medium	Very likely low
HBA2	Likely medium	Very likely low
SFTPA1	Very likely high	Likely medium
SFTPA2	Very likely high	Likely medium
HBB	Very likely high	Very likely low

performed matching operation with normal classes with the tumor classes. After matching we have identified the genes that moved significantly from one class of normal to another class of tumor or vice versa.

Applying the above method on a lung adenocarcinoma data set containing gene expression values, we have grouped those genes into three different classes *low*, *medium*, *high*. The *low*, *medium* and *high* classes for normal consist of 6288, 835 and 6 genes respectively. Similarly, the *low*, *medium* and *high* classes of tumor consists of 6251, 871 and 7 genes respectively. We have compared all three classes of normal with *low*, *medium* and *high* classes of the tumor. On this comparison, we have found 293 genes that are significantly changed their expression level from normal to the tumor. Among these 293 genes, we have reported nine genes that have changed significantly based on the rule we have got. These 9 genes include RPLP0, ADH1, UGB, FMO2, HBA2, SFTPA1, SFTPA2, HBB. Among those 9 genes we have found, the gene HBB that have moved most significantly from class *high* of normal to the class *low* of the tumor. We have reported that this gene is a down-regulated gene that is mediating the development of lung carcinoma. Therefore, over or under expression of these 9 genes are responsible for the development of lung carcinoma. These results have been validated by a number of earlier investigations.

References

1. http://www.camda.duke.edu/camda03/datasets/
2. http://groups.yahoo.com/group/camdadata/
3. Abo, Y., Hagiya, A., Naganuma, T., Tohkairin, K. S. Y., Kajiura, Z., Hachimori, A., Uchiumi, T., Nakagaki, M.: Baculovirus-mediated expression and isolation of human ribosomal phosphoprotein p0 carrying a gst-tag in a functional state. Biochem Biophys Res Commun. **322** (2004) 814–819

4. Azuaje, F.: A computational neural approach to support the discovery of gene function and classes of cancer. IEEE Transactions on Biomedical Engineering **48** (2001) 332–339
5. Beer, G. D., et. al.: Gene-expression profilespredict survival of patients with lung adenocarcinoma. Nature Medicine **8** (2002) 816–823
6. Bhattacharjee, A., et. al.: Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc. Natl. Acad. Sci., USA **98** (2001) 13790–13795
7. Brazma, A., et. al.: Minimum information about a microarray experiment (miame)-toward standards for microarray data. Nat Genet **29** (2001) 365–371
8. Brown, J. M., Leach, J., Reittie, J. E., Atzberger, A., Lee-Prudhoe, J., Wood, W. G., Higgs, D. R., Iborra, F. J., Buckle, V. J.: Coregulated human globin genes are frequently in spatial proximity when active. J Cell Biol. **172** (2006) 177–187
9. Buzina, A., Aladjem, M. I., Kolman, J. L., Wahl, G. M., Ellis, J.: Initiation of DNA replication at the human beta-globin 3' enhancer. Nucleic Acids Res. **33** (2005) 4412–4424
10. Dannenberg, L. O., Chen, H. J., Edenberg, H. J.: Gata-2 and hnf-3beta regulate the human alcohol dehydrogenase 1a (adh1a) gene. DNA Cell Biol. **24** (2005) 543–552
11. Dudoit, S., Yang, Y. H., Callow, M. J., Speed, T. P.: Statistical methods for identifying genes with differential expression in replicated cDNA microarray experiments. Stat. Sin **12** (2002) 111–139
12. Garber, M. E., et. al.: Diversity of gene expression in adenocarcinoma of the lung. Proc. Natl. Acad. Sci. USA **98** (2001) 13784–13789
13. Haider, M., Adekile, A.: Alpha-2-globin gene polyadenylation (AATAAA → AATAAG) mutation in hemoglobin H disease among kuwaitis. Med Princ Pract. **14 Suppl 1** (2005) 73–76
14. Hartemink, A., Gifford, D., Jaakkola, T., Young, R.: Maximum likelihood estimation of optical scaling factors for expression array normalization. SPIE BIOS (2001)
15. Jiang, F., Caraway, N. P., Bekele, B. N., Zhang, H. Z., Khanna, A., Wang, H., Li, R., Fernandez, R. L., Zaidi, T. M., Johnston, D. A., Katz, R. L.: Surfactant protein a gene deletion and prognostics for patients with stage I non-small cell lung cancer. Clin Cancer Res. **11** (2005) 5417–5424
16. Jiang, Y., Xu, X. S., Russell, J. E.: A nucleolin-binding 3' untranslated region element stabilizes beta-globin mrna in vivo. Mol Cell Biol. **26** (2006) 2419–2429
17. Krueger, S. K., Siddens, L. K., Henderson, M. C., Andreasen, E. A., Tanguay, R. L., Pereira, C. B., Cabacungan, E. T., Hines, R. N., Ardlie, K. G., Williams, D. E.: Haplotype and functional analysis of four flavin-containing monooxygenase isoform 2 (FMO2) polymorphisms in hispanics. Pharmacogenet Genomics. **15** (2005) 245–256
18. Liotta, L., Petricion, E.: Molecular profiling of human cancer. Nature Rev. Genet **1** (2000) 48–56
19. Lipshutz, R., Fodor, S., Gingeras, T., Lockart, D.: High density synthetic oligonucleotide arrays. Nat. Genet **21** (1999) 20–24
20. Machado, L., Vinterbo, S., Weber, G.: Classification of gene expression data using fuzzy logic. Journal of Intelligent and Fuzzy Systems **12** (2002) 19–24
21. Madan, T., Saxena, S., Murthy, K. J., Muralidhar, K., Sarma, P. U.: Association of polymorphisms in the collagen region of human sp-a1 and sp-a2 genes with pulmonary tuberculosis in indian population. Clin Chem Lab Med. **40** (2002) 1002–1008

22. Martin, A. C., Laing, I. A., Khoo, S. K., Zhang, G., Rueter, K., Teoh, L., Taheri, S., Hayden, C. M., Geelhoed, G. C., Goldblatt, J., LeSouef, P. N.: Acute asthma in children: Relationships among cd14 and cc16 genotypes, plasma levels, and severity. Am J Respir Crit Care Med. **173** (2005) 617–622
23. Mitra, S., De, R. K., Pal, S. K.: Knowledge-based fuzzy mlp for classification and rule generation. IEEE Transactions on Neural Networks **8** (1997) 1338–1350
24. Rodriguez-Zavala, J. S., Weiner, H.: Structural aspects of aldehyde dehydrogenase that influence dimer-tetramer formation. Biochemistry **41** (2002)
25. Saxena, S., Kumar, R., Gupta, T. M. V., Muralidhar, K., Sarma, P. U.: Association of polymorphisms in pulmonary surfactant protein a1 and a2 genes with high-altitude pulmonary edema. Chest **128** (2005) 1611–1619
26. Shashikant, B. N., Miller, T. L., Welch, R. W., Pilon, A. L., Shaffer, T. H., Wolfson, M. R.: Dose response to RHCC10-augmented surfactant therapy in a lamb model of infant respiratory distress syndrome: physiological, inflammatory, and kinetic profiles. J Appl Physiol. **99** (2005) 2204–2211
27. Sokhansanj, B. A., Fitch, J. P., Quong, J. N., Quong, A. A.: Linear fuzzy gene network models obtained from microarray data by exhaustive search. BMC Bioinformatics **5** (2004) 108–119
28. Warrington, J. A., Dee, S., Trulson, M. (editors): Large-scale genomic analysis using Affymetrix genechip. M. Schena (2000) 119–148
29. Woolf, P. J., Wang, Y.: A fuzzy logic approach to analyzing gene expression data. Physiol Genomics **3** (2000) 9–15
30. Wu, Y. Z., Manevich, Y., Baldwin, J. L., Dodia, C., Yu, K., Feinstein, S. I., Fisher, A. B.: Interaction of surfactant protein a with peroxiredoxin 6 regulates phospholipase a2 activity. J Biol Chem. **281** (2006) 7515–7525
31. Zhang, H. et. al.: Recursive partitioning for tumor classification with gene expression microarray data. Proc Natl Acad Sci USA **98** (2001) 6730–6735
32. Vinterbo, S. A., Kim, E. Y., Machado, L.: Small, fuzzy and interpretable gene expression based classifiers. Bioinformatics **21** (2005) 1964–1970

A Counting Technique based on SVM-RFE for Selection and Classification of Microarray data

Jose Crispin Hernandez Hernandez, Béatrice Duval, and Jin-Kao Hao

LERIA, Université d'Angers,
2 Boulevard Lavoisier, 49045 Angers, France
{josehh,bd,hao}@info.univ-angers.fr

Abstract. Selecting informative genes for the classification of tumor tissues from microarray data is a challenging problem in bioinformatics. Recursive Feature Elimination with Support Vector Machines (SVM-RFE) is known to be an effective method for selection and classification of microarray data. In this paper, we introduce a novel gene ranking method based on a frequency analysis in gene subsets processed by SVM-RFE. This novel ranking leads to select as informative genes the top ranked genes. The effectiveness of this method is assessed using two well-known benchmark data sets from the literature, showing highly interesting results.

Key words: Microarray data analysis, Support vector machines, Adatron, Recursive feature elimination

1 Introduction

Recent advances in DNA microarray technologies enable to consider molecular cancer diagnosis based on gene expression. Classification of tissue samples from gene expression levels aims to distinguish between normal and tumor samples, or to recognize particular kinds of tumors [9, 1]. Gene expression levels are obtained by cDNA microarrays and high density oligonucleotide chips, allowing to monitor and measure simultaneously gene expressions for thousands of genes in a sample. So, data that are currently available in this field concern a very large number of variables (thousands of gene expressions) relative to a small number of observations (typically under one hundred samples). This characteristics, known as the "curse of dimensionality", is a difficult problem for classification methods and requires special techniques to reduce the data dimensionality in order to obtain reliable predictive results.

Feature selection is one of the most popular ways for reducing data dimensionality. Feature selection aims at selecting a (small) subset of informative features (genes) from the initial data in order to obtain high classification accuracy [10, 6, 18]. A feasible approach to feature selection is to rank all the features according to their interestingness for the classification problem and to select the

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 99-107

Received 10/07/06
Accepted 03/10/06
Final version 12/10/06

top ranked features. The feature score can be obtained independently for each feature, as it is done in [9] which relies on correlation coefficients between the class and each feature. The drawback of such a method is to score each feature independently while ignoring the relations between the features. More recently, [11] presents recursive feature elimination using support vector machines (SVM-RFE), which relies also on ranking criteria but takes into account the relations between features.

SVM-RFE is a backward feature elimination method [14] that searches among the n initial features a subset of d features that maximizes the performance of a SVM classifier. To achieve this, one starts with all the features and iteratively removes one feature until d of them are left. The removal is based on a ranking criterion obtained from a SVM classifier trained on the current subset of features. One of the difficulties with such an approach is the choice of the appropriate number d of selected features [17]. Moreover, SVM-RFE is basically a greedy method that studies nested subsets of features, since the selected subset of size m is included in the previously selected subset of size $m+1$. As pointed out in [11], there is no guarantee that this search strategy leads to optimal results.

The most important motivation of this work is to better understand how the selected features occur during the iterative steps of SVM-RFE. So this paper presents a novel approach of gene selection that combines the recursive feature elimination algorithm with a counting method that allows to identify genes that have a recurrent importance at different stages of the elimination process.

The paper is organized as follows: in Section 2, we review the main characteristics of SVM, and explain how RFE ranks and selects the genes according to their importance for class discrimination. In Section 3, we describe our counting technique for gene ranking. Experimental results are presented in section 4, before the conclusion given in Section 5.

2 Feature ranking with SVM-RFE

2.1 Support Vector Machines (SVM)

The principle of a SVM classifier is to find an optimal hyperplane as a decision function in a high-dimensional space [3]. Thus, let us consider a training data set $\{x_k, y_k\} \in \Re^n \times \{-1, 1\}$ where x_k are the m training examples and y_k the class labels. At first, the method consists in mapping x_k in a high dimensional space owing to a function Φ. Then, it looks for a decision function of the form:

$$f(x) = w \cdot \Phi(x) + b \tag{1}$$

and $f(x)$ is optimal in the sense that it maximizes the distance between the nearest points $\Phi(x_k)$ and the hyperplane (this distance is called margin). The class label of x is then obtained by considering the sign of $f(x)$. This optimization problem can be transformed, in the case of L2 soft-margin SVM classifier (misclassified examples are quadratically penalized), in this following one:

$$\min_{w,\xi} \frac{1}{2} \|w\|^2 + C \sum_{k=1}^{m} \xi_k^2 \tag{2}$$

under the constraint $\forall k,\ y_k f(x_k) \geq 1 - \xi_k$. The solution of this problem is obtained using the Lagrangian theory and one can find that the vector w is of the form:

$$w = \sum_{k=1}^{m} \alpha_k^* y_k \Phi(x_k) \tag{3}$$

where α_k^* is the solution of the following quadratic optimization problem:

$$\max_{\alpha} W(\alpha) = \sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k=1,l=1}^{m} \alpha_k \alpha_l y_k y_l \left(K(x_k, x_l) + \frac{1}{C} \delta_{k,l} \right) \tag{4}$$

subject to $\sum_{k=1}^{m} y_k \alpha_k = 0$ and $\forall k,\ \alpha_k \geq 0$, with $\delta_{k,l}$ being the Kronecker symbol and $K(x_k, x_l) = \langle \Phi(x_k), \Phi(x_l) \rangle$ is the Gram matrix of the training examples.

The SVM is commonly trained using either mathematical programming (MP) approaches such as quadratic programming or by strategies that avoid the use of the MP techniques. The latter techniques have the advantage of being easier to implement, while providing similar levels of performance as their MP counterparts. One non-MP based approach is the kernel-Adatron (KA) algorithm [5, 13]; this algorithm iteratively updates the Lagrange multipliers, α, associated with each example. The Adatron has been demonstrated as a useful tool for analyzing microarray data (see [4]).

2.2 Recursive feature elimination (RFE)

Molecular biologists and oncologists seem to be convinced that only a small subset of genes are responsible for particular biological properties, so they want to identify the most important or informative genes. Guyon et al. [11] propose an SVM-based learning method, called SVM Recursive Feature Elimination (SVM-RFE) for gene selection. The motivating idea is that the orientation of the separating hyperplane found by a linear SVM can be used to select informative features: if the plane is orthogonal to a particular gene dimension, then that gene is informative, and vice versa. Specially, given a SVM with weight vector $w = \sum_{k=1}^{m} \alpha_k y_k x_k$, the ranking criterion for gene i is $c(i) = (w_i)^2$. Thus, one starts with all the genes; at each step of the RFE procedure, a classifier is trained on the given sample set, which determines a ranking value for each gene; then the gene with the smallest value is discarded and the process is iterated. RFE thus removes one feature at a time until a single gene is left.

The computational cost of RFE is a function of the number of features, because a classifier must be trained each time a feature is removed. The removal of chunks of features at each loop represents a feasible approach. Another alternative is to eliminate a chunk of least important features per step until a small number of features is left, from which point, one feature is removed per step [6, 7].

SVM-RFE is a powerful method for selection and classification, but the reasons of its effectiveness are not completely known yet. Several authors have studied this technique to better understand its properties. In [15], the authors

present experiments of recursive and non recursive selection procedures applied with different types of classifier in order to see whether the success depends on the specific classifier or on the recursive procedure.

3 A combined technique for gene ranking using SVM-RFE

As explained in the introduction, SVM-RFE works on nested subsets: from the selected subset of size m, the system builds a SVM classifier associated to an hyperplane determined by a vector of weights w. The coefficients w_i^2 are the ranking coefficients of the genes and the gene with the smallest coefficient is removed to obtain the subset of size $m-1$ on which this process is iteratively applied. Even if one can determine a unique feature that gives the best separation between two classes, there is no guarantee that the best pair of features for the classification must contain this feature. So the search strategy of SVM-RFE does not guarantee an optimal result.

This work proposes to analyze the different subsets of genes selected during the successive steps of SVM-RFE. More precisely, we propose a model where the ranking of genes during each step of SVM-RFE determines an importance weight for each gene (the first gene has the greatest importance weight); then we compute the sum of these weights which assigns to each gene an importance value that takes into account the role of this gene during the whole process of SVM-RFE. Finally, this counting, based on weighted occurrences, is used to establish a new ranking of the genes. This model is presented in Algorithm 1.

The algorithm implements the different steps of a recursive feature elimination. At each step, we use the kernel-Adatron algorithm to compute the coefficients of the SVM classifier and we obtain the ranking coefficients c_i. The feature associated to the smallest ranking value is eliminated from the subset of "surviving features".

Moreover, the ranking is used to determine the importance coefficient assigned to each feature: at each step, the feature that is top ranked receives a weight of n, where n is the initial number of features, the second ranked gene receives a weight of $n-1$, and so on... Finally the sum of these weights records the different positions of the feature during the process and produces a novel ranking of the set of features.

4 Experimental results

4.1 Data sets

We applied our approach on two well-known data sets: the leukemia data set and The colon cancer data set. The leukemia data set consists of 72 tissue samples, each with 7129 gene expression values. The samples include 47 acute lymphoblastic leukemia (ALL) and 25 acute myeloid leukemia (AML). The original data are divided into a training set of 38 samples and a test set of 34 samples. The

1. The data set is initialized with the m examples.
Each example is described by n features (gene expression values)
Training samples : $X = [x_1,\ x_2,\ ...\ x_k,\ ...x_m]^T$
Class labels : $y = [y_1,\ y_2,\ ...\ y_k,\ ...y_m]^T$
2. Initialize the following variables:
Subset of surviving features : $s = [1,\ 2,\ ...\ i,\ ...n]$
Feature ranked list : $r = [\,]$
Importance of features : $Imp\,[1,\ 2,\ ...\ i,\ ...n] = 0$
3. **repeat**
   Restrict training samples to good feature indices : $X = X(:,\ s)$
   Train the classifier : $\alpha = Adatron\_train(X,\ y)$
   Compute the weight vector of dimension $length(s)$: $w = \sum_k \alpha_k y_k x_k$
   Compute the ranking criteria : $c_i = w_i^2$, for all i in s
   Sort the features according to c_i; let $position(i)$ be the place of feature i in this sorted list: 1 for the last feature associated to the smallest c_i and length(s) for the best feature associated to the greatest c_i
   Find the feature with smallest ranking criterion: $f = argmin(c)$
   Update the importance value of all features

$$Imp(i) = \begin{cases} Imp(i) + (n - length(s)) + position(i) & : \quad i \in s \\ Imp(i) & : \quad i \in r \end{cases}$$

   Update feature ranked list : $r = [s(f),\ r]$
   Eliminate the feature with smallest ranking criterion :
   $s = s[1 : f-1,\ f+1 : length(s)]$
until $s = [\,]$;
4. Sort the n features, in decreasing order, according to their importance value $Imp(i)$: the first feature is associated to the greatest value $Imp(i)$ and it is the index of the top ranked gene

Algorithm 1: Feature Ranking by frequency counting and SVM-RFE

data were produced from Affymetrix gene chips. The data set was first used in [9] and is available at http://www-genome.wi.mit.edu/cancer/.

The colon cancer data set contains 62 tissue samples, each with 2000 gene expression values. The tissue samples include 22 normal and 40 colon cancer cases. The data set is available at http://www.molbio.princeton.edu/colondata and was first studied in [1].

A linear normalization procedure transforming the gene expression to mean value 0 and standard deviation 1 was applied to each dataset.

4.2 Experiments and results

We applied our model on each data set. In each case, we obtained a ranked list of genes, from which we could select a subset of interesting genes. For a selected subset, we used a leave-one-out cross-validation (LOOCV) estimate of the misclassification rate, which means that each observation of the training set is successively omitted from the data and then classified by a classifier trained on

the remaining observations, the mean of these experiments gives the estimated misclassification rate. For the classification task, we used a SVM-classifier.

Leukemia classification We ran our gene ranking method on the entire data set of 72 samples. The resulting misclassification rate is 0/72 (100% of correct prediction) when we use the subset of the four top ranked genes. This result is similar to the best published results: this maximal rate of correct classification is obtained with 6 genes in [16], but only with 2 genes in [11]. Table 1 shows the four top ranked genes obtained with our method. The first two ranked genes have been reported in [6], the third has been reported in [8] and the fourth has been reported in [9].

Ranking	Gene Accession Number	Description
1	M27891	CST3 Cystatin C (amyloid angiopathy)
2	M19507	MPO Myeloperoxidase
3	M26708	PTMA Prothymosin alpha
4	M63138	CTSD Cathepsin D (lysosomal aspartyl protease)

Table 1. The 4 best-ranked genes of Leukemia data

Colon classification We put all the Colon cancer data into one training set of 62 samples. The LOOCV misclassification rate obtained is 1/62 when we use the subset of four top ranked genes (see table 2). The first ranked gene has been reported in [12], the second and third ranked genes have been reported in [11], and the fourth ranked gene has been reported in [7]. The accuracy of our method is equal to the best result known for this data set and reported in [11]; the unique misclassified sample is sample number 6, which is a normal tissue reported by [1]. As we will show in the next section, the misclassification rate of 1/62 can be improved by combing our frequency counting technique with an exhaustive search, leading to a 100% correct classification.

Ranking	Gene Accession Number	Description
1	H06524	Gelsolin Precursor, plasma (HUMAN).
2	T62947	60S ribosomal protein L24 (Arabidopsis thaliana)
3	H64807	Placental folate transporter (Homo sapiens)
4	R62549	Putative serine/threonine-protein kinase b0464.5

Table 2. The 4 best-ranked genes of colon cancer data

More experiments on colon data set

When we further analyzed the gene subsets processed by RFE, we observed a significant change at the stage that considers 32 genes. For the previous steps, i.e. from the step that considers 2000 genes up to the step that considers 32

genes, a particular gene, X69550, is always the best ranked one according to the c_i coefficients. But in the following steps (with 32 genes and less than 32 genes), this particular gene is no longer in the first position and is eventually eliminated when only 16 genes are retained.

This fact represents thus an important change in the subsets of relevant genes and indicates that it may be useful to take into account the role of a gene along the different steps of a RFE process. This consideration goes with what our counting-based ranking method does. In order to understand what really happens from the step which considers the subset of 32 top ranked genes, we decided to have a closer look at these genes and performed a combinatorial search based on them.

Concretely, from these 32 genes, we examine all the subsets of k genes with increasing values of k=1,2... until a 100% classification rate is reached. This experiment proves to be successful since we found that it is possible to classify the whole data set without error (0/62) with only k=4 genes (using a leave-one-out cross-validation estimation). Once this result is reached, the exhaustive search process is terminated.

Quantitatively, this 62/62 classification rate with only 4 genes for the colon cancer data set constitutes the best possible result and has never been reported in the literature before. The set of 4 genes leading to this result contains (T62947, J02854, T57882 and D14812). T62947 has been reported in [11], J02854 has been reported in [1], T57882 has been reported in [6], and D14812 in [7].

According to our ranking, these genes have respectively received the positions (2,6,10 and 19). This result is very interesting since it shows that the 20 top ranked genes using our ranking contain a subset of genes that enables a perfect classification of the colon data set. Notice that this is not the case if we consider the 20 top ranked genes using SVM-RFE alone. This indicates that some important information is obtained by our ranking method when observing the multiple occurrences of a gene across different selection processes.

Comparison with SVM-RFE with correction of bias selection

The classification rates given in the previous parts are obtained by a LOOCV process that assesses the performance of the classifiers on a set of features previously selected, as it was done in [11]. As it was later pointed out in [2], error estimation and gene selection are not independent processes because both are based on the same training set and this way to conduct LOOCV induces a bias in the results, that are too optimistic. The proper way to evaluate a feature selection method is to perform a cross validation that is external to the selection process.

Therefore we carried out a comparison between SVM-RFE and our counting method with the following evaluation process. The initial data set is splitted into two subsets, a training set and a test set. Each method (SVM-RFE and our counting method) is applied on the training set to select a subset of relevant genes and to build a classifier; then this classifier is used on the test set to estimate the performance. 50 trails are performed, with a new split of the data

into a training set and a test set each time. Table 3 summarizes the results for our counting method, for SVM-RFE and two other methods from the literature. We report the average and the standard deviation of the correct classification rate when n genes are selected; we give the results for the number n that gives in these experiments the best expected classification rate for our counting method and for SVM-RFE. For the colon data set, each of the 50 experiments splits the data into two groups of 31 samples, while ensuring that each group has 11 normal and 20 cancerous tissues. For the leukemia data set, the data are splitted into two groups, one of 38 samples (25 acute lymphoblastic leukemia and 13 acute myeloid leukemia) and one of 34 samples (22 acute lymphoblastic leukemia and 12 acute myeloid leukemia) for each of the 50 splits.

As it can be observed, on the two data sets, our counting method provides the best classification rate.

	Our Counting Method	SVM-RFE	[2]	[12]
Leukemia $n = 64$	98.18% ±1.45%	97.47% ±1.88%	≈95%	-
Colon $n = 32$	89.36% ±0.03%	88.71% ±0.04%	82.5%	88.84%

Table 3. Mean and standard deviation for classification rate over 50 splits

5 Conclusions and future work

This paper deals with the problem of selection of relevant genes for the classification of microarray data. We propose a novel gene ranking method, based on the SVM-RFE process, that provides encouraging results. Our ranking relies on an analysis of the different subsets of gene selected by RFE and takes into account the role of a gene along the different steps of the selection process. With the top ranked genes, we obtain classification results, which is better than that SVM-RFE. Moreover, on the colon cancer data set, we show that we can extract from the 32 top ranked genes a subset of 4 genes that gives a 100% classification accuracy. This classification accuracy (with a bias selection) was never reported before in the literaure.

References

1. U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci USA*, 96:6745–6750, 1999.
2. C. Ambroise and G.J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci USA*, 99(10):6562–6566, 2002.

3. B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152, 1992.
4. T. S. Davison, S. Mehta, I. J. Burgetz, and O. Huner. Support vector machine classification of data quality in microarray experiments. In *Critical Assessment of Microarray Data Analysis*, 2001.
5. T. Friess, N. Cristianini, and C. Campbell. The kernel adatron algorithm: a fast and simple learning procedure for support vector machine. In *15th International Conference on Machine Learning, Morgan Kaufman Publishers*, 1998.
6. L.M. Fu and C.S. Fu-Liu. Evaluation of gene importance in microarray data based upon probability of selection. *BMC Bioinformatics*, 6, 2005.
7. C. Furlanello, M. Serafini, S. Merler, and G. Jurman. Entropy-based gene ranking without selection bias for the predictive classification of microarray data. *BMC Bioinformatics*, 4, 2003.
8. G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene expression data. *Proc Natl Acad Sci USA*, 97(22):12079–12084, 2000.
9. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
10. I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 545–552. MIT Press, 2005.
11. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.
12. T. M. Huang and V. Kecman. Gene extraction for cancer diagnosis by support vector machines - an improvement. *Artificial Intelligence in Medicine*, 35(1-2):185–194, 2005.
13. V. Kecman, M. Vogt, and T. M. Huang. On the equality of kernel adatron and sequential minimal optimization in classification and regression tasks and alike algorithms for kernel machines. In *11th European Symposium on Artificial Neural Networks ESANN*, 2003.
14. R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, 1997.
15. F. Li and Y. Yang. Analysis of recursive gene selection approaches from microarray data. *Bioinformatics*, 21(19):3741–3747, October 2005.
16. S. Peng, Q. Xu, X.B. Ling, X. Peng, W. Du, and L. Chen. Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. *FEBS Lett.*, 555(2):358–362, 2003.
17. A. Rakotomamonjy. Variable selection using svm-based criteria. *Journal of Machine Learning Research*, 3:1357–1370, 2003.
18. J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *NIPS*, pages 668–674, 2000.

Software for a X Ray Tomograph

Rosario Martínez Gómez, Israel Vite Silva, and Luis G. de la Fraga

Cinvestav, Department of Computing.
Av. Instituto Politécnico Nacional 2508. 07360 México D.F., México
Phone (+52) 55 50613755, E-mail: fraga@cs.cinvestav.mx

Abstract A software tool-box has been developed, which allows, not only the volume reconstruction from X ray tomography images, but also the visualization of reconstructed isosurfaces. Although it may be considered that the software needed is tightly linked to the hardware, we'll prove that we need five different (hardware independent) software components for the whole process: (1) acquisition, (2) projection's centering, (3) reconstruction, (4) isosurfaces segmentation and (5) visualization. The reconstruction component was taken from Xmipp, an old software for 3D reconstruction of biological macromolecules, segmentation was developed using k-means algorithm and the visualization was built using the splatting technique. In addition we compare splatting with another two surface visualization techniques such as simple voxels and deformable simplex meshes. The most of the software components were developed in C, C++ and perl. The graphical user interface for the visualization component was developed in C++ using Qt and OpenGL libraries.

Keywords: 3D visualization, splatting algorithm, tomography.

1 Introduction

The objective of a three-dimensional reconstruction is to obtain information about nature and structure of materials that conform the inside of an object. The are several applications, for example data from electron microscopes are used to reconstruct the molecular structure of proteins or to reconstruct the x-ray structure of an astronomical object such as supernova remnant, however one of the most important applications has been in medicine to obtain the density distribution within the human body from multiple x-ray projections [1]. This process is referred to as computerized tomography and is a widely used technique, which enables the reconstruction from projections and the visualization of the internal structure of objects.

A tomograph generates images, named projections, of cross sections of the scanned object, from data obtained by measuring the attenuation of x rays along a large number of lines throughout specimen under study. Despite the fact that acquisition part, in a process of computerized tomography, is tightly linked to the hardware, another components of software are needed in order to solve the

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 109-120

Received 12/07/06
Accepted 04/10/06
Final version 13/10/06

whole problem of visualizing reconstructions. Thus, we have analyzed five main parts of software:

1) projection's acquisition, 2) projection's centering, 3) specimen's reconstruction, 4) segment the different densities of the reconstructed object, and the last, 5) visualization of the surface for each segmented density. Every part is independent of the others and consists of one or several programs, but all parts must be used to resolve the whole problem of visualizing a tomographic reconstruction.

This paper is organized as follow: in section 2 are described part (1) to (4) of the developed software. In section 3 is described the visualization part developed with splatting technique and it is compared with others two visualization techniques, one with voxels faces, and the other one with deformable simplex meshes. Finally the conclusions are presented.

2 Software Parts

2.1 Acquisition

In this part we acquire projections of the specimen under study. In Fig. 1 is shown a photograph of the tomograph for which we developed the software described in this work. The X ray source generates radiation that goes throughout the specimen. X rays are invisible to human eye, therefore to detect them, their photons must hit a phosphorescent screen, by this way the intensity of the light viewed on the screen is proportional to the density and composition of the scanned object. In order to obtain several specimen's projection views, the revolving plate rotates the specimen around a fixed axis. The projections are collected with a CCD camera, it uses an optical device to get the views on the phosphorescent screen. The process geometry depends on a single rotation angle, it is known as *single axis geometry* [1,2].

All tomograph's devices are controlled by an old PC with ISA slots and Windows 98 operating system. The card's drivers to transfer the CCD's images to the PC, and to shut the X ray beam, only work in that operating system. There is not enough information about the PC's cards, therefore we were unable to migrate this program to the GNU/Linux operating system.

2.2 Reconstruction

We built a synthetic volume of mathematically described objects (it is called phantom), placed at desired positions, at desired orientations and of desired size and density, in order to separate possible problems which can not be separated physically such as acquisition's noise. In addition we tested two reconstruction methods: weighted backprojection and ART (Algebraic Reconstruction Technique), both were adapted from Xmipp collection [3], which is an old free-software for reconstructing biological macromolecules. Since backprojection's algorithm calculates the inverse process for generating a projection the result is

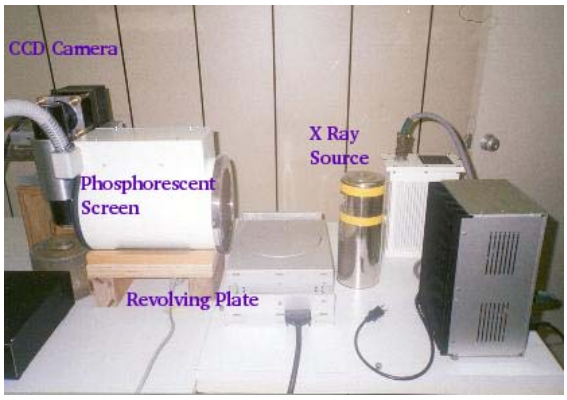


Figure 1: A tomograph's photograph.

blurred, because the most of information is concentrated in the low frequencies. Then a filter must be applied to correct this problem. We implemented the Wiener and Ramp filters and these are applied to each projection in the reciprocal space. Ramp-filter is typically used in weighted backprojection hence it produces more weight to high frequencies. On the other hand, Wiener-filter has the aim of filtering noise which it is tightly linked to the acquisition process.

Ramp-filter is very simple, in two dimensions it is specified as in Eq. 1. Where r is equal to the circle's radius i.e. $\sqrt{x^2+y^2}$, and k is a constant proportional to the Nyquist frequency [4].

Although the reconstruction weighted by Ramp-filter is acceptable, it become worse when noise is present since this filter amplifies high frequencies. In order to solve it Wiener-filter could be used.

$$q(r) = \begin{cases} r\ , & \text{if } r \quad k \\ 0\ , & \text{if } r > k \end{cases} \tag{1}$$

To design the Wiener-filter, we require knowledge about noise and object statistics. Eq. 2 express the Wiener-filter, where $f = \sqrt{(x^2+y^2)}$, and typical used values are $\alpha = 0.5$ and $SNR = 60$; the noise commonly is modeled as white additive noise [5].

$$A(f,\theta) = \frac{f\ 2\pi\alpha(SNR)}{2\pi\alpha(SNR) +\ f\ (\alpha^2 + 4\pi^2 f^2)^{\frac{3}{2}}} \tag{2}$$

To test the developed programs we generated seventy two phantom's projections (taken every 5 degrees on the tilt angle), then we performed three reconstructions using the reconstruction algorithms: backprojection, weighted backprojection, and ART. To compare the reconstructions, we drew the changes in the density along the 15th row on the 30th slice as is shown in Fig. ref-fig:eficiencia. The simple line represents the values of the original phantom's densities, the backprojection-reconstruction is represented by the square-line, it

is improved when Ramp-filter is used (asterisk-line), however this filter decrease the dynamic amplitude, thus the best reconstruction is developed by ART, which it is represented by the cross-line.

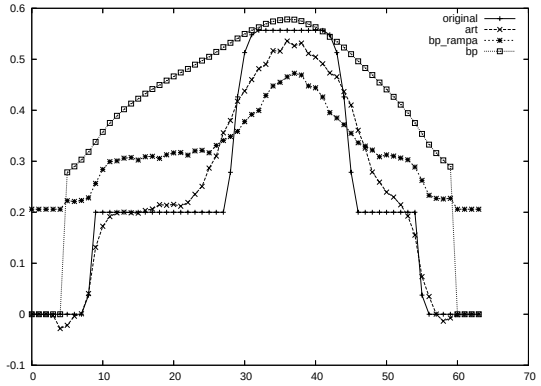


Figure 2: Graph of voxels' density values along the 15th row and 30th slice of the reconstructed volumes. The best reconstruction is performed by ART, which it is represented by the green line.

2.3 Projections centering

Before performing a 3D reconstruction, all projections must be located in a coordinate system fixed to the object, because if we take projections without being aligned, we will reconstruct a distorted volume [1,2]. It is possible to set experimentally the coordinate system origin, first we need to put the specimen on the revolving plate, after rotating the specimen we are able to verify if the still-specimen is in the plate's center. This is a boring trial-error process, which could be solved by software, then we going to show how we solved this problem.

The centering problem can be divided in two cases: (a) when object is small, and (b) when object is big. Fig. 3 shows the (a) case, a noncentered small object, the object's center is a l distance from the revolving plate center.

To solve the second case, noncentered projections of a small object, we can assumed that the coordinate system origin was in the object's center of mass, hence we could estimate it with great accuracy from the projection's centroid because these are free of noise.

The projection's center is a good reference when we can calculate it, that is, when each projection shows the whole object, however we can not calculate the centroid when the object is bigger and their projections do not show the whole object. Fig.4 is a diagram of this case: a big object is not in the revolving plate's center and the object's center is at a l distance of the plate's center. As we know

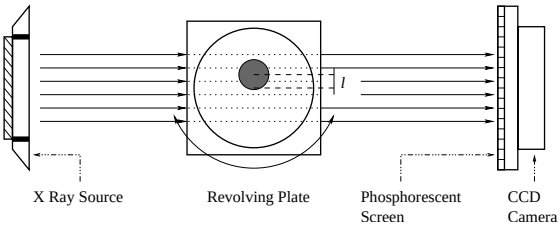


Figure 3: The object is not in the plate's center. There is a distance l from the plate's center to the specimen's coordinates origin.

the tilt angle for each projection, the displaced distance for each projections is $l \cos \phi$, where ϕ is the projection's tilt angle.

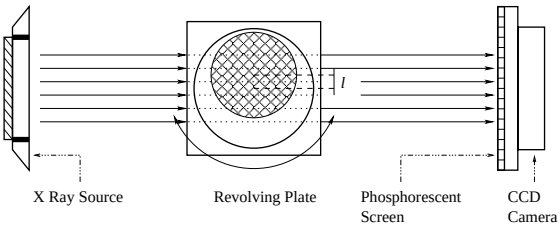


Figure 4: A projection does not show the complete object's information when the object is bigger that the width of the X ray beam.

To center the big object projections we assume not only that their projections are taken with a small increment in the tilt angle but also the displacement is only on the x axis, therefore the projection taken at 90 degrees must have a displacement of value zero, and that projection can be used as a reference to centering the others projections. Besides, we calculate, using cross-correlation, the relative displacement between consecutive projections (so they have a small displacement in the tilt angle). The increment in the tilt angle must be less than 5 degrees to get an error in the calculation of the displacement less than 1 pixel.

To automatically to know which is the reference projection from the displacement data, we need to apply the Eq. (3). In Eq. (3) x is the relative displacement between two projections, j is a index to the reference projection, d is the absolute displacement for the k projection.

$$d_k = \begin{cases} -\sum_{i=k}^{j-1} x_i & \text{if } k < j, \\ 0 & \text{if } k = j, \\ \sum_{i=j+1}^{k} x_i & \text{if } k > j, \end{cases} \tag{3}$$

In order to apply Eq. (3), we suppose that we have all projections in the same order as were acquired and we have calculated the x values, the displacement

between a projections with respect to the previous projection. The reference projection is the corresponding to the minimum value of function s in Eq. (4), that represent the sum of all absolute displacements for n projections and d_k calculated as Eq. (3).

$$s_i = \sum_{k=0}^{n-1} d_k \tag{4}$$

2.4 Segmentation

The reconstruction stage produces a three-dimensional matrix with a density value for each of its elements. To separate the different volumes with the same densities, we decide using the k-means algorithm as a clustering kernel, since it classifies n objects into k disjointed groups, based in some object's features. Thus, we use the cluster-minimal distance between a density value and a group's center. Note that we need only separate the different densities, or to separate the objects composed by the different densities; to separate the objects by their structural components it is necessary to use a clustering algorithm that takes into account the spatial coordinate (x, y, z) for each voxel.

The k-means algorithm perform the following steps:

- Step 1: Initialize the group's centers
- Step 2: Each element is assigned to a group according to the minimal distance between a group and the element.
- Step 3: For each group the center is recalculated with the new assigned elements.
- Step 4: While there are changes in the group's elements, go the step 2.

In the developed program, the number of clusters and the groups centers must be initialized by hand from the density-histogram information. The number of clusters is equal to the number of peaks. Each group's center is defined by two threshold values around the histogram peaks. The segmentation program produces binary volumes as number of groups were specified.

3 Surface Visualization Using Splatting

Splatting is a novel technique for rendering a volume based in points, it is used to render surfaces or volumes in its surface representation, that is, a 3D object is represented as a collection of samples, which they are lying over its surface [6,7]. Each sample point has position, normal, color and some others parameters, which allow to know the distance among sample points and their neighbors. The projection of a point (and its contribution around) is known as the rendering primitive *splat*. In order to represent the splats we used triangles, squares, and circles (actually a circle is a polygon of twelve sides). Each one of them seems

a small planar patch, which are assigned to a point oriented along the object's surface.

The aim of our software component is to visualize the surface of the reconstructed volumes using the splatting technique. So as to perform this part the following steps were made:

- Step 1: Extract the surface for each segmented volume using mathematical morphology. The center of each voxel will be the coordinates of the point and this point will be used to represent the surface.
- Step 2: Calculate the normal to each point over the surface.
- Step 3: Associate a splat to each surface point and set its position according with its normal.

By subtracting an eroded volume (it built with a structural element which consists of a voxel and its six neighbors), to the original volume we performed step 1, the extraction of the voxels in the binary volume surface; so that we developed an program which receives a binary volume and produces another binary volume only with the surface voxels set to 1.

The step 2 is solved in very simple way: A plane is fitted to a point and its neighborhood, then the point's normal is obtained. The plane's equation is given by $Ax + By + Cz + D = 0$, in addition if we divide this equation by D, we will obtain $A'x + B'y + C'z = -1$, that is, we only need three points for obtaining the values of the three unknowns, however we could have 27 points in the neighborhood; therefore we decide to use the SVD method (Singular Value Decomposition) to solve the overdetermined system, moreover by using SVD we assure that we have gotten the best plane fitted in the least squares sense. Once we calculated the plane we know that the point's normal is (A, B, C). Now we need the correct normal direction, so we checked the values of the two neighbors voxels in the normal's position of the original volume; the correct direction is where the neighbor voxel has the value of 0 (outside the volume no density exists).

Finally, to perform the step 3, a splat is represented using OpenGL-primitives; for that reason we designed and programed an graphical user interface (GUI) made in Qt [8] and OpenGL [9] to visualize the volumes. Qt is a library to rapid prototyping developed in C++, it has an excellent documentation, and it allows to assign an OpenGL-widget to the GUI. As input to the GUI we have a file with the format: a number in a single row that represents the maximum distance between two voxels, 6 numbers per row of the three coordinates values to every point and three of their normal's values. Coordinates-values must be normalized between -0.5 and 0.5. The splat-rendering primitive was performed when the visualization is made, that is, a splat is assigned to every point, located according to its normal, and with a size of the maximum distance of two voxels in order to avoid holes in the visualization. Furthermore, we represented splats as triangles, squares, and circles, besides it allows the visualization of points as a quick visualization help.

3.1 Results

The visualization of the trial-phantom and its ART-reconstruction are presented respectively in Fig. 5 and Fig.6. The first row shows the visualization with points; we can watch splat's representations with triangles, squares, and circles corresponding to the second, third, and fourth row. In spite of rendering with triangles should be the cheapest, in the sense of computational resources, we weren't able to notice that, since we tested a relatively small phantom; in other words, we didn't felt a loss of performance with any splat primitive. However, the best splat is represented with circles, because we can see holes on images on second and third row, which they correspond to the triangle and square splat implementations.

The reconstruction's visualization is good as we can see in Fig. 6, also the phantom's size was lightly increased since we performed a morphological closing operation in order cover holes in the reconstructed surface.

We also made a comparison of the phantom's visualization between splatting, simple voxels (using Scubes program [10]) and a deformed simplex mesh [11]. As we can see in Fig. 7 the visualization with simple voxels is not good because the normals are calculated to every voxel face. The visualization with the deformable simplex mesh is not good, because it is unable to represent an object with genus grater than 0 (with one o more holes), in addition is not a simple task to assign initial values to the mesh and the deformation's program is not easy to use. The visualization with splats not only is easy to use and fast to calculate but also produces the best quality; however it is necessary for splatting that the surface is covered with a enough number of points to avoid holes in the visualization.

4 Conclusions

We developed the software for working with a x-ray tomograph. We proposed a software made of five components: acquisition, projection's centering, 3D reconstruction, isosurface's segmentation and visualization. We designed and built four of the five parts. The 3D reconstruction component was adapted from Xmipp, we used weighted backprojection and ART algorithms of this collection. In our test, ART algorithm is better than weighted backprojection: the former produces not only a better quality but also a better dynamic amplitude. We solved by software the problem of projection's centering.

For the segmentation part, we use the k-means algorithm so as to cluster the density values of the voxels around k densities. The number of densities and the initial values for the groups centers are given by hand from the 3D-reconstruction's densities-histogram.

To visualize the isosurfaces on the reconstructed volumes we designed a simple algorithm to calculate the surface's normals: a plane is fitted using SVD to the neighborhood to each surface voxel. We used the voxel's center as the point that sampling the surface. In addition, we used OpenGL primitives to represent

a splat, we tested the primitives of triangles, squares, and circles. The best visualization in provided by the circles (actually a circle is a polygon of twelve sides). We show that splatting is easy to use give us the best quality visualization.

We think that all the programs developed must be tunned according to a specific application, therefore they must be changed to adjust the visualization to an specific reconstruction application task.

References

1. Gabor T. Herman. *Image Reconstruction From Projections.* ACADEMIC PRESS,INC, Orlando, Florida 32887, 1980.
2. John C. Russ. *The Image Processing Handbook*, chapter 9, 10. CRC PRESS & IEEE PRESS, 3. edition, 1999.
3. R. Marabini, I.M. Masegosa, M. C. San Martin, S. Marco, J.J. Fernandez, L.G. de la Fraga, C. Vaquerizo, and J. M. Carazo. Xmipp: An image processing package for electron microscopy. *Journal of Structural Biology*, pages 237–240, 1996.
4. Edwin L. Dove. Notes on computerized tomography. In *Bioimaging Fundamentals.* Dove Â Physics of Medical Imaging, 2001.
5. Image projections and the radon transform. http://www.owlnet.rice.edu/ elec539/Projects97/cult/node4.html.
6. Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 371–378. ACM Press / ACM SIGGRAPH, 2001.
7. Christopher S. Co, Bernd Hamann, and Kenneth I. Joy. Iso-splatting: A point-based alternative to isosurface visualization. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications.* Computer Society, 2003.
8. Qt. http://doc.trolltech.com/3.0/index.html.
9. Opengl. http://www.opengl.org/.
10. Luis Gerardo de la Fraga and Feliú Sagols Troncoso. Scubes: A program to visualize vox-solids. In *VII Conferencia de Ingeniería Eléctrica.* CINVESTAV, 2001.
11. Jorge Eduardo Ramírez Flores and Luis Gerardo de la Fraga. Basic three-dimensional objects constructed with simplex meshes. In *Electrical and Electronics Engineering, 2004. (ICEEE). 1st International Conference.* CINVESTAV, 2004.

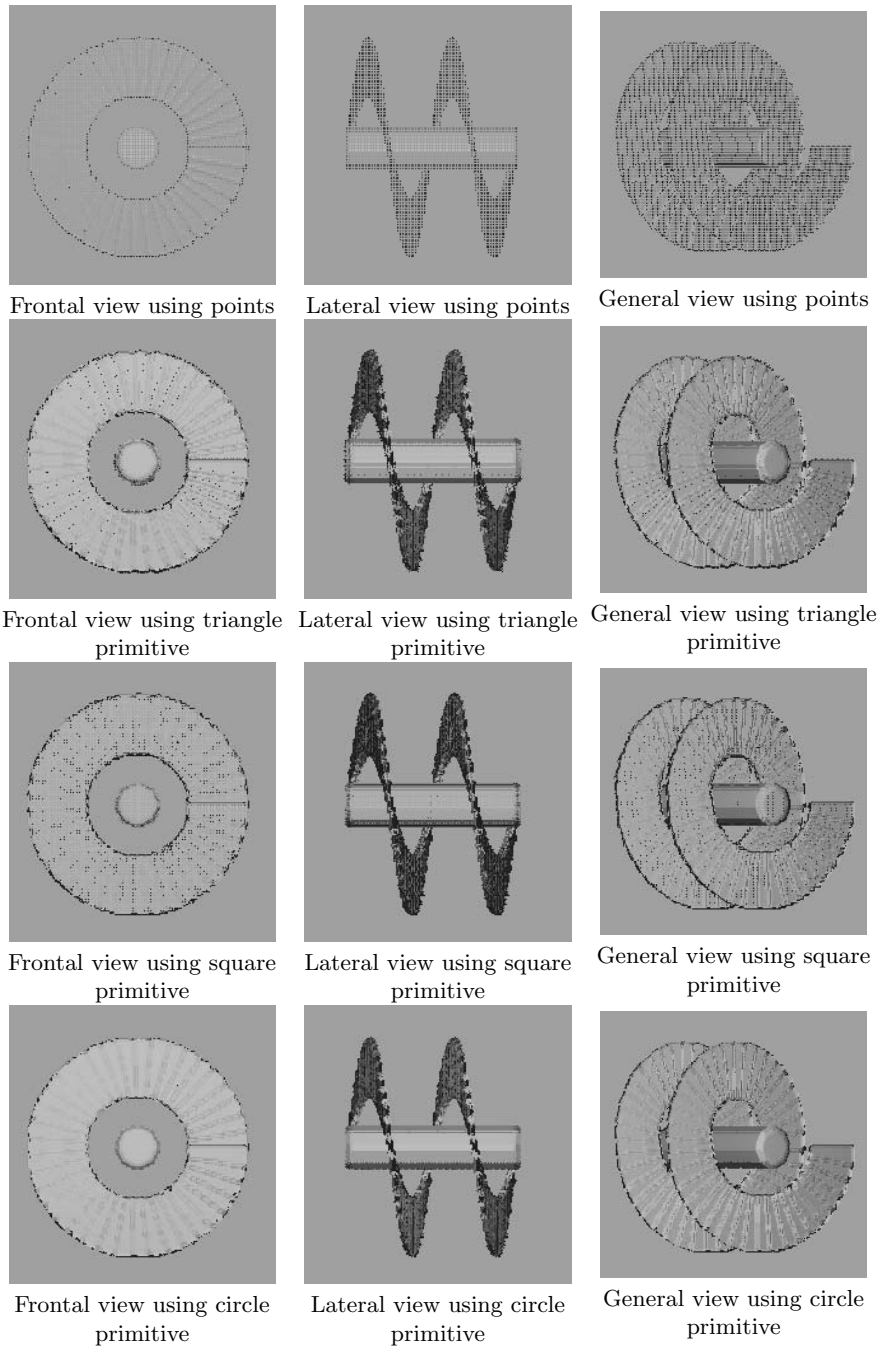
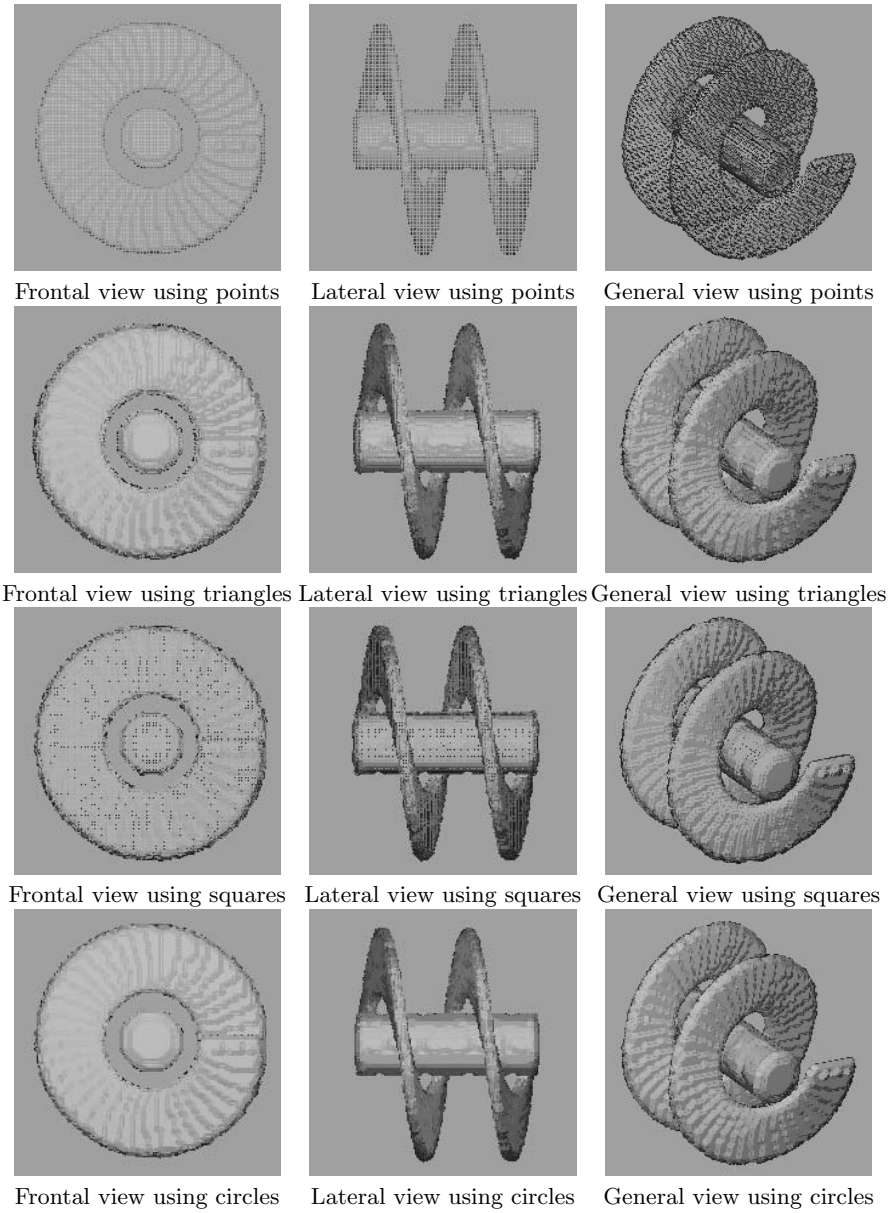


Figure 5: Comparison of the visualization with points and three primitives, triangle, square and circle, to render the splats.



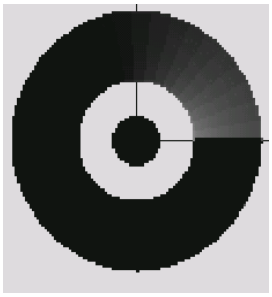
Frontal view using points | Lateral view using points | General view using points

Frontal view using triangles | Lateral view using triangles | General view using triangles

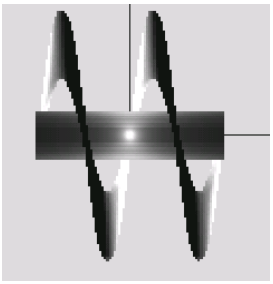
Frontal view using squares | Lateral view using squares | General view using squares

Frontal view using circles | Lateral view using circles | General view using circles

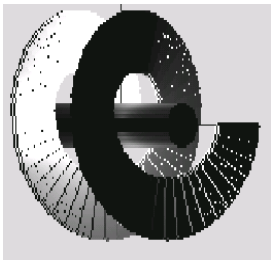
Figure 6: Visualization using different splats of the reconstructed phantom.



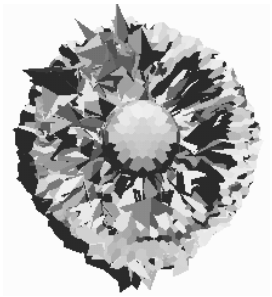
Frontal view using simple voxels



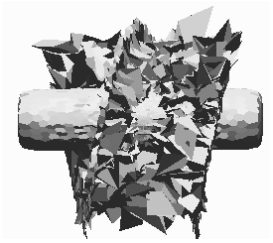
Lateral view using simple voxels



General view using simple voxels



Frontal view using a deformable simplex mesh



Lateral view using a deformable simplex mesh

General view using a deformable simplex mesh

Figure 7: Comparison of the visualization with other two techniques: using simple voxels and a simplex mesh deformed to cover the phantom surface.

Software Engineering

Evaluating the Evolution of Small Scale Open Source Software Systems

Chanchal Kumar Roy and James R. Cordy

Queen's University, Kingston, Ontario, Canada K7L3N6
{croy, cordy}@cs.queensu.ca
http://www.cs.queensu.ca/~cordy

Abstract. For real-world software to remain satisfactory to its stakeholders requires its continual enhancement and adaptation. Acceptance of this phenomenon, termed software evolution, as intrinsic to real world software has led to an increasing interest in disciplined and systematic planning, management and improvement of the evolution process. Almost all of the previous work on software evolution has been concerned with the evolution of large scale real-world software systems developed within a single company using traditional management techniques, or with the large scale open source software systems (LSOSSS). However, there is to our knowledge little or no work that has considered small scale open source software systems (SSOSSS). This paper presents an analysis of the evolution behavior of two small size open source software systems, the *Barcode Library* and *Zlib*. Surprisingly, unlike large scale open source software systems, the evolution behavior of these small size open source software systems appears to follow Lehman's laws for software evolution.

Key Words: Small Scale Open Source Software Systems, Case Study, Software Evolution, Metrics, Lehman's Law of Software Evolution, Software Maintenance.

1 Introduction

The software maintenance phase of the software life cycle is often underestimated, although it consumes remarkable resources. During the 1970s, maintenance was estimated to account for thirty-five to forty percent of the software budget for an information systems organization. By the 1980s, estimates of the expenditure had reached sixty percent of a software system, from initial vision to abandonment [8].

Software systems evolve during the software life cycle due to product improvement and additional development to include new features. This growth is accompanied by increasing complexity of the software. At a certain point, every new release introduces new complexity and renders development more difficult. As this point is reached it becomes necessary to analyze the structure of the system to identify potential shortcomings which may be candidates for restructuring and as a consequence, the term *Software Evolution* is introduced. *Software*

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 123-136

Received 25/07/06
Accepted 04/10/06
Final version 27/10/06

evolution, i.e. the process by which programs change shape, adapt to the marketplace and inherit characteristics from preexisting programs, has become a subject of serious academic study in recent years. Partial thanks for this goes to Lehman and other pioneering researchers. Major thanks, however, goes to the increasing strategic value of software itself [16].

The larger a software product grows, the more important the ability of the architecture of the system to support evolvability becomes. Focusing on the analysis of software evolution has revealed a high potential for improvement of the software development process. Software evolution is more important as systems become longer lived. However, evolution is challenging to study due to the longitudinal nature of the phenomenon in addition to the usual difficulties in collecting empirical data [8].

Often the real structure of software system is not fully known to engineers. Therefore, it is difficult to reason about any possible deficiencies and how they may be corrected. Thus, it is critical to capture software architecture for the maintainability of a software system. After such an analysis, restructuring can be applied, which may help to improve the discovered situation. In the last few years numerous methods for the analysis of software architecture have been developed. Different sources of information, for example release information, design documents, and source code, are used to draw conclusions about the structure of software systems. Many approaches base their analyses on the micro level of program source code. For very large systems these approaches can reach a level where it is difficult to make any reliable statements about the quality of the software. Instead, methods that focus on the macro level (for example number of modules/subsystems, number of classes, and so one) must be applied to analyze these large systems.

A study of the literature reveals that most of the previous work on software evolution has looked primarily at large real-time software systems [11] or large open source software [13], [14], such as the Linux kernel. Surprisingly, little or no work has been concerned with the evolution of smaller size software systems, which leads us to think about observing the evolution behavior of small size open source software systems. This paper presents a study of the evolution behavior of two such small size open source software systems, the *Barcode Library* and *Zlib*. We consider the evolution for both the systems, with particular emphasis on the *Barcode Library*. We have applied both micro and macro level analysis to these systems. However, as our target systems are very small in size, we particularly focus on the micro level, measuring evolution of the uncommented non-blank line of codes (UNBLOC or ULOC for short).

The rest of this paper is organized as follows. In Section 2, some significant related work is reviewed. In Section 3, a brief description of the target open source software systems is provided. Section 4 presents the methodology and measuring metrics used in this work. In Section 5, a detailed case study observing the evolution behavior of the target software systems is presented, and finally Section 6 summarizes our observations and conclusions.

2 Related Work

An extensive study on software evolution has been carried out by Lehman et al. [9], [10], [11], [12], over the last 35 years. Based on their experience we have several laws of software evolution, known as Lehman's Laws of Software Evolution [11], [12]. The work by Godfrey et al. [7] has shown that Linux continuously exhibits a global super-linear growth pattern, and recently Robles et al. [14] conclude that Lehman's laws, especially the 4^{th} law [11] is not well fitted to large size open source software systems. However, a more detailed empirical study [13] does not support the hypothesis that open-source development fosters faster system growth than closed-source development as opposed to [7], [14]. In [6], the authors examined the structure of several releases of a Telecommunications Switching System (TSS) stored in a database of product releases and found that there is a significant difference in the behavior of the whole system versus its subsystems. In [2], Burd et al. have evaluated the evolution of the GCC compiler and found interesting results. Capiluppi et al. have also authored several works on the evolution of open source software projects [4] and have proposed some models [3] working with mid-size projects. Succi et al. [17] also observed super-linearity in the evolution of the Linux kernel, but found linear growth for both GCC and Apache.

3 Target Open Source Software: *Barcode Library* and *Zlib*

In our work we have studied two useful small scale open source software systems, the *Barcode Library* and *Zlib* Tool. The Barcode package [1] is mainly a C library for creating bar-code output files. It includes both command line and graphical front-ends. The package is designed as a library because the main use for barcode-generation tools is embedding in other applications. The library addresses bar code printing as two distinct problems: creation of bar information and actual conversion to an output format. Based on the functionality, the *Barcode Library* is divided into five modules: the *Front-End Module*, the *Output Module*, the *Basic Encoding Module*, the *Advanced Encoding Module* and the *Header File Module*. More detail about Barcode can be found in [1] and in the relevant documentation of the different versions.

Zlib [15] is a well known compression tool which is also designed to be a free, general-purpose, legally unencumbered (i.e., not covered by any patents), lossless data-compression library for use with virtually any computer hardware and operating system. The *Zlib* data format is itself portable across platforms. In this work we study both of these tools but focus particularly on the evolution of the *Barcode package*.

4 Methodology

We have collected 9 different releases of the *Barcode Library* and 43 different releases of *Zlib* from their home pages. We have calculated the total size, total

lines of code (LOC), and number of uncommented non-blank lines of code (UNB-LOC or ULOC for short) for each of the releases of both the software systems. For the case of *Barcode Library*, we have also done the same for the five modules mentioned in Section 3 above. The open source program Numlines [18] is used to assist in these calculations. For each of the releases of the *Barcode Library* and *Zlib* we have also calculated the number of global functions, variables and macros using Ctags [5]. For drawing the least-squares fit of the plotted data a small java tool is developed and called the Least-Squares Fitter (LSF).

There are several kinds of metrics being used by different researchers. In this work we have used the following metrics for observing the evolution of *Barcode Library* and *Zlib*. In the following, we use the usual notation Release$_i$ for the i^{th} release, Growing Rate$_i$ for the *growing rate* of the i^{th} release, and so on.

Size$_i$ = Size of Release$_i$ in bytes i.e., total size of ith release in bytes
UNB-LOC$_i$ = Number of uncommented non-blank LOC in Release$_i$
Added$_i$ = Number of UNB-LOC of the newly added files in Release$_i$
Changed$_i$ = Number of UNB-LOC of the changed/modified files in Release$_i$
Growing Rate$_i$ =(Added$_i$ *X* 100) / (UNB-LOC$_{i-1}$)
Changing Rate$_i$ =(Changed$_i$ *X* 100) / (UNB-LOC$_{i-1}$)
Handled$_i$ = (Changed$_i$ + Added$_i$) - (Changed$_{i-1}$ + Added$_{i-1}$)
UnHandled$_i$ = (UNB-LOC$_i$)- (Handled$_i$)
Handling Rate$_i$ =(Handled$_i$ *X* 100)/ (UNB-LOC$_i$)

Once the data was collected for each of the releases of *Barcode* and *Zlib* libraries, we plotted the data over time, for *Zlib* against the release serial number (RSN) as suggested by Lehman, and for the *Barcode Library* against calendar time (number of months/days since first release) following the style of several other researchers. We then considered whether our plotted data appears to follow Lehman's laws of software evolution [11] or not.

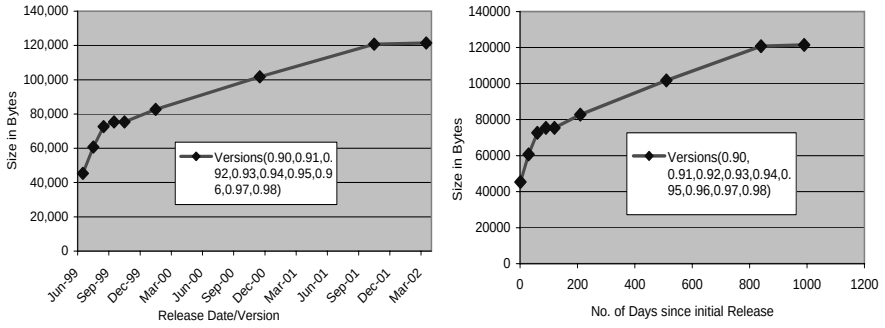
5 Observations on the Evolution of *Barcode Library* and *Zlib*

During the analysis of the software systems we have made some interesting observations. In the following subsection we concentrate on the global systems and their evolution first. Then we zoom in on the subsystems. The outliers that may be recognized in the subparts are discussed in more detail. The evaluation covered releases of the *Barcode Library* from June 1999 until March 2002, which contains 9 releases over the period of thirty three months. For *Zlib*, we considered 43 releases covering a period of about 11 years.

5.1 Overall Growth w.r.t. Size

To begin, Fig. 1(a), shows the overall growth of *Barcode Library* beginning from the first version released in June 1999. We can see that over time the size of the system has increased to meet additional user requirements. We have manually

analyzed the *Barcode Library* documentation for all of the versions and found that in most of cases the growth was required to add new functionally to meet user requirements. Thus later versions have more functional capability, covering more user needs. In the early days of the *Barcode Library* there were a number of releases with fewer new functionalities added, and therefore we can observe (Fig. 1(a)) a period of several versions with comparatively little growth in size compared to the first three. However, we see that the growth follows a monotonic process and therefore we can say that the growth of *Barcode Library* does follow Lehman's 6^{th} Law of continuing growth of system size.



(a) Over Months (b) Over Days

Fig. 1. Growth of Barcode Over Time (months + days) w.r.t Size of the Released Versions

This is even clearer in Fig. 1(b), where we have plotted the size data against the number of days since the first release. We have used our LSF tool to compute a least squares fit of the data and found that the linear equation $Y = 65.92X + 63138.28$ where X is the number of days, is a good predictor of Barcode system size over time. This linear growth is in contrast to the super-linear growth of Linux, reported in [7], [14], and can be termed continuing growth, Lehman's 6^{th} law of evolution.

In Fig. 2, we have created a similar plot for *Zlib*, but in this case we have plotted the data against the release serial number (RSN) as suggested by Lehman rather than time. Here, we also find that system growth continues over time. We have analyzed the changed log histories of the different versions and also found that, like Barcode, the growth seems to be primarily due to the addition of more functionalities to meet additional user requirements.

As we see, even when plotted against RSN, growth of this system also seems to be liner in nature and can be fitted to the linear equation $Y = 6568.53X + 175518.76$ with correlation coefficient 0.9668. Therefore, we can say that the growth of *Zlib* also follows the Lehman's 6^{th} law of continuing growth. However,

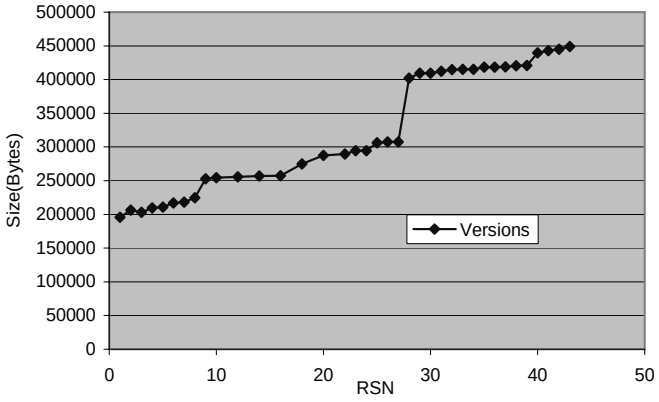


Fig. 2. Growth of *Zlib* Over RSN w.r.t Size of the Released Version

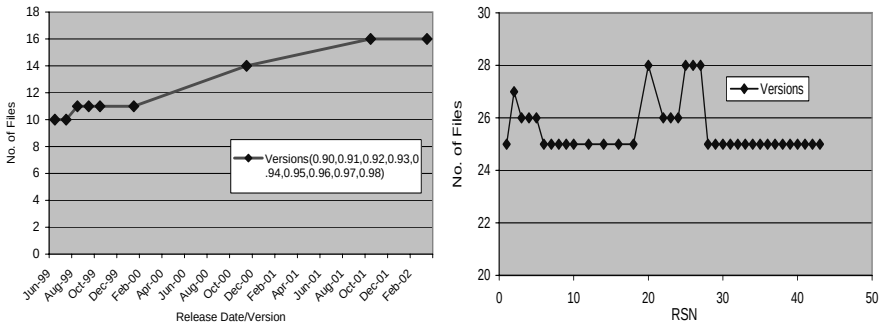
we can also see that although growth is continuing, occasionally it is very high compared to other releases, as we also observed in the case of the *Barcode Library*. In both cases we found similar growth with respect to size of the development releases, and both seem to follow Lehman's 6^{th} law of evolution.

5.2 Growth w.r.t. Number of Files

Measuring growth with respect to the size of released versions might be questionable in the sense that a system could grow in size without adding new functionalities, for example due to addition of unnecessary lines of code (e.g., cloned or redudant code added by a poor coder) or a large number of new comments or blank lines. For this reason recent researchers often use other kinds of metrics, such as number of modules, number of subsystems, number of classes and so on, to measure object-oriented source code. As our target systems are written in C and are quite small in size, we have considered number of files as another of our metrics.

In Fig. 3(a), the number of files in the releases of the *Barcode Library* have been plotted against calendar time. For this system it seems that there has been little significant growth in the number of files, and only 6 files have been added over the entire period of evolution from the initial version to the latest version

In Fig. 3(b), we have found more interesting results for *Zlib*. In this case we see that the number of files remains relatively constant over all releases, except for some earlier releases where the size of the system has actually been decreased over time. In the case of Barcode (Fig. 3(a)), we could say that system is growing continuously in number of files over time, following the 6^{th} law of Lehman, but the same is definitely not true in the case of *Zlib* (Fig. 3(b)) where it is clear that number of files remains constant over most of the latest releases. This is very much contrasted with the literature, especially for large open source



(a) Barcode (b) Zlib

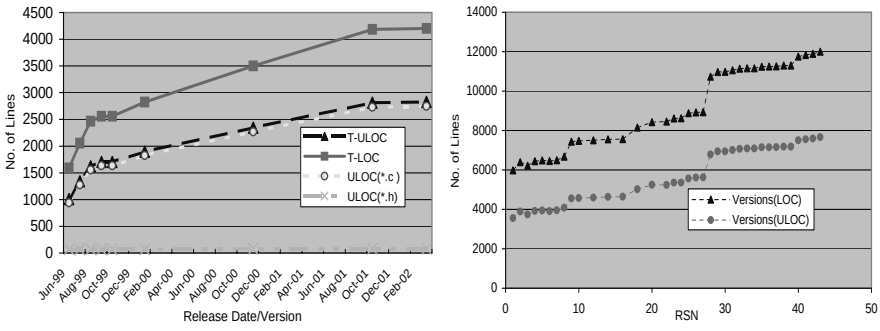
Fig. 3. Growth Over Time (Months) w.r.t the Number of Files

systems, where the reported growth has even been super-linear with respect to the number of modules/subsystems [7], [14]. We therefore decided to consider other kinds of metrics as well.

5.3 Growth w.r.t. LOC and UNB-LOC

Considering the unusual (lack of) growth in *Zlib* with respect to the number of files as shown in Fig. 3(b), we decided that perhaps simply counting the number of files might not be a good metric for evolution of these systems. When target systems are very small, perhaps in most cases the number of files remains more or less the same over the evolution period. We have considered the number of lines over system releases. As the total number of lines may also contain commented lines or blank lines, we decided to try observing evolution by considering only the number of uncommented, non-blank lines of code (UNB-LOC). UNB-LOC is an old technique for measuring evolution, but we have found that for small size software systems, especially those are written in non-object-oriented context, considering the UNB-LOC may be a better or perhaps even the only appropriate metric for evolutionary growth.

In Fig. 4(a), we have plotted the total lines of code (T-LOC) and total UNB-LOC (T-ULOC) for the *Barcode Library*. We have also shown the total UNB-LOC for the *.c files and *.h files separately. As we see, there is a linear growth over time except in the case of the *.h (header) files. Our main concern is the T-ULOC of the releases and we see that the growth of this curve is a good fit to the linear equation $Y = 44.55X + 1115$. This leaves little doubt that this measure of growth of releases of this system seems to follow Lehman's 6^{th} law of evolution The LOC and UNB-LOC (ULOC for short) for *Zlib* have also been plotted in Fig. 4(b) against RSN and we can observe the same linear nature of continuing growth. In the case of the UNB-LOC curve, the linear equation



(a) Barcode (b) Zlib

Fig. 4. Growth Over Time (months) w.r.t. T-LOC and UNB-LOC

$Y = 103.41X + 3356.25$ is found to fit with correlation coefficient 0.977, definitely consistent with Lehman's 6^{th} law of evolution.

5.4 Handled and Unhandled UNB-LOC for the *Barcode Library*

Until now we have discussed only the 6^{th} law of evolution. However, what fraction of the system is handled or unhandled is also a major factor in software evolution and is directly related to some of the Lehman's laws. In this section we have studied what fraction of the system is handled or remains unhandled over time, in order to estimate the *changing rate* and *growing rate* of the system.

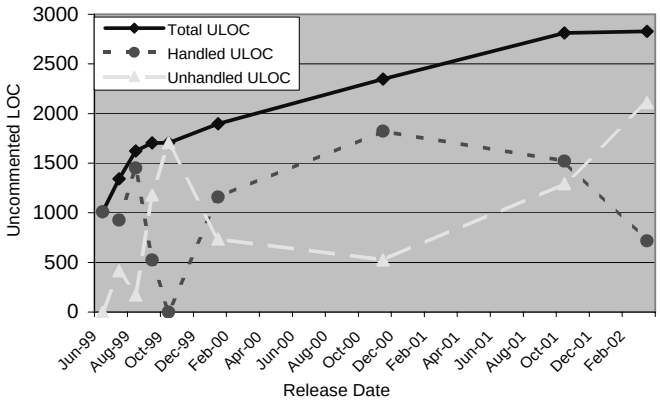


Fig. 5. Handled and Unhandled UNB-LOC Over Time (months) for Barcode.

In Fig. 5, we have plotted the ULOC, the handled ULOC, and *unhandled ULOC* for the total system. We have found interesting results for the case of *handled ULOC*. We see that except for some early releases, the *handled ULOC* consistently increased for a time and since then has decreased significantly. We can also see the same effect in the corresponding *unhandled ULOC* curve. The *handled ULOC* and *unhandled ULOC* curves can be fitted with the linear equations $Y = 11.58X + 892.20$ and $Y = 36.48X + 519.19$ respectively.

One interesting observation to note here is that there is no change from version 0.93 to version 0.94 (during the time period October 1999 to December 1999), as we see that the *unhandled ULOC* curve touches the total ULOC curve and *handled ULOC* curve is at the X-axis. It would be interesting to find out how and why version 0.94 evolved from version 0.93.

Next, in Fig. 6 we have plotted the number of files that are changed and newly added over time (although we know number of files is not a good metric for such small size software systems).

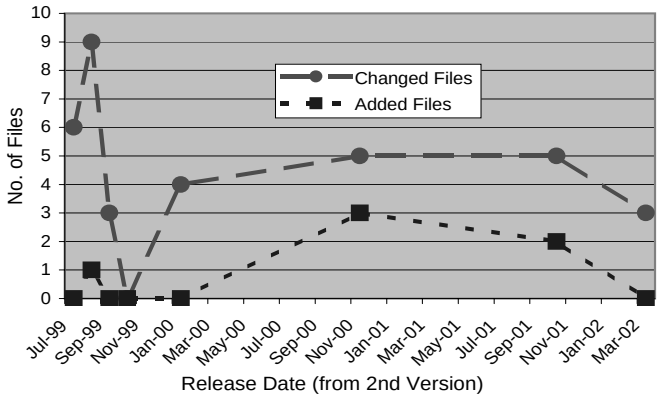


Fig. 6. Changed/Added Files Over Time(months) of Barcode.

We see that very few files are added over time and this adding phenomenon is also decreasing in the latter versions. However, initially the number of changed files is more than in later versions, and the number is also decreasing, following some properties of Lehman's 1^{st} and 5^{th} laws. We know Lehman's 1^{st} law predicts continuing change and the 5^{th} law concerns conservation of familiarity. To focus on these two laws, we have plotted the *changing rate* and *growing rate* of *Barcode* in Fig. 7 by taking into account the *handled ULOC*. In this Fig. 7, we see that the *Barcode Library* system is changing continuously over time, with some exceptions in the earlier versions, especially in the case of version 0.94 from version 0.93. Therefore, we can say that this changing nature of *Barcode Library* follows the Lehman's 1^{st} law. On the other hand, both the *changing*

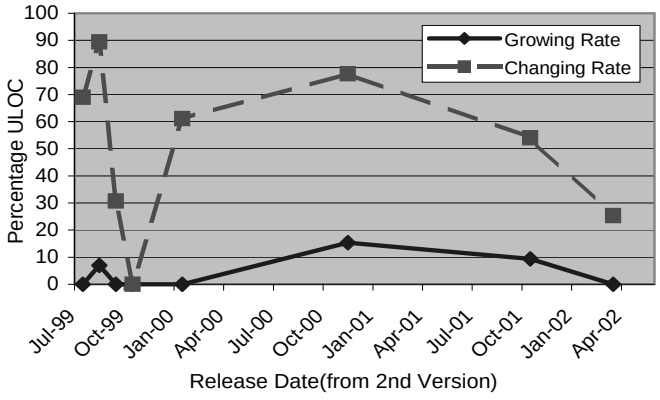


Fig. 7. Growing Rate and Changing Rate of Barcode Over Time (months).

rate and *growing rate* tend to decline over time which lead us to the 5^{th} law of evolution, the *Conservation of Familiarity*. However, as we mentioned earlier, we really could not predict what actually added/modified to version 0.94 from its previous version 0.93 and therefore, in the next subsection, we go into more depth on the *Barcode Library* by observing its evolution at the module level.

5.5 Module Level Observations of the *Barcode Library*

As we know from Section 3, five modules are considered in the *Barcode Library* corresponding to parts of the two main purposes of the library. To observe the evolution of the module level, we have plotted the UNB-LOC for each of the five modules of the system over time in Fig. 8. We see that except for the *Header File module*, all others are increasing in the number of UNB-LOC over time, following Lehman's 6^{th} law of continuing growth and signaling the 1^{st} law of continuing change. We have found that the most promising module is the *Basic Encoding Module* which is increasing more rapidly over time than any other module with respect to both size and rate of growth. We have also noticed that except for the *Basic Encoding Module*, all other modules tend to decline in incremental growth which brings us in mind of Lehman's 5^{th} law of evolution on the module level also. We can therefore say that with few exceptions, the evolution of the Barcode Library modules follows Lehman's 1^{st}, 5^{th} and 6^{th} laws of evolution well.

To be sure about our predictions, in Fig. 9 we have plotted the percentage UNB-LOC of the modules with respect to the total system over time. Here we see that except for the *Advanced Encoding Module*, all other modules either change in parallel or tend to decline in incremental size with respect to the percentage of the total system. A parallel curve, like the later releases of the *Basic Encoding Module*, implies that this module is growing at the same rate as the total system. However, we can also see that except for the *Advanced Encoding*

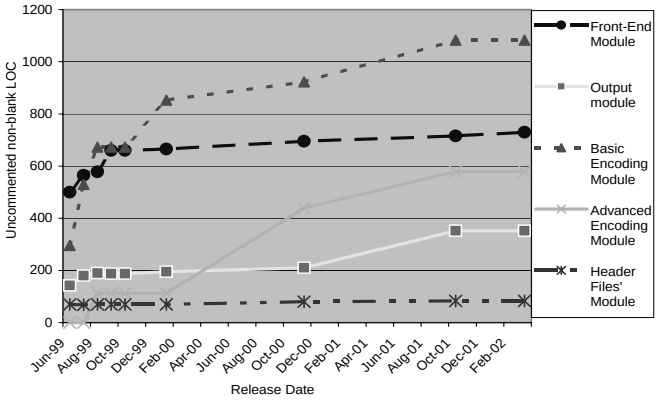


Fig. 8. UNB-LOC of the Modules of Barcode Over Time.

Module, all others are decreasing in their rate with respect to the whole system, demonstrating the predictions of Lehman's 5^{th} law.

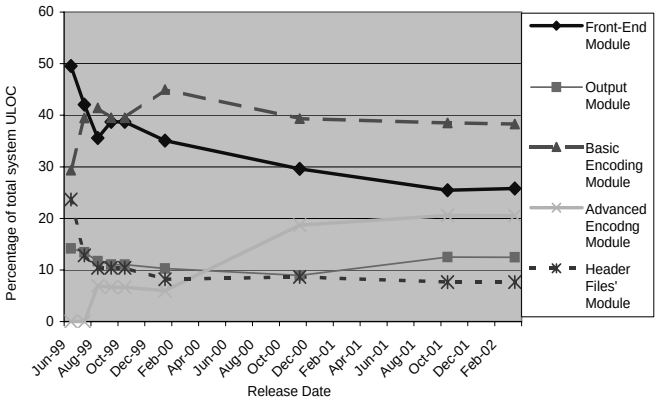


Fig. 9. Percentage of ULOC of the Modules w.r.t the Total System Over Time.;

We mentioned earlier the problem with version 0.94, where we could not figure out how or what had actually been changed or added in version 0.94 from version 0.93. We have assumed that something might happened in the *Basic Encoding Module* as this seems the most promising module among the others. Therefore, we have done a more in-depth focus on this module by plotting its individual files in Fig. 10. As we can see from this figure, files *code39.c* and *code128.c* are growing over time, while *ean.c* is not. We can also see that

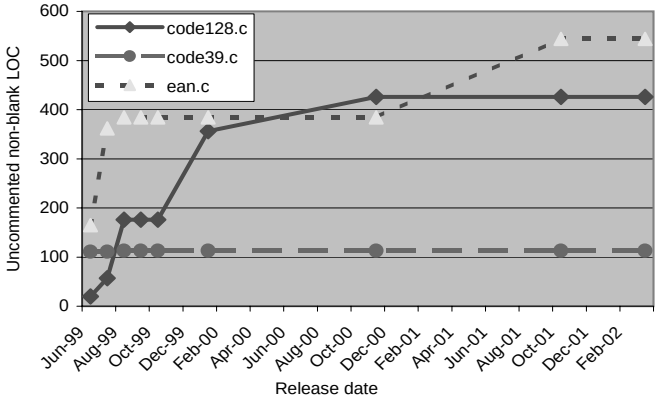


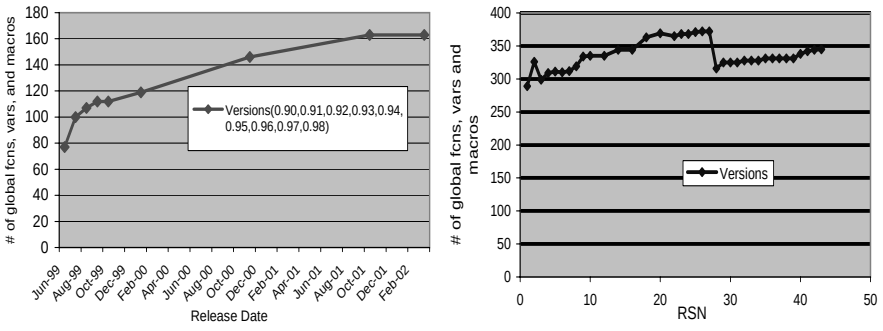
Fig. 10. File-Level Growth of the *Basic Encoding Module* Over Time of Barcode

code128.c has a smoother growth than the others, which helps us imagine that *code128.c* might be one of the main files of this module, or even of the whole system (since *Basic Encoding Module* is the most growing module and *code128.c* is in this module). Perhaps it changes over versions at a smooth rate and most of the system changes are being made in this file.

However, we still cannot predict clearly how and why version 0.94 evolved from version 0.93 as yet, even with these file level observations. With our strong belief that evolution might be associated with *code128.c* (as most promising file, in most growing module), we have considered the size of this file over versions 0.93 to 0.94 and found that there are only 2 bytes of difference between these two files. Even more interesting, there are only 2 bytes of overall size difference between versions 0.93 and 0.94. For our further interest, we have used the Linux command *diff* to see the real differences between *code128.c* of version 0.93 and that of version 0.94, and we have found that a few lines of code have been replaced with the same number of lines but with more complicated code in the same lines of *code128.c* in version 0.94 from version 0.93. We have also found that some comments are added with the same lines.

5.6 Observing the Complexity

Complexity of a system is always a major concern. Lehman also has the 2^{nd} law of evolution, relating to the complexity of the system. The 2^{nd} law states that as a system evolves the complexity of the system increases unless work is done to maintain it. As we know, since the C language has a single flat namespace, the complexity of a C program depends largely on the number of global functions, variables and macros (unlike C++ or Java). We have counted the total number of global functions, variables and macros for each of the releases of Barcode and *Zlib* using Ctags [5]. We have plotted this for *Barcode Library* over time in Fig. 11(a) and that of *Zlib* against RSN in Fig. 11(b)).



(a) Barcode (b) Zlib

Fig. 11. Number of Global Functions, Variables and Macros Over Time of Releases

From Fig. 11(a), it can be seen that as Barcode has evolved, the complexity of the system has increased, following the Lehman's 2^{nd} law of evolution. We have also found a good fit to the linear equation $Y = 2.22X + 98.62$ using LSF, which indicates that the complexity of the system has been increased with roughly linear growth over the months since its first release in June, 1999.

We can also observe more or less same behavior in the case of *Zlib* in Fig. 11(b). Here it can be seen that initially the complexity of the system increases, consistent with the 2^{nd} law of evolution, and then later special work has been done to control the complexity. As we result, after the 21^{st} release, the complexity of the system decreased drastically, and then again increased gradually over the next releases, again following Lehman's 2^{nd} law of evolution.

6 Conclusion

In this work, the evolution behavior of two small scale open software systems has been observed. It has been found that the evolution of these small scale software systems is consistent with Lehman's laws of software evolution, unlike the reported evolution of larger open source software systems, which have been reported to show evolution behaviors inconsistent with Lehman's laws. In our study we have mainly focused on Lehman's 1^{st}, 2^{nd}, 5^{th} and 6^{th} laws of evolution and found that for the most part the evolution of both the *Barcode Library* and *Zlib* follows these laws with some minor exceptions.

While more research would be required to make any firm conclusions, these observations lead us to believe that perhaps small size open source software in general may follow Lehman's laws of evolution more consistently than do larger systems such as those reported by Robles et at. [14] and Godfrey et al. [7]. However, there are critical differences in our studies, including both that Robles and Godfrey have worked with very large scale open source software systems

developed by very large groups of people, and that they have concentrated on the 4^{th} law of evolution. Although we have not specifically focussed on the 4^{th} law, from the observations of this case study, it is however clear that neither *Barcode Library* nor *Zlib* has super-linear growth in their evolution and hence, it can be concluded that the evolution of both *Barcode Library* and *Zlib* does follow Lehman's 4^{th} law of evolution in some sense.

References

1. The Barcode: http://www.gnu.org/software/barcode/barcode.html
2. Burd, E., Munro, M.: Evaluating the evolution of a C application. Internation Workshop on Principles of Software Evolution(1999), pp. 401-410, Fukuoka, Japan.
3. Capiluppi, A.:Models for the evolution of os projects. Proceedings of the International Conference on Software Maintenance(2003), pp. 65-74, Amsterdam, The Netherlands.
4. Capiluppi, A., Morisio, M., Lago, P.: Evolution of understandability in oss projects. Proceedings of the 8th European Conference on Software Maintenance and Reengineering(2004),pp. 58-66, Tampere, Finland.
5. The Ctags: http://www.die.net/doc/linux/man/man1/ctags.1.html
6. Gall, H., Jazayeri, M., Klösch, R., Trausmuth, G.: Software evolution observations based on product release history. Proceedings of the International Conference on Software Maintenance(1997), pp. 160-169, Berlin, Germany.
7. Godfrey, M. W., Tu, Q.: Evolution in Open Source software: A case study. Proceedings of the International Conference on Software Maintenance (2000), pp. 131-142, San Jose, California.
8. Krajewski, J.: QCR - A Methodology for Software Evolution Analysis. Master Thesis(2003), Technical University of Vienna.
9. Lehman, M. M., Belady, L. A.: Program Evolution: Processes of Software Change, Academic Press, 1985.
10. Lehman, M. M., Perry, D. E., Ramil, J. F.: Implications of evolution metrics on software maintenance. Proceedings of the International Conference on Maintenance (1998), pp. 208-217, Bethesda, Maryland.
11. Lehman, M., Ramil, J., Wernick, P., Perry, D.: Metrics and laws of software evolution - the nineties view. Proceedings of the Fourth International Software Metrics Symposium(1997), Portland, Oregon.
12. Lehman, M. M., Ramil, J. F.: Rules and tools for software evolution planning and management. Annals of Software Engineering, **11**(2001):15-44.
13. Paulson, J. W., Succi, G., Eberlein, A.: An empirical study of open-source and closed-source software products. Transactions on Software Engineering, *30*(4): 246-256, April 2004.
14. Robles, G., Amor, J. J., Gonzalez-Barahona, J. M., Herraiz, I.: Evolution and Growth in Large Libre Software Projects. Proceedings of the International Workshop on Principles of Software Evolution(2005), pp.165-174, Lisbon, Portugal.
15. Roelofs, G., Gailly, J., Adler, M.: The Zlib home page. http://www.Zlib.net/
16. Salon: http://www.salon.com/tech/feature/
17. Succi, G., Paulson, J., Eberlein, A.: Preliminary results from an empirical study on the growth of open source and commercial software products. In EDSER-3 Workshop(2001), co-located with ICSE 2001, pp.14-15 Toronto, Canada.
18. The Numlines home page. http://www.gammadyne.com/cmdline.htm, 2005

Cryptography and Security

Genetic Algorithm Attack on Simplified Data Encryption Standard Algorithm

Poonam Garg

Institute of Management Technology
poonam@imt.edu

Abstract. With the exponential growth of networked & electronic systems, the demand of efficient and fool proof internet security is increasing. Security has emerged as a critical concern in wide range of electronic system. Cryptology is at the heart proving these securities. It consists of two complementary fields of study: cryptography and cryptanalysis. Cryptanalysis is one of the major challenging areas of intense research in the discipline of security. In this paper, we explored the use of genetic algorithm to break a simplified data encryption standard algorithm (SDES). To test its performance, we compared the implemented genetic algorithm attack with brute force search algorithm attack. Through extensive experiments and analysis it can be concluded that 1) genetic algorithms attack run ten times faster than brute force search algorithm attack with accuracy, and 2) genetic algorithms attack are 13% more accurate than brute force search algorithm attack, with same running time. A generalized version of cryptanalysis of SDES will give better insight into the attack of DES and other cipher.

Keywords: Simplified data encryption standard, Genetic Algorithm, Key search, brute force search algorithm

1 Introduction

The demand for effective internet security is increasing exponentially day by day. Businesses have an obligation to protect sensitive data from loss or theft. Such sensitive data can be potentially damaging if it is altered, destroyed, or if it falls into the wrong hands. So they need to develop a scheme that guarantees to protect the information from the attacker. Cryptology is at the heart of providing such guarantee. Cryptology is the science of building and analyzing different encryption and decryption methods. Cryptology consists of two subfields; cryptography & cryptanalysis. Cryptography is the science of building new powerful and efficient encryption and decryption methods. It deals with the techniques for conveying information securely. The basic aim of cryptography is to allow the intended recipients of a message to receive the message properly while preventing eavesdroppers from understanding the message. Cryptanalysis is the process of recovering the plaintext and/or key from a cipher. In other words cryptanalysis can be described as the process of searching for flaws or oversights in the design of ciphers.

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 139-149

Received 01/08/06
Accepted 03/10/06
Final version 17/10/06

This paper proposes the cryptanalysis of simplified encryption standard algorithm using genetic algorithm. The cryptanalysis of simplified data encryption standard can be formulated as NP-Hard combinatorial problem. Solving such problems requires effort (e.g., time and/or memory requirement) which increases with the size of the problem. Techniques for solving combinatorial problems fall into two broad groups – traditional optimization techniques (*exact* algorithms) and non traditional optimization techniques (*approximate* algorithms). A traditional optimization technique guarantees that the optimal solution to the problem will be found. The traditional optimization techniques like branch and bound, simplex method, brute force search algorithm etc methodology is very inefficient for solving combinatorial problem because of their prohibitive complexity (time and memory requirement). Non traditional optimization techniques are employed in an attempt to find an adequate solution to the problem. A non traditional optimization technique - Genetic algorithm, simulated annealing and tabu search were developed to provide a robust and efficient methodology for cryptanalysis. The aim of these techniques to find sufficient "good" solution efficiently with the characteristics of the problem, instead of the global optimum solution, and thus it also provides attractive alternative for the large scale applications. These nontraditional optimization techniques demonstrate good potential when applied in the field of cryptanalysis and few relevant studies have been recently reported.

In 1993 Spillman [12] for the first time presented a genetic algorithm approach for the cryptanalysis of substitution cipher using genetic algorithm. He has explored the possibility of random type search to discover the key (or key space) for a simple substitution cipher. In the same year Mathew [10] used an order based genetic algorithm for cryptanalysis of a transposition cipher. In 1993, Spillman [13], also successfully applied a genetic algorithm approach for the cryptanalysts of a knapsack cipher. It is based on the application of a directed random search algorithm called a genetic algorithm. It is shown that such a algorithm could be used to easily compromise even high density knapsack ciphers. In 1997 Kolodziejczyk [8] presented the application of genetic algorithm in cryptanalysis of knapsack cipher .In 1999 Yaseen [14] presented a genetic algorithm for the cryptanalysis of Chor-Rivest knapsack public key cryptosystem. In this paper he developed a genetic algorithm as a method for Cryptanalyzing the Chor-Rivest knapsack PKC. In 2003 Grundlingh [7] presented an attack on the simple cryptographic cipher using genetic algorithm. In 2005 Garg [2] has carried out interesting studies on the use of genetic algorithm & tabu search for the cryptanalysis of mono alphabetic substitution cipher. In 2006 Garg [3] applied an attack on transposition cipher using genetic algorithm, tabu Search & simulated annealing. In 2006 Garg [4] studied that the efficiency of genetic algorithm attack on knapsack cipher can be improved with variation of initial entry parameters.

The SDES [11] encryption algorithm takes an 8-bit block of plaintext and a 10-bit key as input and produces an 8-bit block of cipher text as output. The decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used as input to produce the original 8-bit block of plaintext. The encryption algorithm involves five functions; an initial permutation (IP), a complex function called f_K which involves both permutation and substitution operations and depends on a key input; a simple

permutation function that switches (SW) the two halves of the data; the function f_K again, and a permutation function that is the inverse of the initial permutation (IP^{-1}). The function f_K takes as input the data passing through the encryption algorithm and an 8-bit key. Consider a 10-bit key from which two 8-bit sub keys are generated. In this case, the key is first subjected to a permutation P10= [3 5 2 7 4 10 1 9 8 6], then a shift operation is performed. The numbers in the array represent the value of that bit in the original 10-bit key. The output of the shift operation then passes through a permutation function that produces an 8-bit output P8=[6 3 7 4 8 5 10 9] for the first sub key (K1). The output of the shift operation also feeds into another shift and another instance of P8 to produce subkey K2. In the second all bit strings, the leftmost position corresponds to the first bit. The block schematic of the SDES algorithm is shown in Figure 1.

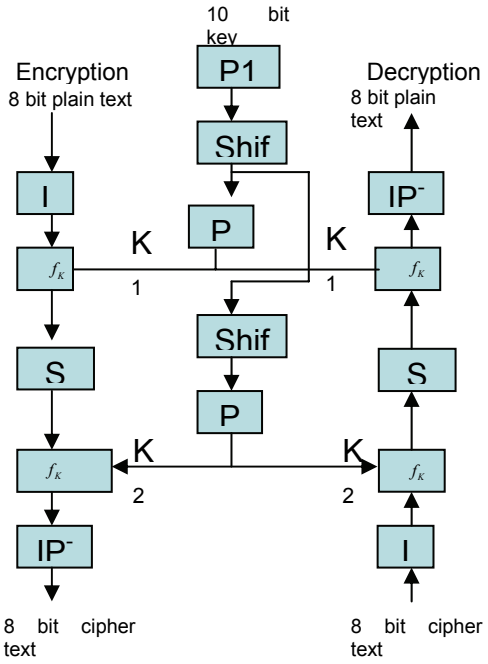


Figure 1. Simplified Data encryption scheme

Encryption involves the sequential application of five functions:

1. Initial and final permutation (IP).

The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function IP= [2 6 3 1 4 8 5 7]. This retains all 8-bits of the plaintext but mixes them up. At the end of the algorithm, the inverse permutation is applied; the

inverse permutation is done by applying, IP^{-1} = [4 1 3 5 7 2 8 6] where we have IP^{-1} (IP(X)) =X.

2. The function f_K, which is the complex component of SDES, consists of a combination of permutation and substitution functions. The functions are given as follows.
   Let L, R be the left 4-bits and right 4-bits of the input, then, f_K (L, R) = (L XOR f(R, key), R)
   where XOR is the exclusive-OR operation and key is a sub - key. Computation of f(R, key) is done as follows.
   1. Apply expansion/permutation E/P= [4 1 2 3 2 3 4 1] to input 4-bits.
   2. Add the 8-bit key (XOR).
   3. Pass the left 4-bits through S-Box S_0 and the right 4-bits through S-Box S_1.
   4. Apply permutation P4 = [2 4 3 1].

The two S-boxes are defined as follows:

$$S_0 \qquad\qquad S_1$$

$$\begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix}$$

The S-boxes operate as follows: The first and fourth input bits are treated as 2-bit numbers that specify a row of the S-box and the second and third input bits specify a column of the S-box. The entry in that row and column in base 2 is the 2-bit output.

3. Since the function f_K allows only the leftmost 4-bits of the input, the switch function (SW) interchanges the left and right 4-bits so that the second instance of f_K operates on different 4- bits. In this second instance, the E/P, S_0, S_1 and P4 functions are the same as above but the key input is K2.

2 Objective of the Study

Cryptanalytic attack on SDES belongs to the class of NP-hard problem. Due to the constrained nature of the problem, this paper is looking for a new solution that improves the robustness against cryptanalytic attack with high effectiveness.

The objective of the study is:

- To determine the efficiency and accuracy of genetic algorithm on the cryptanalysis of SDES.
- To compare the relative performance of genetic algorithm with brute force search algorithm.

3 Genetic Algorithm

Genetic algorithms are considered as one of the most efficient search techniques. Although they do not offer an optimal solution, their ability to reach a suitable solution in considerably short time gives them their respectable role in many AI and searching techniques. The following section introduces genetic algorithms and describes their characteristics.

3.1 Genetic Algorithms Description

The genetic algorithm is based upon Darwinian evolution theory. The genetic algorithm is modeled on a relatively simple interpretation of the evolutionary process; however, it has proven to a reliable and powerful optimization technique in a wide variety of applications. Holland [9] in 1975 was first proposed the use of genetic algorithms for problem solving. Goldberg and Dejong [5] were also pioneers in the area of applying genetic processes to optimization. Over the past twenty years numerous application and adaptation of genetic algorithms have appeared in the literature. During each iteration of the algorithm, the processes of selection, reproduction and mutation each take place in order to produce the next generation of solution. Genetic Algorithm begins with a randomly selected population of chromosomes represented by strings. The GA uses the current population of strings to create a new population such that the strings in the new generation are on average better than those in current population (the selection depends on their fitness value). Three processes which have a parallel in biological genetics are used to make the transition from one population to the next (selection, crossover, and mutation) see Figure 2

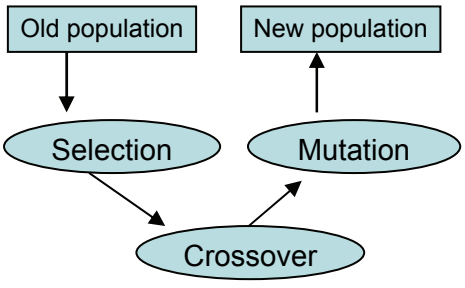


Figure 2. The basic genetic algorithm cycle

The selection process determines which string in the current will be used to create the next generation. The crossover process determines the actual form of the string in the next generation. Here two of the selected parents are paired. A fixed small mutation probability is set at the start of the algorithm.

4 Methodology: Genetic Algorithm Attack on SDES Algorithm

4.1 Problem Formulation

The possibility of using the genetic algorithm in key search is very attractive due to the ability of genetic algorithm to reduce the complexity of the search problem, and since the cryptanalysis problem is a search problem in principle, the security of a cipher is based in many cases on the complexity of the type of attacks to this particular cipher system.

An attack on the SDES using Genetic Algorithm is presented here. Three problems arises in the use of GAs in cryptanalysis

- Key representation
- Fitness function
- Stopping criteria of genetic algorithm search

4.1.1 Key Representation

The binary string is used to represent a chromosome, as key is also a binary word. The mating, crossover and mutation processes are applied directly on the candidate keys to generate new keys directed to the correct key.

4.1.2 Fitness Function

The ability of directing the random search process of the genetic algorithm by selecting the fittest chromosomes among the population is the main characteristic of the algorithm. So the fitness function is the main factor of the algorithm. The choice of fitness measure depends entirely on the language characteristics must be known. The technique used to compare candidate key is to compare n-gram statistics of the decrypted message with those of the language (which are assumed known). Equation 1 is a general formula used to determine the suitability of a proposed key(k), here ,K is known as language Statistics i.e for English, [A,…….,Z_], D is the decrypted message statistics, and u/b/t are the unigram, bigram and trigram statistics. The values of α, β and γ allow assigning of different weights to each of the three n-gram types where $\alpha + \beta + \gamma = 1$.

$$C_k \approx \alpha.\sum_{i\in A} K^u_{(i)} - D^u_{(i)} \mid + \beta.\sum_{i,j\in A} K^b_{(i,j)} - D^b_{(i,j)} \mid + \gamma.\sum_{i,j,k\in A} K^t_{(i,j,k)} - D^t_{(i,j,k)} \tag{1}$$

When trigram statistics are used, the complexity of equation (1) is O(P3) where P is the alphabet size. So it is an expensive task to calculate the trigram statistics. Hence

we will use assessment function based on bigram statistics only. Equation 1 is used as fitness function for genetic algorithm attack. The known language statistics are available in the literature [12, 13].

4.1.3 Stopping Criteria

Two variants of stop conditions are applied. In the variant-I, the algorithm will stops when the fitness function reaches to the value 1. In the variant - II the algorithm will stop either the fitness function reaches to the value 1 or generates 200 populations.

4.2 Proposed Genetic Algorithm Framework

A description of a series of test rounds is as follows.

1. Run step 2 - 11 for the S-DES.
2. Randomly select an encryption key from the main solution pool and encrypt the known plaintext message.
3. Create a new solution pool from the pool of common syllables and calculate the fitness value using equation 1 for all the solution in the new solution pool by decrypting the known-cipher text with each solution key by calculating the number of character matches in the subkey.
4. Choose two solutions with the best fitness value. Each solution should minimally able to recover at least 50% of the original plaintext message.
5. Randomly select a "crossover" point and swap the content between the two solution key arrays.
6. Evaluate the fitness for each new child (solution key) by decrypting the known-ciphertext with each "child" key and calculating the number of character matches in the subkey.
7. Choose the "child" solution with the highest fitness value.
8. Randomly select locations and mutate the selected locations with arbitrarily chosen unigram, bigram and trigram from the pool of common syllables.
9. Evaluate the fitness of the solution. If the fitness value is better then the current fitness value then update the current fitness value and the best solution variable.
10. Repeat step 8-9 until the fitness value is 1 or there is no change is best fitness for a predetermined number of iterations.
11. Repeat step 2-10 twice to produce a test series of 10 test rounds. Obtain the average number of search keys required to decrypt the full message correctly.

4.3 Implementation of the Attack

The above mentioned point was taken into consideration and implemented the attack using "C" language. The attack is implemented by generating independent two 8 bit sub-keys to represent the target key. The first generation is generated randomly using a simple uniform random number generator, to cover the bit space of the last round sub key. Then for each chromosome, a candidate sub key, the fitness is set to zero, then the entire test pair are used and the output difference is compared to the expected

differential. If the correct difference is found the fitness value is incremented and finally normalized to the number of pair.

The genetic algorithm then goes in the normal way to generate new generations. The roulette wheel is used as a selection method. The algorithm is stopped based on the criteria described earlier. The algorithm has been implemented to get key bits of the last round; essentially the attack shall continue upward to get the entire key bits.

5 Experimental Results

Experimental results obtained from genetic algorithm was generated with 100 runs per data point using 'C' language e.g. ten different messages ere created for genetic algorithm and each attack was run 10 times per message. The best result for each message was averaged to produce data point. The attack had been implemented several times, initial genetic algorithm parameters were used in the experiment are listed in Table 1.

Table1. Parameters of genetic algorithm

Population size in each generation	200
Probability of crossover	.8
Probability of mutation	.02
Minimum number of generations	100
Convergence measure	.99

To test the performance of genetic algorithm for the attack of S-DES, we compared it with brute-force search algorithm. Brute-force search algorithm tries every possibility; it always finds the best correlation. By comparing with brute-force search algorithm, we want to know how often genetic algorithm finds the optimal solution and how well it performs in average. The correlation coefficient is used as the measure of finding number of bits matched in key with accuracy.

We tried the different amount of cipher text in this experiment. Both algorithms were run on different amount of cipher text. As we are increasing the amount of cipher text, we essentially increase the search space of genetic algorithm. This enables us to track how well genetic algorithm works in different search space sizes.Table 2 list the result for genetic algorithm and brute force search algorithm, when the different amount of known cipher text is used.

By comparing the correlation column for the two algorithms, we can also see that the genetic algorithm performs better then brute force algorithms. The accuracy of finding number of bits matched in key (correlation) for both algorithms is very close for all the 10 messages. Even some of them are exactly the same. This can be better viewed using the graphical representation, as shown in Figure 3.

Comparing the running time of the two algorithms, we found that running time of genetic algorithm is not sensitive to the size of the search space. For different amount of cipher text, genetic algorithm converges at average period of 3 minutes. In contrast brute force search algorithm is more sensitive to amount of cipher text, as it takes up

to 30 minutes. From Figure 4, we can immediately see that the running time of brute force search algorithm is nearly ten times as much as that of genetic algorithm.

Table 2. Comparison of genetic algorithm and brute force search algorithm

Amount of Cipher text	Genetic Algorithm			Brute force search algorithm		
	TIME (M)	Correl.	Number of bit matched in the key (N)	TIME (M)	Correl.	Number of bit matched in the key (N)
100	2.62	.65	7	24.3	.5	8
200	4.5	.57	6	25.1	.53	6
300	2.13	.29	3	24.67	.31	4
400	2.35	.46	6	25.23	.43	7
500	2.52	.45	6	24.42	.41	6
600	2.07	.76	8	24.73	.77	9
700	4.07	.41	7	25.57	.53	7
800	2.82	.53	8	24.4	.49	8
900	2.53	.27	5	24.15	.16	6
1000	2.17	.37	9	24.35	.34	9

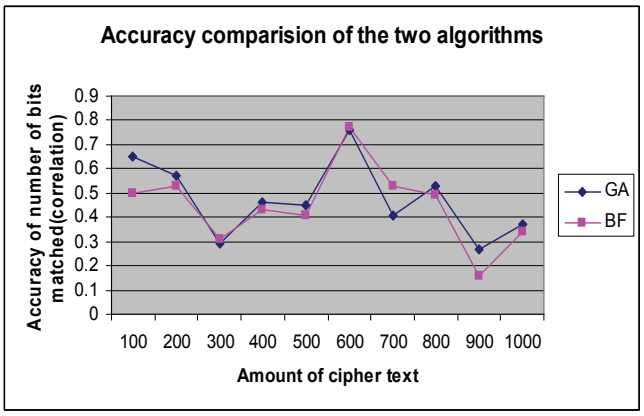


Figure 3. The accuracy of genetic algorithm and brute force search algorithm is very close. But genetic algorithm runs ten times faster (see Figure 5). The accuracy is the actual correlation found by each algorithm

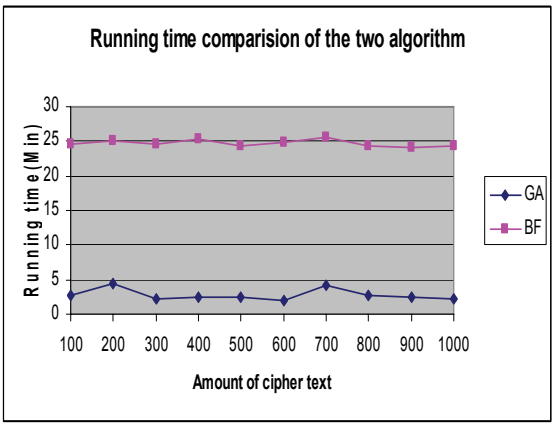


Figure 4. The running time brute force search algorithm is 10 time longer then genetic algorithm

6 Conclusion

The paper explains the genetic algorithm attack on the simplified data encryption standard algorithm. In this paper we have compared genetic algorithm and brute force search algorithm results for the attack of S-DES. Our experimental result shows that genetic algorithm is very efficient method for the cryptanalysis of S-DES. Genetic algorithm takes only 3 Minute to find the correct solution. In contrast, the brute force search algorithm takes up to 30 Minutes. When we are amount of cipher text increases, the running time of genetic algorithm keeps almost the same, but the brute force search algorithm takes more than half an hour. Genetic algorithm is an extremely effective approach for the attack S-DES. It found the best correlation for different amount of cipher text, the performance of genetic algorithm is really good. The overall average accuracy of genetic algorithm (against the best correlation found by brute-force search algorithm) is 95.04%, while average accuracy of brute force search algorithm is only 82.2%. So genetic algorithm improved the accuracy of finding the bits in the key by 13% with same running time.

References

1. Davis, L. , “Handbook of Genetic Algorithms”, Van Nostrand Reinhold, New York ,1991.
2. Garg Poonam, Sherry A.M., Genetic algorithm & tabu search attack on the monoalphabetic substitution cipher, Paradime Vol IX no.1, January-June 2005,pg 106-109

3. Garg Poonam, Shastri Aditya, Agarwal D.C, Genetic Algorithm, Tabu Search & Simulated annealing Attack on Transposition Cipher ,proceeding of Third AIMS International conference on management at IIMA – 2006, pg 983-989
4. Garg Poonam, Shastri Aditya, An improved cryptanalytic attack on knapsack cipher using genetic algorithm approach, International journal of information technology, volume 3 number 3 2006, ISSN 1305-2403, 145-152
5. Goldberg, D.E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, Reading, 1989.
6. Giddy J. P and Safavi-Naini R., Automated cryptanalysis of transposition ciphers, The Computer Journal, Vol 37, No. 5, 1994.
7. Gr¨undlingh, W. & van Vuuren, J. H., Using Genetic Algorithms to Break a Simple Cryptographic Cipher, Retrieved March 31, 2003 from http://dip.sun.ac.za/˜vuuren/abstracts/abstr genetic.htm, submitted 2002.
8. Kolodziejczyk, J., Miller, J., & Phillips, P. ,The application of genetic algorithm in cryptoanalysis of knapsack cipher, In Krasnoproshin, V., Soldek, J., 1997
9. Holland, J., "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975.
10. Methew, R.A.J. (1993, April), The use of genetic algorithms in cryptanalysis, Cryptologia, 7(4), 187-201.
11. Schaefer E, A simplified data encryption standard algorithm, Cryptologia, Vol 20, No 1, 77-84, 1996.
12. Spillman et. al., Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers, Cryptologia, 17(1):187-201, April 1993.
13. Spillman R.,Cryptanalysis of knapsack ciphers using genetic algorithms. Cryptologia, 17(4):367–377, October 1993.
14. Yaseen, I.F.T., & Sahasrabuddhe, H.V. (1999), A genetic algorithm for the Cryptanalysis of Chor-Rivest knapsack public key cryptosystem (PKC), In Proceedings of Third International Conference on Computational Intelligence and Multimedia Applications, pp. 81-85, 1999.

A File Protection Method for Peer-to-Peer Systems

Antonio Liotta[1], Rossana Motta[2] and Ling Lin[1]

[1] Department of Electronic System Engineering
University of Essex, Colchester, CO4 3SQ, UK
{aliotta, llini}@essex.ac.uk
[2] Florida State University, Tallahassee, FL 32306, USA
motta@cs.fsu.edu

Abstract. The benefits of peer-to-peer (P2P) systems as vehicles for information dissemination are overwhelming. They beat the conventional client-server (C-S) approach in terms of scalability, availability, and network efficiency to mention a few. Nevertheless, the problem of protecting copyrighted material in P2P systems is far from solved, whilst several solutions have been proposed and are in use in a C-S environment. In this article, we propose a framework which allows 'legal' P2P file exchange, making it possible for copyright holders to employ P2P as a distribution vehicle. We illustrate a scenario where all parties involved, including the end user, benefit from our approach. Lessons learned and conclusions are drawn from a prototype implementation.

1 Introduction

The Internet as we know it today has become the major vehicle for distributing multimedia digital material directly and conveniently to the end user. But what is the ultimate technology for content provisioning?

The main mechanisms for reaching the user terminal are known as client-server (or C-S) and peer-to-peer (or P2P). In C-S, the content is made available via computer servers (or server farms). The user can search for relevant information (via any of the popular search engines) and finally retrieves the content through the browser (i.e. the client application). Clearly this approach is limited in terms of scalability, availability, and efficiency, since the server (or server cluster) constitutes a computational and network bottleneck. We have all experienced occasions when a web page (or server) is not accessible or when we cannot connect to a real-time streaming service.

Most limitations of C-S have been overcome in P2P systems. In this case, the user terminals themselves form a sort of self-organized content distribution network. Any of the terminals joining the P2P network can in fact play not only the role of passive user (or client) but also, and seamlessly, the role of content provider (or server) - because of this, all terminals are peers. Assuming that suitable publish and search mechanisms are in place, content can be downloaded from any terminal (holding a copy of it) to any other terminal. In the most common P2P systems, whenever the demand and popularity of a file increases, more and more copies of that file will be

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 151-160

Received 13/07/06
Accepted 04/10/06
Final version 10/10/06

spread in the P2P network. In fact, many peers will have that file among their resources and other peers will have the chance to download it from several sources, feeding a positive circle chain. Thus, we can easily see why this approach is considerably more powerful than its C-S counterpart (there is no single server, no single point of failure, and no bottleneck).

Despite its superiority as a mechanism for moving content around, P2P presents a major problem when it comes to protecting copyrighted material. In fact, we can confidently say that none of the largest media production companies is currently conceiving P2P as a vehicle for distributing copyrighted material. P2P systems have in fact become popular in the realm of free (or illegal) content distribution, while most of the non-free content reaches the user via C-S transactional systems for which suitable security means are available.

In this paper, we propose a system which allows content providers to use P2P as a vehicle for distributing copyrighted material, addressing the three key legal challenges of the copyright law: copy infringement; distribution infringement, and vicarious and contributory liability (a thorough explanation of these issues is given in [1]).

In our approach, a trusted entity (for instance the network operator) handles user authentication, registration, authorization and transactions, acting as an intermediary between content provider and users. This portion of the system works in C-S mode, whereas copyrighted content circulates among users in a P2P fashion. We explain how existing encryption and key management technologies can be used in a new way in order to protect files in P2P systems. Our approach is combined with an incentive-based scheme where all parties involved benefit from the 'legal' circulation of information.

Conclusions are drawn from a prototype implementation which has been developed by an academic institution in collaboration with a major network operator. Our prototype runs on thin mobile terminals (personal digital assistants), in addition to ordinary personal computers, and is able to operate not only in wired and wireless networks but also over UMTS.

2 Related Work

2.1 Existing Solutions

While there is a huge variety of different applications and protocols when considering P2P and Digital Right Management (DRM) contexts separately, it appears that no effective solutions are actually available, when trying to merge such contexts.

The great most of P2P systems simply do not have any features at all that allow DRM policy to be integrated. To our knowledge, the only P2P system that claims to distribute DRM-protected content is Kazaa [4]. The latter, however, exploits software from Altnet company [5], which has been recognized by any antivirus as responsible of spywares and other malwares (*i.e.*, "software installed without the user's informed consent, is difficult or impossible to uninstall, and transmits information about the user's activities without notice or consent" [3]). It is even not completely clear if

those files are downloaded in a C-S manner, as some sources report (for instance [8]), or via P2P, as Kazaa claims. During trials that we have performed ourselves after installing Kazaa, we have not been able to find any DRM-protected file residing on peers, whilst all of them were on central servers. Thus we are not able to confirm that Altnet system embedded in Kazaa is actually P2P. In any case, the installation of spywares as an integral part of DRM policy is clearly unacceptable and not viable, if seeking a business approach.

A number of different protocols exist for DRM itself. Actually one of the biggest problems that DRM is facing is the lack of interoperability [9], so that users are tied to very specific applications every time they want to render their files. This clearly feeds a large-scale mistrust toward DRM systems in general. The situation has been made even worse after the clamorous case of one of the most popular worldwide record companies, which has grossly exploited spyware techniques, in the intention of achieving efficient copyright respect [3].

There is actually a growing interest from several major companies dealing with DRM for the so called OMA-DRM project [7]. This represents an attempt toward a large-scale standardization of DRM and at the same time it includes several features that would make it portable in a P2P environment. OMA-DRM is actually still at a development stage and further studies on its implementation in large-scale real systems would be needed, in order to draw more precise conclusions [10].

2.2 Requirements for DRM-enabled P2P

The P2P realm is evolving at an unprecedented pace, which makes it very difficult to capture a complete snapshot of P2P systems and technologies. To the best of our knowledge, none of the existing P2P systems satisfies the following requirements, which are fundamental to effective DRM for commercially-viable P2P:

- *Content provider protection*: prevents the user from illegally duplicating and distributing digital material. Current encryption schemes used in C-S systems cannot be easily exported to P2P - users can currently easily inject content (obtained either legally or illegally) into the P2P network. Some P2P systems such as *Freenet* [2] guarantee user's anonymity, further exacerbating the illegal file distribution problem.
- *P2P provider protection*: the P2P platform developer incurs in vicarious liability if it can be proved that they can exert some form of control over direct infringers' behavior [1]. This is one of the reasons why *Napsters* lost a lawsuit and was shut down.
- *User protection*: some P2P systems make use of *spyware* to monitor user actions and achieve some sort of DRM. As mentioned, this is the case of *Kazaa*.

Our solution, as proposed below, satisfies these three requirements, while also having the following advantages:

- It makes use of well-established encryption technologies;
- It is generic, and may be used with different devises and file formats other than the ones used to develop our prototype;

- It is based on user incentives (it allows the implementation of different economic and rewarding schemes), and ensures that all parties involved (user, content provider, P2P service provider) benefit from joining the service;
- It does not require an unbreakable DRM system (at present it would be unrealistic to assume that such a system can be realized) since the user (i.e. the potential hacker) would stop receiving rewords in that case (this will become clear after the following section). Our system is sufficiently robust to prevent involuntary copyright infringement and would require substantial expertise and facilities (beyond the abilities of the typical expert Internet user) to be broken.
- It is based on a robust authentication framework handled by the network operator, which instills a sense of trust and encourages user participation to the P2P playground.

3 DRM-enabled P2P system

3.1 System entities

In a real-case scenario there will be several entities that will compete for a slice of the media selling market and others who will collaborate to enhance the user accessibility to content. However, for the sake of simplicity we assume that the key players of our P2P content distribution systems are:

- *A Content Provider (CP)*. This will be a media production or broadcasting company producing music, videos, films, *etc.* and holding the copyrights.
- *A Trusted Entity (TE)*. This will be a network operator, service provider or any other company that manages a large amount of subscribers and has gained their trust. The TE has the means to handle user authentication, authorization, charging and billing (via a transactional system).
- *The users (or Peers)*. These will be ordinary users with a terminal connected to the Internet (subscribing to a TE) and who want to purchase copyright protected material. Peers will seamlessly act as content distributors in the P2P network.

In general there will be several content providers advertising their products (music files, videos, etc) via brokers (yellow or white pages). Users will register with different (competing) trusted entities (for instance their own mobile network operator) that will also have to federate in order to allow user and service portability (similarly to what mobile network operators do to support the roaming user). Our approach to DRM-enabled P2P has been designed to work across network operator boundaries (inter-domain) and independently from access network technologies and user terminal type. The CP may also act as TE or *vice versa*.

3.2 Bootstrapping

The distribution of digital content from producer to the end user will be 'initially' mediated by the TE who, for each file, will produce the encrypting/decrypting key and encrypt the file itself. One of the state-of-the-art symmetric encryption techniques will be used. We'll see how, once the P2P network gets populated with copies of the original individual files, the TE will not have to act as a content distribution hub anymore, since content will be distributed directly by the peers. The TE will still hold a master copy of all files injected in the P2P system.

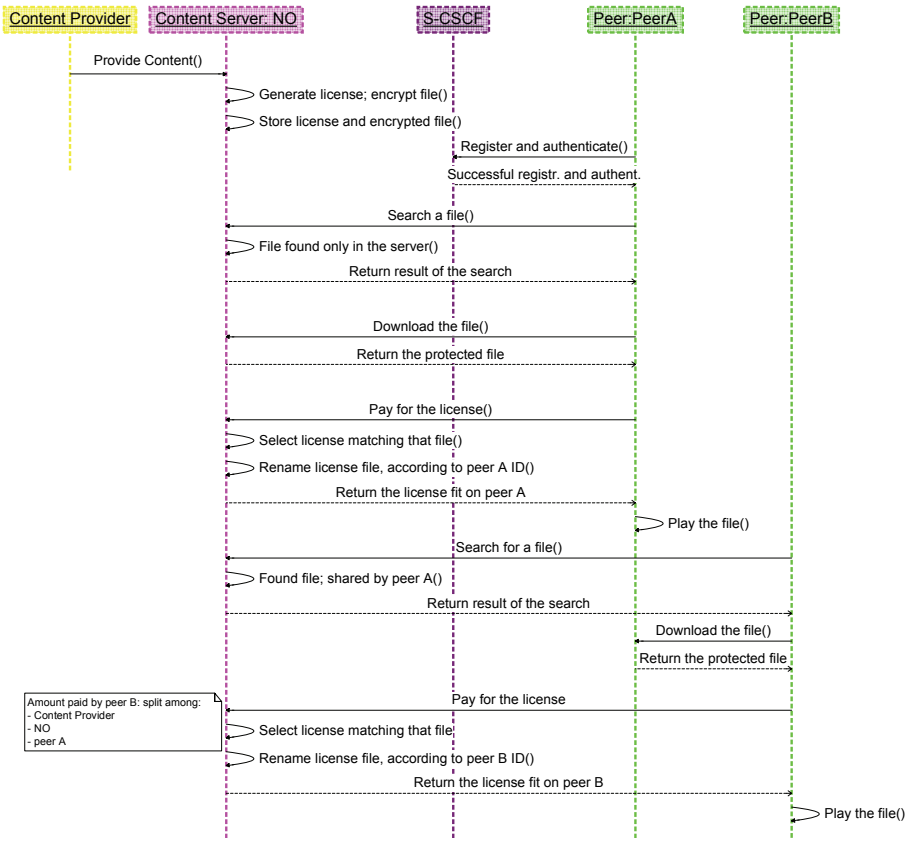


Fig. 1. Sequence of diagram describing main entities and interactions**.**

Figure 1 illustrates the bootstrapping of the content distribution process, starting from the scenario where there are no peers and all content is stored in the content server. We assume that this is managed by the Network Operator (NO) who is also in charge of user and transaction management.

The first operation any peer will have to perform before getting access to the services/content available in the P2P network is registration/authentication. `PeerA`

will then join the P2P network and have access to the search facilities which will provide the same sort of functionality of ordinary Internet search engines.

3.3 File protection hurdles

If the retrieved file is copyright protected, `PeerA` will have to be debited the relevant cost before obtaining the license key that will decrypt the file. This process would be straightforward in a C-S system but is extremely detrimental in the context of P2P. Once `PeerA` gets hold of the key, it could easily share the license key of the file with any other peer. This is in fact the crucial problem with current P2P systems which we want to counter since this is what generates the copy and distribution infringement issue.

The technical problem we have to address is how to allow the same encrypted file to circulate in the P2P network, allowing authorized users to read (or play) it freely but preventing illicit swapping of licenses. We tackle this problem from different fronts. First, we protect the key (Sect. 3.4). Then, since it is quite hard to create an unbreakable DRM system (especially in the context of P2P), we couple the P2P system with a strong user authentication and authorization system (provided by the network operator in our prototype). Finally, we encourage lawful peer participation via an incentive mechanism, which, on the other hand, demotivates hacking actions (Sect. 3.5).

3.4 P2P file protection

The application running on the peer (*e.g.*, the player if the file is a music or video file) expects the key to have a specific filename. The filename is generated by the server using a secret function, *F,* that gets as input the tuple `{peer ID; ID of the encrypted file}`. *F* can be specified starting from state-of-the-art cryptographic algorithms and will incorporate a set of operations consisting of bits manipulations, shifts, bitwise mathematical operations etc. Every bit of the input tuple is considered by *F*, so any variation of the input would result in a different output.

Going back to Figure 1, upon receiving the payment for the relevant license, the server renames the license file univocally to `PeerA` and forwards it to the peer. `PeerA` will now use *F* to compute the location (filename) where the license key will have to be in. It will, then, feed the actual key and source file to the player. A further advantage of our approach is that the only proprietary module is the one running *F*, while it will work with any type of file and application. For instance, if we are using the system to distribute music files, our system will operate with any (locally installed) player application.

Let us see what happens when a second peer, `PeerB` is searching the same file that `PeerA` has just obtained (for simplicity Figure 1 does not illustrate `PeerB`'s registration phase). Now there are two copies of the same encrypted file, one in the server and the other in `PeerA`'s shared resources. Clearly, this time `PeerB` will download the file from `PeerA`, so the server will be relieved from this task. Given that the file is protected, `PeerB` will have to pay for the license, expecting the key to

be stored in a file whose filename is uniquely tailored to `PeerB`. This is something that `PeerA` cannot produce, unless it successfully reverse-engineers *F*. Then, even if `PeerA` were to pass its own license key to `PeerB`, this would not be in the correct filename and path and would be unusable by the decryption routine.

As more peers join the P2P system and express interest in certain files, the latter would gradually populate the system. The licenses, though, will always have to come from the server, which is how we obtain a DRM-enabled P2P. In addition, the more peers are interested in a given file, the more copies of that file will be circulating into the system, leading to increasing levels of scalability and availability.

3.5 Trust, policing, and incentives

In order to further increase the level of security, the file protection mechanism described above is carried out in conjunction with a strong user authentication and authorization component. Our prototype, as specified in Section 4, is integrated in the context of the IP Multimedia Subsystem (IMS), which is the service provisioning framework adopted by the major network operators [6]. Because of that, the IMS comes with a strong emphasis on user management and with specific service components for transaction management, charging and billing. The operator will, therefore, be in a good position to act as trusted entity, monitor the user activity, and handle the necessary key management process.

Effectively, what the operator's platform gives us is the ability to know that the peers in our P2P system are who they say they are. Peers will have also accepted appropriate terms and conditions at registration time. In this way it will be possible to prosecute copyright infringers.

Experience has shown that law enforcement *per se* is never sufficient to stop the compulsive hackers. Economic incentives, instead, appear to be more successful at deterring illegal activities. In our system, a P2P file exchange is handled as an atomic transaction which is concluded when the license has been paid for. Referring to Figure 1, once `PeerB` pays, the operator charging system will credit a portion to itself, a portion to the copyright holder and another portion to `PeerA`, as a reward for acting as file distributor.

Therefore, peers have concrete advantages from 'legal' file circulation. Even if they managed to hack the system, obtain an unencrypted copy of the original file, and get away with the network operator's policing system, they would damage themselves by injecting the unencrypted file in the P2P system – i.e., they would not receive any economic incentive in doing so.

4 The Prototype

What we have actually prototyped is a P2P system built on the IMS (Figure 2). We believe to be first to have accomplished a concrete example of an operator-mediated P2P system, where P2P operations are supervised, traced and charged by the network operator (*i.e.*, the trusted entity).

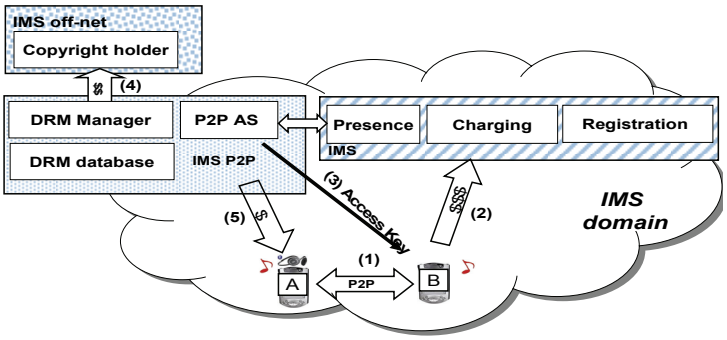


Fig. 2. Logical view of prototype implementation**.**

Figure 2 presents a simplified view of the system which is, in fact, considerably more complicated, since the IMS is a comprehensive service provisioning platform. The *presence*, *charging* and *registration* components have been implemented according to standards specifications [6]. We have built our DRM-P2P system as an extension to the IMS. The content provider does not belong, in general, to the operator's (IMS) domain, although this is not a requirement.

Our IMS P2P set of functionalities has been implemented in Java 1.5.06 (December 2005 release) and has been tested on Linux and Windows XP. We have also prototyped a peer-side application for mobile terminals – specifically the Hewlett-Packard iPAQ h5555 Pocket PC - based on J2ME and WindowsMobile. In this way we have proved that the proposed approach works on relatively thin terminals in addition to ordinary computers.

For encryption and key management we have adopted the AES (advanced encryption standard) symmetric block cipher algorithm - CBC (cipher block chaining) cipher mode and PKCS7 padding mode. CBC needs not only a key but also an Initialization Vector (IV). Hence, in our prototype, the license includes two strings (128 bits each), which are pseudo-randomly generated by the server at encryption time.

For the input parameters of the encryption function, *F* of Sect. 3.4 we have adopted the SIP identifier (to uniquely identify the peer ID) and as file ID the name of the encrypted file. A more robust solution would be to use hash functions (*e.g.* based on MD5). But this operation would have been too heavyweight (especially as file size grows), considering that our peers maybe thin mobile clients. *F* can be defined starting from ordinary cryptographic algorithms, performing an original set of operations, manipulations and shifts on the input parameters.

Just to get an idea of the order of magnitude of the decryption overheads incurred on a terminal with a 2.4 GHz CPU, 1 Gbyte RAM, we obtained the following figures (similar on Linux and Windows XP):

File size (Mbytes)	Overheads (seconds)
3.7	1
6	2
15	4
50	14

Finally, the user rewarding scheme has been implemented according to the functionalities specified by OMA DRM v2 [7].

The aforementioned technological choices are merely an indication of how our system may be readily realized. However, other similar (or future) technologies can be chosen.

5 Conclusions

The community interested in media and information dissemination is split between those who advocate and those who oppose the copyright law. The widespread appearance of P2P systems has re-fuelled this controversy, since the P2P approach is significantly more efficient than its C-S counterpart. P2P is, however, being largely used for distributing free content while also sparking the illegal distribution of copyrighted material.

The perspective presented in this article is that the copyright infringement issue would naturally disappear if all parties involved (including the user) could benefit from the legal distribution of copyrighted material. We have presented a framework which exploits P2P as a content distribution platform (giving the user the role of distribution hub) while also addressing the DRM issue.

Our system can be used to experiment with different types of economic (incentive-based) models. An interesting exercise would be to see what happens when different incentive schemes are deployed in a very large scale. We plan to run simulations to unveil the effects that different incentives have on system scalability.

We do expect to see an increase in user participation to the P2P distribution role. Current P2P systems suffer from the so-called *free-riding* issue, whereby peers obtain content from the network but do not act as distributors. They do that in order to save the consumption of their own computer and network resources. But, in doing so, they fail the very purpose of P2P and lead to a dramatic reduction in performance – the performance of P2P systems degenerates dramatically when free-riders dominate the scene. By providing economic incentives to user participation, our approach will combat the free-riding issue.

Having integrated our prototype with a state-of-the-art (standardized) service provisioning platform (*i.e.*, the IMS), we are effectively showing an evolutionary path towards commercially-viable P2P content distribution. We have also demonstrated that such approach can work across network boundaries (we have tested it on UMTS, as well as on wired and wireless access networks) and for different terminals (PDAs and personal computers).

Hence, our findings indicate that P2P can, indeed, become the ultimate technology for content provisioning. A lot of work lays ahead in order to fully address a number of legal, standardization and interoperability issues.

Acknowledgements. This work has been carried out in the context of the PeerMob project, which is funded by Vodafone Group Plc. We are particularly grateful to Patrick Brick, Max Gasparroni, and Marco Ballette (all from Vodafone) for their insightful comments.

References

1. S.T. Logan, "Peer-to-Peer Technology and the Copyright Crossroads". In "Peer-to-Peer Computing", Idea Group Publishing, 2005, pp.166-193.
2. I. Clarke et al., "Protecting Free Expression Online with Freenet", *IEEE Internet Computing*, IEEE, Jan-Feb 2002, pp. 40-49.
3. E.W. Felten, J. A. Halderman, "Digital Rights Management, Spyware, and Security", *IEEE Security and Privacy*, IEEE, Jan-Feb 2006, pp. 18-23.
4. The Kazaa system is available at www.kazaa.com
5. The Altnet software is available at www.altnet.com
6. G. Camarillo, M.A Garcia-Martin "The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds". John Wiley and Sons Ltd, December 2005.
7. The Open Mobile Alliance Digital Right Management specifications are accessible at www.openmobilealliance.org/release_program/drm_v2_0.html
8. http://en.wikipedia.org/wiki/Altnet
9. Heileman, G. L. and Jamkhedkar, P. A. 2005. DRM interoperability analysis from the perspective of a layered framework. In Proceedings of the 5th ACM Workshop on Digital Rights Management (Alexandria, VA, USA, November 07 - 07, 2005). DRM '05. ACM Press, New York, NY, 17-26.
10. Thull, D. and Sannino, R. 2005. Performance Considerations for an Embedded Implementation of OMA DRM 2. In *Proceedings of the Conference on Design, Automation and Test in Europe - Volume 3* (March 07 - 11, 2005). Design, Automation, and Test in Europe. IEEE Computer Society, Washington, DC, 46-51.

Computer Networks

Integrating the Enterprise Service Bus in the Service Oriented Architecture for Middleware Extension

Guillaume Koum[1], Tam Sangbong[2], Raoul M. Tsamo[2],
Emmanuel Etoundi[2] and Augustin Yekel[1]

[1] Ecole Nationale Supérieure Polytechnique, Université de Yaoundé,
P.O. Box. 8390 Yaoundé, Cameroon
[2]Soft-Tech International Inc,
P.O. Box. 15412 Douala, Cameroon
g_koum@yahoo.fr; tam@soft-techint.com, rtsamo@yahoo.com; etoundiaboa@yahoo.fr;
a_yekel@yahoo.fr

Abstract. Our principal concern lies in designing and implementing a model of integration and communication of services within SOA environment, elaborated enough to allow several services to use a single input/output GSM (Global System for Mobile Telecommunication) channel. This model must offer a high flexibility in integrating new services (or new applications). This model is based on the characteristics of SOA and on communication of type publication/subscription from Event Driven Architecture (EDA).

Keywords: Service Oriented Architecture (SOA), Enterprise Service Bus (ESB), Event Driven Architecture (EDA), middleware.

1 Introduction

SOA is the philosophy that views software systems as multiple services connected to each other. In the context of SOA, a service is simply an autonomous application that runs and communicates to other services by exchanging messages. But an implementation of SOA suggests the use of an Enterprise Services Bus (ESB) because of complexity in managing nxn pairs of communication that could exist if we have n services. The goal of this article consists in designing and implementing an ESB that enable several services to use the same GSM channel for input/output and offers a high flexibility of integration of new services.

The ESB is something that plays a role of router between the services. More explicitly, the bus ensures the transport of messages between the various services which are connected to in order to carry out business functionalities. Because the data used by each service are not always in the same format, the ESB consequently needs to have a function of transformation, data conversion between the services. Moreover, as a router, it will have to ensure the connectivity and the transport of messages. These functionalities are in fact the necessary sub-set. For these purely functional aspects and since we are in a distributed environment, it would be necessary that this bus supports the needs for rise in load, huge volume of data, safety and reliability of the messages, also real time, etc.

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 163-171

Received 07/07/06
Accepted 04/10/06
Final version 13/10/06

The communication model used for these services is of type event-driven. So, we will use the Event Driven Architecture (EDA) that proposes the development frameworks we have in the market. EDA is the type of architecture in which all the components are communicating to others by sending events. There is the publisher, the receiver and/or the broker of events. The publisher publishes the event; the receiver subscribes to the event before and specifies the action to be taken in the case of this event. The broker manages events. As such, it sends it to the good receiver after receiving it from the publisher. Event is like a signal. It naturally has its data. For example, you have a new mail is an event which has the mail content as its data. So, event transports simultaneously its data. Events are mostly caused by changing some data.

2 Our Context

In our case, connections to the bus are done with a SMS (Shorts Message Service). SMS is a technology in the GSM that allows sending small text to a mobile phone. SMS is used in multiple applications nowadays to ensure mobility of the users. [16] is the standard that specifies the format of SMS.

The figure 1 shows the conceptual architecture of this bus based on the use of SMS messages.

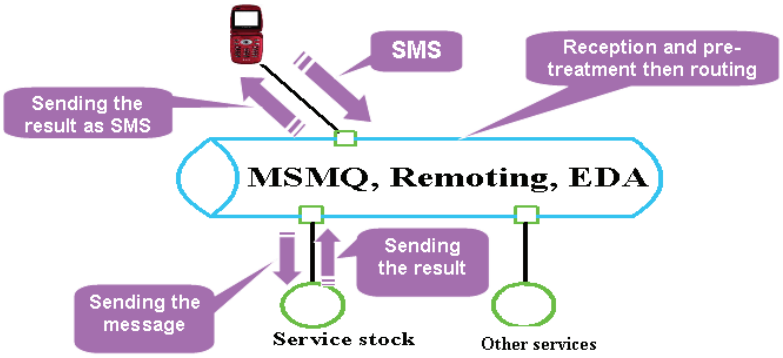


Fig. 1. Conceptual architecture.

The application that we built on this architecture meets the need of mobility of the users above. It will be possible to whomever to reach the information system (IS) of his company, with a certain number of rights naturally, to collect desired information. A specific case would be for example to give the possibility to an itinerant salesman to query the IS of his company in order to get for example the stock of a certain product. This query will be done from a mobile phone. Specifically, he will send a SMS to the system. Once the SMS is received, the ESB will send it to the suitable service which has been given the responsibility to analyze it and to carry out the required processing. At the end of the process the result from this service will

automatically be returned to the applicant. The applicant will at this time receive it directly on his mobile phone.

This form of communication is more advantageous because the user continues to have access to the IS of his company at anytime and anywhere (as long as the GSM network permits it). He is able to get the real time information which is essential for his business! In those IS where there are multiple updates, the user always has the possibility of getting actual data (for example case in the stock exchange place).

Our objective is to give to users the access to a multitude of services while passing by the same channel as figure 2 shows. Because there is only one channel and many services will access it, it is now necessary to have something (an application) that will manage this resource. We call it: a Manager. Its aim is to allow services to receive messages dedicated to them. These services can run as standalone applications. These services can be built by other teams; and, in order to use the channel, they will easily be grafted on the system. The mode of communication between these services and the manager should thus be by exchanging messages: a communication base on program interfaces would lay more problems and would be more constraining.

Indeed, while communicating by the publication of the interfaces, the services and the manager will have mutually to know themselves; this would oblige the manager to know the interfaces of all the services. One could imagine the amount of difficulties that arises when one wants to add a new service. Thus, the manager must be modified each time a new service is added. This adds maintenance problems and shows a considerable lack of flexibility in adding and removing services. In addition, it also poses a real problem of performance because the manager will be blocked at each service call since the communications would need to be synchronous in that case. If the performance is reduced then is also the quality of service (QoS). The huge volume of solicitation of the manager (what will be the case besides) will lead directly to freezing system what puts everything in danger!

Compared to the above discussion, we built a single manager that takes directly into account installation of the new services into the system. This manager is completely autonomous.

However, the access to the system needs to fulfill some criteria of security in order to guarantee the safety and the integrity of the system. For this purpose, it will be necessary to define rules of authentication and authorization. And given the fact that some data are confidential in the IS, it would be interesting to apply some concepts of cryptography and encryption to these data. These aspects were not largely applied here because of the prototype case of this application. But they must always be in mind once in a distributed or opened environment.

The figure below shows the necessary infrastructure that enables the system to operate. But it is interesting to present the models of communication used to make components communicate to each other. The dotted lines in this figure delimit the various circuits that exist for the transport of the requests from the mobile phone to the services; then answers from the services to the requester mobile phone. This figure shows also the technological options in circuits 2 and 3:

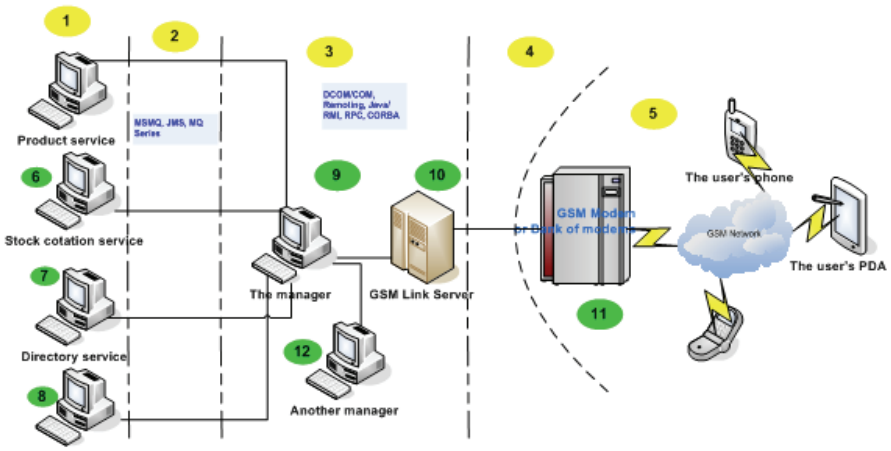


Fig. 2. Physical architecture and technological options.

The circuits sections in question are marked by yellow rounds 1, 2, 3, 4 and 5. They are described below:

1. This portion contains the various services which can execute requests from the users. New services can be integrated by using the infrastructure offered into 2.
2. This circuit represents the channel used so that these services communicate with the Manager. An analysis enabled us to note that this communication is not only of client/server type but should also be made by exchanging messages in order to meet the objectives referred above. The technological options are as follows: MSMQ (Microsoft Message Queue), MQ Series, and JMS (Java Message Queue). Services will subscribe to the events of arrived messages on these queues. So, the manager will write message on the queue. But the manager also subscribe to the same event on another queue in other to get results from services.
3. This circuit represents the channel used so that the Manager communicates with the GSM Link Server. It is a traditional type of communication between applications. Hence the use of directed object middlewares. The technological options are as follows: DCOM/COM+ (Distributed Component Object Model), Remoting, CORBA (Common Object Request Broker), Java/RMI (Remote Method Invocation). But since the GSM Link Server generates also events and that these events must be captured by the Manager, we implemented a model of EDA architecture between the two services that enables capturing events produced by GSM Link server. All communications are based on events and data are simply serializable objects that pass through the middleware.
4. This channel is used for the connection of the Modem to the serial port of GSM Link Server.
5. It is the GSM network which communicates with the GSM Link Server through the use of the Modem that serves as the interface with. The modem is connected to a COM port of the computer. And when an SMS arrives the modem writes data to the COM port and the listening program can then read the incoming data.

The data received on this port are analyzed against [16]. After this analysis, the program composes an object that is made up of the incoming SMS elements as stated in [16]. The system then fires an event and includes the composed object as the event's data. The event constitutes what the manager will the event.

On the figure above, applications involved in the system are identified by green rounds and each machine conceptually consists of one application.

Rounds 6, 7 and 8 contain respectively the Product service, Stock quotation service, the directory services. Rounds 9 and 12 are managers who send/receive messages for services; they consume also the services offered by the GSM Link Server (round 10). This machine has the functionalities that enable it to manage a bulk of modems represented by the component numbered 11.

The preceded sections enable us to show the physical architecture necessary to the implementation of our system having in mind the needs of mobility and real time. This architecture enables us to control the physical infrastructure and the ways that the various messages will get to their recipients. The following section presents the implementation of the prototype which turns for the execution of the main scenario by requesting a service from a SMS.

3 Implementation of the Prototype

3.1 Tools and Languages for Development

1. Our application is a multi-tiers architecture. We used:
2. Microsoft SQL Server 2000 as the Database Management System;
3. Microsoft MSMQ™ for the exchange of messages between the services and the Manager;
4. Microsoft Visual Studio.NET with the Microsoft NET Framework as the development environment;
5. The implementation language used for the business objects is Visual C#.NET;
6. Microsoft NET Remoting for the communication of the Manager and GSM Link Server;
7. A modem for the sending/receiving SMS.

3.2 Tests

The correct functioning of the system consists in launching GSM Link Server initially. This application can run in background. It communicates directly with the modem which can be connected at any time on the serial port of the computer it is running on.

After starting this server, one then now launches the Manager who will undertake to communicate with the server. The starting of the service is characterized by the presence of a notification icon Windows taskbar.

The launching of the Modem is done from the Manager; what shows the remote control we have on the Modem which is the material from which events are all from. Events from the modem can be displayed as follows:

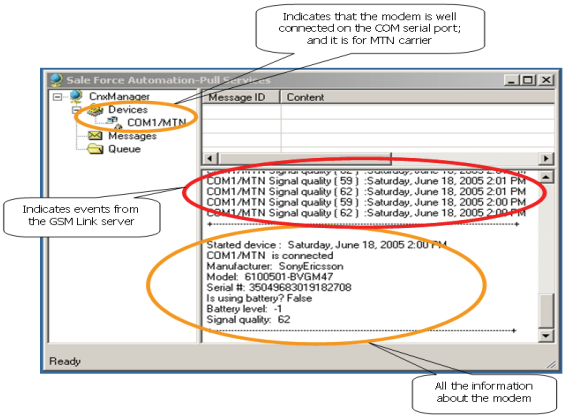


Fig. 3. The Manager main form.

The business services can be launched at any time.

We present the execution of the principal scenario described in the preceding sections:

1. Composition of the SMS by a user starting from his mobile phone. For example, the query to be sent is: "# STOCK#PRD00001 # " i.e. "I want the quantity in stock of the product code PRD00001.(STOCK: ID of the service, PRD00001: Produced ID, #: separator)
2. Reception by the modem, then by GSMLink Server;
3. Generation of the event corresponding to the reception of the SMS; and transfer of the data;
4. Reception of the data transmitted by the Manager and displaying; then insertion in the messages queue;

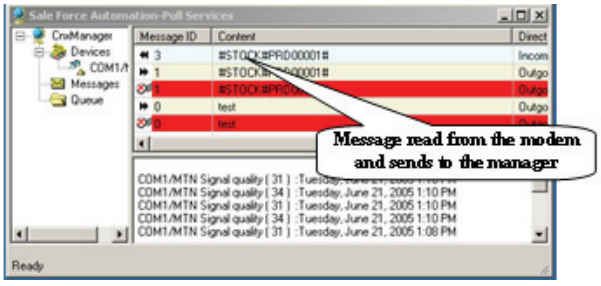


Fig. 4. Reception of the SMS by the Manager.

5. Automatic reception of the message by the concerned service and displaying, processing follows;

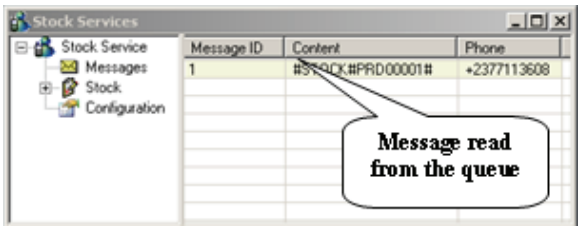


Fig. 5. Reading the message from the message queue.

6. Insertion of the result in the messages queue;
7. Automatic reception of the message by the Manager, then sending of the SMS to the server (GSM Lin Server);
8. Reception and sending of the SMS by the server (GSM Lin Server);
9. Reception of the result by the user through the portable. For example: # PR00001#45000#60 # From Sale Forces Automation # (PRD00001: Produced ID whose stock is required, 45000: Unit Price, 60: Quantity in stock, From Sale Forces Automation: For the signature of the source).

4 Conclusion

Services oriented architectures are today one of the elaborate solutions brought to solve the problem encountered during the integration of the components deployed in the different application servers. The services offered by this architecture are built on standards and because of that purpose could thus be consumed from any platform. This characteristic offers enormous advantages at the level of systems integration as well as at the level of constructing new solutions. The construction of new solutions will take less time because it is possible to compose a new service from those which already exist. This decreases considerably the T2M (Time-to-Market). However, it poses a small problem when one wants to integrate real time aspects in the solutions of company because the services are connected to each other and communicate in a traditional way (question/answer). This style of solution badly supports real time. However, company's IS are generally in a constant state of changes. Each change of state can be seen as an event. The fact that the quantity in stock of a product reaches a threshold is an example. This event can then automatically orchestrate a whole of consequent. Thus EDA are styles of architectures which implement this type of process offering solutions on a real time basis. Consequently, the coupling of the SOA with the EDA offers a more modern architecture than traditional ones.

In the context of the integration of services, the use of point-to-point can easily lead to a mesh network of services. This type of network is difficult to manage or maintain because of the high number of services and connections. The buses of services bring to this problem not only flexibility in the connection of the services but also reliability in sending message safely, the management of the priorities and the scalability (which is the capacity to be extended), etc. The research made in this direction enabled us to enrich the design of our system which is indeed a kind of bus.

Our research in this paper consisted in putting together the concepts of SOA, EDA, EAI and middlewares. These concepts enabled us to develop a model of bus of services allowing services considered as back-end to be able to use only one input/output GSM channel. This bus allows a flexible integration of new services (the services back-end). This system allows the services considered as back-end to give answers to queries carried out by the users of a given IS.

The problem we faced at the beginning of this work was that it was not possible that several services (or applications) use only one modem at the same time (or a bulk of modems). But now that is quite possible. This system can be distributed at the level of the Intranet as well as over the Internet i.e., the modem can be controlled from anywhere! The immediate economic impact is the resource sharing (modem in this case) and the market broadening.

With this purpose, we have exploited SOA, EDA and especially we could enable EDA to be distributed (and we called it d-EDA for Distributed Even Driven Architecture) by developing a certain architecture of classes on a middleware typically object-oriented.

The d-EDA has many advantages since from a network standpoint which generates events, several applications running on various machines can now process these events differently. These types of architectures are useful for example in Telecom, in hospitals (alarms of the medical equipment, etc), and in any field where one uses a device that generates events to be treated by an application.

The perspectives for this work go in the implementation of all the functionalities offered by a bus of services since we built only prototype not a complete bus. We used it only as a router with not as much intelligence as needed. We will enrich the system by the introduction of a messages broker and an intelligent routing process in order to be able to compose the results coming from several services. We will have to also integrate into this bus the capacity to receive Simple Object Access Protocol – SOAP messages. Naturally, it would be interesting to deal with the web services.

References

1. Duane Nickull et al.: Service-Oriented Structure. Whitepaper, Adobe Systems, Inc. http://www.adobe.com
2. Keen Martin et al.: Patterns: Implementation and SOA using year Enterprise Service Drunk. RedBook, IBM®, http://www.ibm.com
3. Mark Andrei et al.: Patterns: Service-Oriented Structures and Web Services, RedBook, IBM®, http://www.ibm.com
4. Hans-Erik Eriksson and Magnus Penker: UML Toolkit, John Wiley & Son' S, Inc, 1998.
5. J.P. Mueller: COM+ Developper' S Guides, Osborne, McGraw-Hill, 2000.
6. Davit S. Platt: Understanding COM+, Microsoft® Press, 1999.
7. Zoran Stojanović: A Method for Component-Based and Service-Oriented Software Systems Engineering. Doctoral Essay, Delft University of Technology, The Netherlands, January 2005, and ISBN: 90-9019100-3.

8. Raoul M. Tsamo: E-Intelligence integrated into the concept of e-marketplace in a distributed environment. Application to an e-procurement system. Master of engineering's end of course thesis in computer science. National Advanced Polytechnic School. Cameroon. 2001.
9. Service Oriented Structures Solution Accelerator Guide, www.bea.com, May 2005
10. Jeffrey R. Shapiro: SQL Server™ 2000: The complete reference, Osborne/McGraw-Hill, 2001.
11. Cesare Pautasso: Message Oriented Middleware (MOM), Computer Department Sciences, Federal Swiss Institute of Technology (ETHZ), http://www.iks.inf.ethz.ch /, April 2005.
12. William Tse: Enterprise Integration Application, www.cs.ucl.ac.uk/staff/W.Emmerich/lectures/3C05-03-04/EAI.pdf, May 2005
13. David Sprott and Lawrence Wilkes: Understanding Service-Oriented Structures, CBDI Forum, January 2004, http://www.cbdi.org/.
14. Distributed Systems, ITEC801. http://www.comp.mq.edu.au/units/itec801/lectures/. June 2005
15. Cyrille MBASSI: Design and implementation of a directed architecture service: News Aggregator Full Service. Master of engineering's end of course thesis in computer science. National Advanced Polytechnic School. Cameroon. 2005.
16. Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS) Point-to-Point (PP) (GSM 03.40). European Telecommunications Standards Institute.

QoS Assignment with GSS for Incoming IP Sessions in UMTS

Carelia Gaxiola-Pacheco[1], Ernesto Quiroz-Morones[1] and Oscar-Iván Lepe-Aldama[2]

[1] Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI-IPN). Tijuana, Baja California, México
[2] Universidad Autónoma de Baja California (UABC). Ensenada, Baja California, México
{cgaxiola, eequiroz}@citedi.mx, oscarlepe@uabc.mx,

Abstract. This article analyzes and characterizes the Generic Service Specification (GSS) framework proposed by Maniatis et. al [3]. GSS is constituted by a generic scheme of specification for the service class and an algorithm for intelligent mapping. The scheme of generic specification allows network administrators to describe the classes of their domain according to GSS. An intelligent mapping algorithm deployed at the point where different networks converge performs the correspondence of the requested session specification to a specific service class when crossing a domain on the end-to-end path. The proposal of Maniatis is extended by applying fuzzy logic for the calculation of the degree of correspondence between two network service specifications.

Keywords: QoS, GSS, UMTS, mapping.

1 Introduction

The existing quasi-ubiquitous Internet, the increasing reliance in data transport over IP networks in preference to other technologies (i.e. circuit switched networks), and the worldwide expansion of the wireless IP based network, has brought about a Quality of Service (QoS) provision dilemma. Actual IP networks are unable to allocate consistent QoS characteristics to a user's session going through diverse administrative domains. Universal Mobile Telecommunications System's (UMTS) classes of service [1], differ from those of the legacy wireline world. Integrated Services (IntServ), Differentiated Services (DiffServ) and Multiprotocol Labeling Switching (MPLS) technologies provide partial solution in the inter-domain realm of QoS provision. The ever-growing types of multimedia applications exert a burden on the capability of the networks to provide the expected performance to users applications.

In view of this panorama, the Internet Engineering Task Force (IETF) started in 2004 the initiative "Next Steps in Signaling" (NSIS) [2], a signaling system for IP networks aimed to provide QoS assurance. Over this foundation, Maniatis, et. al. propose the Generic Service Specification (GSS) framework, consisting of a "universal" template of QoS and traffic attributes, which is passed on from one domain to another, and

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 173-180

Received 07/07/06
Accepted 07/10/06
Final version 11/10/06

interpreted at the edge of the networks by an Intelligent Mapping Algorithm (IMA), in charge of deciding the best correspondence for the incoming session with one of the native classes of service. This paper deals with a GSS framework implementation for incoming inter-domain sessions in UMTS.

2 The Generic Service Specification Schema

The GSS framework defines a GSS schema stating a Network Service Specification, comprised of QoS objectives and traffic classification of a session that needs to traverse two or more administrative domains. The schema is handed on from one domain to the next. With this information the IMA can calculate and decide the best native service correspondence for the incoming session. Fig. 1 shows the Network Service, QoS, and Traffic Specifications of the GSS schema, and the particular parameters that describe them.

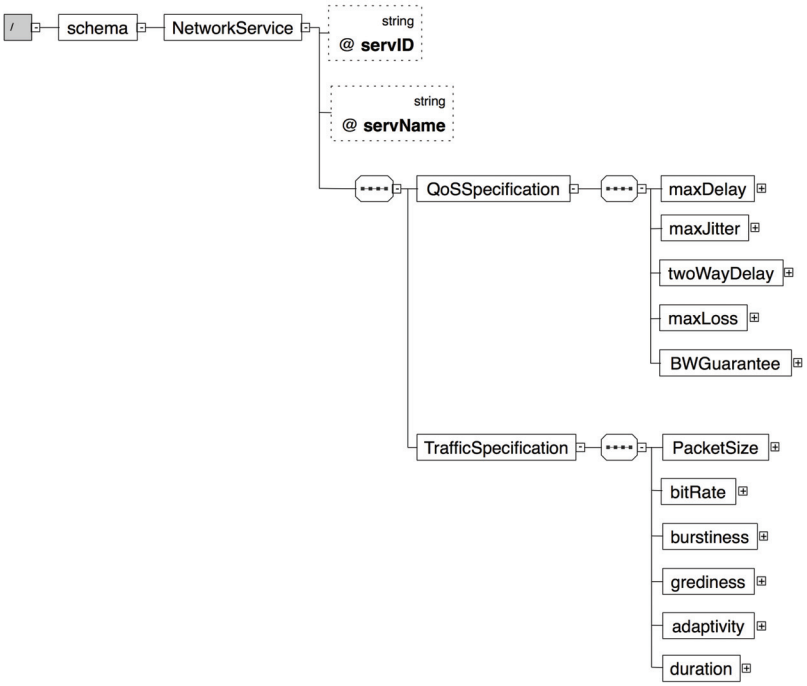


Fig. 1. Logic Diagram of the GSS schema

The QoS specification can have either numeric or descriptive values, when using the latter, the range of each parameter has six levels: veryLow, low, medium, high, veryHigh, and unspecified.

Traffic specification descriptors can adopt the following values: Bit rate (constant or variable {veryLow, low, medium, high, veryHigh, and unspecified }), packet size

(constant or variable {verySmall, small, medium, big, veryBig, and unspecified }), burstiness (veryLow, low, medium, high, veryHigh, and unspecified), greediness (yes, no, unspecified), adaptativity (yes, no, unspecified), and duration (shortLife, longLife, and unspecified). This data allows the receiving domain to effect operations of admission control, policing and resource management.

To convey the relative importance among the eleven unitary components of the network service specification, they are assigned a weight, whose integer values span from zero (not relevant), to ten (capital importance). The weights differentiate applications, i.e. voice needs a weight ≈ 10 in maxDelay, whereas web requires a weight ≈10 on maxLoss.

The GSS schema is structured with the Extensible Markup Language (XML) [4], as suggested by [3].

3 The Intelligent Mapping Algorithm

Since different QoS models specify non-homogeneous classes of service, the IMA at the edge router of a UMTS domain has the purpose of deciding (with a criteria described in this section) which of its native classes is the one that more resembles that of the GSS schema received.

Each class of traffic in the UMTS domain is described in a Network Service Specification by means of a XML document, following the same pattern of the GSS scheme.

The fundamental operation for the selection of a certain network service specification is the calculation of the correspondence degree, which denotes the grade of similarity between two different network service specifications. Absolute or relative conformance can be used to calculate the correspondence degree. In the case of absolute conformance, the attribute's values are compared, if they are identical the weight associated to the attribute is added to the total correspondence degree, otherwise it is subtracted. In the case of relative correspondence, the calculation involves a more complex process, stemming from the fact that numerical values and descriptive values (rank of numerical values, i.e. 200 ms to 600 ms) may exist for the same attribute [3]. Furthermore, the correspondence degree implies an uncertainty factor, because it makes reference to a degree of similarity between two data sets. Fuzzy logic is used to deal with the uncertainty present in the structure of the Network Service Specification data set. The IMA models the QoS specification's parameters (maxDelay, twoWayDelay, maxLoss, maxJitter, bwGuaranteed) as linguistic variables, whose descriptive values (veryLow, low, medium, high, veryHigh, and unspecified) represent a linguistic value. For each linguistic value numerical figures (known as support) must be defined. In this work, the support for each linguistic value is allocated on the basis of the recommendation [6]. The numerical value within each support that holds the greater degree of similarity is known as core. Table 1 gives an example the support and core for maxDelay.

Table 1. Numerical values correspondent to maxDelay

maxDelay	Min. limit	Core	Max. limit
verylow	0 ms	100 ms	300 ms
low	200 ms	400 ms	600 ms
medium	500 ms	1 s	2 s
high	2 s	4 s	6 s
veryhigh	5 s	10 s	5 min

The membership degree takes on values between 0 (non-member) and 1 (identical member) [5]. In order to ascribe a degree of similarity to the discrete values of the support, a Membership Function has to be defined. A Triangular MF (Fig. 2) is invoked by the IMA to compute the membership degree. The MF is expressed by Equation 1, where a < b < c take up the values of recommendation [6], as exemplified in Table 1.

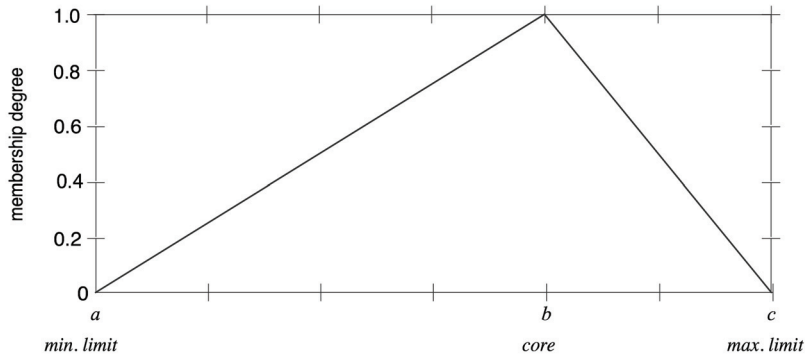


Fig. 2. Triangular membership function [5]

$$\text{triangular}(x;a,b,c) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c < x \end{cases} \quad (1)$$

Upon obtaining the membership degree, the IMA can figure out the relative correspondence. First it calculates the relative weight multiplying the numerical value of the membership degree times the attribute's weight. Then an addition of the relative weights is performed. When the membership degree equals 0, the weight is

subtracted (the whole value of the weight, or a fixed fraction of it). Once the relative correspondence estimate is obtained, the IMA yields the name of the local network service specification that has the greater correspondence degree. Whenever the relative correspondence outcome is negative, Best Effort service should be assigned.

Fig. 3 depicts the action diagram of the IMA.

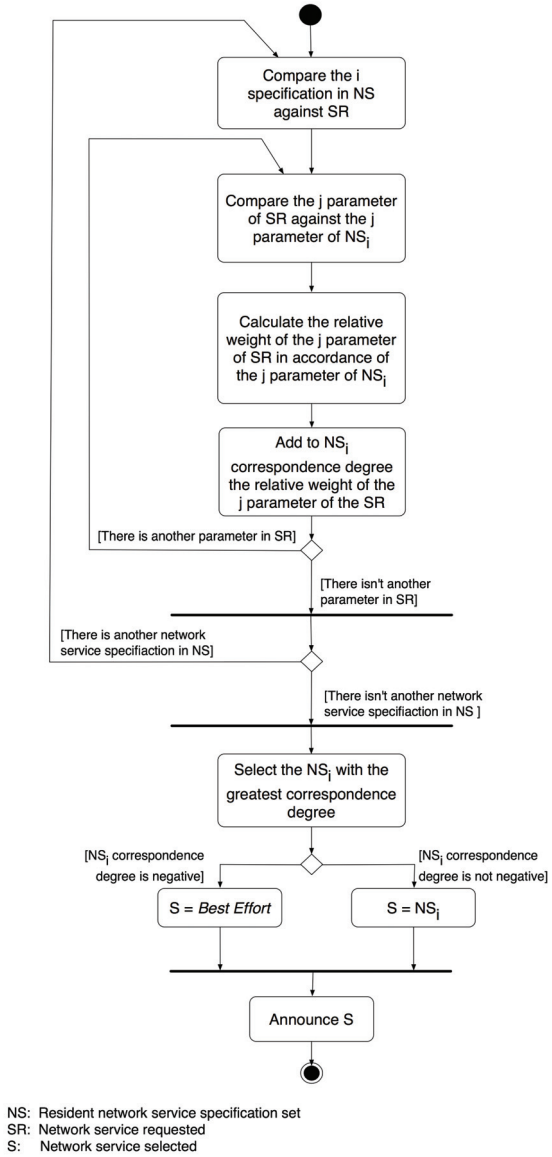


Fig. 3. Intelligent mapping algorithm action diagram

4 GSS Framework Results

To evaluate the GSS framework implementation, an iterative program that generates network service specifications was elaborated and tested with the following conditions: When the membership degree between two values is zero: (a) the parameter's weight is subtracted (b) a fraction of the weight is subtracted. The value and weight of the QoS attributes (maxDelay, maxJitter, maxLoss, twoWayDelay and bwGuarantee) were set according to Table 2. The UMTS domain holds 17 network service specifications. Because of space limitations, a sample of these specifications is enlisted in Table 3.

Table 2. Set of network service specifications attribute's values generated

Attribute	Content	Weight		
	Value	Min. Value	Max. Value	Increment
maxDelay	from 100 to 5000 with increments of 10%	1	10	1
maxJitter	[1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 7.5, 10]	2	10	2
maxLoss	[0, 0.001, 0.01, 0.03, 0.05]	2	10	2
twoWayDelay	unspecified	0	0	0
bwGuarantee	unspecified	0	0	0

Table 3. Network service specifications sample

Application	*maxDelay*			*maxJitter*		
	Descriptive value	Weight	Numerical value	Descriptive value	Weight	Numerical value
1. Voice	veryLow	10	100	veryLow	10	1
2. Video-conference	veryLow	10	100	veryLow	10	1
3. Interactive games	low	10	250	unspecified	0	
4. Telemetry/two way	low	10	250	unspecified	0	

Upon receiving a service request, the IMA selects a network service specification from this set of alternatives. For negative correspondence, though best effort is suggested, the algorithm yields the network service specification whose correspondence degree associate is equal to the maximum calculated.

Trials were run for 0%, 25%, 50% and 100% of the attribute's weight being subtracted for negative correspondence. In this paper we exhibit results for the two latter instances. Table 4 displays the overall results for the case when 50% is subtracted.

Table 4. Test 1: 50% of the weight is subtracted

A network service specification was identified as adequate for the requested service	The use of the *BestEffort* class was recommended for the requested service
330,675 [57.26%]	246,825 [42.74%]

Table 5 shows a set (randomly chosen) of the results obtained.

Table 5. Service specification outcomes (50%)

maxDelay		*maxJitter*		*maxLoss*		Correspondence degree	Specification selected
ms	Weight	ms	Weight	FER	Weight		
146.41 ms	2	1	8	0.010	6	12.535	2
235.79	5	5	8	0.001	10	-0.420	5
459.50	3	10	10	0.030	6	-0.500	1
895.43	10	5	6	0.030	2	-6.000	1
1311.00	4	2	2	0.000	6	3.000	5, 4, 17
1919.43	1	4	8	0.000	6	1.500	5, 14, 17

ms: miliseconds FER: frame error rate

Table 6 shows overall results for the 100% case.

Table 6. Test 2: 100% of the weight is subtracted

A network service specification was identified as adequate for the requested service	The use of the *BestEffort* class was recommended for the requested service
170,465 [29.52%]	407,035 [70.48%]

In the Table 7 is a set of the results obtained during the test 2 of the algorithm

Table 7. Service specification outcomes (100%)

maxDelay		*maxJitter*		*maxLoss*		Correspondence degree	Specification selected
ms	Weight	ms	Weight	FER	Weight		
146.41	2	1	8	0.010	6	9.535	2
235.79	5	5	8	0.001	10	-4.420	5
459.50	3	10	10	0.030	6	-7.000	1
895.43	10	5	6	0.030	2	-14.000	1
1311.00	4	2	2	0.000	6	0.000	5, 4, 17
1919.43	1	4	8	0.000	6	1.500	5, 14, 17

By comparison of the outcomes in Tables 4 and 6, it can be seen that subtracting the whole weight narrows the number of correspondence instances with respect to subtracting 50% of the weight. The decision on which one to apply lies on the system administrator, since on the one hand the figures indicate that the first case is more "accurate" or selective, but on the other hand, that sessions which could be given a set of QoS characteristics will be either rejected or assigned best effort service.

5 Conclusions

End-to-end QoS assurance without degradation in an interdomain scenario is still an unattained goal for IP networks. The IETF has ongoing work to define the NSIS, with the aim of providing a respectable solution to this goal.

In this paper we describe a GSS and IMA implementation, which in the context of the NSIS, yields QoS conformance for IP sessions coming from any administrative domain into a UMTS mobile network.

The IMA defines fuzzy logic sets out of each network service specification's attribute, to compute an overall relative correspondence benchmark, to search for the local service with the greater likeness to the specified by the GSS.

Results show that in many cases a one to one correspondence is achieved, nevertheless, there are frequent multiple selections, because the relative correspondence degree obtained from overall weight calculation can be met by various local services. These results lead to the need of further research on the definition of the fuzzy group's membership function applied and the weight manipulation criteria, for the purpose of obtaining more convergence in the local service selection

References

1. C. Chioariu, "QoS in UMTS," Helsinki University of Technology Seminar on Internetworking, 2004.
2. Next Steps in Signaling (NSIS) charter, http://www.ietf.org/html.charters/nsis-charter.html
3. Sotiris I. Maniatis, Eugenia G. Nikolouzou, Iakovos S. Venieris, "End-to-end QoS specification issues in the converged all-IP wired and wireless enviroment", IEEE Communications Magazine, vol. 42, no. 6, pp. 80-86, Jun 2004.
4. 4 Extensible Markup Language (XML), http://www.w3.org/XML
5. Jyh-Shing, Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Pearson Education, 1st edition September 26, 1996
6. 3GPP, ''Service Aspects; Services and Service Capabilities (Release 6)'', 3GPP TS 22.105, version 6.2.0, June 2003.

A New NBEMV-AODV Energy Aware Routing Protocol in Mobile Ad-Hoc Networks

Jin-Man Kim and Jong-Wook Jang

Dept. of Computer Engineering, Dong-Eui University,
995 Eomgwangno, Busanjin-gu, Busan, Korea
iricejm@nate.com, jwjang@deu.ac.kr

Abstract. The mobile ad-hoc network consists of nodes driven by the battery with a limited life time. So, studies in relation to minimization of the total amount of energy consumed in the network are deemed to be mandatory. Until now, there haven't been sufficient studies on routing algorithms that can be specifically applied to the environment of the ad-hoc network even if many studies have focused on energy-efficient routing algorithms. Therefore, in this paper, we propose the online energy aware routing algorithm that is most suitable for the ad-hoc environment. The proposed algorithm is an improved version of the conventional AODV algorithm and it is referred to as the NBEMV-AODV (Neighborhood Energy Mean Value–AODV). In this algorithm, the overall network lifetime is significantly extended using the energy mean values of the neighboring nodes. The simulation results indicate that our energy-aware routing protocol achieves significant improvements in network lifetime by compared with existing AODV by using NS-2.

1 Introduction

A mobile ad-hoc network (MANET) is composed of a group of mobile, wireless nodes which cooperate in forwarding packets in a multi-hop fashion without any centralized administration. In MANET, each mobile node acts as both a router and an end node which is a source or destination, thus the failure of some nodes' operation can greatly impede performance of the network and even affect the basic availability of the network, i.e., routing, energy depletion of nodes has been one of the main threats to the availability of MANET. Since the mobile nodes have limited battery power, it is very important to use energy efficiently in MANET.

Ad hoc wireless network has applications in emergency search-and-rescue operations, decision making in the battlefield, data acquisition operations in hostile terrain, etc. It is featured by dynamic topology (infrastructureless), multihop communication, limited resources (bandwidth, CPU, battery, etc.) and limited security. These characteristics put special challenges in routing protocol design [1]. The one of the most important objectives of MANET routing protocol is to maximize energy efficiency, since nodes in MANET depend on limited energy resources. Several routing protocols for MANET's have been suggested in late 90's: DSR, AODV, DSDV, TORA and others (see [2] for comprehensive review of these

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 181-190

Received 03/07/06
Accepted 04/10/06
Final version 20/10/06

protocols). The classical MANET settings assume that neither node locations nor relative locations of other nodes are available. In this paper, we consider only protocols which do not rely on location knowledge – even if each node is supplied with GPS, the node mobility implies significant communication overhead caused by location updates. The primary objectives of MANET routing protocols are to maximize network throughput, to maximize energy efficiency, maximize network lifetime, and to minimize delay. The network throughput is usually measured by packet delivery ratio while the most significant contribution to energy consumption is measured by routing overhead which is the number or size of routing control packets. The general consensus based on simulations (e.g., in the network simulator NS2 [3]) is that reactive protocols, i.e., those finding routes on fly by request with no work in advance, perform better than proactive routing protocols, which try to maintain the routs for all source-destination pairs (see [4]). In hop-by-hop reactive routing protocols (e.g. used in Ad-hoc On-demand Distance Vector routing (AODV) [2]), every intermediate node decides where the routed packet should be forwarded next. Rout requests are generated at each hop by local broadcasting in case of path discovery. A simple flooding broadcast for route requests generates a considerable redundant packet overhead which is a major cause of inefficiency of MANET routing protocols.

In this paper, we propose an improvement of AODV routing protocol of node energy-awareness. With a simple modification in applying the Energy Mean Value algorithm of AODV, our proposed routing protocols can achieve better load balancing among nodes, which results in a longer network lifetime.

The rest of the paper is organized as follows. In Section 2, we discuss to On-demand routing protocols in MANETs. Section 3 explains a proposed NBEMV-AODV which applies Neighborhood Energy Mean Value algorithm. Section 4 includes performance evaluation results of NBEMV-AODV compared with existing AODV using NS-2 (Network Simulator 2). Section 5 concludes the paper.

2 Description of AODV

In this section, we start with a general review of On-demand routing protocol, Ad-hoc On-demand Distance Vector (AODV), in MANETs. This protocol initiate route discovery only when a route is needed and maintain active routes only while they are in use. Unused routes are deleted.

AODV is based on distance vector algorithm. However, AODV, unlike other proactive distance vector algorithms, does not use periodic or triggered updates to disseminate routing information. AODV requests for a route only when needed and does not require nodes to maintain routes to the destinations that are not actively used in communications. Routing tables are built using route discovery techniques and maintained using Route maintenance, which are described as follows.

2.1 Route Discovery

When a node needs to send a Packet to a destination to which it does not have a routing entry. It broadcasts a route request (RREQ) packet. To prevent unnecessary broadcasts of RREQs the source node uses an expanding ring search technique as an optimization. In an expanding ring search, the source node initially uses a time-to-live (TTL) = TTL_Start in the RREQ packet IP header and sets a timeout for receiving a reply (RREP). Upon timeout the source retransmits a RREQ with TTL incremented by TTL_increment. This continues until TTL reaches the highest value. Source will retransmit the RREQ with the highest TTL if it does not receive any reply within time-out period. Maximum number of retransmission must be determined. Nodes receiving RREQs setup reverse paths to sources of RREQs in their routing tables, and either reply to the RREQ if they already have an entry for the destination or forward the RREQ. If there isn't any intermediate node, having a route to the destination, finally the destination will reply. Nodes receiving RREPs setup paths to the destinations and in this way desirable routes will be discovered.

2.2 Route Maintenance

An existing routing entry may be invalidated if it is unused within a specified time interval, or the next hop node is no longer reachable. In that case, the invalidation is propagated to neighbors that have used this node as their next hop. Each time a route is used to forward a data packet. Its route expiry time is updated when a node detects that a route to a neighbor is no longer valid. It will remove the invalid entry and send a route error message to the neighbors that are using the route. The nodes receiving route errors will repeat this procedure. Finally source will request a new route if it still needs the route.

3 NBEMV-AODV Routing Protocol by Applying an Neighborhood Energy Mean Value Algorithm

The NBEMV-AODV algorithm that we propose is a protocol based on the AODV, an on-demand routing protocol and it maximizes the overall network lifetime by allowing the network to effectively consume the limited battery of the moving nodes, which is one of the main limitations in the ad-hoc network. The NBEMV-AODV protocol is a very realistic online energy aware algorithm that is suitable for many applications in the actual ad-hoc network and it extends the network lifetime to its maximal permissible duration by using the energy mean values of the neighboring nodes and allowing the network to consume its energy more efficiently. The existing AODV protocol does not consider the energy status of each node but it rather sets the route according to the basic AODV route setting method. When a certain node has a higher rate of node participation, it will participate in many nodes and the battery energy will be consumed at a rapid pace unless it carefully considers how much energy it has. Eventually, the node won't be able to participate in the network.

In order to extend the network lifetime by preventing the energy from concentrating on a certain node and distributing it efficiently over the network, the status of energy of the network should be considered as well as the status of energy of each node. In the NBEMV algorithm that we propose is basically inspired by the fact that the communication is achieved between the neighboring nodes. In the route setting method of the AODV protocol, the RREQ message uses the status of energy of the node and the energy mean values of the neighboring nodes in order to delay the RREQ flooding. Consequently, the possibility for the node, whose status of energy is vulnerable, to participate in the route is greatly reduced through the method such that it eventually leads to the maximized network lifetime.

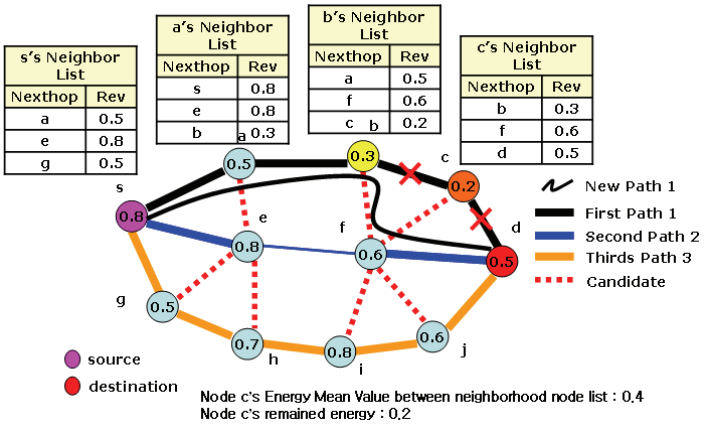


Fig. 1. NBEMV-AODVs Path Calculate and each node one-hop neighbor list

3.1 Operation of the Proposed AODV

In an existing AODV, if a source node is going to communicate with a destination node, the source node broadcasts RREQ messages first and a path is established by the RREP reply to RREQ arrive first.

Our proposed method is based on the fact that the actual communication is achieved between the neighboring nodes. We improved the two aspects of the existing AODV protocol in order to extend the entire network life.

First, the existing AODV protocol does not manage the list for the neighboring nodes. In addition, it doesn't manage the information of their energies either. Hence, we modified the existing AODV protocol and allow it to have the neighboring node list and manage the information on energy. The RREQ message is made available, so that each node can manage its neighboring node list. When a node broadcasts the RREQ message to the network for the purpose of route discovery for the data transmission, the RREQ messages includes the status of its energy in the 11 bits of the reserved fi eld. When the RREQ message being broadcasted to the network arrives at the intermediate node, that is, the neighboring node, the intermediate node browses its

neighboring node list and determines whether the message originates from the neighboring node. If it is not listed in the neighboring node list then it will be newly added to the list. Otherwise, the list will not be updated. The neighboring node list and the energy information accompanying it are maintained for the duration specified by ACTIVE_ROUTE_TIMEOUT and the neighboring node list will be updated with the new RREQ message obtained from the route discovery of other source nodes.

Second, the Energy Mean Values, E(mv) for the current node and the neighboring nodes are obtained using the energy information of the neighboring node described above.

Energy Mean Value process can be expressed in a numerical formula as follows.

- Each node remained energy : $Ne\_R_{(ev)}$
- The number of neighbor nodes : $Neighbors_{(cnt)}$
- The neighbors node Energy Mean Value : $E_{(mv)}$

$$E_{(mv)} = ((\sum_{i=1}^{Neighbors_{(cnt)}} Ne\_R_{(ev)}) + Self_{(ev)}) / (Neighbors_{(cnt)} + 1)$$

The Energy Mean Values obtained as above are compared with the remaining energy in the node when the node is searching for another route after the initial route discovery. If it is below the mean value then the RREQ message is delayed by 2.0ms. If it's above the mean value, then the RREQ message is delayed by 0.01ms. This reduces the possibility for a certain node to participate in the route if it has a lower energy than the neighboring node. Therefore, the neighboring node with a relatively large energy participates in the route so that the overall network life can be extended.

Fig. 1 illustrates an example of the route discovery by the proposed NBEMV-AODV protocol that uses the neighboring node list and compares the energy mean values of the neighboring nodes. The remaining energy of the node c is 0.2 and the energy mean value for the nodes in the neighboring node list including itself is 0.4. So, when the RREQ message passes by the node b and finally arrives at the node c, the time delay is set to 2.0ms such that the possibility for it to participate in the subsequent route is reduced. In other words, the possibility for the route participation of a node with a relatively less energy is reduced. So, the energy consumption is distributed over the network by allowing the node with a relatively larger energy to participate in the route. As a result, the overall network life is extended significantly.

4 Performance Evaluation

In this section, we evaluated NBEMV-AODV routing protocols by comparing its performance with existing AODV and New-AODV[5]. We implemented our improved AODV protocol using NS-2, which is a discrete event-driven network simulator with support for various mobile ad-hoc network routing algorithms. We used a simulation model, an ad-hoc network model using the CMU (Carnegie Melon University) [6] of Monarch Research Group for NS-2. The used Channel is a Wireless channel/Wireless Physical, Propagation Model is Two-ray Ground reflection Model, Queuing Model is Drop-Tail/Priority Queue and MAC protocol which is 802.11. The

radio network interface card (NIC) model includes collisions, propagation delay and signal attenuation with a 2Mbps data rate and a radio range of 250 meters.

4.1 Simulation

In our simulation model, a network with 49 mobile nodes is placed in a rectangular region size of 800 x 800 unit^2 where 49 nodes are uniformly distributed in the region. The initial energy of all the nodes is 10J. The transmission power is 600mW and the receiving power is 300mW. During the simulation, there are 100 CBR (constant Bit Rate) traffic streams between the source and destination pairs.

We evaluated the performance of our proposed NBEMV-AODV protocol (using Neighborhood nodes Energy Mean Value) by comparing the traditional AODV (Existing-AODV) and New-AODV (using Entire Network Energy Mean Value).

We analyzed the two main areas of performance of the proposed scheme through our simulation model. First, the number of nodes fully consuming its energy is analyzed. Second, the number of packets received by the entire receiving nodes is analyzed. The proposed NBEMV-AODV is compared with the existing AODV and New-AODV protocols. The New-AODV protocol is the result from preceding studies and it is a type of AODV protocol where the network life is extended by using the energy mean value of the entire network. The New-AODV shows an ideal performance in obtaining the energy mean value of the entire network under a specific constraint. Nevertheless, it is a very unrealistic and difficult matter to obtain the energy mean value for the entire network in reality. Thus, we proposed the NBEMV-AODV that is applicable to the actual ad-hoc networks and comparatively analyzed its performance vs. the existing protocols.

4.1.1 Simulation Model

The traffic model consists of five links, <0, 48>, <3, 45>, <6, 42>, <27, 21> and <43, 4>. it means that <"source","destination>. The formally AODV protocol's RREQ messages delay time of NS-2 is determine to the formula "0.01 * Random::uniform()", that is not fixed specifically.

We used three kinds of simulation models. The models are as follows.

First, we applied the simulation model, which was mentioned above, leaving the existing-AODV protocol unchanged and counted (for 100second) the number of nodes that consumed all energy in the entire network.

Second, we applied the simulation model, which was using Entire Network Energy Mean Value (New-AODV[5]. It compared energy states of each node with the energy state of the entire network, taking into consideration the Energy Mean value. If a node's energy state is higher than that of the entire network, then the RREQ delay time is set at 0.01ms. And if a node's energy state is lower than that of the entire network, then the RREQ delay time is set at 2.0ms.

Third, we applied the simulation model, which was our proposed NBEMV-AODV (using Neighbor Energy Mean Value). It compared energy states of each node with the energy state of the entire network, taking into consideration the Energy Mean value. If a node's energy state is higher than that of the entire network, then the

RREQ delay time is set at 0.01ms. And if a node's energy state is lower than that of the entire network, then the RREQ delay time is set at 2.0ms.

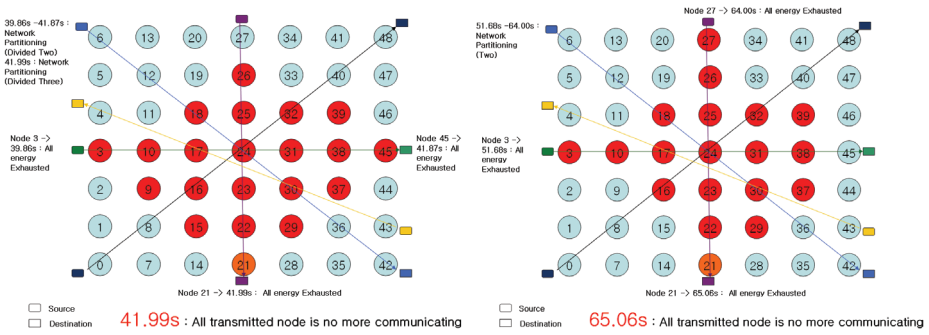
4.2 Simulation Results

Table 1 represents the nodes that consumed its full energy as time elapses.

As shown in Table 1, the existing AODV and New-AODV protocols and the NBEMV-AODV protocols become unable to communicate in 41.9907 seconds, 65.0685 seconds and 61.03859 seconds, respectively.

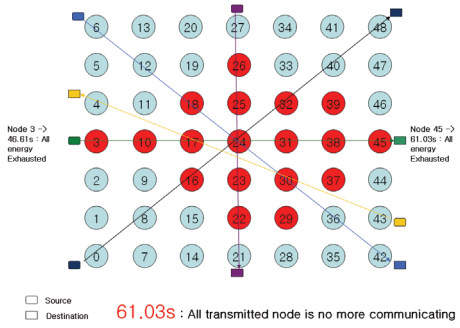
Fig. 3 is a graphical representation of these data. For our further analysis, we analyzed in which ways the nodes fully consume its power in the network.

Fig. 2 is a graphical representation of each case. In case of the existing AODV protocol in Fig. 2(a), the node 3 and the node 45 consume all of their energies in 39.986 seconds and 41.87 seconds, respectively. As a result, the network is divided in two and the source nodes can not transmit the data anymore.



(a) Existing-AODV

(b) New-AODV(using Entire Network Energy Mean Value)



(c) NBEMV-AODV(using Neighborhood node Energy Mean Value)

Fig. 2. Simulation results for different ways of the node consuming all of the energy

Fig. 2(b) corresponds to the case of New-AODV protocol. The node 3 and the node 27 consume all of their energies in 51.68 seconds and 64.00 seconds, respectively. The network is divided in two as well in this case. However, unlike the existing AODV protocol, several source nodes are still able to communicate. When the node 21 consumes its energy in 65.06 seconds, the rest of the source nodes cannot transmit the data anymore.

Table1. The number of nodes consuming all of the energy vs. time

Existing AODV			New-AODV (Using Entire Network Energy Mean Value)			NBEMV-AODV (Using Neighborhood Nodes Energy Mean Value)		
Time	Node	count	Time	Node	count	Time	Node	count
35.61907	24	1	48.01164	24	1	44.1379	24	1
36.36335	23	2	50.08048	17	2	46.37262	23	2
.....	.....	.....	.....	.....	.....	.....	.....	.....
39.80572	38	14	63.23194	3	14	57.93274	39	14
39.86968	3	15	63.98093	37	15	58.78723	3	15
40.36544	26	16	64.00492	27	16	58.84867	37	16
40.46846	37	17	64.68894	39	17	59.16575	29	17
40.6181	29	18	65.02778	29	18	61.03859	45	18
40.73804	39	19	65.06925	21	19			
41.87853	45	20						
41.99076	21	21						

Fig. 2(c) corresponds to the NBEMV-AODV protocol proposed by us. In this case, the node 3 and node 45 consume all of their energies in 46.61 seconds and 61.03 seconds. Consequently, the network is divided in two and none of the source nodes can transmit the data

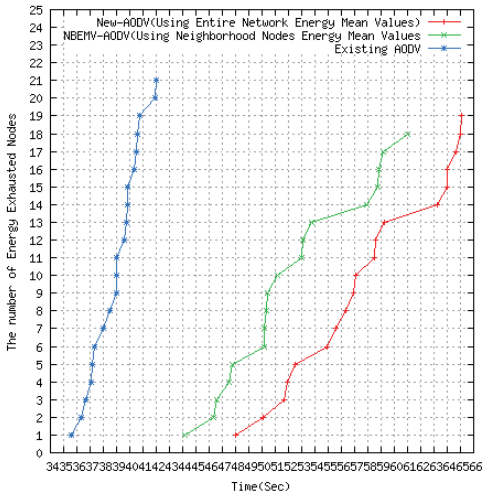


Fig. 3. Network life comparison of the NBEMV-AODV protocol, the existing AODV and New-AODV protocols

To summarize, in the existing AODV protocol, a total of 21 nodes consume all of their energies in 41.99 seconds. As a result, the network is divided in three and becomes unable to communicate. In the New-AODV protocol, 19 nodes consume all

of their energies in 65.06 seconds. As a result, the network is divided in three and becomes unable to communicate. In the NBEM-AODV protocol, 19 nodes consume all of their energies in 61.06 seconds. As a result, the network is divided in two and becomes unable to communicate

The following is an analysis of the packet transmitted in each type of AODV protocol.

Table 2. The number of packets received by the entire receiving nodes

Link : (0,48) (3,45) (6,42) (27,21) (43,4)			
Destination	**existing-AODV**	**New-AODV (Using Entire Network Energy Mean Value)**	**NBEMV-AODV (Using Neighborhood Nodes Energy Mean Value)**
48	386	156	250
45	747	1012	1050
42	69	96	212
21	760	878	920
4	633	832	541
Total	2595	2974	2973

Table 2 represents the total number of received packets. According to this table, the total number of the received packets in the existing AODV protocol, the New AODV protocol and the NBEM-AODV protocol are 2595, 2974 and 2973, respectively

5 Conclusions

We propose a NBEMV-AODV protocol which extends the entire network lifetime in an Ad-hoc network environment through NS-2. Our study does not state that the New-AODV is necessarily the best solution. Since the New-AODV uses the energy mean value for the entire network, it is quite impractical to apply it to an actual network environment. Therefore, based on the fact that the actual communication is achieved between the neighboring nodes in the network, we proposed the NBEMV-AODV that is applicable to an actual network environment. From our simulation result, it is clear that the NBEMV-AODV protocol is suitable for applications in an actual network environment and much more energy-efficient than the existing AODV protocol as well as its performance is almost as good as the New-AODV protocol.

Acknowledgements. This research was supported by the Brain Busan 21 Project and the Korea Industrial Technology Foundation in 2006.

References

1. P. Sinha, R. Sivakumar, and V. Bharghavan, “Cedar: Core extraction distributed ad hoc routing,” in Proc. of IEEE INFOCOM, 1999.
2. C. Perkins, Ed., Ad Hoc Networking. Addison-Wesley, 2001.

3. K. Fall and K. Varadhan, "The vint project, uc berkeley, lbl, usc/isi, and xerox parc," 1997. [Online]. Available:http://www-mash.cs.berkeley.edu/ns/
4. J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom98), 1998.
5. Jin-Man Kim, Jong-Wook Jang, "AODV based Energy Efficient Routing Protocol for Maximum Lifetime in MANET," in Proc, of the AICT/IEEE International conference, 2006.
6. The CMU Monarch Project, "The CMU Monarch Project's Wireless and Mobility Extensions to ns," Carnegie Mellon University, Aug, 1999.

FPGA Based Efficient Interface Model for Scale-Free Computer Network using I2C Protocol

P.Venkateswaran, Arindam Sanyal, Snehasish Das, S.K.Sanyal and R.Nandi

Electronics & Tele-Communication Department, Jadavpur University, kolkata-700 032 INDIA
pvwn@yahoo.co.in, arindam_3110@yahoo.co.in, snehasishetce@yahoo.co.in,
s_sanyal@ieee.org, robnon@ieee.org

Abstract. Devices communicating with each other over a serial bus must have some protocol to avoid data loss, as well as enabling faster devices to communicate with slower ones. In a network, there may be several electronic modules across which a communication link has to be established. There must be some algorithm to solve conflicts between the contending nodes in the network. In this design, we have implemented an efficient network interface model which can prevent data loss due to collisions. The proposed interface model follows the I2C protocol with a slight change in the manner in which the data transfer is initiated or terminated. The I2C bus is a true multimaster bus which includes arbitration safeguards against data collisions. The interface model described can be used to connect any number of devices to a network.

1 Intoduction

The I^2C bus has become the de-facto world standard that is now used in different Integrated Circuits(ICs). The advantages of using I^2C are numerous and by employing the I^2C protocol in a design, much of the auxiliary support circuitry such as address decoders and standard logic gates needed for communications can be eliminated. In an I^2C bus there is no central server to resolve the data conflicts. The collisions are prevented using the wired-and configurations of the Serial DAta (SDA) line and the Serial CLock (SCL) line, and the data loss is prevented by the fact that every byte on the SDA line has to be followed by an acknowledge. The important I^2C bus specifications[1] are described in Section 2. The network interface design presented in this paper can be used to interconnect any number of devices on a network efficiently as long as the total capacitance limit is not exceeded. The model can be a master or a slave or a combination of both. The model can be used as an initiator to start/stop all possible data transfers. The model can be used as target (slave) device which detects start/stop conditions and perform data transfer according to the master's request. The model is capable of handling traffic and detecting faults, if any, on the bus. Moreover, if data transfer is to be done intermittently, the chip can be used in parallel for different applications. The implementation[2] methods are given in Section 3 and the performance results of the proposed model using simulation is given in Section 4.

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 191-198

Received 07/07/06
Accepted 03/10/06
Final version 12/10/06

2 I2C Bus Specification

The I^2C bus is built around a two-wire serial bus, SDA (serial data) and SCL (serial clock). Each device is recognized by a unique address, and can operate either as a transmitter or as a receiver. The I^2C master is the device that initiates a transfer and generates the clock for the same. Any device addressed by the master is the slave. If more than one master attempts to transmit at the same time, there will be a conflict. The I^2C specification resolves this conflict by its arbitration process. Before explaining the arbitration process, the basic I^2C characteristics is given in Section 2.1.

2.1 I2C Characteristics

The SDA and SCL lines are bi-directional lines connected to a positive voltage supply through a pull-up resistor. The bus is free when these lines are high. The data on the SDA line is valid only when the SCL line is low. During data transfer, the master generates the START and STOP conditions, which are unique conditions. A high to low transition on the SDA line while the SCL line is high, indicate the START condition, while a low to high transition on the SDA line while the SCL line is high, indicate the STOP condition (Fig. 1). The START and STOP conditions are always generated by the master. The bus is considered to be busy if a START condition is generated.

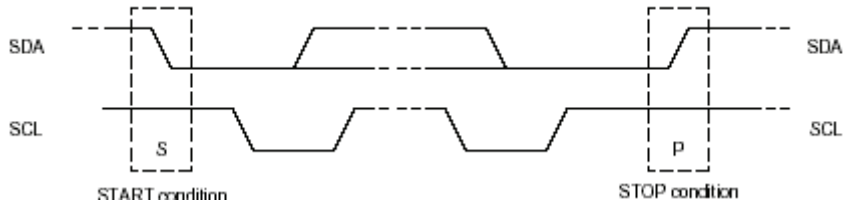


Fig. 1. START and STOP condition

2.2 Acknowledge

Every data put on the SDA line is 8-bits long. Each byte has to be followed by an acknowledge bit (Fig.2).

After transferring each byte, the master frees the SDA line for the slave to send the acknowledge bit. The slave sends the ACK by pulling the SDA line low. The master reads the SDA line in the next clock pulse and senses the acknowledge. If the slave cannot accept the data due to some reason, it sends a NAK (No-AcKnowledge) by leaving the SDA line high. The master sends the NAK and can either stop the transfer or initiate a restart

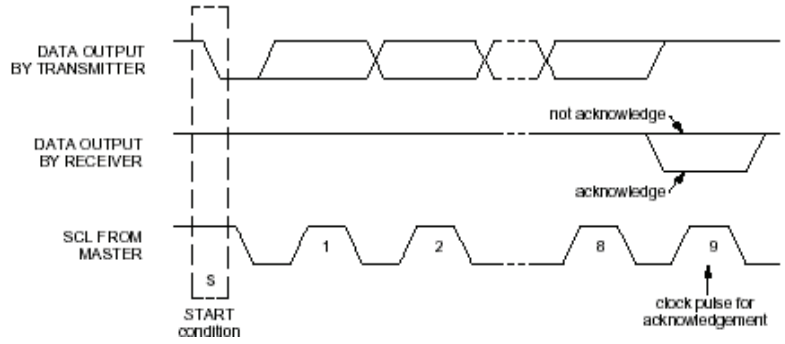


Fig. 2. Acknowledge on the I^2C bus

2.3 Clock Synchronization

The SCL lines of the I^2C bus have wired-and connection. This plays an important role in clock synchronization. The wired-and connection implies that the I^2C device with the lowest low period will hold the SCL line low. If the slave is busy to service an internal request and needs more time to respond to the master's request, it can pull the SCL low. As long as the SCL is held low by the slave, the master cannot bring it to the high state. This is known as 'clock stretching'.

2.4 Arbitration

The SDA lines also have wired-and configurations like the SCL lines. This fact is used in arbitration of the I^2C bus. A master may transfer data only if the bus is free. Now two or more masters may attempt to transfer data at the same time by initiating a START condition simultaneously. Arbitration takes place on the SDA line while the SCL line is at the high level such that the master transmitting a high while another master is transmitting a low, will lose the control of the bus. Arbitration may continue for several bits. Its first stage is the comparison of the address bits. If both the masters are trying to address the same slave, then the arbitration will continue into the data bits till one of the masters attempt to transmit a high while another is transmitting a low. The losing master will then release its SDA line and will revert to the slave mode if it is being addressed by the winning master. The winning master's address and data are the only valid items on the I^2C bus and nothing is lost in the arbitration process.

2.5 Addressing Format

The multiple devices on the I^2C bus can be differentiated using their addresses. The I^2C devices can be addressed using a 7-bit addressing or a 10-bit addressing. The 7-bit addressing format is described as we have used this format in our network inter-

face model. The first byte sent on the I^2C bus after the start is usually an address byte. One exception involves sending a "general call" address following the start condition. The "general call" addresses everyone on the I^2C bus. Any device not wishing to listen to the "general call" can do so by not sending an acknowledge to the master. The addressing scheme is shown in Fig. 3.

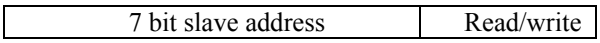


Fig. 3. The first byte after START

The bits 7 through 1 of the address byte carry the I^2C address information and the last bit, bit 0 determines if the I^2C operation will be a read or write. A zero in bit 0 tells the slave that the master will be writing data to the slave device, and conversely a 1 in the lsb (least significant bit) tells the slave that the master will read information from the slave. Only the device that contains a match for the first seven bits will ultimately respond to the master

3 Implementation of the Proposed Interface

The proposed network interface consists of a master section and a slave section(Fig. 4). The internal signals have not been shown for the sake of simplicity. The details of the master and slave blocks are shown in figs. 5 and 6 respectively. The device can function as a master or a slave depending on the stimuli provided. In this design, we have modified the I^2C protocol by changing the generation of START and STOP conditions. In the I^2C protocol the START condition is denoted by the master pulling down the SDA line while the SCL is high. Similarly, a master signals end of transmission by taking back the SDA to high state while the SCL is high. We have made the interface respond only to the rising and falling edges of the SCL[3]. Checking for the START or STOP conditions involves continuously monitoring the SDA line for changes while the SCL is high. This implies that the interface will be busy for half of the clock period doing practically nothing. So we have used an alternate method to utilize the high state of the clock in performing data shifting operations between the user and the interfacing device so that data can be transferred between the communicating ICs with minimum of delay. The data initiation and termination signals are instead generated and conveyed to all the ICs in the network through a common BUS line. The interface is a three wire device consisting of the SDA, SCL and the BUS lines. The state of the bus at any time is reflected by the BUS line. If a transfer is in progress, the BUS line is at a low state to signify that the bus is busy. The master senses the state of the I^2C bus through a signal called BUS. All the BUS lines of the devices have a wired-and configuration like the SDA and SCL lines. The I^2C bus is free if the BUS is at a high state. Any master attempting to transmit, first senses whether the BUS is at a high state. If the BUS is high, the master immediately captures the I^2C bus by pulling down the BUS line to a low state. If two masters attempt to capture the bus at the same time, then the winning master is decided by following the I^2C arbitration logic. The START and STOP signal generation and detection can be performed in another way which follows the I^2C specification. This involves gen-

eration of another clock signal which is delayed by a fixed time period from the SCL signal. This delayed clock signal is locally generated by all the devices connected to the network from the SCL signal which is clocked by the master. Then instead of having to continuously monitor the SDA line for changes when the SCL is high, the SDA has to be checked for a change only after a fixed time from the rising edge of the SCL. But this method involves generation of very accurate delays and it also has to take the propagation delay of the SDA and SCL signals into consideration. From our experiments we have seen that delay in SDA is usually not equal to the delay in the SCL line and hence this second method requires greater coding effort. The unequal delays are also to be expected as the SDA changes only a certain time after rising or falling edges of the SCL. The master and slave are further divided into smaller functional units. The details of the master and slave blocks and their functioning are described in Section 3.1

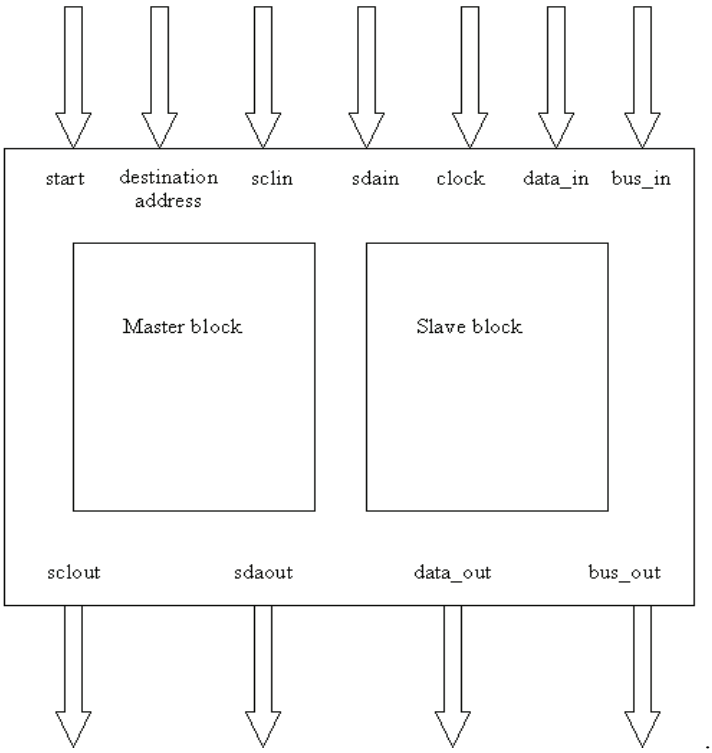


Fig. 4. Functional block diagram of the network interface

3.1 Master Block

The block diagram (Fig. 5) below shows the details of the master block.

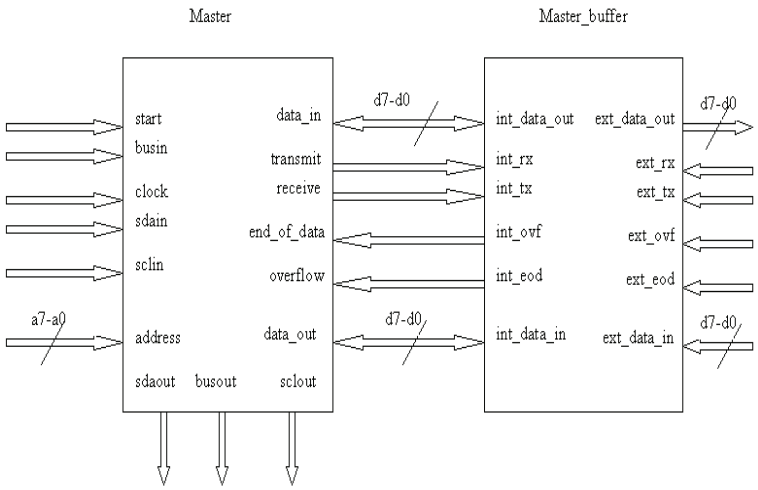


Fig. 5. Functional block diagram of the master

The master block has its own buffer where data to transmitted can be stored by the user(external data) or the data received from the slave can be stored by the master (internal data). The buffer has a fixed capacity, and if the master or the external user tries to send more data to the buffer while it is full, the buffer will generate overflow signal(ext_ovf or int_ovf). Similarly, the buffer will generate the end_of_data signal(ext_eod or int_eod) if any attempt is made to retrieve data from the buffer while it is empty. The master has five main functional units: a) Initiator b) Address Block c) Write Block d**)** Read Block e) Clock generator**.** The functioning of the master block is given below:

Step 1: When the master has to perform any data transfer, the Initiator senses whether the bus is free. If the bus is free, it pulls the BUS line to a low and enables the clock generator which generates the clock for data transfer, and the Address Block.

Step 2: The Address Block sends each bit of the address byte on the SDA line on falling edges of the SCL and also checks whether the data on the SDA line is what it has sent. If the data on the SDA line does not match with the address bits, the Address Block senses that it has lost control of the bus to another master and sends the signal "Control-add" to the Initiator which then frees its SDA, SCL and BUS lines. After completion of address sending, the Address Block frees the SDA line for the slave to send acknowledge (ACK). If the slave sends an ACK, then it sends a signal "over-add" to the Initiator which then enables the read or write block. If the slave sends a NAK, it sends a signal "error-add" to the Initiator which then terminates the transfer.

Step 3: The Write Block transfers data to the slave on the SDA line at falling edges of the SCL. It also checks whether the data on the SDA line is the same as the data it has sent, and waits for the ACK from the slave after transfer of each byte. If there is a NAK from the slave, it reports to the Initiator which then terminates the transfer.

Step 4: The Read Block reads data from the slave at falling edges of the SCL. After reception of each byte, it sends an ACK to the slave or NAK if it wishes to terminate the transfer.

3.2 Slave Block

The block diagram (Fig. 6) below shows the details of the slave block

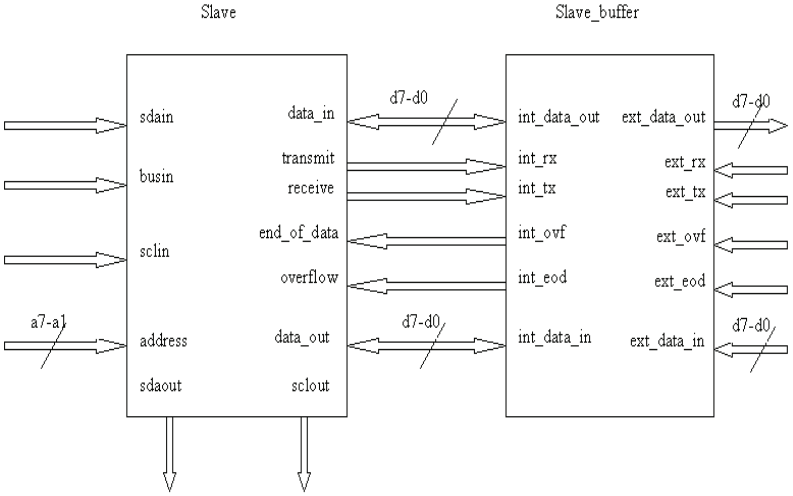


Fig. 6. Functional block diagram of the master

The Slave Block has four main functional units: (a) Monitor (b) Address Block (c)Receiver (d)Transmitter. The functioning of the slave block is given below:

Step 1: The Monitor continuously senses the state of the BUS line. If it senses that the bus line has been pulled to a low state, it senses that a master has captured the bus. It then enables the Address Block. Step 2: The Address Block checks whether the address sent by the master on the SDA line matches with it address. In case of a match, it sends the "add-match" signal to the Monitor which then enables the transmitter or receiver. In case of no match, the Monitor waits for the next "initiate-transfer" condition.

Step 3: The Receiver is responsible for receiving the data from the master and sending the ACK (or NAK) to the master. It reads the data at positive edges of the SCL. On detecting a release of the bus by the master , the receiver is disabled by the monitor.

Step 4: The Transmitter sends data to the master on the SDA line at positive edges of the SCL. It also checks for the ACK(or NAK) sent by the master. It terminates the data transfer on receiving a NAK from the master.

4 Simulation Results and Conclusion

We have used Modelsim 6.0 IIIa to simulate the I^2C devices. We have simulated a network containing three devices. Each of the devices have their own unique address by which they can be addressed. The address of the devices are given below:

Table 1. Address map of the devices

Device name	Device address
dev1	1111100
dev2	1111101
dev3	1111110

We have simulated the traffic in the network by using a testbench[4]. The masters of devices 1 and 2 were given a 8-bit data and their destination addresses were set to the address of device 3. Both the masters were given the start signal at the same time so that a conflict occurs. Due to the proposed I^2C arbitration specifications, one of the masters (device 1) wins control of the bus, and its data is transferred to the slave of device 3 successfully. This can be seen from the simulation results given below (Fig. 7).

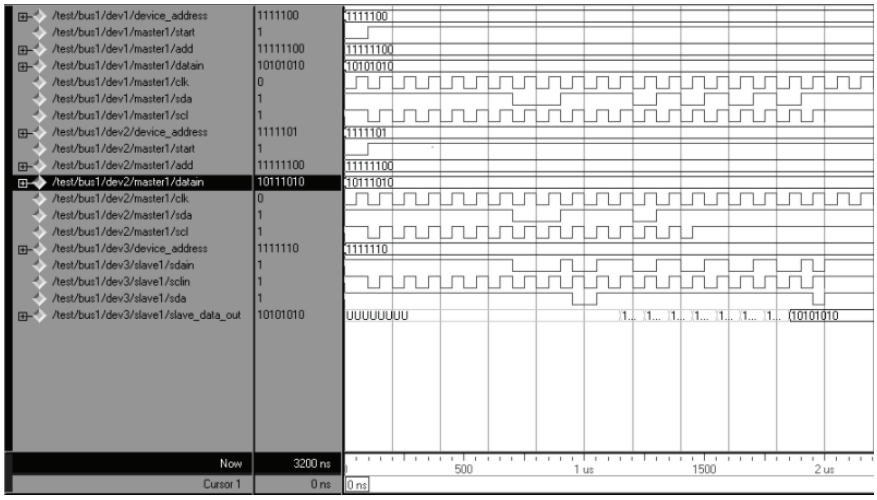


Fig. 7. Simulation waveform

References

1. THE I^2C BUS SPECIFICATION VERSION 2.1 January 2000, Philips Semiconductors
2. *IEEE Standard VHDL Language Reference Manual*, IEEE std 1076-1993 (revision of 1076-1987)
3. Fred Eady, *Networking and Internetworking with Microcontrollers*, Elsevier ,2004
4. J Bhasker *A VHDL Synthesis Primer,* BS Publications 2nd Edition, 2003 , pp. 132

Educational Software

The Measurement of the Student's Basic Knowledge Obtained by Means of Computer Assessments

Gennadiy N. Burlak[1], José Alberto Hernández[1],
Alberto Ochoa[2] and Jaime Muñoz Arteaga[3]

[1] Centro de Investigación en Ingeniería y Ciencias Aplicadas,
Universidad Autónoma del Estado de Morelos,
Av. Universidad 1001 Col. Chamilpa, Cuernavaca, Morelos México, CP. 62209
gburlak@uaem.mx

[2] Universidad Autónoma de Zacatecas
Jardín Juárez 147 Centro Histórico, Zacatecas, México, CP. 98000

[3]Universidad Autónoma de Aguascalientes

Abstract. We discuss the application of the computer technology on assessing the level of basic knowledge for a wide spectrum of subjects. This approach consists in questioning and then evaluating the validity of the students' answers. The employment of the Monte-Carlo technique for estimation of the optimal parameters allows us to develop a new approach to increase the accuracy of the valuation and determination of the stability of the knowledge. As implementation we use the Client-Server technology based on the natural evaluation process, where the students (Clients) are tested by the remote examiner (Server) in online regime. We focus on the Question and Testing Interoperability stage and the possibility of creation of the learning objects. We find that the optimal regime of assessment can be achieved at the specific random generation of tasks (assessment items) in the test. The construction of new adaptive methods and the program realizations of such technology are discussed also.

Keywords: Online assessments, Monte Carlo Method, Client-Server Technology

1 Introduction

The goal of this paper is to consider the evaluation of the level of student's knowledge at computer testing. We discuss the following questions: (i) What is the level of knowledge and how can it be estimated (ii), Why the random numbers must be used, (iii) How the statistically stable conclusions can be obtained, and (iv) Which are the optimum conditions for the evaluation of the student knowledge level.

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 201-213

Received 08/07/06
Accepted 03/10/06
Final version 17/10/06

This paper is organized as follows. In Sec.1 we discuss the basic strategies and methods of the tests; we present some numerical study of the student's knowledge level. In Sec.2 we present the algorithms and the program implementation of the computer testing. First, we analyze the online testing system (OTS) to assess the students' basic knowledge through online tests implementation with the use of learning objects [12] on the base of Client-Server technology. We focus on the e-learning Question and Test Stage, and the generation of learning objects in XML framework. Our goal is to accomplish the standard IMS Question & Test Interoperability Specification version 1.2.1 [14]. The OTS is not competition for well-known systems similar Blackboard [20] that offers an integrated solution for e-learning. We consider a similar solution as the Questionmark Perception Assessment software [19]. This enables educators to write, manage and report about assessments on a secure manner in Windows and web environments on a secure manner, storing the results for further processing in e.g. Oracle or SQL-Server Databases. The difference with respect to Questionmark is that presented OTS is platform independent, and it can save the database license expenses since storing the results in XML - MySQL databases. In Sec. 3 we present the results of practical implementation, and finally, we discuss and summarize our conclusions.

2 Approach to Methods of the Test

First we discuss the structure of the testing knowledge. Let us assume the knowledge base (KB) as a collection (set) of the coupled questions-answers (x_i, $\overline{f}_{ij}$, T) (records). In such a collection x_i is the asked question and $\overline{f}_{ij}$ are proposed answers to this question, i is the current number, j is number of answer in the list of proposed answers, and T is time to test. Here i<=N, j<=N_a , N is number of records in the collection, and N_a is number of proposed answers. To the sake of simplicity we further refer to the experts (teachers) knowledge base as the etalon KB. We assume that answers $\overline{f}_{ij}$ are ordered on the closeness to the correct answer. Then $F_i = \overline{f}_{i1}$ is the correct answer to i-th question. (We note that before the presentation all data undergo to random mixing.) At evaluating, the question is addressed to the student and his answer is compared with the corresponding correct answer. After testing the initial collection is extended with obtained results, e.g., obtained answers and other relevant information (response time, rating, etc.). After test, the initial collection is extended by the obtained answers, and there contains all necessary information for the estimation of the student's knowledge [7].

There are two different types of examination. In first type the student must recognize the correct answer from several predefined variants $\overline{f}_{ij}$ (the closed-ended form of the test). In the second type the system allows student to write his answer freely (the open-ended form of the test). Obviously that in first case (closed-ended test) the problem of verification of the answer can be solved easily. But this problem requires much more efforts in the case of open-ended test. Since the correct answer

can be written in various but equivalent forms, the problem of reduction of the answer to the unique referenced form has obvious solutions only for simplest cases. But in general such a problem may turn out quite involved. Here we pay main attention to the closed-ended test case.

We assume that the levels of the knowledge are ordered as follows: the quantity F_i is greater or equal with respect to other answers $F_i >= \overline{f}_{ij}$. We suppose that it is only the unique right answer to every question in the etalon collection [3].

Further we renormalize $\overline{f}_i$ and F_i with some coefficients w_i. With the use $w_i=1/F_i$ the etalon knowledge is represented by a straight line $F_i=1$, while the student knowledge is rewritten as $\overline{f}_{ij} / F_i = f_{ij}$, where $0 \le f_{ij} \le 1$.

In Fig.1 are shown schematically both distributions: the etalon knowledge and the student knowledge. The line at the top represents the level of correct (etalon) knowledge (knowledge of the teacher), and the variant line the student's knowledge. ***X***-axis corresponds to the number of asked question, while in the ***Y***-axis the level of student's answer is postponed.

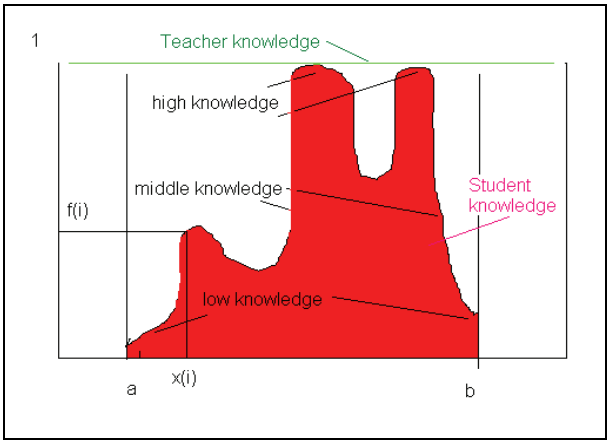


Fig. 1. Distribution of normalized etalon KB (green color) and distribution of student's KB (correct, wrong) in general (red color).

Here ***N*** is total number of records (tasks) in the collection, while ***n*** is the number of records (questions) in the current test.

The details of the graphic representation f_{ij} depend on the rules used at the examination. In the advanced cases the quantity (proposed answers) f_{ij} may have a value proportional to a closeness of the student answer to the etalon answer F_i. In a practical important case, one may use the following simple rule of the estimation: $f_{ij}=1$ for correct answer, and $f_{ij}=0$ if the answer is incorrect. For a small number of questions ***n*** such graphical representation is the histogram. At great number $n>>1$ the discrete picture can be generalized to the case of continuous distribution of the knowledge as

$$dZ = f(x)dx, \tag{1}$$

In this case ***f(x)*** is the local density of knowledge at vicinity of ***x*** and ***x+dx***.

In general the normalized knowledge ***Z*** may be defined as

$$Z = \frac{1}{N}\sum_{i=1}^{N} f_i \quad , \tag{2}$$

Then the normalized etalon knowledge is reduced to

$$Z_p = 1 \quad , \tag{3}$$

while the normalized knowledge of student is

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

In the framework of the knowledge measurement, the following factors are important: (i) For the objectivity of the estimation the student should not be foreknown on the sequence of the asked questions. This means that the order of the records in the task (both questions and predefined answers) should be randomized. (ii) The number of questions should be large enough to achieve of the desired accuracy of evaluation.

In result the student's knowledge after the test may be represented as

$$Z_s = \frac{1}{n}\sum_{i=1}^{n} f(\xi_i) \ . \tag{5}$$

where the integer numbers ξ_j already are not the successive integer numbers of the questions in the initial collection, but the random integers distributed in [***1 ,N***]. With Eq.(5) the problem of the knowledge measurement may be redefined as the problem of numerical evaluation of the sum $\boldsymbol{Z_s}$.

Mentioned above allows us to apply for such an evaluation the well-known technique of the statistical modeling (the Monte Carlo method [2]). Then the idea of evaluation takes following form: Since ξ_j are random numbers the Z_s the Eq.(5) is random also. From Eq.(5) and with the use of the central limit theorem[5] we can write the density of probability $p(Z_s)$ as follows:

$$p(Z_s) = \frac{n}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(Z_s - \overline{f}\right)^2}{2\sigma^2}n\right), \tag{6}$$

where $\overline{f} = \frac{1}{n}\sum_{i=1}^{n} f(\xi_i)$ and σ is the standard deviation for $f(\xi_i)$. From (6) the probability for the quantity $\left|Z_s - \overline{f}\right|$ to be less than 3σ/√n is given by

$$P\left(\left|Z_s - \overline{f}\right| < 3\frac{\sigma}{n^{1/2}}\right) = \frac{1}{\sqrt{2\pi}}\int_0^3 \mathrm{e}^{-z^2/2}\, dz \approx 0.997 \tag{7}$$

One can see that for a large number of the records (questions) in the test ***n>>1,*** the student's knowledge $\boldsymbol{Z_s}$, may be evaluated as $Z_s \approx \overline{f}$ independently on the details of the used random distribution of ξ_i , as it is written in Eq.(5). But at limited number of questions ***n,*** we have to optimize the distribution p(ξ) to minimize σ in Eq.(6a), and to improve the accuracy of the evaluation. The question arises: what distribution ***p(x)*** is good enough? To see that, we have to optimize parameter σ in Eq.(6a).

We rewrite Eq.(5) as follows

$$nZ_s = I = \sum_{i=1}^{n} f(\xi_i) = \sum_{i=1}^{n} g(\xi_i)p(\xi_i) = \overline{g}\,, \tag{8}$$

where ***g(x)=f(x)/p(x)*** and ***p(x)*** is the unknown normalized probability density ($\sum_{i=1}^{n} p_i = 1$) of the random integers $\boldsymbol{\xi_i}$ distributed in the interval ***[1, n]***, here $\overline{g}$ is the mean value of the g.

Calculating the standard deviation for random $\boldsymbol{\xi_i}$ with the use of Eq. (8) is given by

$$\sigma^2 = \overline{(g-\overline{g})^2} = \overline{g^2} - \left(\overline{g}\right)^2 = \sum_{i=1}^{n} g_i^2 p_i - \left[\sum_{i=1}^{n} g_i p_i\right]^2 = \sum_{i=1}^{n} g_i^2 p_i - I^2 \tag{9}$$

Since in (9) the quantity ***I*** is independent on the distribution ***p(x)***, we have to find such ***p(x)***, which minimizes the standard deviation σ in (9). With the use of the Cauchy-Buniakowski-Schwarz theorem [18], from (8)-(9) we obtain the following inequality

$$\sum_{i=1}^{n} g_i^2 p_i \equiv \sum_{i=1}^{n} \frac{f_i^2}{p_i} \le \sum_{i=1}^{n} \frac{f_i^2}{p_i^2} \cdot \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} \frac{f_i^2}{p_i^2}\,, \tag{10}$$

where normalizing of ***p(x)*** was taken into account. The optimum of $\boldsymbol{\sigma^2}$ arrives when in Eq. (10) the exact equality is achieved. Let us try to use the distribution $\boldsymbol{p_i}$ in the form

$$p_i = \frac{f_i}{\sum_{i=1}^{n} f_i}. \tag{11}$$

Substituting (11) in (10), we obtain in the left side

$$\sum_{i=1}^{n} \frac{f_i^2}{p_i} = \sum_{i=1}^{n} \frac{f_i^2}{f_i} \sum_{j=1}^{n} f_j = \left[\sum_{j=1}^{n} f_j \right]^2 = I^2, \tag{12}$$

while the right side becomes

$$\sum_{i=1}^{n} \frac{f_i^2}{p_i^2} = \sum_{i=1}^{n} \frac{f_i^2}{f_i^2} \left[\sum_{j=1}^{n} f_j \right]^2 = \left[\sum_{j=1}^{n} f_j \right]^2 = I^2 \tag{13}$$

We observe from (11), (12) that for distribution (11) in Eq. (10) the exact equality becomes. For this case in (9) we have σ=0 , therefore now the maximal accuracy of the evaluation achieves.

However it is clear that such a choice $\boldsymbol{p_i}$ in general is unachievable, since the function $\boldsymbol{f_i}$ is unknown before the student finishes the test. Therefore, we can only recommend that to improve the test estimation, we have to choose the distribution $\boldsymbol{p_i}$ proportional to $\boldsymbol{f_i}$. In context of the knowledge estimation this means that the probability $\boldsymbol{p_i}$ to ask a i-th question has to be higher in the area, where the knowledge of the student is better.

This observation allows us to propose the following two-step adaptive algorithm. In the first step the normal exam must be made to obtain answers $\boldsymbol{f_i}$, and in the second step on the base obtained f_i, we can construct the desired distribution $\boldsymbol{p_i}$ (11), which generates new exam. Such approximation will improve the final accuracy of the evaluation. It is worth noting that the area, where the student's knowledge is poor ($\boldsymbol{f_i=0}$), already does not contribute in the total measured knowledge $\boldsymbol{Z_s}$.

Furthemore, the use of this information yields the possibility to evaluate the stability of the student's knowledge. We have to compare the student's answers received in the first step $f_i^{(1)}$ with the answers obtained in second step $f_i^{(2)}$, and then calculate the quantity S as follows

$$S = 1 - \frac{1}{n} \sum_{i=1}^{n} (f_i^{(2)} - f_i^{(1)}). \tag{14}$$

The quantity $\boldsymbol{S}$ in (14) may be regarded as a stability of the student knowledge. If $S \approx 1$ the student has answered equally on the same questions both times, so he/she has a stable knowledge. Otherwise, if S is small the knowledge is not stable: in such a case the student's answers to the same questions were different.

The possibility to measure the stability of the student's knowledge is a very important characteristic of every evaluating system, which has to allow the estimation both quality and the level of the education obtained in an University.

3 Algorithm and Program Realization of the Online Testing System

In this section we analyze the realization of our Online Testing System (OTS) and its importance at the pedagogical process. First, we study the use of learning objects and the IMS QTI standard version 1.2.1 in our system. Then we analyze the Client-Server schema on which such OTS is based. We review the Server side, the Client side and the tools used for implementation. Finally, we show preliminary results obtained with the use of this system and discuss further work.

It is convenient to separate the program realization in two parts. The student part (Clients side) can be putted in the networks computers, while the part of code with the etalon knowledge is placed into other computer (Server side). The Server evaluates the result of exam in response of the data receiving from a student (Clients part).

3.1 The Application of Learning Objects into the OTS

In this section we analyze the meaning of learning object, how to apply this concept according the Question and Test Interoperability (QTI) standard version 1.2.1 de IMS Global to the OTS, as well as the defined migration schemas adequately to this standard.

3.1.1. What is a Learning Object?

There is a variety of definitions for learning objects that sometimes results very width, meanwhile other times are customized for the tool, system or organization that use it. The IEEE (Institute of Electrical and Electronics Engineers)'s Learning Technology Standards Committee (LTSC) defines learning object like "any entity, digital or non-digital, that can be reused or referenced by the technology supporting learning" [15]. This definition is extremely broad, and when we analyze fails to exclude any person, place, thing, or idea that has existed in any moment of the history of the universe, due any of them supports in some way learning.

Now we analyze some definitions that use the object oriented vision for computer aided instruction that turns concept confused. David Merrill uses the term "object learning" [16]. The educative software financed by the NSF (National Science Foundation) uses the term "educative software component" [10], and only accepts the Java's Applet (little Application) as learning objects [9]. The MERLOT (Multimedia Educational Resource for Learning and On-Line Teaching) Project makes reference to them as "online learning materials" [17]. Finally, Apple Learning Interchange simply refers to them as "resources" [1].

In our report we refer to learning objects as "any digital resource that can be reused to support learning" [22]. This definition includes any thing that can be delivered through the web on demand, be it large or small. Examples of smaller digital resources include digital images or photos, live or prerecorded videos, animations, or applets delivered via server, as a Java's Applet calculator [6].

3.1.2. The IMS QTI Standard Version 1.2.1 and the OTS

A proper used online testing system can result useful for the organization and users that employ it, due it makes the process faster, saves costs and generates repositories of reusable learning objects. However if the system does not adequate to any interoperability standard, the reach is local to the organization that use only. We will define a task as a composite element which integrates the question, the possible answers, the assigned time to answer, as well as the feedback or help for the respondent. The typical task is given by:

```
1. La velocidad de la luz en un bloque de vidrio con índice refracción 1.5 es:
<html> (D)2X10<sup>8</sup> m/s</hmtl>
<html> (A)3x10<sup>8</sup> m/s</html>
<html> (B)3x10<sup>8</sup> m/s</html>
<html> (C)4.5x10<sup>8</sup> m/s</html>
Todas
No se
60
Consulta la bibliografía recomendada para este curso
16. La densidad de los datos por unidad de energia Z(E), de electrones de un metal es representad
<html><div align='center'><img src='http://localhost/test/semiconductors/images/graficazea.gif' n
<html><div align='center'><img src='http://localhost/test/semiconductors/images/graficazeb.gif' n
<html><div align='center'><img src='http://localhost/test/semiconductors/images/graficazec.gif' n
<html><div align='center'><img src='http://localhost/test/semiconductors/images/graficazed.gif' n
Todas
No se
60
Consulta la bibliografía recomendada para este curso
2. El Sulfuro de Zinc es transparente en luz visible, esto implica que su brecha de energia es:
(A)>3 eV
(B)<2 eV
(C)<2.5 eV
D)<1.77 eV
Ninguna
Todas
60
Consulta la bibliografía recomendada para este curso
```

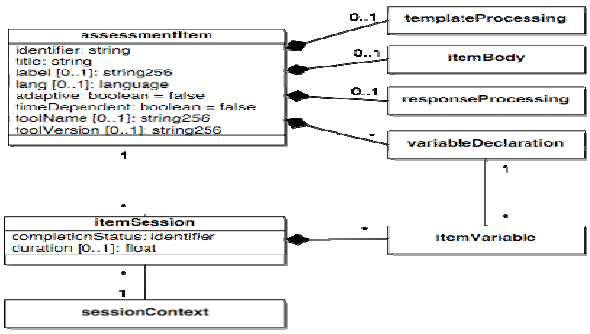
Figure 2. Segment of test tasks using proprietary standard on text-HTML

Now we discuss how we use the idea of Learning Objects in our system. The above figure represents our proprietary standard on text-HTML code for tasks management, however to being successful on knowledge economy “common standards for metadata management, learning objects and learning architectures are obligatory” [12]. Among the most important standards can be found ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe) supported by the European Union, and the American standard established by the IMS Global Consortium [8]. Keeping in mind to share our learning objects repositories, we started the migration process from our proprietary standard (see Figure 2) toward the IMS QTI standard version 1.2.1. The documentation in this standard indicates that an exam can be separated into questions or assessment items [IMS GLOBAL 2006], the rule to create these items is very simple: if the item is too large to fit the screen, then it will be necessary to create shorter items. Considering above exposed and based on the recommended UML (Unified Modeling Language) schema in Figure 3(a), first we identified required information (info.) for the assessmentItem class. Then we developed software that obtains it from the tasks declared in our proprietary standard, and automatically substitutes the information inside required sections delimited by XML tags: <assessmentItem required info. <correctResponse> required info.</correctResponse>, <outcomeDeclaration> required info. </outcomeDeclaration>, <itemBody> required info. </itemBody>,

<responseProcessing required info. />, </assessmentItem>. Figure 3(b) shows an example of the resulting XML code.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!--
 This example has been adapted from the PET Handbook, copyright University of Cambridge
-->
<assessmentItem xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p0 imsqti_v2p0.xsd"
        identifier="choice" title="Unattended Luggage" adaptive="false" timeDependent="false">
<responseDeclaration identifier="RESPONSE" cardinality="single" baseType="identifier">
<correctResponse>
<value>ChoiceA</value>
</correctResponse>
</responseDeclaration>
        <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="integer">
<defaultvalue>
        <value>0</value>
</defaultvalue>
</outcomeDeclaration>
<itemBody>
<choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
<prompt>La velocidad de la luz en un bloque de vidrio con indice refracción 1.5 es:</prompt>
<simpleChoice identifier="ChoiceA"> 2X10<sup>8</sup> m/s </simpleChoice>
<simpleChoice identifier="ChoiceB"> 3x10<sup>8</sup> m/s </simpleChoice>
<simpleChoice identifier="ChoiceC"> 3x10<sup>-8</sup> m/s </simpleChoice>
<simpleChoice identifier="ChoiceD"> 4.5x10<sup>8</sup> m/s </simpleChoice>
<simpleChoice identifier="ChoiceE"> Todas          </simpleChoice>
<simpleChoice identifier="ChoiceF"> Ninguna        </simpleChoice>
</choiceInteraction>
</itemBody>
<responseProcessing
        template="http://www.imsglobal.org/question/qti_v2p0/rptemplates/match_correct"/>
</assessmentItem>
```

(a) Assessment item schema



(b) XML code for single task

Figure 3. Assessment item schema (a) and the XML code for a single task (b), according the IMS Question and Test Interoperability standard version 1.2.1

3.2 Capabilities of the OTS

We have constructed our computer testing program with the following requirements. It must:

- Enable the repeatedly estimation of a basic knowledge level and a quickness of the correct answer finding.
- In case of problems (wrong answer) program has to show the correct answer, reference to the textbook, being training program.
- Show the final protocol, which contained the given questions, answers of the student, correct answers, and dynamics of his rating.
- Have a friendly and clear interface.

- To provide security by using DES algorithms to cipher communications (We will analyze the implementation of this specification in future paper).

In base of the above requirements, a Client-Server technology was developed, see Figure 4. The Client side represents the student or students and the server (the examiner). First, the client request for registration at the Server. The valid Client receives a collection of tasks containing the questions for the exam [13]. The client is asked for each task, and system stores all partial answers. When the test finished, such information is sent to the Server and processed, then the results are shown at the client side. The process of evaluation can be repeated several times in training mode and just one time in control mode.

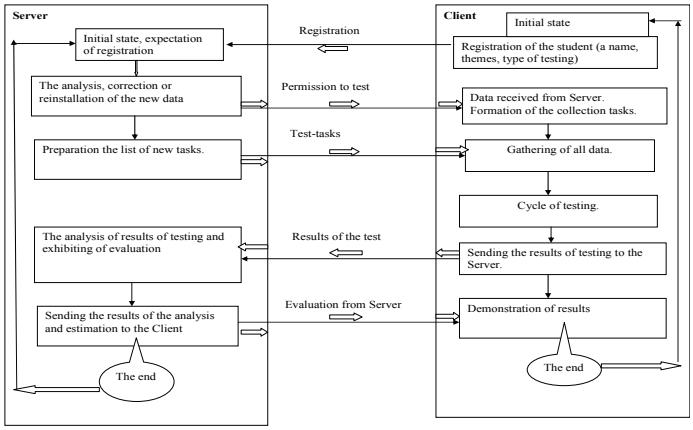


Figure. 4. Client-Server schema.

3.2.1. The Client-Server Technology

The implementation was done with the use of Object-oriented technology. We constructed the hierarchy of classes; and have code it in Java programming language [11]. The resulting technology is shown in Figure 5. At the left, is running the server program, listing the clients connected and the duration they was connected (see the AskServer section); the results obtained during examination process (see Finished Jobs); and the history of the information transferred (see historyForm) between the server and students connected.

The client is registered at the Server side, receives a collection of tasks and after that the evaluation process starts (see right side of Figure 5). The order of questions is determined by the generator of random numbers and is unpredictable. When the test is finished, the results are sent to server to be processed. Finally, the Client receives from the Server a grade, a rating and the final result of the examination.

To accumulate the results of exams, at the end of the 2005 year, we integrated MySQL Server as Data Base Management System to the OTS to avoid expensive licenses costs. Nowadays we are creating tests for several subjects to store the information in Databases and obtain the feedback from students to perform further analysis with Data Mining.

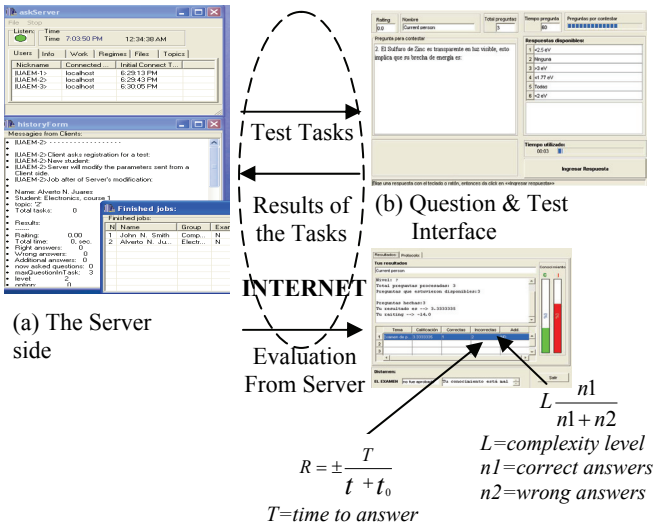


Figure 5. The server side (left) attending clients (right) [4].

3.2.2. Results of the OTS Implementation

For a simple estimation we have carried out N = 10 tests in the real tasks consisting of n = 27 questions. The average of the correct answers has appeared m = 4.5 (from 10 maximal). On the other hand, assuming the test as a gauss process, we find mathematical expectation m0=N/p, where p - probability of success in individual test, in our case p = 1/6 (6 closed answers). For the above given numbers one can find m0=4.5. The number of successes, equals 5 for this case (10 is maximum grade), is far below of the least satisfactory grade. Therefore we can declare the impossibility to obtain the satisfactory grade in absence of knowledge.

3.2.3. Further Work

To reinforce the innovative approach of our reasearch, we are going to use data mining [23] over resulting databases to determine student's learning patterns, and to avoid cheating (making frauds) on online tests. To improve the measurement of student's knowledge, we are planning to expand our system reachness from the use of question and test learning objects (assestment stage) to incoporate full learning objects that includes objective, theory, simulation and assessment. Simultaneaously, we are developing a tool to evaluate the quality of LOs since an integral approach that includes student and expert-professor (qualitative and quantitative) perspectives, as well as the use of intelligent agents and visual data mining to improve the LO's and student's learning experience [21].

4 Conclusions

The studied system can be customized for each student and carried out at any time, any where at a minimum cost. Thanks to the use of internet free of examiner; topics from different knowledge areas can be easily adapted to work within the system. The interfaces of programs (server side and client side) were designed in a user friendly way and implemented for various operating systems to avoid the platform incompatibility. System can evaluate mathematical, chemical formulas, images and even video, however performance is limited by the available Internet broad band access. The system can be used to implement surveys, to train personnel, and to assess at the successful candidates for job positions.

We can conclude that: The choice of the random order of asked questions allows estimating the level of student's knowledge at any distribution of random order of testing. At the fixed number of asked questions the accuracy of evaluation is better, if more questions are asked from the field, where the knowledge of the student is good.

The local form of the student's knowledge representation is not important, what is important only is the proportionality to the correct knowledge. For example, in a simplest case one can use the choice ***f(x)*** as following: ***f(x)*** =1 at right answer and ***f(x)=0*** otherwise. Such the method of the testing can be easily algorithmized and can be used at parallel computer evaluations of various students' groups separately in any time any where.

Acknowledgements. This work was partially supported by CONACyT granted by project 47220.

References

1. ALI. (2000). "Apple learning interchange website". In: http://ali.apple.com/
2. Bernd A. Berg (2004), "Markov Chain Monte Carlo Simulations and Their Statistical Analysis (With Web-Based Fortran Code)", World Scientific, ISBN 981-238-935-0
3. Bugai, Y., Burlak, G., Demchenko, A., Kuz'menko, N. (1997) "On Application of Computer Technology Testing In Educational Process", In: Conference Role of Universities in the Future Information Society RUFIS'97, Prague, p. 141-144.
4. Burlak, G. Hernández, J. Zamudio, A. (2005) "The Application Of Online Testing For Educational Process In Client-Server System", In: Proceedings IADIS International Conference ICWI-05, Portugal, Lisboan.
5. Box, George E.P., Hunter, William G., Hunter, J. Stuart (2002). "Estadística para Investigadores. Introducción al Diseño de Experimentos, Análisis de Datos y Construcción de Modelos", Editorial Reverté, México, D.F. p. 41.
6. Cavalli, A., Magg, S., Papagiannaki, S., Verigakis, G. (2005), "From UML models to automatic generated tests for the dotLRN e-learning platform", In: Electronic Notes in Theoretical Computer Science http://www.elsevier.com/locate/entcs, Vol. 116. p. 133-144.

7. C.P. Robert, G. Casella (2004). "Monte Carlo Statistical Methods" (second edition). New York: Springer-Verlag, 2004, ISBN 0-387-21239-6
8. Duval, E. (2004) Learning Technology Standardization: Making Sense of it All, International Journal on Computer Science and Information Systems, Vol. 1, Issue 1, p. 33-43.
9. EOE. (2000). "Educational objects economy website", In: http://www.eoe.org/eoe.htm
10. ESCOT. (2000). "Educational software components of tomorrow website", In: http://www.escot.org/
11. Javax.crypto.spec (Java2) (2005), In: http://java.sun.com/j2se/1.5.0/docs/api/javax/crypto/spec/package-summary.html, July 2005
12. Hodgins, W. (2005), "Learning Objects & Learning Standards", In: http://learnativity.com/standards.html, July 2005
13. Hunt, N., Hughes, J., Rowe, G. (2002), "Formative Automated Computer Testing (FACT)", In: British Journal of Educational Technology, Vol. 33, No 5, p. 525-535.
14. IMS Global Learning Consortium (2006) "IMS Question & Test Interoperability Specification", In: http://www.imsglobal.org/question/, July 2006
15. LTSC. (2000a). "Learning technology standards committee website". In: http://ltsc.ieee.org/
16. Merrill, M.D. (1991) et al. "Instructional transaction theory: An introduction. Educational Technology", 31(6), 7-12.
17. MERLOT. (2000). "Multimedia educational resource for learning and on-line teaching website". In: http://www.merlot.org/
18. Mitrinovic, D.S., Pecaric, J.E. And Fink, A.M. (1993). Classical and New Inequalities, Kluwer Academic Publishers, Dordrecht.
19. Questionmark Software Perception-Product Information. (2006), In: http://www.questionmark.com/us/perception/index.htm, May 2006
20. The Blackboard Learning System™ (Release 6) (2006), In: http://bb1.ctdlc.org/, May 2006.
21. Varughese, J. (2005) "Testing, Testing: IHEs are trying to ensure the success of students by testing their abilities using the latest online assessment software", In: University Business http://www.universitybusiness.com, p. 59-65.
22. Willey, D. (2000). "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy". In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version, In: http://reusability.org/read/chapters/wiley.doc, July 26, 2005
23. Witten, I., Frank, E. (2000) "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations" Morgan Kaufmann Publishing USA p. 369.

Control

Towards The Stability Analysis of Principles of Power Conduct

Julio Clempner[1], Jesús Medel[1] and Alin Cârsteanu[2]

[1]Center for Computing Research
National Polytechnic Institute (CIC-IPN)
Av. Juan de Dios Batiz s/n, Edificio CIC
Col. Nueva Industrial Vallejo, 07738
Mexico City, Mexico
email: julio@k-itech.com/jjmedelj@cic.ipn.mx
[2] Centro de Investigacion y de
Estudios Avanzados del IPN (Cinvestav)
Apartado Postal 14-740, C.P. 07360
Mexico City, Mexico
e-mail:alin@math.cinvestav.mx

Abstract. One of the major problems in the Machiavellian (manipulation) interaction social model is establishing its key property of stability, which is important in behavior evaluation and control. This paper presents a formal framework for Machiavellian dynamic stability analysis and model validation. We assume that Machiavellianism is a stochastic process, and as a consequence, it can be described by a model whose result set is in the basic set of elements of the Machiavellian Stochastic Process. An autoregressive model AR(1) is proposed to represent the dynamics of the Machiavellian interaction social model. An estimation matrix, an identification model and an error estimation model are employed to implement the stability properties of the Machiavellian interaction social model. System stability of a Machiavellian individual is being considered. An illustrative example with a simulation of the model is given.

Key words: interaction social model, Machiavellianism, stability.

1 Introduction

In this work we focus on the interaction principles of conduct laid down by Machiavelli. The importance of Machiavelli in the development of organizational theory consists in that he was among the first to identify the natural laws that govern how effective leaders of hierarchical organizations exercise power over the human resources within such an organizational system.

Machiavellianism has been used to categorize individuals in terms of a belief that persuasive, manipulative behavior will help to achieve personal goals. People holding this viewpoint will extol the use of guile and deceit to reach

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 217-227

Received 10/07/06
Accepted 11/10/06
Final version 17/10/06

their objectives ([14], [15], [16]). Although Machiavelli did not recommend lying, he assumed its necessity in an imperfect world. To achieve one's objective, an individual can sacrifice truthfulness, and therefore, ethics would be sacrificed. Machiavellian individuals should not be viewed as consistently untruthful or unethical, but rather as individuals who are willing to sacrifice ethics, if necessary, to attain their objectives.

Although most of what Machiavelli had to say was intended to provide advice on how successful leaders exercise power over political organizations, his views can (and should) be applied to today's business executives and organizations, given that all organizations are subject to power politics.

Christie and Geis [3] were the first to study manipulation as a fundamental parameter in human behavior variation, calling it "Machiavellianism". They showed that Machiavellians tend to have a "utilitarian rather than a moral view of their interactions with others". There is an inconsistency between earlier opinions that Machiavellians lack a sense of morality, and current research showing how Machiavellians may indeed behave in a relatively moral fashion [13].

The paper is structured in the following manner. After this introduction, section 2 gives the basic background in Machiavellianism. Section 3 extends the definition of Machiavellianism. We describe the Machiavellian interaction social model in section 4. In Section 5 we introduce all the formalisms needed for the stability analysis of the Machiavellian system. Consequently, we present the estimation error model in section 6. In section 7 we describe the Estimator and Identifier Filters. Section 8 presents an application example with the corresponding simulation. Finally, section 9 concludes the paper which presents the current status of the work and future research directions.

2 Background in Machiavellianism

Machiavellianism has been associated with different variables, given a wide range of interpretations related to psychological components. Smith [20] argued that the descriptors of the psychopath and those of the Machiavellian must have common domains, because they are similar (manipulative style, poor affect, low concern about conventional moral, low ideological compromise, and others). In agreement with Cleckley [4] other Machiavellian tendencies coincide with some components of anomie (cynicism, low interpersonal credibility, external locus of control).

Researchers have investigated the relationship between locus of control and Machiavellianism. Solar and Bruehl [21] were the first in establishing a relationship between Machiavellianism and locus of control, considering both as aspects of interpersonal power. Their study reported a significant relationship between Machiavellianism and locus of control for males, but not for females. Prociuk and Breen [18] supported this result. Mudrack [17] conducted a meta-analytic review of 20 studies determining the relationship between Machiavellianism and external locus of control. Gable, Hollon and Dangello [9] sustain this result. They related locus of control, Machiavellianism and managerial achievement; their results did

not show significant correlations between locus of control and achievement, but found a positive correlation between Machiavellianism and external control.

With respect to influence tactics, Falbo [8] showed that persons with high Machiavellianism are associated with the use of rational indirect tactics (i.e., lies), while those with low Machiavellianism are associated with the rational use of direct tactics (i.e. rewards). Grams and Rogers' [11] research confirms this result and also shows that persons with high Machiavellianism are more flexible when it comes to breaking some ethical rules. Vecchio and Sussman [22] suggested that Machiavellianism and the tactics selection are related to sex and organizational hierarchy; the use of influence tactics is common in males and females with high-level positions.

3 Definition of Machiavellianism

In accordance with different studies of social psychology, manipulation is placed among the forms of social influence as part of the social interaction behavior. Raven [19] argued that power can be psychologically studied as a product of behavior, including personal attributes, with the possibility to affect others through interaction, and the environment structure. Dawkins [6] proposed that, in terms of selfishness, altruism, cooperation, manipulation, lie and truth, genetically there exists a selfishness and manipulation gene. Dawkins and Krebs [7] classified manipulation as a natural-selection state benefiting individuals able to manipulate others' behavior. Vleeming [23] denotes a personality dimension in which people can be classified in terms of being more or less manipulated in different interpersonal situations. Wilson, Near and Millar [24] define Machiavellianism as a social strategy behavior involving the manipulation of others to obtain personal benefits, frequently against others' interests. They clarify that anybody is able to manipulate others to different degrees, and they also explain that selfishness and manipulation are behaviors widely studied in evolutionary biology. Hellriegel, Slocum Jr. and Woodman [12] define Machiavellianism as a personal style of behavior in front of others, characterized by: the use of astuteness, tricks and opportunism in interpersonal relationships; cynicism towards other persons' nature; lack of concern with respect to conventional morals. Christie and Geis [3] propose three factors to evaluate high or low Machiavellianism: tactics, morality and views. Tactics are concerned with planned actions (or recommendations) to confront specific situations with the purpose of obtaining planned benefits at the expense of others. Morality is related to behavior that can be associated with some degree of "badness" with respect social conventions. Views involve the idea that the world consists of manipulators and manipulated. In this sense we introduce the following definition.

Definition 1. *Machiavellianism is a social interaction model supposing that the world can be manipulated by applying (Machiavelli's) strategies and tactics with the purpose of achieving personal gains according (or not) to a conventional moral.*

Immorality is a un-arrangement of customs. One of the best-known concepts is the immorality described by Nietzsche. Therefore, the factor of morality proposed by Christie and Geis [3] is not appropriate, because in the evaluation of the factor, immorality is considered the opposite to a "conventional moral".

For the purposes of this paper, we will consider the terms views, tactics and immorality defined as follows:

Views: The belief that the world is able to be manipulated.

Tactics: The use of a manipulation plan guided by strategies to achieve specifica power situations (goals). Plans, strategies and goals are analyzed in Machiavelli's ThePrince [14], The Discourses [15], The Art of War [16] and the psychological behavior patterns (section2).

Immorality: The disposition to not be come attached to a conventional moral.

These three factors are statistically evaluated to a certain degree through a set of variables considered in the tests Mach IV and Mach V (Christie & Geis, 1970). Such results, in agreement with Christie and Geis (1970), are described by a well-known distribution function. Other tests could be used to measure the amount of Machiavellianism, but it is important to verify their coefficients for internal consistency and stability.

4 Machiavellian social interaction model

In agreement with Gnedenko [10] and Caines [2], considering its properties as a part of an innovation process, Machiavellianism could be described as a stochastic process. The problem is to measure the Machiavellian variables and to analyze their properties [**?**].

In the Machiavellian stochastic process we find a set of personality variables (random variables) with correlation to the Machiavellianism, as defined in Christie and Geis [3]; Falbo [8]; Gable, Hollon, & Dangello [9]; Grams & Rogers [11]; Mudrack [17]; Prociuk & Breen [18]; Solar & Bruehl [21]; Vecchio & Sussmann [22]; Vleeming [23]; and others. Some random variables are: influence tactics, leadership, locus of control, cynicism, organizational commitment, impulsivity, and others.

A Machiavellian stochastic process could be described in recurrent times using a filter theory as a descriptor of real processes, which is considered to generate a good enough answer. This model is also described by a probability distribution function of the process, defined by its two probability distribution moments.

In the present work, we would like to verify whether an autoregressive model captures the essential concept of a Machiavellian system and, given the environment dynamics, allows to satisfy a stationary distribution function.

The simplest form of this kind of systems is an AR(1) model with additive perturbations. In symbolic form, the Machiavellian social interaction model could be described as:

$$y_k = Ay_{k-1} + B_k\Psi_k \tag{1}$$

where A is the gain matrix of the internal properties. The set of its eigenvalues are inside of the unit circle, which guarantee the distributional invariance conditions and hereby stability in terms of mean and variance of the process. represents the influence of the immediate past state of the individual. B is the matrix of the random condition of the individual's environment, and is an interaction vector with the environment.

The system described in (1) is obtained from:

$$x_k = Sx_{k-1} + b\omega_k \ , \tag{2}$$

$$y_k = x_k + c\upsilon_k \ , \tag{3}$$

where: $B_k = \left[-Sc \vdots cb \vdots c\right]$, $\Psi_k = \left[\upsilon_{k-1} \vdots \omega_k \vdots \upsilon_k\right]^T$ and $A = S$ which describes the internal dynamics of the system.

5 Machiavellianism stability analysis

The evolution of the recurrence (1) depends of the matrix . But, in fact, its values are a priori unknown in the real applications, because in the system (1), only the values of external perturbations and the set of the answer described by the Machiavellian social interaction model are measurable. Since we do not know the dynamics behavior, we make use of an estimator in recursive form. The estimator is a function of the answer of the individual considering a specific test, that obeys a set of conditions established in probability sense.

The vector describes the real interaction of the individual in the environment and describes the optimal interaction conditions of some individual in the environment. So, the difference of and is described by a vector predefined by the variance of the random variables with respect its inputs and outputs. Formally, we have the following definition:

Definition 2. *An individual is Machiavellian uniformly asymptotically stable if*

$$\lim_{k\to\infty} \{\|\Psi_k^* - \Psi_k\|\} \to 0 \tag{4}$$

In agreement to Ash [1] this condition is true if $0 < \|\Psi_k^* - \Psi_k\| < \upsilon_k$, and $\underset{k\leq m}{lim}\ \{\upsilon_k\} \leq \varsigma_m$. If Ψ_k^* is optimal in the Lyapunov sense, then Ψ_k is good enough and we have $\underset{k\leq m}{lim}\ \{\varsigma_m\} \to 0$ and the set of answers of the individual are uniformly asymptotically stable too.

Arbitrarily we choose an estimator $\hat{A}$ and an identifier $\hat{y}_k$ able to describe a Machiavellian individual behavior. If we have a 2nd-probability moment with respect to y_k such that the system described in (1) is bounded in recursive

sense, and we have that $\hat{y}_k$ (identifier) is Machiavellian bounded by the same probability moment, then $\hat{y}_k$ is uniformly asymptotically stable. In this sense, if we consider that $\hat{y}_k$ is Machiavellian stable and bounded by a recursive 2nd-probability moment and we have that $\hat{y}_k$ is also Machiavellian bounded by the set of answer of the same recursive 2nd-probability moment, as a result we have that the answer of the filter is asymptotically stable. y_k will be bounded by the 2nd probability moment, because in the AR(1) model we chose in (1) the variance asymptotically converges in the 2nd probability moment.

It is important to note that the estimator filter $\hat{A}$ is stable if $\hat{y}_k$ is a Machiavellian bounded input. The stability exponent Λ of $\hat{A}$ asymptotically converge. Then, $\hat{A}$ is asymptotically stable if it is bounded and, in other case, $\hat{A}$ is invariant in time sense. In addition, the estimator filter $\hat{A}$ could be asymptotically stable if it is bounded with respect to 2nd probability moment.

Moreover, the identifier filter $\hat{y}_k$ is asymptotically stable with respect a Machiavellian bounded input. The Kalman-Yakubovic-Popov lemma suggests that the stability exponent Λ of $\hat{y}_k$ asymptotically converge. Then, $\hat{y}_k$ is asymptotically stable if it is bounded.

As a result let us consider the answer of the estimator $\hat{A}$ and the answer of the identifier $\hat{y}_k$ asymptotically stable with respect to the answers of the Machiavellian individual. Then the Machiavellian system is uniformly asymptotically stable if in agreement with Lyapunov:

$$\Psi_k^* = B_k^+ f(y_{k-1}; A_k; \Psi_k) - \hat{A}_k \ \hat{y}_{k-1}, \tag{5}$$

where Ψ_k is bounded.

In this sense, let us consider

$$y_k = f(y_{k-1}; A_k; \Psi_k) \tag{6}$$

as a generic model with respect to (1).

Then, the vector input described by Ψ_k is optimal if $\dot{\Delta}_k = 0$. Considering the error is described as

$$\Delta_k = |\hat{y}_k - y_k|, \tag{7}$$

Its gradient developed is $\dot{\Delta}_k = \hat{A}_k \ \hat{y}_{k-1} + B_k \Psi_k - f(y_{k-1}; A_k; \Psi_k)$,
and accomplished the optimal conditions with respect to input:

$$\Psi_k^* = B_k^+ f(y_{k-1}; A_k; \Psi_k) - \hat{A}_k \ \hat{y}_{k-1}, \tag{8}$$

where $B_k^+ := B_k^T (B_k^T \ B_k)^{-1}$.

Note that the matrix of estimated parameters $\hat{A}$ and the vector of identification states $\hat{y}_k$ established the stability conditions of the real system.

6 Error estimation

We consider that the convergence rate between the Machiavellian interaction model described in (1) and his identifier is bounded by the 2nd probability moment expressed by $J_n = E(\Delta_n \ \Delta_n^T)$ and converge asymptotically to the equilibrium point, i.e., $\lim_{n\to\infty} J_n = 0$. Where the identification error is $\Delta_n := |y_n - \hat{y}_n|$, $n \in \aleph$.

In this sense, let the 2nd probability moment be expressed in simple form as $J_n = \sum_{k=1}^{n} \Delta_k \ \Delta_k^T \ P(\Delta_k/\Im_{k-1})$. Considering that the set of identification errors we have $J_{n-1} = \frac{1}{n-1} \sum_{k=1}^{n-1} \Delta_k \Delta_k^T$
then, in recursive form with respect to J_n, we obtain

$$J_n = \frac{1}{n} \left[\Delta_n \Delta_n^T + J_{n-1}(n-1) \right] \tag{9}$$

Considering that: $\Delta_n \in [0, 1)$, $\forall n \in \aleph$, then evidently $\lim_{n\to\infty} J_n = 0$ is true and J_n decreases exponentially.

7 Estimator and Identifier Filters

In this section we will choose an estimator and an identifier able to fulfill the concepts described in previous sections.

From (2) and (3) we have an estimator using an instrumental variable such that $E\left\{\omega_k \ \upsilon_k^T\right\}$ and the interaction of the Machiavellian interaction in the environment is bounded by the 2nd probability moments:

$$E\left\{y_k \ \omega_k^T\right\} = (\Theta_\omega)^2, \ E\left\{y_k \ \upsilon_k^T\right\} = (\Theta_\nu)^2, \tag{10}$$

The matrix parameter estimator $\hat{A}$ using instrumental variable is described in recursive form:

$$\hat{A}_k = \left(\hat{A}_{k-1} T_{k-1}^{-1} + y_k \ \Theta_k^T \right) T_k^{-1} \tag{11}$$

where: $T_k^{-1} = E\left\{y_{k-1} \ \Theta_k^T\right\}$, and Θ_k^T is an instrumental variable such that it is a function of real interaction of the Machiavellian interaction y_k.

In accordance with the Machiavellian interaction model, the estimator described in (11) has an asymptotic convergence in agreement with definition 2. The model satisfies A, defined as the matrix gain of the internal properties of Machiavellian interaction model.

To guarantee the stability and the invertibility conditions it is necessary proceed with a dynamic following the behavior of the individual, given a specific environment and using an identification model. In general this kind of system

could be described by identification techniques, but in any case, each of them require the transition matrix or the estimation techniques.

The final goal of an identification model is to find a mean square stabilizing, minimum variance control law. To do this we note that by viewing as a state process, the conditional expectation $\tilde{y} = E\{y_k/\Im_{k-1}\}$ is computable using the Kalman filter, providing as result a Gaussian distribution.

In this case, the Kalman filter identification is obtained recursively by the following equations. Form (2) and (3) let $Cov(b\ \omega_k) = Q_k$ and $Cov(c\ \upsilon_k) = R_k$, then

$$\hat{y}_k = \hat{y}_{k-1} + K_k(\hat{A}_k - \hat{A}_k\ y_{k-1}), \tag{12}$$

where $K_k = \left(R_k + \hat{A}_k\ P_k\ \hat{A}_k^T\right)^{-1} P_{k-1}\hat{A}_k^T$, and $P_k = P_{k-1} - \left(R_k + \hat{A}_k\ P_k\ \hat{A}_k^T\right)^{-1} P_{k-1}\hat{A}_k\hat{A}_k^T P_{k-1} + Q_k$.

8 Simulation Example

The Machiavellian social interaction model (1) and its error estimator J_n is described in recursive form by (9).

In graphs 1a, 1b, and 1c is described the matrix parameter estimator $\hat{A}$.

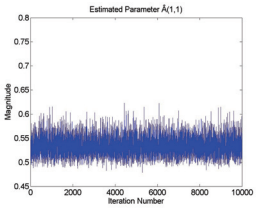


Figure 1a. Estimated Parameter (1,1)

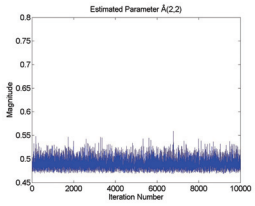


Figure 1b. Estimated Parameter (2,2)

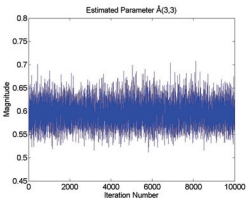


Figure 1c. Estimated Parameter (3,3)

In figures 2a, 2b and 2c we represented the estimated parameters for the variables tactics, immorality and views. The corresponding filtered graphs are presented in figures 3a, 3b and 3c.

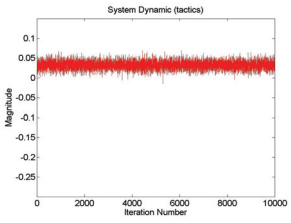


Figure 2a. System Dynamic (tactics)

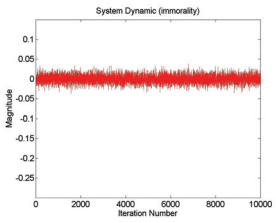


Figure 2b. System Dynamic (immorality)

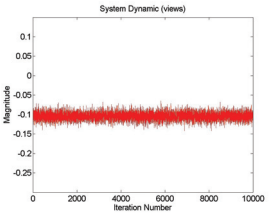


Figure 2c. System Dynamic (views)

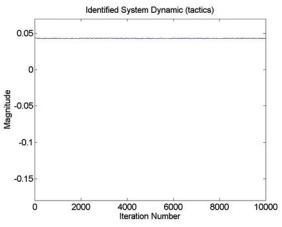


Figure 3a. Identified System Dynamic (tactics)

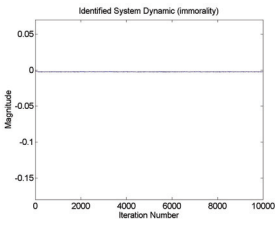


Figure 3b. Identified System Dynamic (immorality)

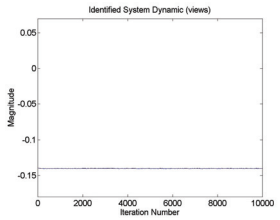


Figure 3c. Identified System Dynamic (views)

Machiavellianism could be conceptualized as being influenced by different variables. On the one hand, by the three concepts that define Machiavellianism (M^3): tactics, immorality and views (Figures 4a, 4b). These variables are correlated with the Machiavellianism, but have been shown to have no correlation between them, resulting in a diagonal transition matrix A.

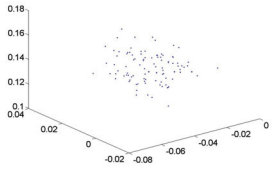


Figure 4a. Simulation with 3 variables (tactics, immorality, vision)

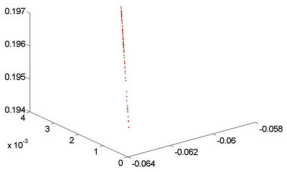


Figure 4b. Simulation filtered with 3 variables (tactics, immorality, vision)

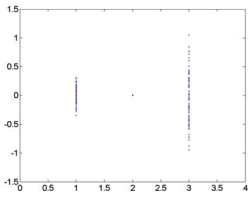


Figure 5a. Simulation with 3 variables (tactics, immorality, vision)

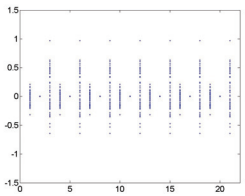


Figure 5b. Simulation with 21 personalities variables

On the other hand, Machiavellianism (M^n) is correlated with different personality variables like: aggression, achievement, dogmatism, conscientiousness,

trust, defensive, authority, exploitativeness, etc. These variables present a correlation between them that must be expressed in a transition matrix A with nonzero off-diagonal elements of the Machiavellianism model (1). The eigenvalues obtain after diagonalizing such a matrix play the same role in the stability of the system as do the eigenvalues of a diagonal transition matrix.

The simulation of both scenarios is presented in Figure 5a and Figure 5b[1]. In both graphs we can establish a stability region between (-1, 1.1) and (-0.7, 1) respectively. So it is possible to suppose that Machiavellian M^3 and M^n display a similar behavior, as we expected. Obviously, the graphs in figures 5b will be closed to the graphs in Figures 5a when the number of personality variables will be increased.

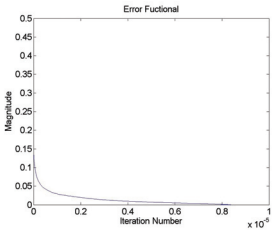


Figure 6a. Error asymptotically stable

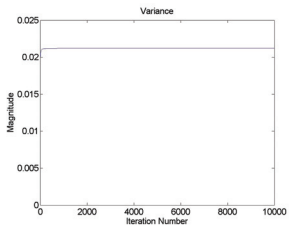


Figure 6b. Variance convergence

The error, represented in figure 6a, decrease monotonically for the three variables (tactics, immorality and views). So, the Machiavellian system is uniformly asymptotically stable, i.e. when the identifier index k tends to infinity, J_n tends to zero (9). In figure 6b we have a representation of the variance, where it is graphically shown that the Machiavellian system converges in its 2nd probability moment.

9 Conclusions and future work

A formal framework for stability analysis of the Machiavellian interaction social model has been presented. A new definition of Machiavellianism that extends the definition of Christie & Geis [3] was introduced. The stability of the process describing the behavior of the Machiavellian individual was analyzed in terms of mean and variance. We present the condition under which the error and the variance converge. The Machiavellian stochastic process was also described in terms of a filter as a descriptor of real processes. The simulation illustrates the applications of the Machiavellian procedure in practical data analysis. As future work it will be of interest to develop a software tool which will take care of the stability for real Machiavellianism life problems.

[1] Note that in the graphs the stability area of some variables is represented by points. This is related with the choice of the stable initial values of these variables for simulation purpose.

References

1. Ash R. (1970). Real Analysis and Probability. Ed. Academic Press.
2. Caines P. (1986). Linear Stochastic Systems. Ed. Wiley.
3. Christie, R. & Geis, F. (1970). Studies in Machiavellianism. Nueva York: Academic Press.
4. Cleckley, H. (1976). The Mask of sanity. (5th ed.). St. Louis: Mosby.
5. Clempner J., Medel J. and Cârsteanu A. (2005). Toward Representing Social Interaction Principles of Power Conduct in a Game Context. WSEAS Transactions on Business and Economics, 2, 2, 96-104.
6. Dawkins, R. (1976). The selfish gene. Oxford, England: Oxford University Press.
7. Dawkins, R. & Krebs, J.R. (1978). Animal signals: Information or manipulation. En J.R. Krebs & N.B. Davies (Eds.), Behavioral ecology: An evolutionary approach (pp.282-309). Oxford, England: Blackwell.
8. Falbo, T. (1977). Multidimensional Scaling of power strategies. Journal of Personality and Social Psychology, 35, 537-547.
9. Gable, M., Hollon, Ch., & Dangello, F. (1992).Managerial Structuring of Work as a Moderator of the Machiavellianism and Job Performance Relationship. The Journal of Psychology, 126, 3, 317-325.
10. Gnedenko, B. V. (1962). The Theory of Probability. Ed. Chelsea Publishing Company.
11. Grams, W.C. & Rogers, R.W. (1990). Power and Personality: Effects of Machiavellianism, Need for Approval, and Motivation on Use of Influence Tactics. The Journal of General Psychology, 117, 1, 71-82.
12. Hellriegel, D., Slocum, Jr., J.W. & Woodman, R.W. (1997). Organizational Behavior. South-Western Educational Publishing. July.
13. Leary M., Knight R. & Barns B. D. (1986). Ethical ideologies of the Machiavellian. Personality and Social Psychology Bulletin, 12, 75-80.
14. Machiavelli N. (1513a). The Prince. In The Prince and Leviathan. Chicago: Encyclopedia Brittanica, Inc. 1952.
15. Machiavelli N. (1513b). "Discourses in the first ten books of Titus Livius". In chiefwork and other. Durham, NC: Duke University Press. 1965.
16. Machiavelli N. (1520). The Art of War. Da Capo Press. 2001.
17. Mudrack, P.E. (1989). Machiavellianism and Locus of Control: A Meta-Analytic Review. The Journal of Social Psychology, 130, 1, 125-126.
18. Prociuk T. J. & Breen L. J. (1976). Machiavellianism and Locus of Control. The Journal of Social Psychology, 98, 141-142.
19. Raven, B.H. (1993). The Bases of Power: Origins and Recent Developments. Journal of Social Issues, 49, 4, 227-251.
20. Smith, R.J. (1979). The psychopath in society. New York: Academic Press.
21. Solar, D. & Bruehl, D. (1971). Machiavellianism and Locus of Control: Two Conceptions of Interpersonal Power. Psychological Reports, 29, 1079-1082.
22. Vecchio, R.P., & Sussmann (1991). Choice of influence tactics: Individual and organizational determinants. Journal of Organizational Behavior, 12, 73-80.
23. Vleeming, R.G. (1979). Machiavellianism: A preliminary review. Psychological Reports, 44, 295-310.
24. Wilson, D.S., Near, D.C. & Miller, R.R. (1996). Machiavellianism: A Synthesis of the Evolutionary and Psychological Literatures. Psychological Bulletin, 119, 2, 285-299.

Admissibilization of Continuous Descriptor Systems

M. Chaabane[1], O. Bachelier[2], R. A. Ramírez-Mendoza[3], and D. Mehdi[2]

[1] Automatic Control Research Unit, Electrical Engineering Department, Sfax National Engineering School, B.P. 805 Route Menzel Chaker km 0.5, 3038 Sfax, Tunisia
mohamed.chaabane@ipeis.rnu.tn

[2] L.A.I.I, ESIP, 40 Avenue du Recteur Pineau, 86022 Poitiers Cedex, France,
{olivier.bachelier,mehdi.driss}@univ-poitiers.fr

[3] Tecnológico de Monterrey, Campus Monterrey, Avenida Eugenio Garza Sada 2501 Sur, Monterrey, N.L., México
ricardo.ramirez@itesm.mx

Abstract. This paper deals with the problems of robust admissibility and admissibilization for uncertain continuous descriptor systems. We propose a new necessary and sufficient condition in term of a strict Linear Matrix Inequality (LMI) for a nominal continuous descriptor system to be admissible (stable, regular and impulse free). Based on this, the state feedback admissibility problem is solved and the solution extended to the case of uncertain descriptor systems. Finally numerical examples are given to illustrate the results.

Key words : Continuous descriptor systems, Admissibility, Lyapunov equation, Feedback control,

1 Introduction

Intuitively, singular state space description of linear systems is more general than conventional state space description. In particular, a descriptor form includes information about static as well as dynamic constraints. Singular system, both continuous and discrete, has been of interest in the literature since they have many applications (see, [4]), for instance in electrical circuits network, robotic and economics. It is fair to say that descriptor models give a more complete class of dynamical models than the conventional state space systems.
Many classical concepts and results obtained for conventional systems have been extended to descriptor systems. Let us quote for instance controllability and observability, pole assignment, stability analysis [8,6]) and stabilization techniques as well as results including robustness aspects [9,7,13].
The natural *Generalized Lyapunov Equation* (GLE) [8] was proven in [6] to fail unless the system is in its Weirstrass form and the authors of [6] proposed a new GLE equivalent to that given in [11]. In [9], the authors modified the GLE given

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 229-239

Received 07/07/06
Accepted 07/10/06
Final version 13/10/06

in [11] and proposed an equivalent matrix inequality condition.
In the available literature on descriptor systems, there are two kinds of stabilization problems for singular continuous-time systems. One consists in designing a state feedback controller in such a way that the closed-loop system is regular, impulse-free and stable or equivalently admissible. The other is to design a state feedback controller in order to make the closed-loop system regular and stable.
In many approaches, the system model is transformed into a special form and it is understandable that this way of doing is not very appropriate in the presence of uncertainty.
Concerning the stability analysis and the stabilization problem, a number of approaches assuming or not the regularity of the descriptor system have been proposed in the literature. Let us quote for instance [2,4,12] among those assuming the regularity and [4,12] among those not assuming the regularity.
Robust control of linear state space systems has been the focus of much attention during the past decades and various aspects and approach for analysis and control design for linear uncertain systems have been investigated, see for instance [15]. In the available literature we easily note that quadratic stability and stabilization approaches have taken a lion's share. The quadratic stability or stabilization is characterized by a determination of a unique so-called Lyapunov matrix which gives the approach an inherent conservatism. Many results have been reported in quadratic stability analysis and/or stabilization (see for instance [1,15,14] and the reference therein).
Recently the Parameter Dependent Lyapunov (PDL) approach has been introduced to reduce the conservatism of the quadratic approach. The PDL approach consists in expressing the Lyapunov matrix as a function of the uncertainty and with the help of some slack additional variables the approach yields a significant reduction of conservatism [5,3].
In this paper, a Linear Matrix Inequality (LMI) formulation is adopted to express necessary and/or sufficient conditions for the admissibility of continuous time descriptor systems. The proposed approach can be understood as the LMI-correspondant formulation of the GLE proposed in [6]. It is known that strict inequality conditions are tractable and reliable especially with the available LMI software solver.
The state feedback stabilization problem is solved by means of the admissibility of the closed loop. When the system contains uncertainties, the present method is extended to solve the robust state feedback admissibility problem particularly through a PDL approach.
This paper is organized as follows. Section 2 gives the problem formulation and some preliminary definitions. Section 3 gives the result on admissibility for continuous time descriptor systems. In section 4, main result to solve the static feedback problem for the nominal descriptor systems is given, whereas Section 5 presents the result for uncertain singular systems. Section 6 presents illustrative examples. Section 7 concludes the paper.

2 Problem formulation

Consider the following continuous time descriptor system

$$E\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input. The matrix E may be singular, we shall assume that $\mathsf{rank}(E) = r \leq n$. A and B are known real constant matrices with appropriate dimensions.

Definition 1. *[4]*

1. *The pair (E, A) is said to be regular if $det(sE - A)$ is not identically zero.*
2. *The pair (E, A) is said to be impulse free if $deg(det(sE - A)) = \mathsf{rank}(E)$.*

In the rest of the paper the notation is standard unless it is otherwise specified. $L > 0$ ($L < 0$) means that the matrix L is a symmetric and positive definite matrix (a symmetric and negative definite). In the sequel, $\mathsf{Sym}\{\cdot\}$ is defined as $\mathsf{Sym}\{X\} = \left(X + X^\top\right)$ for any matrix X.

It is worth noting that the stability property for conventional systems is no more sufficient for singular systems but completed by the regularity and the absence of impulses and this lead us to introduce the notion of admissibility.

Definition 2. *[4] [7]*
The continuous time singular system (1) is said to be admissible if it is regular, impulse free and stable.

In order to characterize the admissibility of a singular system let us recall that for a pair (E, A) there exists a transformation couple (U, V) such as

$$\bar{E} = UEV = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \qquad \bar{A} = UAV = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}.$$

It comes then that the singular system is said to be admissible if

$$\text{matrix } \ \bar{A}_{22} \ \text{ is non singular} \tag{2}$$

and there exits a symmetric and positive definite matrix $\bar{X}_{11}$ such that

$$\mathsf{Sym}\left\{((\bar{A}_{11} - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{A}_{21})\bar{X}_{11})\right\} < 0. \tag{3}$$

Indeed, condition (2) means that the considered system is regular and impulse free whereas condition (3) states that matrix $(\bar{A}_{11} - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{A}_{21})$ is stable. Notice that conditions (3–2) are not tractable for uncertain systems and it is preferable to use directly the system matrices. Both conditions will be combined in a unique condition as in [6]. The associated Lyapunov matrix will be neither symmetric

nor positive definite. The positivity will be required only on a fraction of the Lyapunov matrix in order to satisfy condition (3).

To solve the admissibility problem, we propose a Lyaponuv-type admissibility condition, which is expressed by a strict LMI as given in Theorem 1. The goal of this paper is to find a static state feedback controller $u(t) = Kx(t)$ such that the closed loop system $(E, A + BK)$ is admissible. This is defined, in this paper, as the static feedback admissibility problem of descriptor systems. The solvability of the above problems will be characterized by some LMI conditions. If the derived LMI conditions are feasible, the feedback gain matrix can be obtained. If the system contains polytopic uncertainties, the results can be modified to find the static state feedback gain in such a way that the closed loop uncertain system is admissible.

3 Admissibility analysis

Consider the singular system described by the pair (E, A) and define $E^{\perp}$ and $E^{\dagger}$ as follows

$$E^{\perp} = V\,(I - UEV)\,U, \qquad E^{\dagger} = U^{\top}\,(I - UEV)\,U^{-\top}, \tag{4}$$

with U and V, two non singular matrices satisfying

$$UEV = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}.$$

Theorem 1. *The continuous time singular system (E, A) is admissible if and only if there exist some matrices $X = X^{\top}$, Y and G such that condition*

$$\begin{bmatrix} 0 & \left(XE^{\top} + E^{\perp}YE^{\dagger}\right)^{\top} \\ \left(XE^{\top} + E^{\perp}YE^{\dagger}\right) & 0 \end{bmatrix} + \mathsf{Sym}\left\{ \begin{bmatrix} A \\ -I \end{bmatrix} G \begin{bmatrix} U^{-\top}V^{\top} \end{bmatrix} \right\} < 0 \tag{5}$$

is feasible.

Proof of Theorem 1 :
By virtue of [10, Theorem 2.3.12] condition (5) is equivalent to

$$\begin{bmatrix} I & A \end{bmatrix} \begin{bmatrix} 0 & \left(XE^{\top} + E^{\perp}YE^{\dagger}\right)^{\top} \\ \left(XE^{\top} + E^{\perp}YE^{\dagger}\right) & 0 \end{bmatrix} \begin{bmatrix} I \\ A^{\top} \end{bmatrix} = \mathsf{Sym}\left\{AXE^{\top} + AE^{\perp}YE^{\dagger}\right\} < 0 \tag{6}$$

$$\begin{bmatrix} U & -V^{-1} \end{bmatrix} \begin{bmatrix} 0 & \left(XE^{\top} + E^{\perp}YE^{\dagger}\right)^{\top} \\ \left(XE^{\top} + E^{\perp}YE^{\dagger}\right) & 0 \end{bmatrix} \begin{bmatrix} U^{\top} \\ -V^{-\top} \end{bmatrix} = -\mathsf{Sym}\left\{\bar{X}\bar{E}^{\top} + \bar{E}^{\perp}\bar{Y}\bar{E}^{\dagger}\right\} < 0 \tag{7}$$

with

$$\begin{aligned}
\bar{X} &= V^{-1}XV^{-\top},\\
\bar{Y} &= UYU^{\top},\\
\bar{E} &= UEV,\\
\bar{E}^{\perp} &= V^{-1}E^{\perp}U^{-1} = I - UEV,\\
\bar{E}^{\dagger} &= U^{-\top}E^{\dagger}U^{\top} = I - UEV.
\end{aligned}$$

Note that condition (7) implies that the 11-block of $\bar{X}$ is positive definite.

First we will show that matrix $\bar{A}_{22}$ is non singular or in other words the considered system is regular and impulse free. For this, multiplying both sides of (6) by U and $U^{\top}$ respectively, we get

$$\begin{aligned}
&\mathsf{Sym}\left\{UAXE^{\top}U^{\top}\right\} + \mathsf{Sym}\left\{UAE^{\perp}YE^{\dagger}U^{\top}\right\}\\
&= \mathsf{Sym}\left\{\bar{A}\bar{X}\bar{E}^{\top}\right\} + \mathsf{Sym}\left\{\bar{A}\bar{E}^{\perp}\bar{Y}\bar{E}^{\dagger}\right\}\\
&= \mathsf{Sym}\left\{\begin{bmatrix} * & 0\\ * & 0\end{bmatrix}\right\} + \mathsf{Sym}\left\{\begin{bmatrix} 0 & *\\ 0 & (\bar{A}_{22}\bar{Y}_{22})\end{bmatrix}\right\} < 0
\end{aligned}$$

which shows that we have necessarily

$$\mathsf{Sym}\left\{\bar{A}_{22}\bar{Y}_{22}\right\} < 0$$

and this means that $\bar{A}_{22}$ is invertible or in other words that the system is regular and impulse free. Note that the stars here corresponds to terms with no much relevance at this step.

To prove the stability of the singular system, we consider the two matrices

$$\Sigma = \begin{bmatrix} I & 0\\ -\bar{A}_{22}^{-1}\bar{A}_{21} & \bar{A}_{22}\end{bmatrix} \qquad \text{and} \qquad \Gamma = \begin{bmatrix} I & -\bar{A}_{12}\bar{A}_{22}^{-1}\\ 0 & I\end{bmatrix}$$

that transform the matrix $\bar{A}$ in a diagonal form as

$$\bar{\bar{A}} = \Gamma\bar{A}\Sigma = \begin{bmatrix} \bar{\bar{A}}_{11} & 0\\ 0 & I\end{bmatrix}$$

with

$$\bar{\bar{A}}_{11} = \bar{A}_{11} - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{A}_{21}.$$

If we multiply both sides of (6) by ΓU and its transpose we get

$$\mathsf{Sym}\left\{\bar{\bar{A}}\bar{\bar{X}}\bar{\bar{E}}^{\top}\right\} + \mathsf{Sym}\left\{\bar{\bar{A}}\bar{\bar{E}}^{\perp}\bar{\bar{Y}}\bar{\bar{E}}^{\dagger}\right\} = \mathsf{Sym}\left\{\begin{bmatrix} \bar{\bar{A}}_{11}\bar{X}_{11} & 0\\ * & 0\end{bmatrix}\right\} + \mathsf{Sym}\left\{\begin{bmatrix} 0 & 0\\ * & *\end{bmatrix}\right\} < 0$$

with

$$\begin{aligned}
&\bar{\bar{E}} = \Gamma \bar{E} \Sigma = \bar{E}, \\
&\bar{\bar{X}} = \Sigma^{-1} \bar{X} \Sigma^{\top -1} = \begin{bmatrix} \bar{X}_{11} & * \\ * & * \end{bmatrix}, \\
&\bar{\bar{E}}^{\perp} = \Sigma^{-1} \bar{E}^{\perp} \Gamma^{-1} = \bar{E}^{\perp}, \\
&\bar{\bar{Y}} = \Gamma \bar{Y} \Gamma^{\top}.
\end{aligned}$$

Then, it comes that we have

$$\mathsf{Sym}\left\{\bar{\bar{A}}_{11}\bar{X}_{11}\right\} < 0 \tag{8}$$

which means that the singular system is stable since matrix $\bar{X}_{11}$ is symmetric and positive definite according to condition (7) and this ends the proof of the theorem. $\nabla\nabla\nabla$

4 Admissibilization by state feedback

In this section we address the problem of admissibilisation by state feedback for the singular system given by

$$E\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input.

The control law given by a state feedback is then

$$u(t) = Kx(t) \tag{9}$$

where the gain K, of appropriate dimension is computed in such a way that the singular closed loop system is admissible.

Theorem 2. *The continuous singular system is admissibilizable if and only if the* LMI *problem*

$$\begin{bmatrix} 0 & \left(XE^{\top} + E^{\perp}YE^{\dagger}\right)^{\top} \\ \left(XE^{\top} + E^{\perp}YE^{\dagger}\right) & 0 \end{bmatrix} + \mathsf{Sym}\left\{\begin{bmatrix} AG + BR \\ -G \end{bmatrix} \begin{bmatrix} U^{-\top} V^{\top} \end{bmatrix}\right\} < 0 \tag{10}$$

is feasible in the variables X, Y, R and G. Moreover, the state feedback is given by

$$K = RG^{-1}. \tag{11}$$

Proof of Theorem 2

The closed loop singular system is admissible if and only if there exist two matrices X and Y and a state feedback K such that

$$\begin{bmatrix} 0 & \left(XE^\top + E^\perp Y E^\dagger\right)^\top \\ \left(XE^\top + E^\perp Y E^\dagger\right) & 0 \end{bmatrix} + \mathsf{Sym}\left\{ \begin{bmatrix} (A+BK) \\ -I \end{bmatrix} G \begin{bmatrix} U^{-\top} & V^\top \end{bmatrix} \right\} < 0 \quad (12)$$

and it becomes clear that a change of variable $R = KG$ yields condition (10). This ends the proof by virtue of Theorem 1. $\nabla\nabla\nabla$

From Theorem 2 we can easily be aware that this approach has the advantage to be nicely applicable in the case the system is uncertain with a polytopic description of uncertainty which is the subject of the next section.

5 Robust Admissibilisation

In this section, we consider the feedback admissibilization problem for systems containing uncertainties. The descriptor system is characterized by the pair $(E,\ A(\bar{\alpha}), B(\bar{\alpha}))$ where matrices $A(\bar{\alpha})$ and $B(\bar{\alpha})$ belong to a polytope, that is

$$\begin{bmatrix} A(\bar{\alpha}) & B(\bar{\alpha}) \end{bmatrix} = \sum_{i=1}^{p} \alpha_i \begin{bmatrix} A_i & B_i \end{bmatrix}$$

with

$$\alpha_i \geq 0,\ i = 1,\ \ldots,\ p,$$
$$\sum_{i=1}^{p} \alpha_i = 1 \qquad \text{and} \qquad \bar{\alpha} = \begin{bmatrix} \alpha_1 & \ldots & \alpha_p \end{bmatrix}.$$

The problem for a system corrupted by uncertainty is to preserve its performances for all admissible uncertainties or in other terms for every instance of matrices $A(\bar{\alpha})$ and $B(\bar{\alpha})$.

Definition 3. *The uncertain singular system is robustly admissible if and only if it is admissible for every instance of the uncertainty* $\bar{\alpha}$.

Precisely, the continuous singular system will be robustly admissible if the 22-block of $UA(\bar{\alpha})V$ is invertible and all the finite poles are located in the open left half plane, that is, there exists a symmetric positive definite matrix $\bar{X}_{11}(\bar{\alpha})$ such that

$$\mathsf{Sym}\left\{((\bar{A}_{11}(\bar{\alpha}) - \bar{A}_{12}(\bar{\alpha})\bar{A}_{22}^{-1}(\bar{\alpha})\bar{A}_{21}(\bar{\alpha}))\bar{X}_{11})(\bar{\alpha})\right\} < 0.$$

In this case, the continuous singular system will be termed as robustly admissible.

Unfortunately, solving the previous equation is out of reach for the time being and instead one can thought of matrix $\bar{X}_{11})(\bar{\alpha})$ as a unique matrix over the uncertainty set, that is condition

$$\mathsf{Sym}\left\{((\bar{A}_{11}(\bar{\alpha}) - \bar{A}_{12}(\bar{\alpha})\bar{A}_{22}^{-1}(\bar{\alpha})\bar{A}_{21}(\bar{\alpha}))\bar{X}_{11})\right\} < 0$$

holds with of course the invertibility of matrix $\bar{A}_{22}$ then in this case the continuous singular system will be termed as quadratically admissible.

It is worth noting that quadratic admissibility implies robust admissibility but the converse is, in general, false. If quadratic admissibility is only sufficient for robust admissibility, it matters to find (sufficient) conditions for robust admissibility that are less conservative than quadratic admissibility. In other words, it is important to derive conditions implicitly involving matrices X and Y that are not constant but dependent on the uncertainty. One possibility is that matrices X and Y comply with

$$[\, X \quad Y \,] = \sum_{i=1}^{p} \alpha_i \, [\, X_i \quad Y_i \,], \tag{13}$$

where X_i and Y_i are valid to assess the admissibility of extreme models. Based upon these notions, the next theorem is proposed:

Theorem 3. *The uncertain closed loop singular system is robustly admissible if there exist matrices X_i, Y_i, for $i = 1, \ldots, p$, and two matrices G and R such as the* LMI *problem*

$$\begin{bmatrix} 0 & \left(X_i E^\top + E^\perp Y_i E^\dagger\right)^\top \\ \left(X_i E^\top + E^\perp Y_i E^\dagger\right) & 0 \end{bmatrix} + \mathsf{Sym}\left\{ \begin{bmatrix} A_i G + B_i R \\ -G \end{bmatrix} [\, U^{-\top} \, V^\top \,] \right\} < 0 \tag{14}$$

is feasible. The admissible feedback gain is then given by

$$K = RG^{-1}.$$

Proof of Theorem 3 :

Let matrices X and Y be defined as in (13), then multiplying conditions (14) by α_i and summing up from 1 to p one gets the same conditions as in Theorem 2. $\nabla\nabla\nabla$

6 Illustrative example

Example 1. Consider a continuous time descriptor system as in (1) described by the parameters :

$$E = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 2 \\ -1 & 1 & 0 \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 4 & 4 \\ -1 & 2 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.0 & 0.2 \\ 1.0 & 0.2 \\ 0.9 & 0.8 \end{bmatrix}.$$

The finite poles are located at 0.3820 and 2.6180 which implies that the open loop is not admissible.

Taking

$$U = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{bmatrix} \quad \text{and} \quad V = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

and applying Theorem 1 we obtain :

$$X = \begin{bmatrix} -1.3116 & -1.2550 & 1.3530 \\ -1.2550 & -1.1265 & 1.2720 \\ 1.3530 & 1.2720 & -1.3670 \end{bmatrix} \times 10^4,\ Y = \begin{bmatrix} 0.6075 & 2.5598 & 1.4604 \\ -0.3609 & 1.5548 & 0.6442 \\ 1.4604 & 0.6442 & -3.1074 \end{bmatrix} \times 10^4,$$
$$G = \begin{bmatrix} -3.8214 & 7.6429 & -3.8498 \\ -7.2060 & 4.3087 & 4.2920 \\ 3.8544 & -4.2934 & 0.0071 \end{bmatrix} \times 10^4,\ R = \begin{bmatrix} 1.3466 & 0.3268 & -2.0933 \\ -0.2569 & 0.4626 & -0.2203 \end{bmatrix} \times 10^5$$

and the state feedback gain :

$$K = RG^{-1} = \begin{bmatrix} 2.5061 & -2.6310 & 1.0596 \\ -1.9832 & -2.2806 & -6.8965 \end{bmatrix}$$

which renders the resulting closed loop system admissible with the finite closed loop poles located at -0.9965 and -0.2841 and the 22-block of UAV equaling -2.7512.

Example 2. Consider the uncertain singular system defined by

$$E = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 2 \\ -1 & 1 & 0 \end{bmatrix}$$

with matrices A and B belonging to a polytope whose vertices are given by

$$[\, A_1 \quad A_2 \,] = \left[\begin{array}{ccc|ccc} 1 & -1 & 0 & -1.0000 & 0.0680 & -0.8158 \\ 3 & 0 & 4 & -0.2500 & 0.6842 & 2.9136 \\ 1 & 4 & 6 & 1.7500 & 0.5482 & 4.5452 \end{array}\right],$$

$$[\, A_3 \quad A_4 \,] = \left[\begin{array}{ccc|ccc} 2.0000 & 0.0800 & 2.2162 & -2.0000 & 0.0672 & -1.8166 \\ 2.8900 & 0.7562 & 6.3456 & -1.4500 & 0.6834 & 1.4728 \\ -1.1100 & 0.5962 & 1.9132 & 2.5500 & 0.5490 & 5.1060 \end{array}\right].$$

The input matrices

$$[\, B_1\, B_2\, B_3\, B_4 \,] = \left[\begin{array}{cc|cc|cc|cc} 0.0122 & 0.0422 & 0.0122 & 0.0402 & 0.0122 & 0.0422 & 0.0122 & 0.0402 \\ 0.3590 & 0.1652 & 0.3750 & 0.1632 & 0.3990 & 0.1652 & 0.3350 & 0.1632 \\ 0.5291 & 0.2949 & 0.5591 & 0.3289 & 0.5971 & 0.3349 & 0.4911 & 0.2889 \end{array}\right]$$

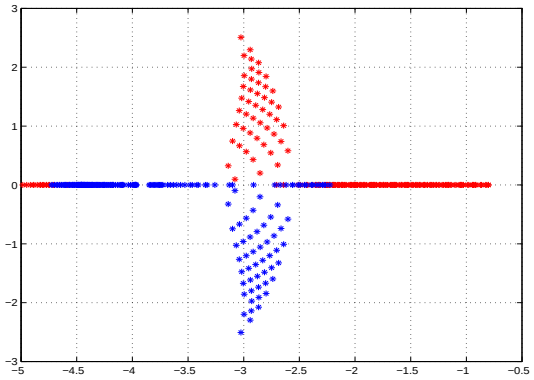


Fig. 1. Finite eigenvalues location of the uncertain closed loop system over the considered polytope

According to Theorem 3, the robust state feedback gain

$$K = \begin{bmatrix} 39.4509 & -16.8080 & 16.9447 \\ -75.3237 & 15.2275 & -59.6286 \end{bmatrix}$$

will renders the uncertain closed loop system robustly admissible.

7 Conclusion

The problem of admissibility and admissibilization for continuous time descriptor systems have been studied. In terms of a strict LMI, a necessary and sufficient condition for continuous descriptor systems to be admissible has been proposed. This condition is an LMI twin formulation of the well known improved generalized Lyapunov equation. LMI conditions are obtained to ensure the admissibility of the closed loop via a state feedback control law. A robust admissible state feedback control law is proposed for polytopic uncertain continuous descriptor systems. Numerical examples are given to illustrate the usefulness of the proposed methods.

References

1. B.-R. Barmish. Necessary and sufficient conditions for quadratic stabilizability of an uncertain system. *J. Optim. Theory Applicat.*, 46:399408, 1985.
2. P. Bernhard. On singular implicit linear dynamical systems. *SIAM Journal of Control Optimization*, 20:612–633, 1982.
3. J. Daafouz and J. Bernussou. Parameter dependent Lyapunov functions for discrete-time systems with time-varying parametric uncertainties. *Syst. Control Lett.*, 43:355359, 2001.

4. L. Dai. *Singular control systems*. Lecture notes in control and information sciences. Springer Verlag, Berlin, 1989.
5. P. Gahinet, P. Apkarian, and M. Chilali. Affine parameter-dependent Lyapunov functions and real parametric uncertainty. *IEEE Trans. Automat. Contr.*, 41:436442, 1996.
6. J.-Y. Ishihara and M.-H. Terra. On the Lyapunov theorem for singular systems. *IEEE Trans. on Automatic Control*, 47:1026–1930, 2002.
7. C. H. Kuo and C. H. Fang. An LMI approach to admissibilization of unceratin descriptor systems via static output feedback. *Proceedings of the American Control Conference*, FP06, 2003.
8. F. L. Lewis. A survey of linear singular systems. *Circuits systems Sig. Prog.*, 5:266–36, 1986.
9. I. Masubuchi and Y. Kamitane. H_∞ control for descriptor systems: A matrix inequality approach. *Automatica*, 33:669–673, 1997.
10. R.-E. Skelton, T. Iwasaki K., and Grigoriadis. *A Unified Algebraic Approach to Linear Control Design*. Taylor-Francis, Bristol-USA, 1998.
11. K. Takaba, N. Morihira, and T. A. Katayama. Generalized Lyapunov theorem for descriptor system. *Systems and Control Letters*, 24:49–51, 1995.
12. A. Varga. On stabilization methods of descriptor systems. *Systems & Control Letters*, 24:133–138, 1995.
13. S. Xu and J. Lam. Robust stability and stabilization of discrete singular systems : An equivalent characterization. *IEEE Trans. on Automatic Control*, 49:568–573, 2004.
14. K. Zhou, J.-C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River, NJ, 1996.
15. K. Zhou and P.-P. Khargonekar. Robust stabilization of linear systems with norm-bounded time-varying uncertainty. *Sys. Control Lett.*, 10:1720, 1988.

Computer Architecture

Design of RBSD Adder and Multiplier Circuits for High Speed Arithmetic Operations and their Timing Analysis

Neelam Sharma, B.S. Rai and Arun Kumar

Sr. Lecturer, E&C Department, Institute of Technology and Management,
Gida, Gorakhpur, India
neelam_sr@yahoo.com

Abstract. RBSD Adder / RBSD Multiplier circuits are logic circuits designed to perform high-speed arithmetic operations. These high-speed arithmetic machines add and multiply numbers using Redundant Binary Signed Digit number system. In RBSD number system carry propagation chain are eliminated which reduce the computation time substantially, thus enhancing the speed of the machine. A novel design of RBSD Adder cell made of NOR gates only are proposed in this paper. A pipelined design of Multiplier cell using RBSD Adders is also suggested. The speed of these high-speed circuits is compared to the speed of CLA adder / Wallace Booth Multiplier (WBM) which are being used conventionally for high-speed calculation. Comparison yields favorable results although RBSD machines may not be convenient for manual computations but they are of great advantage for design of high-speed machines. The design of these circuits is carried out using EDA tools. The designs are simulated and synthesized using VHDL software, more specifically modelsim software is used for simulation, Xilinx project navigator and Leonardo Spectrum for synthesis. Timing analysis reports for these circuits and graphs drawn for delay time give favorable comparative results.

Index Terms: RBSD, CLA adder, WB Multiplier, Carry free addition, high speed arithmetic, carry chain, VHDL.

1 Introduction

Arithmetic operations play an important role in various digital systems such as computers, process controllers, signal processors computer graphics and image processing. Using non-conventional number system such as signed digit numbers for fast arithmetic units is particularly gaining much attention in recent years. Signed digit number system offers the possibility of carry free addition by taking advantage of redundancy associated with the representation. Several papers have been published in recent years exploring the use of signed digit numbers for fast arithmetic units. With recent development in technology of integrated circuits, various high speed circuits with regular structures have been proposed and some of them have been fabricated on VLSI chips since many digital systems are designed to operate fast with

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 243-254

Received 24/06/06
Accepted 07/10/06
Final version 13/10/06

high reliability, not only high speed operations and regular structures but also fault tolerant features should be implemented in arithmetic circuits.

RBSD number representation posses sufficient redundancy to allow for the annihilation of carry or borrow chains and hence result is past, propagation – free addition and subtraction. Signed – digit number not only allows for carry – free addition and borrow – free subtraction but also offers important advantages for the practical implementation of arithmetic functions. These numbers are in practical use for representing intermediate values of in 2's complement and high speed multiplication schemes ever since multiplier recording was introduced by Booth. They have also been used in redundant quotient representation for the S-R-T division algorithm which was proposed independently by Sweeney, Robertson, and Tocher and which drives it's name from their initials

VLSI Technology has given designers the freedom to integrate many complex components, which may not be possible in the earlier days. Multiplication is nothing but repetitive addition of partial products. It involves two basic operations 1). The generation of partial products 2) their accumulation to produce the final product. Hence if fast multiplication is desired either the numbers of partial products are reduced or the time needed to accumulate or add them is decreased.

Modified Booth Encoding (MBE) in which the partial products are reduced to half the original value is used.. After generation of partial products, they must be accumulated to obtain the final product. Fast multi-operand adders such as Wallace trees and Carry Save Adders (CSA) tree should be employed for high speed accumulation [4] . However, tree structures generally have very irregular interconnections that complete the implementation and more importantly result in Area inefficient layouts. RBSD Adders are more suitable for fast addition. They take a finite amount of time to generate the result. Since a three-input/two-output adder (3-2 Compressor, also called CSA) is most commonly used in Wallace trees, the partial product reduction rate is 3/2. The reduction rate of 2/1 can be accomplished by using RBSD Adders. Wallace trees, CLA adders and CSA adders are conventionally used for high speed addition. Similarly Wallace Booth Multiplier is used for fast multiplication. In this paper it is proved by virtue of simulation results that RBSD adder/multiplier are superior to the adders/multipliers used conventionally for high speed addition/multiplication. Adders/Multipliers of 4, 8, 16, 32, 64 bit are simulated and synthesized using EDA (Electronic Design Automation) softwares and the results yielded are discussed in the paper.

Timing analysis of high speed RBSD adder will be compared with that of CLA adder. Similarly, timing analysis of high speed RBSD multiplier will be carried out and compared with conventional Wallace booth multiplier.

2 Salient Features of RBSD Numbers

The following additional advantages are offered by RBSD numbers other than their carry-free addition property.

- Ease of multiplication, division, and other arithmetic operations.
- Ease of zero detection.

- Suitability of use with error codes.
- Pipelined approach.

The stored-carry representation was derived in connection with the design of past multi operand adders. Even though other arithmetic operations such as multiplication and division are performed on stored-carry numbers, the required hardware is much move complex than those of RBSD. More so the regular structure of the RBSD adder cell proposed by Kal and Rajshekhar makes the design regular, pipelined and suitable for VLSI implementation.

RBSD number system's suitability for use with arithmetic error codes is a direct result of the compatibility of this representation with Binary [10]. Particularly the effective studies performed for arithmetic error codes in connection with the binary system carry over with little change to the RBSD system unlike stored-carry system. Zero has a unique RBSD representation and can thus be easily detected when needed. This is also possible in stored carry representation but requires more hardware.

2.1 Background

The credit of redundant signed digit goes to Robertson (1959) and Avizienis (1961) [3]. They suggested a set of rules for RBSD Arithmetic. Chow and Robertson suggested the idea of logic design of the RBSD adders. Takaji and Azima proposed high speed algorithm with RBSD numbers suitable for VLSI implementation.

2.2 Redundant Signed Digit Number System Preliminaries

Signed digit representation limit carry propagation to one position to the left during the operation of addition and subtraction in digital computers. Carry-propagation chains are eliminated. Redundancy in the number representation allows a method of fast addition and subtraction in which each sum digit is a function only of the digits in two adjacent digital position. In RD representation with radix r a digit can have more than r values. SD (Signed Digit) number can assume $2\alpha+1$ values.

$$\Sigma_r = \{-\alpha,\ldots.-1,0,1,\ldots\ldots\ldots\ \alpha\} \tag{1}$$

α must be within the following region

$$\begin{gathered}(r-1)/2 <= \alpha <= r-1 \\ \alpha >= 1\ \&\ r >= 2\end{gathered} \tag{2}$$

A SD number is represented by n+m+1 digits

$$Z_i = (i=-n,\ldots\ldots\ldots -1,0,1,\ldots\ldots\ldots m) \tag{3}$$

has the algebraic value m
And equivalent SD number
$Y=(Y_{n-1},\ldots..,Y_1,Y_0)_r$
For every difference digit

$$d_i = x_i - r b_{i+1} \tag{4}$$

where the borrow digit

$$b_{i+1} = \begin{cases} 0 \text{ if } x_i < \alpha \\ 1 \text{ if } x_i >= \alpha \end{cases} \tag{5}$$

Let the number 12 be changed to RBSD representation and back to binary. Note that X_i stands for RBSD number n and Y_i for binary form of the number.
$X=(1100)_2$ with r=2, n=4 and α=1 for a given SD set (-1,0,1) The SD number Y is $(10\ \bar{1}00)$ as obtained in the Table1.

Table 1 : Binary to RBSD conversion

Digit Position	4	3	2	1	0
X_i		1	1	0	0
d_i		$\bar{1}$	$\bar{1}$	0	0
b_i	1	1	0	0	
Y_i	1	0	$\bar{1}$	0	0

12 is represented as 10 100 i.e.16-4 as bar stands for -1
Here d_i and b_i represent the sum and borrow digits and are same as in equations (4) and (5) respectively.
Conversion back to conventional radix

$Y_i = Y^+ + Y^-$
$Y = (1\ 0\ \bar{1}\ 0\ 0)_2$
$Y^+ = (1\ 0\ 0\ 0\ 0)_2$ -> positive ones are replaced by 1
$Y^- = -(0\ 0\ 1\ 0\ 0)_2$ -> negative ones are replaced by 1

$X_i = (0\ 1\ 1\ 0\ 0)_2$ ->Subtract to obtain X
X_i= back to conventional binary

3 RBSD Adder

RBSD adder structure with a digit set {-1, 0, 1} per addition operation by "Two Transfer Addition Technique" which is computed in three stages

$$x_i + y_i = 2*t'_{i+1} + w'_i \tag{6}$$

$$w'_i + t'_i = 2*t''_{i+1} + w''_i \tag{7}$$

$$s_i = w''_i + t''_i \tag{8}$$

where x_i and y_i e {1, 0, 1}

t''_i and t'_i are intermediate carriers called transfer digits of i^{th} stage. w'_i and w''_i are intermediate sums in the i^{th} stage.

S_i is the final sum [1,2] . The sum of operands is realized using the following steps:

Step-1: The transfer digit $|t'_{i+1}|=1$ wherever $|x_i+y_i|>=1$

Step-2: The transfer digit |t"I+1|=1 only if |t'+w'|>=2 is completed and w" is computed using formula (2)

Step-3: s_i=w"+t"

Under the above conditions w" and t" cannot be both +1 and –1 thus giving carry free addition. To implement the circuit the chosen set (1, 0), (0,0) and (0,1) to represent (-1), (0) and (1) respectively. RBSD adder cell for addition of two bit RBSD numbers was generated regular structures of RBSD adder cell is an added advantage over irregular structure of CLA adder cell in VLSI implementation of the structure. CLA adders circuits implemented for fast addition suffer from fan-in and fan-out limitations. These limitations depend on logic families employed, TTL circuits for example are limited to n=12. If m_i, b_i and d_i are binary variables with digit set {0,1} and x_i* y_i* s_i* are variables of digit set {-1,0,1} represented by two bit binary (1,0),(00),(0,1) respectively. For this chosen representation corresponding to x_i* y_i* and s_i* are $x_i^+ x_i^-$, $y_i^+ y_i^-$ and $s_i^+ s_i^-$ respectively. For this chosen representation, the Boolean expressions for d_i, m_{i+1} b_{i+1}, s_i^+ and s_i^- are as follows :

$$d_i = m_i \oplus \bar{x}_i^+ \bar{x}_i^- \oplus \bar{y}_i^+ \bar{y}_i^- \tag{9}$$

$$m_{i+1} = \bar{x}_i^+ \bar{y}_i^+ \tag{10}$$

$$b_{i+1} = \bar{m}_i \bar{x}_i^+ \bar{x}_i^- + x_i^+ y_i^+ + y_i^+ \bar{y}_i^- \bar{m}_i + \bar{x}_i^+ \bar{x}_i^- \bar{y}_i^+ \bar{y}_i^- \tag{11}$$

$$s_i^+ = d_i \bar{b}_i \tag{12}$$

$$s_i^- = d_i \bar{b}_i \tag{13}$$

The RBSD Adder cell used to add two bit RBSD numbers ($x_i^+ x_i^-$) and ($y_i^+ y_i^-$) suggested by Kal Rajashekhar are shown in figure (1) [2]. The adder structure shown in the figure is used to perform parallel addition of two numbers four bit RBSD numbers.

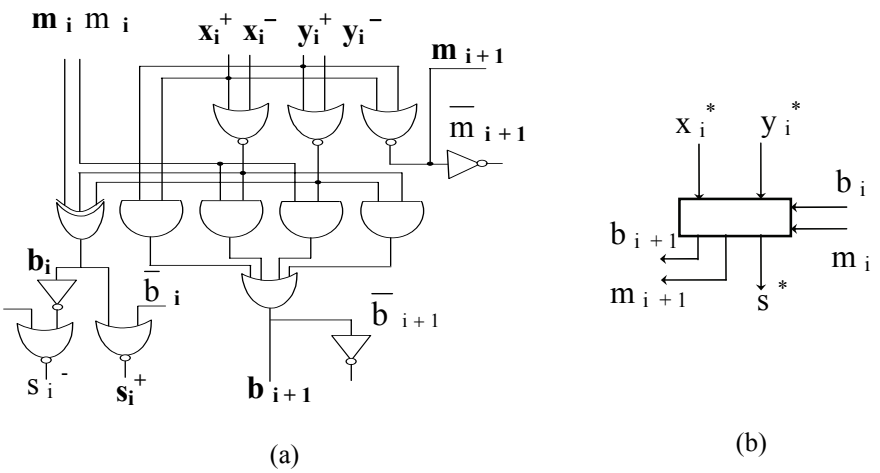


Figure 1. (a) Logic diagram of RBSD adder cell,
(b) Block diagram of RBSD adder cell

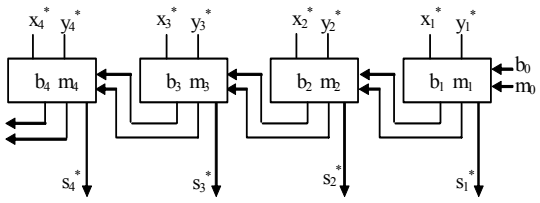


Figure 2. Block diagram of addition of two four bit RBSD numbers

3.1 Logic Design of RBSD Adder using Universal Logic

Kal and Rajashekhar designed an RBSD Adder cell made of AND, NOR, XOR , OR, and INVERTERS. However NAND and NOR operations are each functionally complete. It is highly desirable to construct digital circuits of NAND or NOR Gates because of simplicity and uniformity of circuits which have just a single primitive component [13] . And since these gates can be manufactured quite in expensively they contribute to the major components used today by logic designers. NAND and NOR gates are also available as integrated circuit SSI packages. These packages vary in sizes and may content from one to four gates per package. In digital design techniques, much attention has to be paid to network design in the form of repeated pattern of identical circuits. Having a similar type of gate (NAND or NOR) also makes nMOS and CMOS design easy for the designer says there is a repetition of similar design.

The proposed Adder cell in this paper is made of NOR gates. This Adder design is verified using PSPICE software and it's VHDL code is also generated. VHDL code for the Adder cell proposed by Kal and Rajashekhar is also generated. Both the

designs of RBSD Adder cells are simulated and synthesized and the Timing Reports generated. The time delay for the RBSD Adder proposed by Kal and Rajashekhar is 12.11 ns and for the NOR RBSD Adder cell proposed and given in figure (2) is 10.532 ns. Thus this design proved faster. Adder cells are the building blocks of multipliers and can therefore this design can increase the speed of the multiplier multifold.

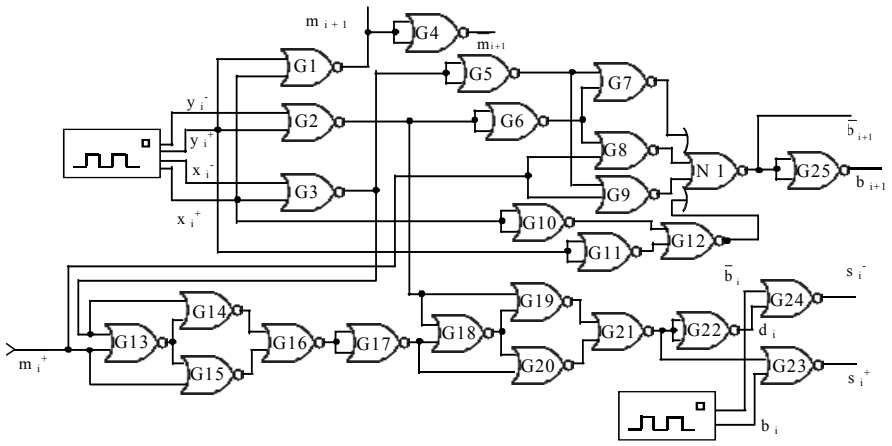


Figure 3. RBSD Adder Cell using NOR-NOR Logic

3.2 Comparison between RBSD Adder Cells

The two different types of RBSD Adder cells 1) Ordinary (the one suggested by Kal and Rajashekhar) 2) NOR-NOR – Proposed cells can be compared. The Adder structure have the same inputs and outputs but they only differ in type of gates that they are made of. The table below list the gate types and no. of gates used for proposed Adder and the Adder suggested by Kal and Rajashekhar. It also gives the time delay and hardware used on FPGA.

3.3 Comparative Timing Analysis of CLA and RBSD adder

VHDL implementations of CLA and RBSD adders were carried out. N bit RBSD adder structure and CLA adder were simulated with modelsim software from FPGA advantage. These adders were synthesized with Xilinx project navigator and their timing reports viewed for values n=4, 8, 16, 32 and 64.

A comparative graph of CLA and RBSD adder structures is shown below. We observe that the delay produced in RBSD adder is independent of wavelength of the string to be added and has regular structure. Hence for higher values of n RBSD adders structure is ideal for VLSI implementation.

It is observed from table 2 that the delay produced by RBSD Adder is less as compared to CLA Adder for bit lengths 8, 16, 32 & 64. RBSD Adders are most suited for higher bit lengths.

Table 2. Comparative table of no. gates and gate types along with the time delay and hardware used for different RBSD Adder cells.

S. No.	Type of RBSD Adder Cell	No. of gates used	Hardware Used on FPGA	Time Delay (ns)
1	Ordinary RBSD Adder Cell	2 I/P AND – 4 4 I/P OR – 1 2 I/P NOR – 5 3 I/P XOR – 1 INVERTERS - 3	No. of Slices – 5 out 768 No. of 4 I/P LUT's– 9 out of 1536 No. of bonded IOB's – 14 out of 96	12.122 ns (7.091 ns logic 5.031 ns route)
2	NOR-NOR RBSD Adder Cell	2 I/P XOR – 25 4 I/P NOR – 1	No. of Slices – 4 out 768 No. of 4 I/P LUT's – 7 out of 1536 No. of bonded IOB's – 12 out of 96	10.538 ns (6.857 ns logic 3.681 ns route)

Table 3. Timing Analysis of CLA and RBSD Adder

S.No.	No. of Bits	Delay (ns) CLA	Delay (ns) RBSD
1	4	13.6	16.51
2	8	19.5	16.51
3	16	24.73	16.51
4	32	27.5	16.51
5	64	30.43	16.51

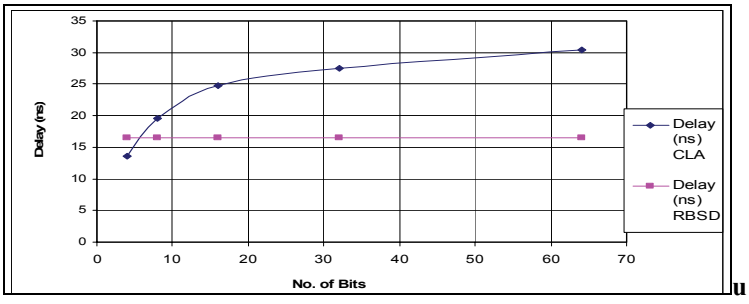


Figure 4. Timing comparison of CLA and RBSD Adder

4 RBSD Multiplier

Multiplication is the most critical operation in every computational system. In the proposed RBSD Multiplier scheme modified booth encoding is used. And n bit multiplication can be carried out with enhanced speed. The multiplier has a regular array structure thus suitable for VLSI implementation [7,8]. The combination of Modified Booth Encoding and Redundant Binary Signed-Digit (RBSD) Adders has been used to implement the multiplier using VHDL.
The various steps involved in the multiplication process are:

- Take two binary numbers one multiplicand and other multiplier
- Convert the multiplicands from two's complement form to RBSD representation (multiplier is retained in 2's complement form)
- Generate n/2 RBSD partial products using Booth's bit pair recording technique on the n-bit two's complement multiplier and the RBSD multiplicand.
- Add the partial products pair wise by means of a binary tree of log2 (n/2) levels of adders.
- Convert the resulting RBSD product into two's complement form.

4.1 Pipelined Design for Multiplication using RBSD Adders

A pipelined system divides the logic into stages separated by a set of latch registers. Pipelining is a technique to provide further increase in bandwidth by allowing simultaneous execution of many tasks. By pipelining the various stages of a RBSD Multiplier structure, its efficiency will increase thus decreasing the computation rate. The major advantage of pipelining over other parallel design techniques is that improvement in performance can be obtained for less cost [11,12] .

The 16 x 16 fixed-point multiplication units can be divided into six pipeline stages. Stage S_1 generates the partial product and stage S_2 converts the partial products to RBSD numbers. Stages S_3, S_4 and S_5 stages add the RBSD numbers. Here of is important to note that since RBSD adder delay time is independent of word length with increased complexity, stages S_3, S_4 and S_5 will have the same delay time. Stage S_6 is a CLA adder used to convert the RBSD number in two's complement form. Delay times of stages S_1, S_2 and S_6 can be further adjusted. The pipeline is thus divided into six stages and the clock rate can be fixed as desired to give appropriate results. Figure (6) shows the pipelined unit of a RBSD multiplier.

4.2 Comparative Timing Analysis of WBM and RBSD Multiplier

VHDL implementation of WBM and RBSD multipliers are carried out and the designs of these multipliers are simulated with the help of modelsim. The designs are synthesized using Xylinx Project Navigator. Timing Analysis report is generated for 4, 4, 16, and 32 bit WBM and RBSD multipliers and a comparative graph is

generated. It is observed that 4 bit WBM is faster than 4 bit RBSD adder but as the bit length of the multiplier and multiplcand to be multiplied increases the delay of RBSD Multiplier is much less than that of WBM. The graph also shows that with the increase of bit length the RBSD Adders prove much faster. Thus even though RBSD operations are unsuitable for manual computations but they are ideal for design of fast ALU.

5 Computation of Elementary Functions

Multipliers can be used as building blocks for computation of elementary functions. Fast RBSD multipliers discussed in this paper can be used for these computations thus enhancing the speed of these circuits as well. The computation of reciprocal and square root has been considered of importance for many years since there functions appear is many applications. Recently inverse square root has also received attention because of increased significance of multimedia and graphics applications [9]. More so, because if its similar characteristics, it is considered advantages to have a single scheme to implement all three functions. Computation of all there elementary functions can be done using small tables, small multipliers, and for since functions, a final large multiplication. The strength of this method is that the same allows the computation of all the functions. Computation of logarithms and Exponentials can also be done using the same scheme. Here it is noted that the same basic computations are performed for all various functions. Computational decay of elementary functions performed by this method is quite close to other related methods, but it is significantly farther for square root and inverse square root.

Table 4. Timing Analysis of WBM and RBSD Multiplier

S.No.	No. of Bits	Delay (ns) WBM	Delay (ns) RBSD
1	4	31.63	41.2
2	8	59.39	55.07
3	16	107.19	63.72
4	32	399.25	229.4

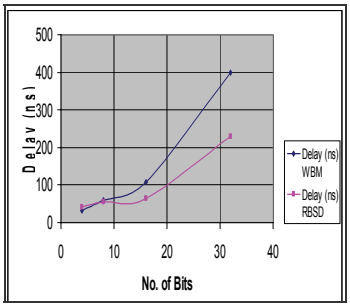


Figure 5. Timing Comparison of WBM and RBSD Multiplier

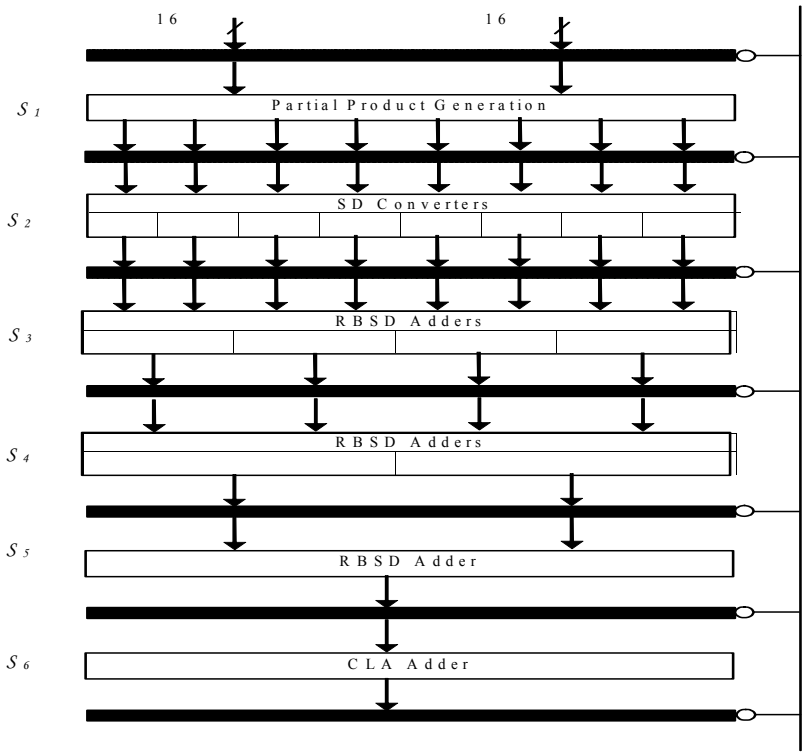


Figure 6. A pipeline unit for fixed-point multiplication of 16 x 16 bit numbers.

6 Conclusion

RBSD Machines are excellent in both the computation speed and regularity in layout. RBSD number system offers faster add times because of carry free addition. Regularity of adder structure is of great advantage for its VLSI implementation. RBSD machines offers higher comparative speed if the number of bits to be added is greater. A 4, 8, 16, 32 and 64 bit conventional high speed machines compared with RBSD arithmetic machines so that higher number of bits show great advantages. Adder and multiplier circuits discussed can be used as building blocks for other RBSD arithmetic machines which is a topic of continued research. If we use ternary logic it offers reduced circuit complexity in both transistor form and interpolations. The design of RBSD Adder cell made of NOR gates only proposed in this paper proved faster than the one proposed by Kal and Rajashekhar. Adder cells are the building blocks of multipliers and can therefore this design can increase the speed of the multiplier multifold. Pipelined Design of Multiplier can also make compact circuits with reduced cost and increased speed. A further effort for implementing RBSD Adder cells with NAND gates only is also been carried out.

Even though this technique can be applied to intermediate computation results in order to speed up conventional binary computers, its main application area is in the design of special purpose arithmetic "engines" which deal with long sequences of computations and / or long operands. Since conventional binary numbers are easily converted to RBSD numbers, and since re-conversion can be accomplished by a single binary subtraction, compatibility between the two (conventional and RBSD) computational systems can be achieved.

References

1. Hwang K.(1979). Computer Arithmetic/Principles, Architecture and Design .New York :Wiley
2. Rajashekhar T.N.and Kal O. (1990). Fast Multiplier Design using Redundant Signed- Digit Numbers, International Journal of Electronics vol .69,No. 3, pp – 359-368.
3. Avizienis A. (1961).Signed digit number representation for fast parallel arithmetic, IRE Trans. Electron Computers.,vol EC-10,pp.389-400
4. Pucknell D.A. and Eshraghian K.(2002).. Basic VLSI Design PHI Third Edition
5. Yeh,Wen-Chang and Jen Chein Wie (2000).High speed Booth encoded parallel multiplier design , IEEE Transaction on Computers,vol.49,No.7 .
6. Darninger J. ,Halthaniany David J.H., Koenemann, B., Lavin M. and others .(2000). EDA is IBM : Past Present and Future, IEEE Transaction on computers Aided Integrated circuit and system, vol. 19, No. 12.
7. Besli N. and Deshmukh R.G(2002). A novel Redundant Binary Signed –Digit (RBSD) Booth's Encoding., Proceedings IEEE Southeastcon
8. Rajashekhar T.N. and I-Shi Eric Chen. (1990). A Fast Adder Design Using Signed-Digit Numbers and Ternary Logic.,Proc.1990 IEEE Southern Tier Technical Conference, pp.187-194,Binngamton,New York.
9. M D Ercegovac, J M Muller, Tisseran ,(2000). Reciporcation, square root, Inverse square root and some Elementary functions using small multipliers, IEEE Transactions on computers, vol 49, No. 7. .
10. Parhami B(1988). 'Carry free Addition of Recorded Binary Signed-Digit Numbers', IEEE Transactions on computers Vol. 37 No. 11, pp. 1470-1476.
11. Hallin T. G. and Flynn M. J. ,(1972)., Pipelining of Arithmetic Functions, IEEE Transactions of computers, pp. 880-885.
12. Jump J. R.and Ahuja S. R,(1978).Effective Pipelining of Digital Systems, IEEE Transaction on computers Vol. C-27, No. 9, pp. 855-865.
13. Kohavi Zvi (2002). Switching and Finite Automata Theory, Tata McGraw-Hill, Second Edition

Author Index
Índice de autores

Editorial Board of the Volume

Comité editorial de volumen

Additional Reviewers

Árbitros adicionales

J.C. Acosta Guadarrama
Carlos Aguilar Ibañez
Bibiane Angerer
Nils Bulling
José Ramón Calvo de Lara
Victor Manuel Darriba Bilbao
Osvaldo Espinosa Sosa
Mario Estrada
Mohamed Abdel Fattah
Edgardo Manuel Felipe Riverón
Karen A. García Gamboa
Lucina Garcia
Edel Garcia Reyes
Fernando Godinez
Abel Gómez
Miguel Gonzalez
Agustín F. Gutiérrez Tornés
Jorge Herrerías
Xiao Hui Jin
Emilio Insfran
Hermann Lehner
Patricio Letelier
Juan Carlos López Pimentel
Fernando Magán Muñoz
Tristan Marc Behrens
Jesús Alberto Martínez Castro
José Francisco Mata Silva
Jaime Mora
Elena Navarro
Martin Nordio
Peter Novak
José Ramón Paramá
Bernhard Peischl
Miguel R. Penabad
Aurora Pérez
Jennifer Pérez
Angeles S. Places
Leticia Flores Pulido
Oleksiy Pogrebnyak
Jose Federico Ramirez
Orion F. Reyes Galaviz
Francisco José Ribadas Pena
Elsa Rubio Espino
Arsenii Rudich
Joseph N. Ruskiewicz
Pedro Sánchez Palma
Francisco José Silva Mata
Marta Soto
Hind Taud
Carlos Vega
Jesus Vilares

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2006.
Printing 500 / Edición 500 ejemplares.