ISSN 1934-7332 (Print)
ISSN 1934-7340 (Online)

Computer Technology and Application

Volume 2, Number 11, November 2011

David Publishing Company
www.davidpublishing.org

From Knowledge to Wisdom

Computer Technology and Application

Volume 2, Number 11, November 2011 (Serial Number 12)

DAVID PUBLISHING

David Publishing Company
www.davidpublishing.org

Publication Information:

Computer Technology and Application is published monthly in hard copy (ISSN1934-7332) and online (ISSN1934-7340) by David Publishing Company located at 1840 Industrial Drive, Suite 160, Libertyville, Illinois 60048, USA.

Aims and Scope:

Computer Technology and Application, a monthly professional academic journal, particularly emphasizes practical application of up-to-date technology in realm of computer and other relevant fields. And articles interpreting successful policies, programs or cases are also welcome.

Manuscripts and correspondence are invited for publication. You can submit your papers via web submission, or E-mail to computer@davidpublishing.org. Submission guidelines and web submission system are available at http://www.davidpublishing.org.

Editorial Office:

1840 Industrial Drive, Suite 160, Libertyville, Illinois 60048
Tel: 1-847-281-9862
Fax: 1-847-281-9855
E-mail: computer@davidpublishing.org

Abstracted / Indexed in:

Database of EBSCO, Massachusetts, USA
Chinese Database of CEPS, Airiti Inc. & OCLC
CSA Technology Research Database
Ulrich's Periodicals Directory
Summon Serials Solutions
Chinese Scientific Journals Database, VIP Corporation, Chongqing, China

Subscription Information:

Price (per year):
Print $450; Online $320; Print and Online $600

David Publishing Company
1840 Industrial Drive, Suite 160, Libertyville, Illinois 60048
Tel: 1-847-281-9862. Fax: 1-847-281-9855
E-mail: order@davidpublishing.org

David Publishing Company
www.davidpublishing.org

Computer Technology and Application

Volume 2, Number 11, November 2011 (Serial Number 12)

Contents

Software and Systems Development

835 **Software Composition of Different Security Level Components**
Frank Tsui, Edward Jung and Sheryl Duggins

843 **Generating Graphic User Interface of Web Applications Using Source Code Generator Based on Dynamic Frames**
Danijel Radošević, Denis Bračun and Ivan Magdalenić

849 **Intelligent Tracking Telescope Using Embedded Microcontrollers**
Bassam Shaer and David Cudney

Computer Algorithm and Modeling

858 **Diagnosing Student Learning Problems in Object Oriented Programming**
Hana Al-Nuaim, Arwa Allinjawi, Paul Krause and Lilian Tang

866 **Computing All Pairs Shortest Paths on Sparse Graphs with Articulation Points**
Carlos Roberto Arias and Von-Wun Soo

884 **Direct Adaptive Fuzzy Based Backstepping Control of Semi Strict Feedback Nonlinear Systems**
Raoudha Ben Khaled, Chaouki Mnasri and Moncef Gasmi

Computer Security and Information Technology

893 **The NP-Complete Face of Information-Theoretic Security**
Stefan Rass and Peter Schartner

906 **Games-Based Learning Framework**
Antonio Garcia-Cabot, Eva García, Roberto Barchino, Luis de-Marcos, José-María Gutiérrez, José-Antonio Gutiérrez, José-Javier Martínez, Salvador Otón and José-Ramón Hilera

Other Issues

912 **An Integrated Approach for Decision Support through Multi-objective Optimization with Application to an Ill-posed Design Problem**
Yoshiaki Shimizu, Yasumasa Kato and Takeshi Kariyahara

926 **Effective Channel Length Degradation under Hot-Carrier Stressing**
Anucha Ruangphanit, Kunagone Kiddee, Rangson Muanghlua, Surasak Niemcharoen and Amporn Poyai

Computer Technology and Application 2 (2011) 835-842

Software Composition of Different Security Level Components

Frank Tsui[1], Edward Jung[2] and Sheryl Duggins[1]

1. Software Engineering, Southern Polytechnic State University, Marietta, Georgia 30060, USA

2. Computer Science, Southern Polytechnic State University, Marietta, Georgia 30060, USA

Received: September 01, 2011 / Accepted: September 29, 2011 / Published: November 25, 2011.

Abstract: In this paper the authors show how software component design can affect security properties through different composition operators. The authors define software composition as the result of aggregating and/or associating a component to a software system. The component itself may be informational or functional and carry a certain level of security attribute. The authors first show that the security attributes or properties form a lattice structure when combined with the appropriate least upper bound and greatest lower bound type of operators. Three composition operators, named C1, C2 and C3 are developed. The system's security properties resulting from these compositions are then studied. The authors discuss how different composition operators maintain, relax and restrict the security properties. Finally, the authors show that C1 and C2 composition operators are order-sensitive and that C3 is order-insensitive.

Key words: Software composition, security, component-design.

1. Introduction

In this paper we investigate the results of composing a system from components of different security levels. We demonstrate that different composition operations can influence the security properties of the software system. Several definitions, including composition rules, are introduced. We explore the resulting system's security characteristics based on the composition rule and discuss how changing the composition rule maintains, relaxes or restricts the security property of the resulting system.

Both software composition [1-7] and software security [8-10] have been heavily studied. We will depend greatly on these past researches and use them as a framework for our work. It is, however, still necessary for us to define and clarify what we mean by these terms. Our definition of security is mostly influenced by the one given by Pfleeger and Pfleeger [11] where computer security must address or enforce (i) confidentiality, (ii) integrity, and (iii) availability.

In section 2 we define the notion of software composition; and in section 3 we provide specific definitions for security levels of software via several sub-attributes of security. In section 4 a metric is prescribed to the security sub-attributes; and we show that these security properties form a partial ordering. Three composition operators, *C1*, *C2*, and *C3* are introduced in section 5, and we show how these operators affect the security levels of the resulting system. Finally, in section 6 we discuss the general characteristics of these three operators and show in two theorems that *C1* and *C2* operators are order-insensitive and that *C3* is order sensitive.

2. Software Composition

Software composition is emphasized here in that it is different from hardware composition. By software composition we mean logical grouping or logical integration of software. Software may be used by

Corresponding author: Frank Tsui, Ph.D., associate professor, research fields: software complexity, software metrics, and software management. E-mail: ftsui@spsu.edu.

making a copy and then moving that copy to another physical location, but it may also be used just by giving a reference to it without ever making another copy or moving it. Thus composing software components into a system may be markedly different from composing physically distinct, albeit logically same, pieces of hardware components.

We define a system *S*, initially as an empty set. As we integrate a software entity *e*, into *S*, then we are composing a system. This simple definition is closer to the concept of aggregation. That is, *e* is an element of *S* but may or may not exist meaningfully as an independent entity outside of *S*. This situation may be thought as making a copy of the software entity *e*, and physically placing it into the system *S*.

But software can be used through reference calls and parameter passing. This logical composition of software may be thought of as association. The entity *e* may exist outside of *S* as a meaningful, independent entity. The software entity *e* may be logically grouped or composed into system *S* by referencing it from *S*. No extra copy is made or physically moved. Entity *e* logically belongs to *S*.

In both situations, our expectation is that after composition of an entity *e* into *S*, the resulting system *S*' may gain more functionality but does not drastically alter its intrinsic system properties. By intrinsic system properties of *S* we mean non-functional characteristics such reliability, security, cohesion, coupling, maintainability, etc. Before we describe the rules of composition, we need to discuss the system property of interest. Here the specific system property of interest is security and the preservation of security property.

3. Security

Security of a software entity or system is not a simple, single property but can best be expressed through multiple sub-attributes. We draw a parallel to earlier mentioned Pfleeger and Pfleeger's security definition where we view confidentiality via readability, integrity via write-ability or modifiability, and availability via evocability. We consider security of a software entity from five sub-properties perspective, represented by a 5-tuple. The five sub-attributes all address security access control of read, write, and evocation. Let security property of a software entity e or a software system *S* be SP. Then SP of *e* or *S* may be represented as follows:

$$SP = < R, W, FR, FW, FE >$$

where:

R = readable;

W = writeable or changeable;

FR = functionally read;

FW = functionally write or change;

FE = functionally evoked.

The equal "=" symbol is used loosely here to mean "represents" as opposed to the mathematical equality. The read sub-attribute *R*, defines whether the software entity is readable by others. As such it may take on the value of *{none}*, *{all}*, *{e1, ---, en}*. Thus the *R* sub-attribute of an entity's security property, SP, specifies whether that entity may be read by no-one (no entity), everyone (all entities), or some specified set of entities *(e1, ---, en)*. *W* is the sub-attribute which defines whether the entity may be written over or changed by either no-one, everyone, or some set of entities. Thus it also takes on the value of *{none}*, *{all}*, or *{e1, ---, en}*. Both *R* and *W* sub-attributes are especially applicable to software entities that are information or data entities such as files or tables. One may consider the software entity to be more secure the more restrictive are the *R* and *W* sub-attributes. The extreme case is an entity whose *R* and *W* values are both {none}, meaning the entity can neither be read nor written to. Such an entity, if it were a data entity, may be quite secure but probably not very useful. However, if it were a functional code entity, then we may want its *R* and *W* sub-attribute values to be *{none}* to prevent undesirable change to it.

The next three sub-attributes, FR, FW, and FE are more applicable for functional entities. FR specifies what an entity may read. Here the term functional read, FR, takes on a broader meaning. It includes the evocation of another entity, asking it to read and return the values. FR also takes on the values of *{none}*, *{all}*, and *{e1, ---, en}* to represent that the entity has the authority and can read no-one, everyone, or some specified set of entities. The same three values are also applicable to FW and FE sub-attributes. In the case of FW, it indicates the authority and the capability of the entity to write and change no-one, everyone, or some set of entities. FW also takes on a broader meaning of evoking another entity, passing it some parameters and asking it to write/change the values of that entity. For FR and FW attributes, one may interpret that the entity is more secure if it has more authority and capability to read and to write/change other entities. FE, while it has the same three sub-attribute values, may be interpreted slightly differently. An entity may be evoked by no-one, everyone or only some set of entities. An entity that may be evoked by no-one may be very secure but is not very meaningful or useful. On the other end, an entity that is evocable by everyone may be considered highly insecure.

The security property SP, of an entity with its 5-tuple representation allows us to gauge the security level of that entity through the values of the five sub-attributes. The level of security as measured by the five sub-attributes may be different. For *R*, *W*, and FE sub-attributes, the security is highest when the value is *{none}* and lowest when the value is *{all}*. For FR and FW, the reverse is the case where security is highest when the value is *{all}* and lowest when it is *{none}*. These interpretations of security and sub-attribute values are depicted as follows in Table 1.

Table 1 Security levels of sub-attributes.

Sub-attribute	Low security	Medium security	High security
R	{all}	{e1,---, en}	{none}
W	{all}	{e1,---, en}	{none}
FR	{none}	{e1,---, en}	{all}
FW	{none}	{e1,---, en}	{all}
FE	{all}	{e1,---, en}	{none}

An entity *e* whose SP 5-tuple is <*all, all, none, none, all*> may be viewed as very insecure or lowest security. The SP 5-tuple of <*none, none, all, all, none*> would be considered very secure or highest security. If we consider every non-empty and non-all set *{e1, ---, en}*, regardless of its cardinality, as one value, then each sub-attribute has three distinctive possibilities. Thus there are a total of 3^5 or 243 combinations of the 5 sub-attributes. SP can take on any of the 243 possibilities. Two of these combinations form the lowest security and the highest security. We need to discuss the security levels of the remaining 241 combinations.

Consider an entity *e1*, whose *SP = < none, none, {ex, ey, ez}, none, {em}>*, and an entity *e2*, whose *SP = < none, none, none, {ey, ez}, {em, en}>*. It is not clear if *e1* has a higher or lower SP than that of *e2*. Except for the highest and the lowest case, the question of which entity, e*1* or *e2*, has a higher security SP is very difficult to answer since the 243 combinations do not have a natural ordering that one may use.

4. Partial Ordering of Security Property SP

A common attempt to create an ordering is to map the sub-attribute value to some numerical value. Then perform a weighted sum of the 5 sub-attributes to generate a single numerical value for the SP. Such a scheme would provide us at least a partial ordering of the 243 combinations of SP values. One way would be to assign numerical values of 1, 3, and 5 to the low, middle, and high security cases respectively. Note that what is low security or middle security is different for the different sub-attributes. For example, for the sub-attribute *R*, *{all} = 1*, *{e1, ---, e2} = 3*, and *{none} = 5*. But for the sub-attribute FR, {*all} = 5*, *{e1, ---, e2} = 3*, and *{none} = 1*.

Of the 243 combinations of potential SP values that an entity may have and using the above weighted sum approach, some entities are going to take on the same SP value. Consider *e1* whose *SP = < {all}, {e3, e7}, {none}, {none}, {none}>* and *e2* whose *SP =*

<{{e5,e6,e7}, {all} , {none}, {none}, {none}>. For *e1*, the SP weighted sum value is 1+3+1+1+5 = 11, and for *e2* the SP weighted sum value is 3+1+1+1+5 = 11. Thus these two combinations of SP have the same weighted sum value and thus may be interpreted as equal in security level. The lowest security combination is the *SP = < {all}, {all}, {none}, [none}, all}>*, whose weights would be <1, 1, 1, 1, 1>, and the weighted sum value of this SP is 5. The highest security combination of SP is <5, 5, 5, 5, 5>, and its weighted sum is 25. All other combinations fall in between these two bounds.

Using a scheme similar to the weighted sum approach demonstrated here, one can map the 243 combinations of SP into a partial ordering. In addition, if one further defines operations of "least upper bound" and "greatest lower bound" over the 243 combinations of SP values, then we will have the lattice structure commonly exhibited in a military security access model such as the Bell-La Padula model [12].

5. Composition Operators, *C1*, *C2*, and *C3*

In this section we define and explore three different composition operators and the consequences of these. The notation, *e C S*, will be used to mean that an entity *e* is composed with system *S* via an operator *C*.

We first define a simple compose operator, *C1*, which is based on the set-union operation over SP's sub-attribute values. *C1* composition operator is defines as follows:

- A system *S* is empty at first and designated as S_0. The SP of S_0, by definition will be classified as the lowest security = < *all, all, none, none, all*>;
- The composition operator *C1*, composes an entity e with a system S and produces a system *S'* whose SP value is based on the set-union rule as applied towards each of the 5 sub-attributes of SP: R, W, FR, FW, and FE.

The following matrix entries, expressed in bold italicized letters, in Table 2 show the SP of *S'* values after "*e C1 S*" operation.

Table 2 ***C1*** **compose operational matrix.**

		[SP of S]		
		{none}	{all}	{em,----, es}
[Sp of e]	***{none}***	***{none}***	***{all}***	***{em,---,es}***
	{all}	***{all}***	***{all}***	***{all}***
	{ex,---,ez}	***{ex,---,ez}***	***{all}***	***{ex, ---,ez} U {em,---,es}***

C1 composition operator, utilizing the set union operation over the values for all five of the sub-attributes moves the SP sub-attributes in different security directions. Consider that we start with the S_0 whose SP values are defined as minimal security, <*all, all, none, none, all*>. Using the weighted sum approach discussed earlier, S_0 has a numerical SP value of <1, 1, 1, 1, 1> or a weighted sum of 5. For illustration, we utilize an entity *e* whose SP values are the highest <*none, none, all, all, none*> with the numerical weighted sum value of 25. The composition, *e C1* $S_{0,}$ will yield a system *S'* whose SP value would be <*all, all, all, all, all*>. The numerical values of SP for *S'* are <1, 1, 5, 5, 1>, and its weighted sum value is 13. The resulting system *S'* is more secure than S_0 after the, *e C1* S_0, composition. Note that the *R* sub-attribute and the *W* sub-attribute remained low security as that of S_0. But the FR and FW sub-attributes changed from low to high security. FE remained low. In general, the *C1* composition operator, based on the set-union operation, will always (a) maintain or lower the security level of *R*, *W* and FE sub-attributes and (b) maintain or increase the security level of FR and FW sub-attributes.

Next we define another simple compose operator, *C2*, based on the set-intersection operation over the values of the sub-attributes. *C2* is very much like *C1* except for this one difference of uniformly using set-intersection instead of set-union and is defined as follows:

- A system *S* is empty at first and designated as $S_{0.}$ The SP of S_0, by definition will be classified as the lowest security = < *all, all, none, none, all*>;
- The composition operator, *C2*, composes an entity *e* with a system *S* and produces a system *S'* whose SP value is based on the set-intersection rule as applied

towards each of the 5 sub-attributes of SP: R, W, FR, FW, and FE.

The following matrix entries, expressed in bold italicized letters, in Table 3 show the SP of *S*' values after "*e C2 S*" operation.

Now, consider we start with the same S_0 system with SP sub-attribute values of <*all, all, none, none, all*> or numerically <1, 1, 1, 1, 1> and the weighted sum of 5. Pick the same e with the highest security SP <*none, none, all, all, none*> or numerically <5, 5, 5, 5, 5> with the weighted sum of 25. The composition *e C2* $S_{0,}$ will yield a system S' whose SP value would be <*none, none, none, none, none*>. The numerical values of SP for *S*' are <5, 5, 1, 1, 5>, and its weighted sum value is 17. Just as the *C1* composition operator, *C2* yielded a system *S*' whose SP weighted sum value moved higher from the minimal 5 to 17. However, in general, the *C2* compose operator differs from *C1* compose operator and has a reverse affect on the sub-attributes. *C2*, based on the set-intersection operation, will always (a) maintain or increase the security level of *R, W* and FE sub-attributes and (b) maintain or decrease the security level of FR and FW sub-attributes.

Both *C1* and *C2* compose operators do not move the security direction of SP of a system uniformly among the five sub-attributes. For *C2,* if *{ex, ---, ez}*∩*{em, ---, es}* is empty, then the result is *{none}*. For the sub-attribute FE, the resulting system *S*', may become evocable by no-one while the system, prior to composition, was evocable by some set and the entity that entered into the composition was also evocable by some set of entities. A similar kind of drawback occurs when an entity *e* that is not-evocable is composed with a system *S* that is evocable by some entities prior to the composition. The resulting *S*' becomes totally not-evocable. A system that may write to a set of entities may become a system that can write to no other entity if composed with an entity that is not allowed to write to anything. Under *C2*, once the system becomes not-evocable, composition with any type of entity will not change that sub-attribute value. It remains not-evocable. Thus *C2* may create some unrealistic or useless system even though it moved the system security level higher.

Table 3 ***C2*** **compose operational matrix.**

	[SP of S]		
	{none}	{all}	{em,----, es}
[Sp of e] ***{none}***	***{none}***	***{none}***	***{none}***
{all}	***{none}***	***{all}***	***{em, ---, es}***
{ex,---,ez}	***{none}***	***{ex,---,ez}***	***{ex, ---,ez}*** ∩ ***{em,---,es}***

A third compose operator *C3,* will be defined and explored next. Our goal for defining *C3* is to bring some uniformity in the directional movement of the five sub-attributes, which was not exhibited by both *C1* and *C2*, and perhaps lessen some of the drawbacks of *C2*. In order to bring the uniformity in directional movement of security level of the sub-attributes, we need to define the compose operation in a non-uniform way across the 5 sub-attributes. We will develop *C3* with set-intersection operation over the sub-attribute R and W. But the set-union operation is used over the sub-attribute FR and FW. For FE a totally different mapping will be defined. C3 composition is defined by multiple rules as follows:

- A system *S* is empty at first and designated as S_0. The SP of S_0, by definition will be classified as the lowest security = < *all, all, none, none, all*>;
- The composition operator, *C3,* composes an entity *e* with a system *S* and produces a system *S*' whose SP value is based on the set-intersection rule as applied towards *R* and *W* sub-attributes;

The following matrix entries, expressed in bold italicized letters, in Table 4 show the SP of *S*' values for only the *R* and *W* sub-attributes after "*e C3 S*" operation.

Table 4 ***C3*** **compose operational matrix for only *R* and *W* sub-attributes.**

	[SP of S]		
	{none}	{all}	{em,----, es}
[Sp of e] ***{none}***	***{none}***	***{none}***	***{none}***
{all}	***{none}***	***{all}***	***{em, ---, es}***
{ex,---,ez}	***{none}***	***{ex,---,ez}***	***{ex, ---,ez}*** ∩ ***{em,---,es}***

- The composition operator *C3*, composes an entity *e* with a system *S* and produces a system *S*' whose SP value is based on the set-union rule as applied towards FR and FW sub-attributes;

The matrix entries in Table 5 show the SP of S' for only the FR and FW sub-attributes after the composition, "*e C3 S*".

- The composition operator *C3*, composes an entity *e* with a system *S* and produces a system *S*' whose SP value is based on a special mapping of FE sub-attributes.

The matrix entries in Table 6 show the SP of *S*' for only the FE sub-attributes after the composition, "*e C3 S*".

The *C3* compose operator, defined in Table 3 will always maintain or move the system towards a higher security level for read and write sub-attributes because it uses the set-intersection rule on the *R* and *W* sub-attributes. Similarly, this *C3* operator will always maintain or move the system towards a higher security level for functional read and functional write sub-attributes because it uses the set-union operation rule for FR and FW sub-attributes.

For FE sub-attribute, *C3* also maintains or moves system *S* towards a higher security level. The exception is when the entity *e*'s FE sub-attribute value is *{all}* and the FE sub-attribute value of system *S* is a set of entities *{em, ---, es}*. In the special situation where entity e is also one of those element in the *{em, ---, en}* set then the resulting system, *S*' after *C3* composition, would have a FW sub-attribute value of *{all}* which is lowest security level for FE sub-attribute. Otherwise *C3* moves the FE security in the same direction as *R*, *W*, FR, and FW. Note that FE as expressed by *{ex, ---, ez} U{em, ---, es}* may have more elements than *{em, ---, es}*, but security level for FE sub-attribute with *{ex, ---, ez, em, ---, es}* is the same middle security level as that of any FE sub-attribute with set *{e1, ---, en}*. Thus the entry *{ex, ---, ez}U{em, ---, es}* maintains the same security level as the system's FW security level of *{em, ---, es}*. Therefore *C3* composition operator, except for the one exception mentioned above, uniformly maintains or increases the security levels of all the SP sub-attributes of the system. Note that the first entity *e1*, which is composed with S_0 will set the baseline security level for the system. From then on *C3*, except for the one special situation, will uniformly push the system towards a higher security level with each composition.

Table 5 *C3* **compose operational matrix for only FR and FW sub-attributes.**

		[SP of S]		
		{none}	{all}	{em,----, es}
[Sp of e]	*{none}*	*{none}*	*{all}*	*{em,---,es}*
	{all}	*{all}*	*{all}*	*{all}*
	{ex,---,ez}	*{ex,---,ez}*	*{all}*	*{ex, ---,ez} U {em,---,es}*

Table 6 *C3* **compose operational matrix for only FE sub-attributes.**

		[SP of S]		
		{none}	{all}	{em,----, es}
[Sp of e]	*{none}*	*{none}*	*{none}*	*{em,---,es}*
	{all}	*{none}*	*{all}*	*{em,---,es} or {all} if e in (em,---,es}*
	{ex,---,ez}	*{none}*	*{ex,---,ez}*	*{ex, ---,ez} U {em,---,es}*

6. Some General Characteristics of SP with *C1*, *C2*, and *C3*

In this section we will delineate some of the observations about SP, *C1*, *C2*, and *C3*.

Note that the 243 combinations of SP sub-attributes are closed under all three *C1*, *C2*, and *C3* composition operators. The minimum SP defined by the five sub-attributes <*all, all, none, none, all*> is the defined SP value for the initial empty system S_0. SP closure under *C1*, *C2*, and *C3* would mean that the composed resulting system S' can never have an SP outside of the 243 combination values. Using the earlier scheme of weighted sum, then any system *S*' composed with *C1*, *C2*, or *C3* operator can never have a SP value less than 5 or more than 25.

Recall that composition operation, *e C S*, is always aggregating or associating an entity *e* with a system *S*. Let e_2 $(e_1\ C\ S)$ denote e_2 *C* $(e_1\ C\ S)$. Thus e_2 $(e_1\ C\ S)$ is the same as *(e_1 C S) =>S'* followed by *(e_2 C S')*. The question here is whether the order in which the entities are composed with the system makes any difference. Would we obtain different results in terms of the SP value for the ending system? In other words, is the system *S* composition order-sensitive or order-insensitive in terms of the SP sub-attribute values.

Theorem 1: A system *S* defined with the SP values of sub-attributes is composition order-insensitive with either *C1* (or *C2*) operator.

Proof: *C1* composition operator uniformly applies the set-union operator over the five sub-attributes of SP. Since set-union operation is commutative, *C1* is commutative in SP. Thus e_2 *(e_1 C1 S) = e_1 (e_2 C1 S)* and *C1* is a commutative. Similarly, *C2* composition is commutative because *C2* uniformly applies the set-intersection operator over the five sub-attributes of SP. Because both *C1* and *C2* composition operators are commutative, the order in which we compose a system using either *C1* (or *C2*) is immaterial. Thus *C1* (or *C2*) operator is order-insensitive.

Theorem 2: *C3* is composition order-sensitive.

Before we provide the proof by showing one example against order-insensitivity, we would point out that for *C3* the FE sub-attribute operation is where the problem arises. The other, *R*, *W*, FR, and FW sub-attributes are under the traditional set operations.

Proof: Consider starting with S_0 whose SP sub-attributes, by definition, is <*all, all, none, none, all*>. Let entity e_1 have the SP sub-attributes values of <$\{e_x, e_y\}, \{e_z\}, \{e_m, e_n\}$ *all*, $\{e_j, e_2\}$> and entity e_2 have the SP sub-attributes values of <$\{e_x, e_l\}, \{e_z\}, \{e_n, e_p\}$ *all, all*>. Then, e_2 *(e_1 C3 S_0)* results in a system *S'* with SP sub-attribute values of <$\{e_x\}, \{e_z\}, \{e_m, e_n, e_p\}$, *all, all*>. On the other hand e_1 *(e_2 C3 S_0)* results in a system s' with SP sub-attribute values of <$\{e_x\}, \{e_z\}, \{e_m, e_n, e_p\}$, *all*, $\{e_j, e_2\}$>. Note that the FE sub-attribute value is different depending on the order of *C3* composition. Thus *C3* is composition order-sensitive.

7. Conclusions

We have defined security property, SP, of an and of a system with five sub-attributes based on the access control of the five read, write, functional read, functional write, and evocation sub-attributes, which are in turn based on Pfleeger and Pfleeger's confidentiality, integrity and availability concepts. We then defined three levels of security for each of the five sub-attributes to form 243 potential values SP. Three composition operators, *C1*, *C2*, and *C3* are constructed, and various characteristics of SP and *C1*, *C2*, and *C3* are examined. *C1* and *C2* compositions are shown to be order-insensitive and *C3* to be order-sensitive.

Composition operators are models of how real systems may be composed and what the characteristics of tools that are needed to help us compose, order-insensitively or order-sensitively. We will need to construct more diverse types of composition operators to broaden our study of how to relax, maintain, and restrain the SP sub-attributes.

Also, more SP sub-attributes need to be included in the future to broaden the study of composition operators.

References

[1] R. Gonzales, M. Torres, Issues in component-based development: towards specification with ADLs, Journal of Systemics, Cybernetics, and Informatics 4 (5) (2006) 49-54.

[2] R. Giguette, Building objects out of plato: applying philosophy, symbolism, and analogy to software design, Communications of the ACM 49 (10) (2006) 66-71.

[3] K.K. Lau, Z. Wang, Software component models, IEEE Transactions on Software Engineering 33 (10) (2007) 709-724.

[4] J.K. Millen, Hookup security for synchronous machines, in: Proceedings of Computer Security Foundations Workshop III, New Hampshire, USA, June 1990, pp. 84-90.

[5] D.L. Parnas, On the criteria to be used in decomposing systems into modules, Communications of the ACM 15 (12) (1972) 1053-1058.

[6] D. Socha, S. Walter, Is designing software different from designing other things?, International Journal of Engineering Education 22 (3) (2006) 540-550.

[7] F. Tsui, O. Karam, Essentials of Software Engineering, 2nd ed., Jones and Bartlett Publishers, Sudbury, Massachusetts, USA, 2011.

[8] K.W. Coertzel, T. Winograd, Safety and Security Considerations for Component Based Engineering of Software-Intensive Systems, available online at: https://buildsecurityin.us-cert.gov/swa/downloads/NAVSEA-Composition-DRAFT-061110.pdf, February 2011.

[9] K.M. Khan, J. Han, Deriving system level security properties of component based composite systems, in: 16th Australian Software Engineering Conference, Brisbane, Australia, March 2005.

[10] F. Xie, J.C. Browne, Verified systems by composition from verified components, in: 4th Joint Meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering, Helsinki, Finland, September 2003.

[11] C.P. Pfleeger, S.H. Pfleeger, Security in Computing, 4th ed., Prentice Hall, 2007.

[12] D.E. Bell, Looking back at the Bell-La Padula model, in: 21st Annual Computer Security Applications Conference, Tucson, Arizona, USA, December 2005.

Computer Technology and Application 2 (2011) 843-848

Generating Graphic User Interface of Web Applications Using Source Code Generator Based on Dynamic Frames

Danijel Radošević, Denis Bračun and Ivan Magdalenić

Faculty of Organization and Informatics, University of Zagreb, Varazdin 42000, Croatia

Received: August 31, 2011 / Accepted: September 09, 2011 / Published: November 25, 2011.

Abstract: This paper deals with a problem of application generation together with their Graphic User Interface (GUI). Particularly, the source code generator based on dynamic frames was improved for more effective specification of GUI. It's too demanding for the developers to have specification of the application that contain all physical coordinates and other details of buttons and other GUI elements. The developed solution for this problem is based on post-processing of generated source code using iterators for specifying coordinates and other values of graphic elements. The paper includes two examples of generating web applications and their GUI.

Key words: Generators, graphic user interface, dynamic frames.

1. Introduction

Dynamic frames generator model, introduced in Ref. [1] is a model of source code generator that we use for building of complete applications following principles of Software Product Lines (SPL). It has some similarities with other SPL approaches, including frames techniques like XML-based Variant Configuration Language (XVCL) [2], Generative Programming [3], Aspect Oriented Programming [4] and approaches based on scripting languages (e.g., Open Promol [5] and CodeWorker [6]). Unlike some other approaches, our model clearly separates three basic components: Specification (S) that describes application characteristics, Configuration (C) that describes the rules for building applications, and Templates (T) that refer to application building blocks [1]. Specification of application exists independently from other two model elements, which enables usage of different target implementation technologies, programming languages, etc.

Different features of application to be generated can be easily specified in Specification, as values of appropriate attributes. The arrangement of elements on Graphic User Interface (GUI) could be also specified by attribute-value pairs, but that could be demanding, making Specification bulky. For this reason, generator model was improved for more effective specification of GUI. The solution based on post-processing of generated code is implemented in two examples of web applications generation. In the first example, the arrangement of GUI elements is defined using HyperText Markup Language (HTML) layers, while the second example generates Extensible Application Markup Language (XAML) document to define the arrangement of GUI elements of Silverlight based application.

Section 2 gives presentation of related work; section

Denis Bračun, B.A., research fields: programming, programming languages.

Ivan Magdalenić, Ph.D., research fields: e-business, web technology, semantic web technology and generative programming.

Corresponding author: Danijel Radošević, Ph.D., associate professor, research fields: programming languages, generative programming and educational software. E-mail: danijel.radosevic@foi.hr.

3 introduces Dynamic frames generator model; section 4 presents the given solution for generation of GUI; section 5 gives an example of generating XAML document and section 6 gives conclusions.

2. Related Work

The most comparable approach to our Dynamic frames generator model [1] is Jarzabek's XVCL [2]. XVCL is a variant mechanism that uses *x-frames* to define building blocks of source code to be generated. All used *x-frames* make a tree structure, as shown in Fig. 1.

XVCL distinguish specification *x-frames* that contain program specification, while other *x-frames* combine program code with break sections to define insertion of variable program parts. Configuration elements are specified implicitly, in break sections, defining different kinds of insertion and adaptation. As described in section 3, Dynamic frames generator model also define a tree structure, but developer has to define only top level frames, while other frames are instantiated dynamically, during the process of generation. Also, each frame contains clearly separated parts regarding to Specification, Configuration and code template (particular template from Templates).

Automatic generation of GUI could be integrated with some other features of target applications, like security. Schlapfer [8] has combined GUI functionality with awareness of the security policy. In traditional approach, GUI elements that are not allowed for the particular user are shown in gray, as shown in Fig. 2.

The role of generators, as desribed by Schlapfer [8] is to generate the GUI without redundant elements, i.e., appropriate to particular user. So, there is a problem of arranging GUI elements, and specification of that arrangement. Schlapfer provided a method to validate the definition of smart, security-aware GUIs by checking the corresponding Object Constraint Language (OCL) expressions over the SecureUML+ GUI models that capture the smart, security aware GUI definitions [8].

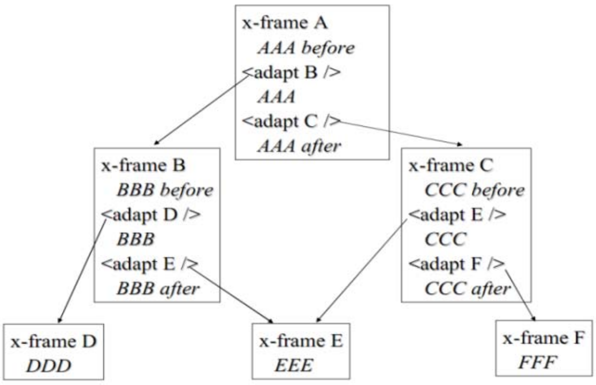


Fig. 1 XVCL *x-frames* [7].

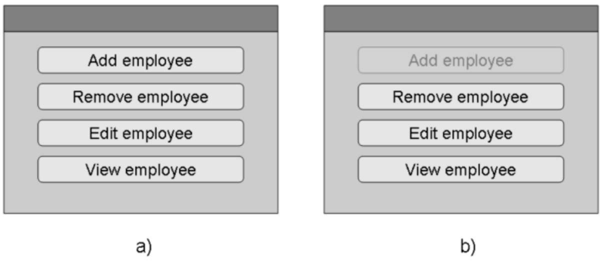


Fig. 2 An example window: (a) View of HR members; (b) View of other employees [8].

Different XML variants are used for defining of GUI layouts. Extensible Formatting language (XF) is a generic high-level formatting language serving as a mediator for contemporary, powerful formatting languages such as Cascading Style Sheets (CSS) and Extensible Stylesheet Language (XSL) [9]. XML User Interface Language (XUL) is the model used by the Mozilla family of browsers and has a rich notation for creating widgets, and uses Box, Grid and other layout models [10]. XAML is a user interface mark-up language for Windows Presentation Foundation (WPF) and consists of features from both Microsoft Windows applications and web applications [11]. There are some other examples XML variants used for defining of GUI layouts, like XForms and Laszlo XML (LZX) [11]. It's important that these XML, HTML and similar documents can be easily generated using our Dynamic frames generator model, as shown in given examples (sections 4 and 5).

3. Dynamic Frames Generator Model

Dynamic frames generator model [1] was developed on a basis of previously introduced Scripting Generator Model (SGM) [12]. The main purpose of SGM was to offer a model for modeling of generators written in scripting languages like Perl and Hypertext Preprocessor (PHP). SGM separates program specification and set of code templates used as building blocks of programs to be generated, but configuration was inlined in a generator itself. That was changed in the Dynamic frames model, which defines generator from three kinds of elements: Specification (S), Configuration (C) and Templates (T). All three model elements together make the SCT frame (Fig. 3):

Specification contains attribute-value pairs that define features of generated application. Template is a set of all code templates. Each code template contains source code in target programming language together with connections (replacing marks for insertion of variable code parts). Configuration defines the connection rules between Specification and Template in a form of triplets (connection-attribute-template). Connections are replacing marks (usually enclosed by special signs '#') in code templates. During the process of generation, connections are replaced by another code templates and/or values from Specification.

Starting SCT frame [1] contains the whole Specification, the whole Configuration, but only the base template from the set of all Templates. Other SCT frames are produced dynamically, for each connection in template, forming generation tree (Fig. 4).

Each frame produces one fragment of source code in process of program generation. The final program code is built from all fragments of source code by source code generator.

Handler:

Handler makes generator scalable, enabling generation of more target program files from same Specification. In other words, Handler prepares inputs for the generator and then collects and saves generator outputs (Fig. 5).

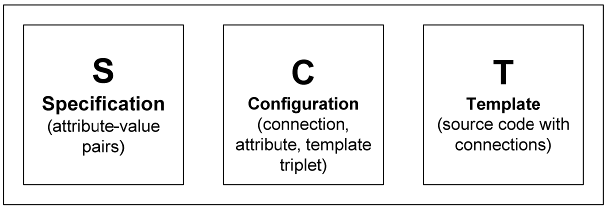

S
Specification
(attribute-value pairs)
C
Configuration
(connection, attribute, template triplet)
T
Template
(source code with connections)

Fig. 3 SCT frame [13].

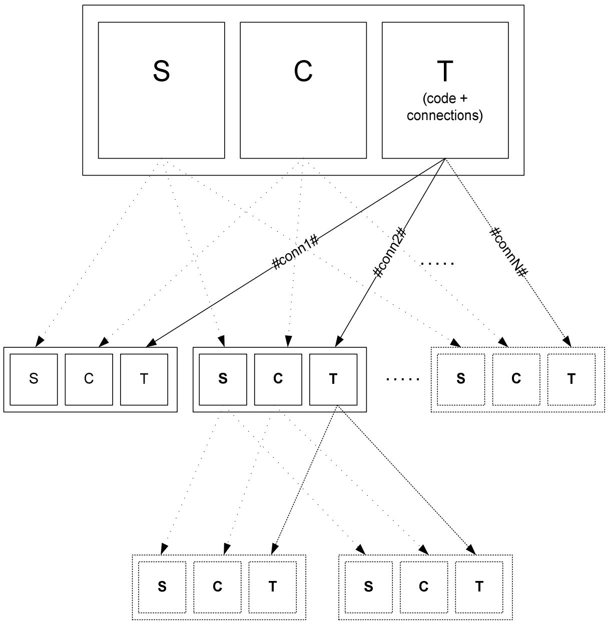


Fig. 4 The generation tree [13].

As shown in Fig. 5, Handler uses initial lines in Configuration to define top-level code templates, as bases for building of appropriate generation trees. This lines of Configuration are connected to appropriate lines in Specification (contain *OUTPUT* attribute), defining kinds of outputs. Each kind of output can be used in generation of more output files (e.g., *output* is used in generation of *output/students.cgi*, *output/courses.cgi*, *output/exams.cgi* and *output/questionnaire.cgi*, as shown in Fig. 5).

4. Generation of GUI Elements

There are different techniques that enable defining position of GUI elements. Some Java layouts, HTML layers, XML layouts like XAML and other enable usage of absolute or relative coordinates to define positions. These positions could be specified to generator by attribute-value pairs, but that could be demanding, with too much coordinates to be specified.

[1] XML schema of SCT frame is available at http://generators.foi.hr/xml_schema.jpg, example of textual representation at: http://generators.foi.hr/SCT_generator_python/form.cgi.

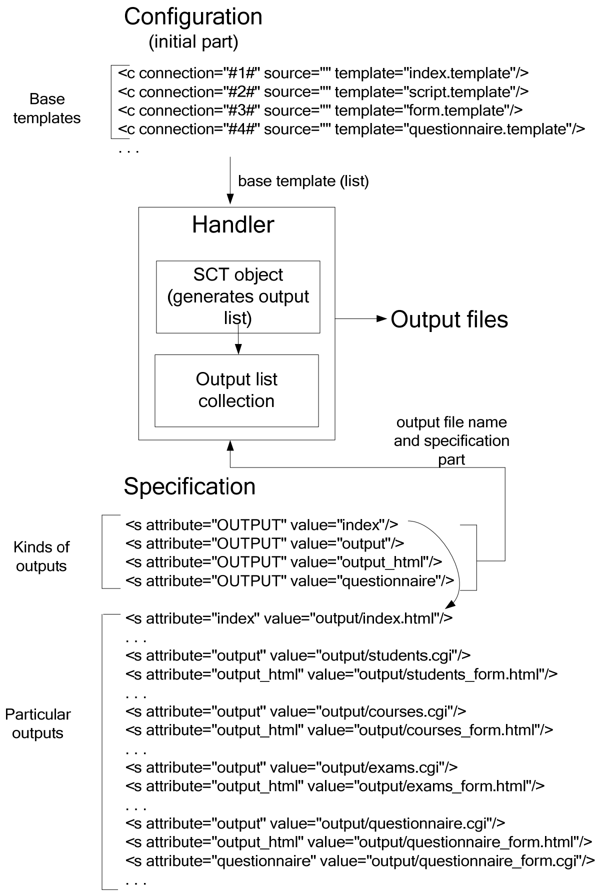

Configuration
(initial part)
Base templates
<c connection="#1#" source="" template="index.template"/>
<c connection="#2#" source="" template="script.template"/>
<c connection="#3#" source="" template="form.template"/>
<c connection="#4#" source="" template="questionnaire.template"/>
. . .
base template (list)
Handler
SCT object (generates output list)
Output list collection
Output files
output file name and specification part
Specification
Kinds of outputs
<s attribute="OUTPUT" value="index"/>
<s attribute="OUTPUT" value="output"/>
<s attribute="OUTPUT" value="output_html"/>
<s attribute="OUTPUT" value="questionnaire"/>
Particular outputs
<s attribute="index" value="output/index.html"/>
. . .
<s attribute="output" value="output/students.cgi"/>
<s attribute="output_html" value="output/students_form.html"/>
. . .
<s attribute="output" value="output/courses.cgi"/>
<s attribute="output_html" value="output/courses_form.html"/>
. . .
<s attribute="output" value="output/exams.cgi"/>
<s attribute="output_html" value="output/exams_form.html"/>
. . .
<s attribute="output" value="output/questionnaire.cgi"/>
<s attribute="output_html" value="output/questionnaire_form.html"/>
<s attribute="questionnaire" value="output/questionnaire_form.cgi"/>
. . .

Fig. 5 Functions of handler [13].

The proposed solution in this paper uses iterators to define position of GUI elements. Iterators are post-processed after the main generation process is finished. Iterators have to be specified in both, Specification and Templates.

4.1 Specification of Iterators in Templates

Iterators are specified in Templates within '#' signs, which is similar to connections, but uses the keyword *iterator_*, as shown in the following example:

```
 <div
style="position:absolute;top:#iterator_Y#%;
 left:#iterator_X1#%; z-index:1">
 <b>#field_display#:</b>
 </div>
```

This example uses two iterators, *#iterator_Y#* and *#iterator_X1#*. On the other hand, the connection *#field_display#* is an ordinary connection that has its line in Configuration (here: *#field_display#, field_display*; means that *#field_display#* has to be replaced by the value of *field_display*, as specified in Specification). Iterators are directly connected to Specification, so there are no appropriate configuration lines.

4.2 Specification of Iterators in Specification

Iterators have its separate area in Specification. Unlike other specification attributes, iterators are given by specifying two values, as shown in the following example:

```
iterator_Y:25,4
iterator_X1:35,0
iterator_X2:51,0
```

The first number right to iterator name is the starting value, and another is the increment value. Thus, the real value depends on the ordinary number of iterator occasion in the generated code, as shown in the example:

```
<div style="position:absolute; top:25%;
left:35%; z-index:1">
<b>Student id:</b>
</div>
<div style="position:absolute; top:29%;
left:35%; z-index:1">
<b>Surname and name:</b>
</div>
<div style="position:absolute; top:33%;
left:35%; z-index:1">
<b>Year of enrollment:</b>
</div>
. . .
```

As shown in the example, the value for *Y* coordinate starts from specified value od 25, with the increment value of 4. The *X1* coordinate stays at 35. The generated user input/edit form is shown in Fig. 6.

5. XAML Example

This example uses XAML to define the layout of Silverlight web application. The generated application

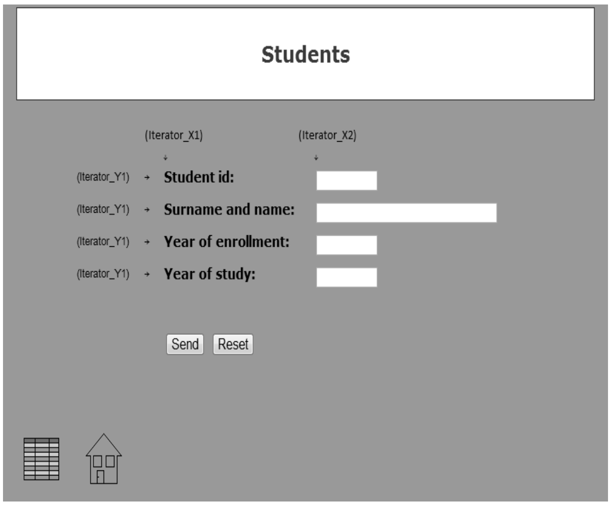


Fig. 6 Generated input/edit form.

is in C# and it is used for editing database table content by using input/edit form. The overall structure of the XAML document to be generated is given in its code template:

```
<UserControl
xmlns="http://schemas.microsoft.com/.."
xmlns:x="http://schemas.microsoft.com/.."
xmlns:d=http://schemas.microsoft.com/.."
Width="640" Height="480" mc:Ignorable="d">
<Grid x:Name="LayoutRoot" Margin="0,0,8,0">
#fields_entry#
<Button x:Name="btnSave" Margin="312,230,268,0"
VerticalAlignment="Top"
Content="Save" Click="btnSave_Click">
<Button.Background>
<LinearGradientBrush EndPoint="6.519,4.273"
StartPoint="6.308,4.318">
<GradientStop Color="Black" Offset="0"/>
<GradientStop Color="White" Offset="1"/>
</LinearGradientBrush>
</Button.Background>
</Button>
</Grid>
</UserControl>
```

The connection *#fields_entry#* is to be replaced during the process of generation with code defining edit fields and their labels. This fields could be different types, like integers, character strings, etc., so corresponding configuration line defines usage of appropriate code template:

```
#fields_entry#,field_*,field_entry_*.template
```

Asterisk means “for all” (e.g., *fields_** =all Specification attributes having name starting with *fields_*). Specific suffix (e.g., number in *field_number*) is used to form filename of the template (here: *field_entry_number.template*). The template is defined for each type of edit field to be used in generation, e.g., *field_entry_number.template*:

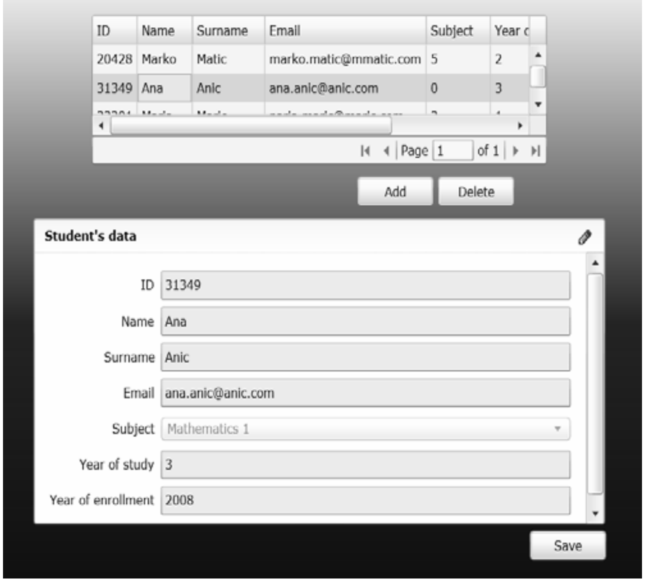
```
<TextBox x:Name="#polje#" Margin="286,
#iterator_1#,234,0" VerticalAlignment="Top"
TextWrapping="Wrap"/>
<dataInput:Label x:Name="#polje_labela#"
HorizontalAlignment="Left" Margin="220,
#iterator_2#,0,0" VerticalAlignment="Top"
Width="62" Content="#polje_prikaz#:"/>
```

As shown, two iterators are used in this template: *iterator_1* to define *Y* coordinate of *TextBox* field and *Iterator_2* to define *Y* coordinate of appropriate label.

In Specification, these two iterators are specified as follows (numbers specify starting value and increment):

```
iterator_1:38,28
iterator_2:46,28
```

Resulting layout of generated Silverlight application is given in Fig. 7.



ID	Name	Surname	Email	Subject	Year c
20428	Marko	Matic	marko.matic@mmatic.com	5	2
31349	Ana	Anic	ana.anic@anic.com	0	3

Page 1 of 1

Student's data

ID 31349
Name Ana
Surname Anic
Email ana.anic@anic.com
Subject Mathematics 1
Year of study 3
Year of enrollment 2008

Fig. 7 Layout of generated Silverlight application.

6. Conclusions

This paper presents our solution for generating applications together with their GUI. For this purpose, our Dynamic frames generator model was improved for more effective specification of GUI layout. The solution is based on post-processing of generated source code by using iterators for specifying coordinates and other values of graphic elements. Iterators are tested on two examples. In the first example, GUI of web application is defined using HTML layers. Second example uses XAML to define the screen layout. For this purpose, a generator of Silverlight application was made.

It is shown that our solution enables effective specification of GUI layout, together with all other features of generated application.

References

[1] D. Radošević, I. Magdalenić, Source code generator based on dynamic frames, Journal of Information and Organizational Sciences 35 (1) (2011) 73-91.

[2] S. Jarzabek, P. Bassett, H. Zhang, W. Zhang, XVCL: XML-based variant configuration language, in: Proc. Int'l Conf. on Software Engineering, Los Alamitos, CA, USA, 2003, pp. 810-811.

[3] K. Czarnecki, U.W. Eisenecker, Generative Programming: Methods, Techniques, and Applications, Addison-Wesley, 2000.

[4] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C.V. Lopes, J.M. Loingtier, J. Irwin, Aspect-oriented programming, in: Proceedings of the European Conference on Object-Oriented Programming (ECOOP), 1997.

[5] V. Štuikys, R. Damaševičius, G. Ziberkas, Open PROMOL: an experimental language for target program modification, in: A. Mignotte, E. Villar, L. Horobin (Eds.), System on Chip Design Languages, Kluwer Academic Publishers, 2002, pp. 235-246.

[6] C. Lemaire, CODEWORKER Parsing Tool and Code generator—User's Guide & Reference Manual, available online at: http://codeworker.free.fr/CodeWorker.pdf, 2008.

[7] S. Jarzabek, XML-based Variant Configuration Language (XVCL), Specification Version 2.10, National University of Singapore, Singapore, 2006.

[8] M. Schlapfer, Automatic generation of smart&secGUIs from security design models, Master Thesis, Information Security Group, Department of Computer Science, ETH Zurich, Zurich, Switzerland, 2009.

[9] H. Tervo, A. Honkaranta, P. Leinonen, Towards generic layouts for multi-channel publishing: "XF—extensible formatting", in: Proceedings of Information Systems Technology and Its Applications, 2006, pp. 165-176.

[10] J. Bishop, Multi-platform user interface construction—a challenge for software engineering-in-the-small, in: Proc. 28th Int. Conf. on Software Engineering, Shanghai, China, 2006.

[11] M. Pohja, Comparison of common XML-based web user interface languages, Journal of Web Engineering 9 (2) (2010) 95-115.

[12] D. Radošević, B. Kliček, J. Dobša, Generative development using scripting model of application generators, in: B. Katalinic (Ed.), DAAAM International Scientific Book 2006, Vienna, Austria, 2006, pp. 489-502.

[13] D. Radošević, I. Magdalenić, Python Implementation of Source Code Generator Based on Dynamic Frames, in: Proceedings of 34th International Conference MIPRO 2011, Opatija, Croatia, 2011, pp. 969-974.

Computer Technology and Application 2 (2011) 849-857

Intelligent Tracking Telescope Using Embedded Microcontrollers

Bassam Shaer and David Cudney

Electrical and Computer Engineering Department, University of West Florida, Shalimar 32579, United States

Received: August 30, 2011 / Accepted: September 09, 2011 / Published: November 25, 2011.

Abstract: This paper presents the design and implementation of the embedded microcontrollers within a telescope control system. The design objectives of the overall system are to automatically find light emitting objects at night within a user specified area, to track a light emitting object for a user specified time given the initial position of the object, and to allow the user access to these functions through a graphical user interface. The embedded system is used to provide a communication link between the graphical user interface and system hardware, to provide angle data processing for a dual axis accelerometer, to provide data processing for an electronic compass, and to provide pulse outputs for the step and direction inputs of three stepper motor drives. This paper will describe the design, implementation, and results of each of these objectives.

Key words: Intelligent, tracking, telescope, electronic compass.

1. Introduction

The system known as the "Intelligent Tracking Telescope" is an autonomous tracking and scanning telescope. The system includes a gimbel type platform which is referred to as the pedestal, a motor and drive assembly, and a graphical user interface. The system will provide the user with right ascension and declination coordinates of any light emitting body within a user specified window of the night sky. The telescope will also automatically track a light emitting object for a user specified amount of time. A high level system block diagram is shown in Fig. 1.

The telescope pedestal assembly includes the telescope, the camera, the stepper motors, an electronic compass, and a dual axis accelerometer. The pedestal is a steel frame which allows gyroscopic movement of the telescope. The pedestal allows movement of the telescope approximately ± 45 degrees in declination and ± 45 degrees in right ascension from a center position. A computer generated drawing of the pedestal is depicted in Fig. 2. In this figure, the telescope is at the center position described above. The upper portion of the pedestal also rotates as to allow the alignment of the right ascension axis with the North and South poles.

Three stepper motors are used to rotate the pedestal about the three axes. These stepper motors are driven using three stepper motor drives. Four Basic Stamp microcontrollers are embedded in the system to provide the step and direction outputs controlling the stepper drives. They provide an interface between the Graphical User Interface (GUI) [1-7] and the pedestal components, and they process peripheral sensor data. A small digital video camera is part of this system as well. The camera is mounted to the eyepiece of the telescope as to allow the camera input data to contain whatever is in the field of view of the telescope. There is also a dual axis accelerometer mounted to the telescope itself. The accelerometer provides an analog pulse signal that is processed by the microcontrollers into a tilt angle measurement. When this data is needed, it is sent to the

David Cudney, electrical engineer, BSEE, research fields: embedded system, image processing.

Corresponding author: Bassam Shaer, Ph.D., assistant professor, research fields: VLSI design and test, embedded system design, and microelectronics. E-mail: bshaer@uwf.edu.

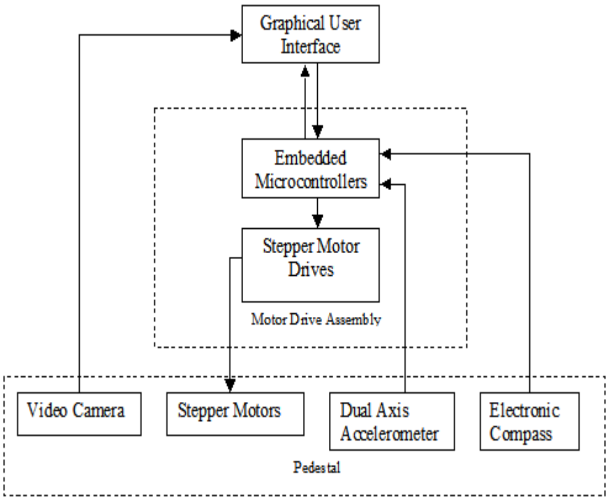


Fig. 1 High level system block diagram.

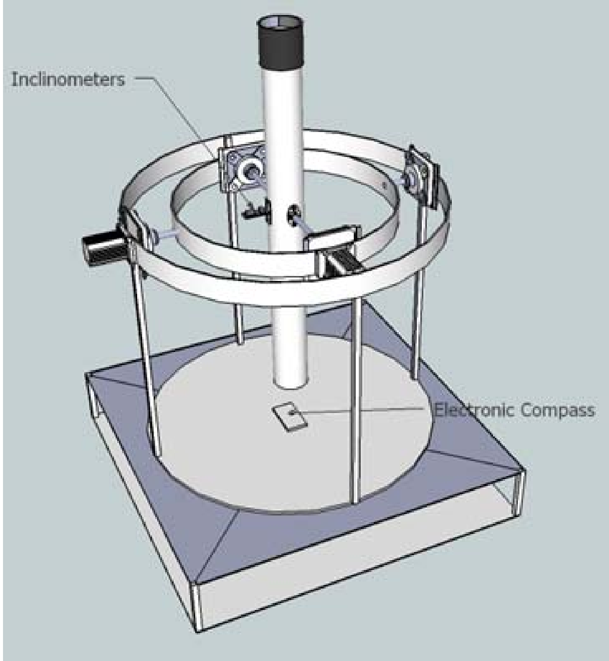


Fig. 2 Pedestal assembly.

GUI function through an asynchronous serial link. The position of the accelerometer can be seen in Fig. 2 and is labeled “inclinometers”. An electronic compass is also mounted to the pedestal. The data from the electronic compass is provided by the compass in the form of a digital angle reading. This data is transferred to one of the microprocessors using the I^2C (Inter-Integrated Circuit) protocol. The heading data is strictly handled within the embedded microprocessor. The user can control the axis alignment function from the GUI through a pushbutton which sends the embedded microcontroller into a calibration routine. This routine uses the compass to align the right ascension axis of the pedestal with the North and South poles. The electronic compass ultimately provides the system with a reference to the earth’s North Pole. The position of the compass can also be seen in Fig. 2. The paper is organized as follows: Section 2 presents an overview of the embedded system; section 3 discusses the axis alignment operations; section 4 discusses the data processing for the angle and tilt operations; section 5 discusses the axis rotation operation; section 6 discusses operational results and testing procedures; section 7 covers the conclusions and future work.

2. Embedded System Overview

Four Basic Stamp 24 pin microcontrollers were used in the embedded system. These microcontrollers are packaged in an easy to use 24 pin type package. This package is relatively small and provides a common footprint which is compatible with almost any hardware configuration. A printed circuit board was designed and built with 24 pin chip sockets to house the microcontrollers and all necessary input/output I/O connectors. The printed circuit board was designed to allow separate in place programming of each of the microcontrollers.

The functions carried out by the microcontrollers include the axis alignment operations, right ascension operations, declination operations, and tilt angle data processing. The axis alignment microcontroller was programmed to process data from the electronic compass and provides step and direction pulse outputs to the axis alignment stepper drive. The right ascension microcontroller was programmed to process the pulse signal from the right ascension axis of the accelerometer and to provide step and direction pulse outputs to the right ascension stepper drive. It was also a necessity of the right ascension microcontroller to communicate with the declination microcontroller during some functions. The declination axis microcontroller was programmed to process the pulse signal from the declination axis of the accelerometer and to provide step and direction pulse outputs to the declination axis stepper drive. The accelerometer microcontroller was programmed specifically to process pulse signals from both axes of the accelerometer and to provide that data to the graphical user interface.

Communication between the embedded microcontrollers and the GUI was done through a 3 position asynchronous serial communication link. The Basic Stamp microcontrollers used in the system were very easy to use for this purpose. The microcontrollers were programmed using a PBasic compiler provided by Parallax Inc. Within the PBasic language, built in serial communication functions were used. These functions, “serin” and “serout” [8], are designed to read and write data from specified pins on the Basic Stamp. The input arguments for these functions are specified as I/O pin number, baud mode, and data to be sent or variable to save incoming data to. Other parameters can also be used with the serin and serout functions such as the “WAIT” parameter. The WAIT parameter can be used with the serin function to specify a string only after data is stored into the specified variable. As an example, the syntax SERIN 8, 84, [WAIT(“ABC”), data] would wait for the string “ABC” to be read from I/O pin 8 and store the data read after the string to the variable “data” [8]. The baud mode in this case which is specified by “84” defines the baud rate, parity bit, and inversion scheme of the incoming data. Although the baud mode can be specified as a number of different combinations, the mode used in this system was 9600 bps, 8 bit non inverted and no parity. This communication scheme was used to communicate between microcontrollers as well.

3. Axis Alignment Operations

Axis alignment operations are used in the system to align the right ascension axis with the North and South Poles. An electronic compass is used to provide the system with a reference to magnetic North. The compass used in this system is the Honeywell HMC 6352 Electronic Compass Module.

The axis alignment microcontroller is programmed to wait for the string “CAL” to be sent through the communication link from the PC on which the GUI is running. The string is sent once the user presses the push button labeled “Calibrate” within the GUI figure window. This pushbutton invokes the calibrate callback function. Upon receiving this string, the microcontroller jumps into a loop which reads the heading data from the compass, compares the current data to 0 degrees or North, defines a direction of rotation for the pedestal, and gives the axis alignment stepper drive a direction and step input. An operational flow chart of this loop is shown in Fig. 3.

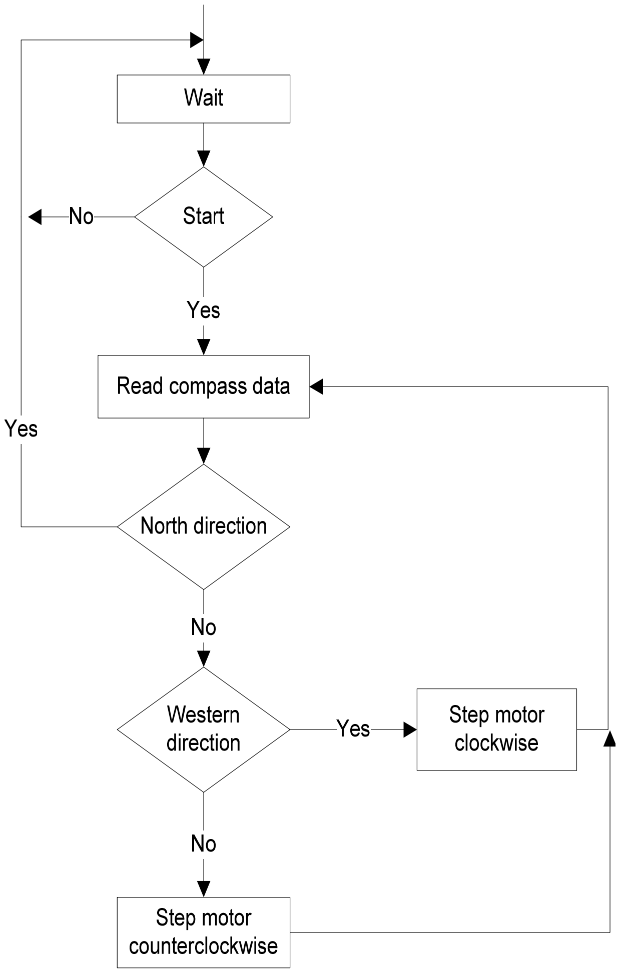


Fig. 3 Axis alignment microcontroller flow chart.

The HMC 6352 electronic compass uses the Inter-Integrated Circuit (I^2C) communication protocol. This interface is easily dealt with using the Basic Stamp to communicate with the compass, and another I/O pin as a transceiver. Using this scheme, the hex address containing the heading data is shifted out of the transceiver pin and the heading data was shifted in. Two more I/O pins on the microcontroller are used as step and direction outputs for the applicable stepper drive. The direction output is either set high (+5VDC) or low (0VDC) depending on the desired direction of step. A 0.2 ms pulse is used as a step input for the stepper drive. A pause between loops of 50 ms is also used. The direction is defined by comparing the current heading read from the compass to 180 or South. This ensures that the pedestal would rotate the shortest distance possible to align the compass to magnetic North.

4. Angle of Tilt Data Processing

To measure tilt, the Memsic2125 dual axis accelerometer is used. The accelerometer is mounted to the telescope using a right angle bracket. The two axes of the accelerometer are used to measure right ascension and declination tilt of the telescope. This component provides a pulse type signal in which the high pulse width indicates the amount of *g* force on the respective axis. Each time the angle data is read from the accelerometer by a microcontroller, it is done within a loop. First, the pulse width is measured from the applicable axis on the accelerometer. This is done using the "pulsin" function within PBASIC [8]. When using the Basic Stamp Module, the "pulsin" function returns the pulse width in 2 μs units. The pulse width was then converted to acceleration using the equation A(g)=(T1/T2-0.5)/12.5% where T1 is the high time and T2 is the period of the signal being read [9]. The acceleration is then multiplied by a gain which depends on the magnitude of the acceleration. The gain factors are given in the Memsic 2125 data sheet. These gain factors are arranged in a table and accessed using the "LOOKUP" and "LOOKDOWN" functions within PBASIC [8]. A constant gain factor is not used to avoid unnecessary error within the angle calculation.

5. Axis Rotation Operations

Within the right ascension and declination axis microcontrollers, the "serin" function is used along with the "WAIT" parameter. This function is used to program the microcontrollers to wait for a mode of operation, wait for angle data, and also to wait for any start or stop signals from the GUI or other microcontrollers. Upon system start up, the right ascension and declination microcontrollers first wait for a mode to be specified by the user. The mode is specified by sending the string "MOD" followed by a 0 or 1 from the GUI function through the asynchronous serial link to the microcontrollers. After the mode has been specified, both microcontrollers wait for the applicable angle data to be received. The angle data is again sent from the GUI function to the microcontrollers through the serial communication link. The angles are specified by the user within text input boxes inside the GUI figure window. The angle data is read by the microcontrollers using the "serin" function in conjunction with the "WAIT" parameter.

To specify the minimum right ascension angle, the applicable GUI function sends the string "MINR" followed by the minimum right ascension or initial right ascension angle depending on which mode has been selected. If scan mode has been selected, the scan callback function also sends the string "MAXR" followed by the maximum right ascension angle. The right ascension microcontroller is programmed to store the data following the strings "MOD", "MAXR", or "MINR".

To specify the minimum declination angles for the declination axis microcontroller, the string "MIND" is sent followed by the minimum declination angle specified by the user. The maximum declination angle is sent following the string "MAXD". This process is similar to that of the right ascension axis microcontroller.

After the mode and angle data has been received, the microcontroller program uses a simple if statement to determine what mode of operation should be used. If mode is specified as 0, the microcontroller goes to the "track" subroutine. Upon entering the track subroutine, operations to determine the sign of the maximum and minimum angles is done. The angles can range from – 50 to + 50 degrees. After the angle data has been processed, a loop is entered that measures the pulse width of the applicable accelerometer axis and converts the pulse width into a tilt angle measurement. This process is done as discussed in the angle of tilt data processing section above. Once a current tilt angle has been determined, the program compares the current tilt angle to the minimum angle specified by the user. The minimum angle is also defined as the starting or initial angle for both the track and scan functions. By comparing the current tilt angle to the desired initial position angle, the program can determine a desired direction output state to set the stepper drive to. After setting the direction pin to the determined state, the loop outputs one 0.2 ms pulse to the step pin. This causes the stepper drive to step the applicable motor one step in the determined direction. This loop continues until the current angle that is read in by the microcontroller from the accelerometer is equal to the minimum or initial angle specified by the user.

The right ascension and declination axis microcontrollers are programmed to communicate with each other during the initial positioning operations. The declination microcontroller is programmed to wait for the string "GO" to be sent from the right ascension microcontroller. The declination microcontroller does not enter the initial positioning loop until this string is received. After the right ascension microcontroller has determined that the initial right ascension angle has been reached, it sends the string "GO" and waits for the string "B". The string "GO" signals the declination microcontroller that the initial right ascension position has been reached and allows the declination microcontroller to enter the initial positioning loop. After the initial declination position has been reached, the declination microcontroller sends the string "START" back to the GUI function. The declination microcontroller then waits for the string "A".

The string "START" allows the GUI function to enter a loop that begins reading the data from the digital video camera mounted on the telescope eyepiece. When a star or some other light emitting object comes into the field of view of the camera, the object is represented by a high valued pixel. Since the pixel values of the camera are represented using 8 bit unsigned integers within the GUI function, a white pixel is represented as 255 while a black pixel is represented as 0. The GUI function reads the values given by the camera until it finds a value greater than a threshold of 200. A value greater than 200 would indicate a light object such as a star or planet that is reflecting light. The GUI function determines the row and column indices of this pixel and determines whether the camera needs to move up or down, left or right in order to center the high valued pixel in the camera frame. If the function determines that the camera needs to move up, it sends the string "A" followed by a 1 to the declination microcontroller. If downward movement is needed, the GUI function sends the string "A" followed by a 0 to the declination microcontroller. If rightward movement is needed, the GUI function sends the string "B" followed by a 1 to the right ascension microcontroller. If leftward movement is needed, the GUI function sends the string "B" followed by a 0 to the right ascension microcontroller. The GUI function then goes back to reading the values given to it by the camera. The GUI function remains in this loop until a user specified time has passed.

When the declination microcontroller receives the string "A" followed by a 1, it sets the direction output pin high and sends a 0.2 ms pulse to the step output pin. This causes the stepper drive to step the declination axis in a positive direction. If the string "A" followed by a 0 is received, the declination microcontroller sets the direction pin low and pulses the step output pin. This causes the stepper drive to move the declination axis motor in a negative direction. The microcontroller then goes back to waiting for the string “A” until the GUI function sends the string “MOD”. This string specifies that the tracking session is finished and another session will start.

When the right ascension microcontroller receives the string “B” followed by a 1, it sets the direction output pin high and sends a 0.2 ms pulse to the step output pin. When the string “B” followed by a 0 is received, the microcontroller sets the direction output pin low and pulses the step output pin. This causes the stepper drive to step the right ascension axis motor in either a positive or negative direction. The right ascension microcontroller then goes back to waiting for the string “B” until the string “MOD” is received. “MOD” specifies that the user has ended the tracking session and wishes to begin another session.

A flow chart for the tracking subroutine within the declination and right ascension microcontrollers is shown in Fig. 4.

If mode is specified by a 1, the microcontrollers will go to the scan subroutine. Upon entering the scan subroutine, the declination microcontroller will determine the sign of the maximum and minimum declination angles. After these operations are complete, the declination microcontroller will wait to receive the string “GO” from the right ascension microcontroller. When the right ascension axis microcontroller enters the scan subroutine, it also determines the sign of the maximum and minimum right ascension angles. After performing these operations, the right ascension microcontroller enters a loop in which it reads the right ascension angle from the accelerometer, determines the direction in which the motor needs to move, then steps the motor in that direction. This loop is continued until the angle read from the accelerometer matches the angle specified by the minimum right ascension angle. Upon completing this loop, the right ascension microcontroller sends the string “GO” to the declination axis microcontroller and waits to receive the string “UP”.

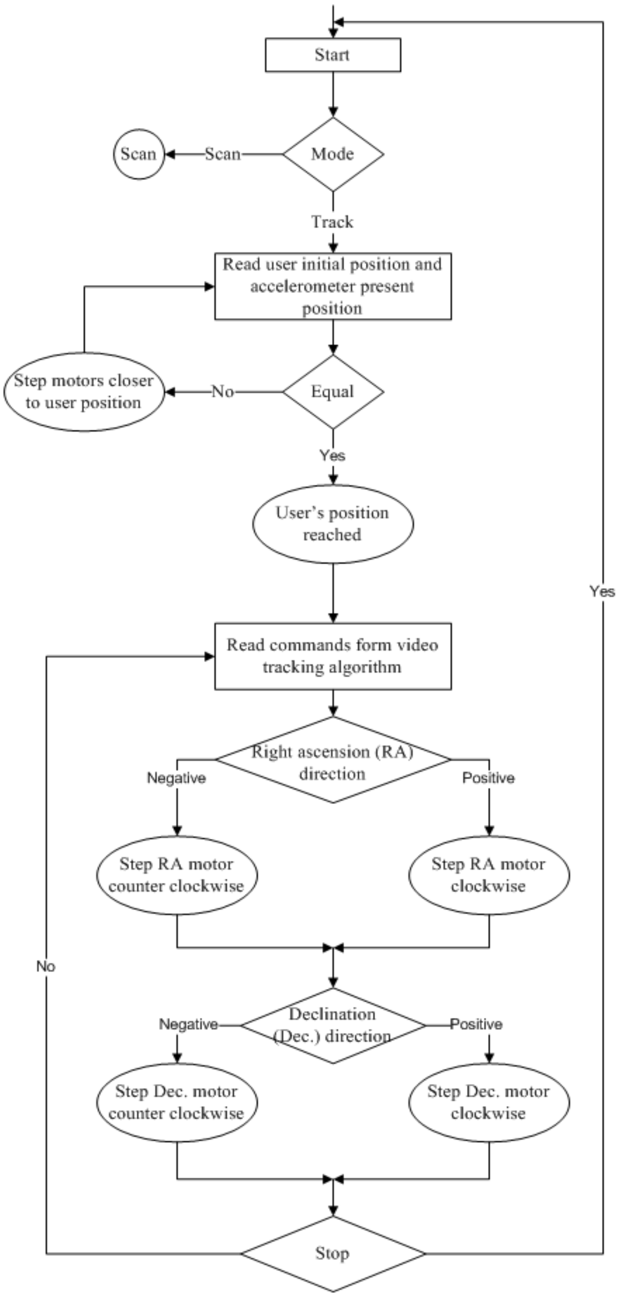

Start
Scan
Scan
Mode
Track
Read user initial position and accelerometer present position
Step motors closer to user position
No
Equal
Yes
User’s position reached
Yes
Read commands form video tracking algorithm
Right ascension (RA) direction
Negative
Positive
Step RA motor counter clockwise
Step RA motor clockwise
No
Declination (Dec.) direction
Negative
Positive
Step Dec. motor counter clockwise
Step Dec. motor clockwise
Stop

Fig. 4 Flow chart for track subroutine.

After receiving the string “GO”, the declination microcontroller moves the declination axis to the initial declination angle. This is done in the same manner that the right ascension microcontroller moves the right ascension axis to its initial position. After the initial declination position is reached, the declination microcontroller sends the string “GO” to the accelerometer microcontroller and waits for the string “START” to be sent from the GUI. The accelerometer microcontroller receives “GO” from the declination microcontroller and sends the string “1234” to the GUI. The string “1234” is received through the scan callback function inside the GUI function. Upon receiving this string, the scan callback function in the GUI sends the string “START” to the declination microcontroller. The GUI function then begins reading the data given to it by the camera. Upon reading a pixel value that is greater than 200, the GUI function sends the string “ANG” to the accelerometer microcontroller. When this string is received by the accelerometer microcontroller, the tilt angle data is read from the accelerometer, processed, and sent back to the GUI function. The GUI function continuously reads the camera data until the maximum right ascension angle has been reached. The GUI determines this by reading the angle data from the accelerometer microcontroller.

Upon receiving the string “START”, the declination microcontroller moves the declination axis to the maximum declination angle specified in the beginning of the program. This is done by entering a loop in which the present angle of the declination axis is determined and compared to the desired angle. The direction of rotation is then determined and a pulse is output to the declination stepper drive +step input. This loop is continued until the current angle and the maximum angle are equal. After exiting this loop, the pedestal declination axis is at the maximum declination angle. The declination microcontroller then sends the string “UP” to the right ascension axis microcontroller and waits to receive the string “GO”.

Upon receiving the string “UP”, the right ascension microcontroller reads the right ascension angle from the accelerometer, compares it to the maximum right ascension angle, and determines the desired direction of rotation. The right ascension microcontroller then sends a pulse to the right ascension stepper drive which steps the motor in the specified direction. The string “GO” is then sent to the declination microcontroller.

After receiving this string the declination microcontroller moves the declination axis back to its minimum angle. This process is repeated until the present right ascension angle matches the maximum right ascension angle. The overall effect of this subroutine is that the telescope moves from minimum declination to maximum declination, up one degree of right ascension, back to minimum declination, and repeats itself until the maximum right ascension has been reached.

Fig. 5 shows a flow chart pertaining to the scan subroutine used in the right ascension microcontroller. Fig. 6 shows a flow chart pertaining to the scan subroutine used in the declination axis microcontroller.

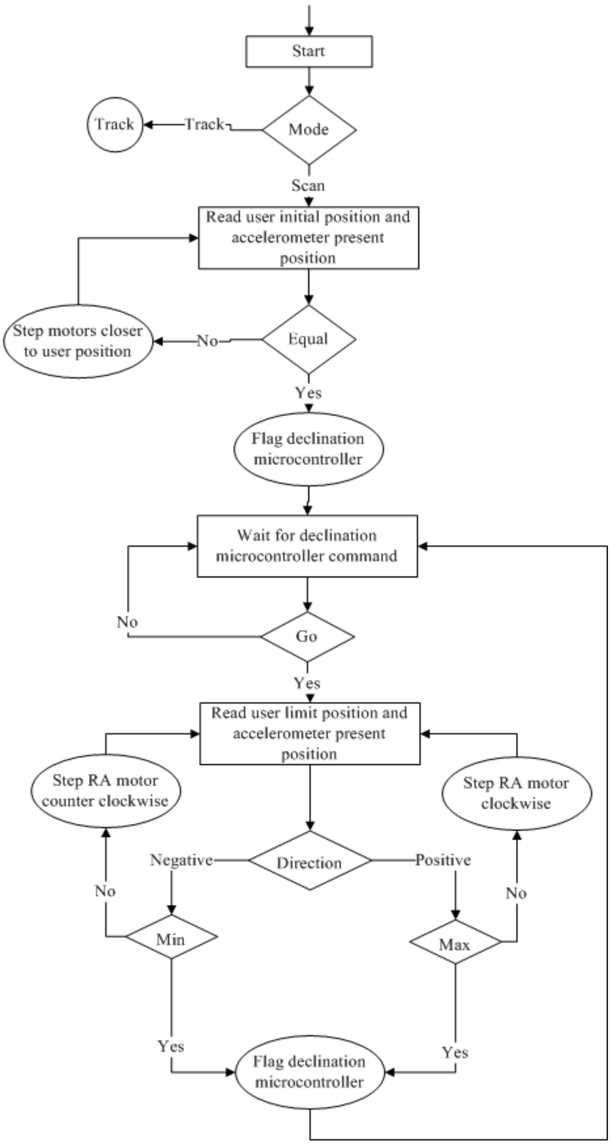

Start
Mode
Track
Track
Scan
Read user initial position and accelerometer present position
Equal
No
Step motors closer to user position
Yes
Flag declination microcontroller
Wait for declination microcontroller command
Go
No
Yes
Read user limit position and accelerometer present position
Step RA motor counter clockwise
Step RA motor clockwise
Negative
Direction
Positive
No
No
Min
Max
Yes
Yes
Flag declination microcontroller

Fig. 5 Flow chart for the right ascension axis microcontroller scan subroutine.

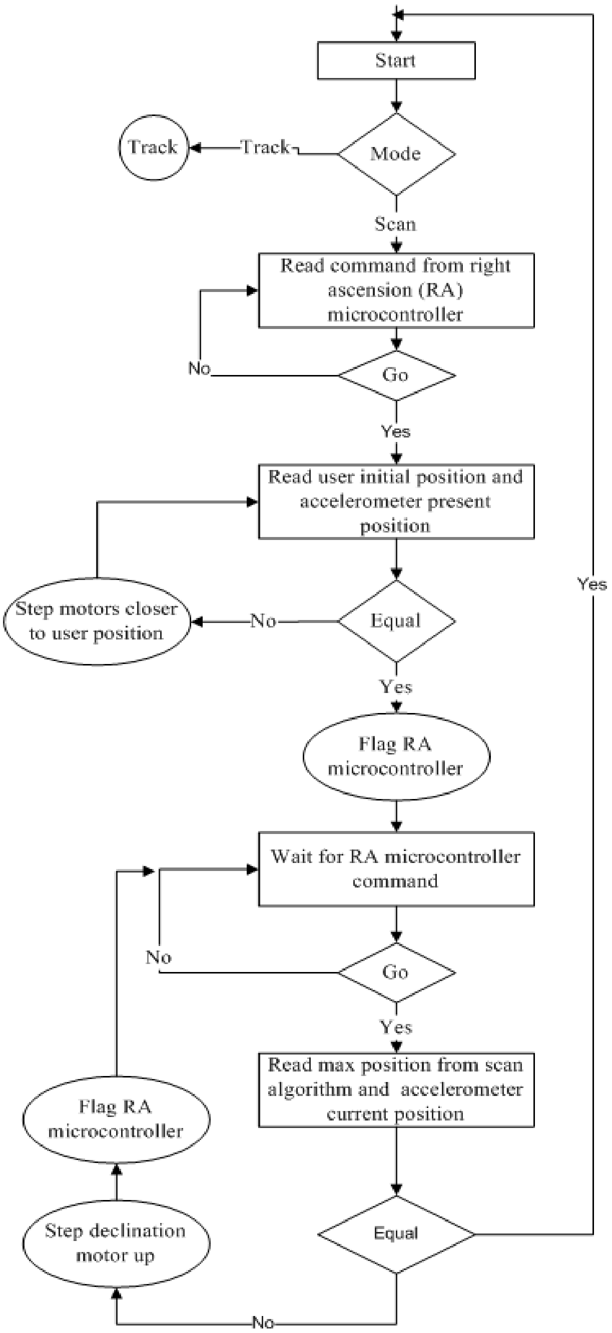

Start
Track
Track
Mode
Scan
Read command from right ascension (RA) microcontroller
No
Go
Yes
Read user initial position and accelerometer present position
Step motors closer to user position
No
Equal
Yes
Flag RA microcontroller
Wait for RA microcontroller command
No
Go
Yes
Read max position from scan algorithm and accelerometer current position
Flag RA microcontroller
Step declination motor up
Equal
No
Yes

Fig. 6 Flow chart for the declination axis microcontroller scan subroutine.

6. Operational Results & Testing

The calibrate function was tested by pressing the calibrate push button within the GUI. It was confirmed that the pedestal rotated about the alignment axis, and the right ascension axis of the pedestal was aligned with the north and south poles. This was confirmed using a magnetic compass.

Next, the tracking function was tested. This was done by first removing the camera from the eye piece of the telescope and placing it over the top of the telescope. This was done since initial testing was indoors and the picture from the camera would show up blurry if the telescope were used at close range. Next, an initial declination and right ascension were entered into the text input boxes in the GUI. The “Start Tracking” pushbutton was then pressed. A common flashlight was used as an object of interest. The light was moved randomly at a moderate speed in front of the camera. As the flashlight was moved, the camera preview window was examined within the GUI. It was confirmed that the light remained within the ten by ten pixel box in the center of the preview axis. This test confirmed that the video tracking algorithm was working properly.

The scan function was then tested. To do this, five lights with different intensities were hung above the telescope pedestal to form a triangular shape. The telescope was manually moved to each flashlight location and the right ascension and declination coordinates were recorded. Next, sufficient maximum and minimum right ascension and declination angles were entered into the text input boxes in the GUI. The “Start Scan” pushbutton was then pressed. The telescope first moved to its initial position then began scanning line by line within the specified area. It was confirmed that the telescope was scanning properly by watching to see if the telescope right ascension and declination angle readings were going from the minimum to maximum as specified. When the maximum right ascension angle was reached, the telescope stopped and the function ended, returning a scatter plot of the coordinates obtained. These coordinates were verified using the list made while manually finding coordinates of the lights. The coordinates were correct. Next the scatter plot was examined. It was verified that the scatter plot did reveal a triangular shape. This confirmed that the scan function was working properly.

7. Conclusions

The intelligent tracking telescope did meet all design specifications. The system provides the user with right ascension and declination coordinates through the use of the accelerometer and electronic compass. The coordinates can be plotted within the graphical user interface. The telescope has also been designed and proven to track a light emitting or light reflecting object. It is able to provide the user with the ability to input system parameters and view system data. The GUI is also able to provide the user with real time video from the camera. The coordinates that the system provides have been proven to be within 5 degrees of accuracy due to the ability and resolution of the chosen components.

Although the main focus of this design was to provide the user with equatorial coordinates, simple changes or additions to this system could easily increase its capabilities and scientific value. A much higher quality telescope could be used to expand the capability of the system. With a higher quality image, more image processing techniques, and better data processing, this general design could lead to the discovery of the very first stars ever formed. One possibility is the addition of another element in the software that would relate the provided coordinates to the equatorial system used by astronomers. This element would take the current longitude and latitude of the pedestal along with the current date and time and use it to correct the coordinates for position and time error. This could provide the system with the ability to be used within a multiple telescope system.

References

[1] Videoinput, Matlab Help Navigator, retrieved: May 7, 2010.

[2] Getdata, Matlab Help Navigator, retrieved: May 7, 2010.

[3] RGB, Matlab Help Navigator, retrieved: May 7, 2010.

[4] Scatter Plot, Matlab Help Navigator, retrieved: May 8, 2010.

[5] Preview, Matlab Help Navigator, retrieved: May 8, 2010.

[6] Graphical User Interface, Matlab Help Navigator, retrieved: May 8, 2010.

[7] Graphical User Inteface, Matlab Help Navigator, retrieved: May 7, 2010.

[8] BASIC StampSyntax and Reference Manual, available online at: http://www.parallax.com/dl/docs/ prod/stamps/ basicstampman.pdf, retrieved: September 30, 2009.

[9] S. Edwards, The Nuts and Volts of Basic Stamps, Column 92: It’s All About Angles, available online at: http://www.parallax.com/dl/docs/cols/nv/vol1/col/nv92.pdf, retrieved: September 30, 2009.

Computer Technology and Application 2 (2011) 858-865

Diagnosing Student Learning Problems in Object Oriented Programming

Hana Al-Nuaim[1], Arwa Allinjawi[1], Paul Krause[2] and Lilian Tang[2]

1. Computer Science Department, King Abdulaziz University, Jeddah 21589, Saudi Arabia

2. Department of Computing, Surrey University, Guildford GU2 7XH, United Kingdom

Received: September 28, 2011 / Accepted: October 18, 2011 / Published: November 25, 2011.

Abstract: Students often face difficulties while taking basic programming courses due to several factors. In response, research has presented subjective assessments for diagnosing learning problems to improve the teaching of programming in higher education. In this paper, the authors propose an Object Oriented conceptual map model and organize this approach into three levels: constructing a Concept Effect Propagation Table, constructing Test Item-Concept Relationships and diagnosing Student Learning Problems with Matrix Composition. The authors' work is a modification of the approaches of Chen and Bai as well as Chu et al., as the authors use statistical methods, rather than fuzzy sets, for the authors' analysis. This paper includes a statistical summary, which has been tested on a small sample of students in King Abdulaziz University, Jeddah, Saudi Arabia, illustrating the learning problems in an Object Oriented course. The experimental results have demonstrated that this approach might aid learning and teaching in an effective way.

Key words: Higher education, programming learning difficulties, object oriented programming, conceptual model.

1. Introduction

Programming is central to any Computer Science (CS) curriculum. Unfortunately, it is often highly challenging, because it is not an easy topic to grasp and requires that students master complex skills [1-2]. Computer Science educators need to fully understand why students struggle to understand programming concepts, which may account for the high failure rate in programming and the subsequent dropout from CS majors [3]. The high attrition and failure rates in introductory programming courses comprise a widespread problem in CS that has motivated many researchers to propose methodologies and tools to help students. It is possible that there are several reasons and factors that cause these learning and teaching difficulties [1], and perhaps the most important reason is the apparent lack of problem solving abilities in many students. When they cannot problem solve, they cannot program, as they do not know how to create algorithms.

Meanwhile, many papers have proposed approaches for improving the teaching of computer programming in Higher Education. However, our observation is that these approaches are normally motivated by a subjective assessment of where the problems lie. It is rare to see (a) any detailed scientific analysis of where the learning difficulties lie and (b) a statistically valid evaluation that affirms a given intervention actually does resolve a specified learning difficulty. This problem is particularly acute in the teaching of Object Oriented programming, a topic that features a wide diversity of opinions on the balance between and sequencing the teaching of object-oriented and procedural concepts. This paper will focus on the

Arwa Allinjawi, lecturer, Ph.D. student, research field: object oriented programming.

Paul Krause, professor, Ph.D. FIMA CMath, research field: software engineering.

Lilian Tang, Ph.D., research field: natural language processing and image processing.

Corresponding authors: Hana Al-Nuaim, D.Sc., research fields: multimedia systems, visualization and HCI. E-mail: hnuaim@kau.edu.sa.

development of a rigorous experimental method that diagnoses learning problems to address weaknesses with the teaching of a module.

The structure of this paper is organized as follows: Section 2 identifies the problems and barriers that arise while teaching and learning Object Oriented Programming (OOP), specifically in the Kingdom of Saudi Arabia (KSA); section 3 discusses the methodology for how the use of a conceptual model might impact the effectiveness of higher education teaching in general, and then describes how we use the conceptual model to diagnose the learning status of students at King Abdulaziz University (KAU) in Saudi Arabia; section 4 presents the results of the case study; In section 5, we conclude the study and recommend future work.

2. Difficulties in Teaching and Learning OOP

Students in many universities often face difficulties in basic programming courses, especially in OOP. A study by an ITiCSE 2001 working group (the McCracken Group) [4] established that in their introductory courses, many students do not know how to program. Many students in the introductory Computer Science (CS) courses begin without adequate problem solving skills or begin with a poor understanding of the basic mathematical and logical tools needed for problem solving. In addition, it appears that rather than learning the basic concepts of the field, the students devote their energies to learning syntax. They resort to trial and error instead of learning real problem solving skills. In addition, rather than getting the big picture of computer science, they narrow their focus to getting a specific program to run [5]. Therefore, many students have a fragile grasp of both basic programming principles and have a limited ability to perform routine programming tasks [4].

In KSA, the widespread acceptance of CS and the fast development of CS in public, private and corporate business areas have led to new, interesting and various opportunities for CS graduates [6]. However, the high-school curricula in KSA are lacking in CS courses and problem solving skills, and they devote inadequate time to teaching the English language. Typically, the majority of students who are admitted into the CS departments in higher education institutions have never had any exposure to CS concepts, and CS is a completely a new domain to them. In addition, in KAU CS departments, teachers find it difficult to instruct object-oriented programming due to the students' lack of English language abilities, limited resources, few labs and few teaching assistants [7]. Moreover, CS faculties in KAU, especially those teaching Object Oriented (OO) courses, face many difficulties while teaching the concepts, such as:

- Students must learn a large number of concepts in OOP while having little time for adequate practice examples;
- Time is insufficient for those who teach introductory programming courses to solve problems that are large enough to demonstrate the benefits of object oriented design;
- Lack of a graphical presentational environment.

Owing to these obstacles in teaching OOP concepts, students:

- Do not fully comprehend the OO development environments;
- Lack understanding of the OOP concepts and have difficulty visualizing the state spaces;
- Cannot implement some of the OO features.

The fact that the students generally have significant difficulties with the concepts in OOP raises the question of what are the specific problems faced by each student for which specific concepts, and which particular concepts do they perceive as being particularly difficult to grasp.

3. Methodology

In higher education, the mission of teaching and supervising students to learn well is complex and multifaceted. In fact, one of the main goals of higher educational institutions is to identify ways for students to be able to think/learn effectively and to develop critical thinking and learning skills. To aid this, the researchers' intention is to improve statistically students' grasp of particular OOP concepts with which they have fundamental problems. Therefore, a statistical summary has been made with a small sample of students to identify a lack of understanding in specific concepts of OOP according to a conceptual model.

The creation of a conceptual model is a part of the process of learning rather than a manifestation of learning itself. Thus, the concept map is becoming an increasingly popular tool in education to represent the meaningful relationships between concepts. Hawang [8] and Chen and Bai [9] propose a conceptual map model that provides learning suggestions by analyzing the subject materials and test results. Their approach offers an overall cognition of the subject contents by identifying the key concepts of the course and the relationships between the concepts. Then, the relationships between subject concepts and test items is determined by analyzing the subject materials and the item bank and determining the learning problems of each student according to these relationships in order to diagnose his or her learning status. This system can provide both objective assessments and personalized suggestions for each student by analyzing the student's answers and the relationships between the subject concepts and the test items. Moreover, it might be useful to help the instructors to focus on particular concepts by indentifying the key concepts that need to be covered in the course, and it may help the instructors to design a particular exam related to the course that might effectively diagnose the students' specific learning problems with particular concepts.

Therefore, to diagnose the learning status of a student, this study employed a concept effect propagation approach to detect the student's learning problems in the OO course. In OO courses, students learn new concepts and new relationships among previously learned concepts. This knowledge can be represented as a conceptual model to illustrate the relations between concepts. Fig. 1 illustrates the hierarchy of related OO concepts and may produce adequate methods of measurement and assessment. These concepts were taught in OO course (CPCS203) in the first semester, year 2010/2011, at the Computing and Information Technology College, female section, KAU, Jeddah, KSA. The flow of the conceptual model in Fig 1 illustrates the concepts' relationships according to the course syllabus and the text books, "A Comprehensive Introduction to Object-Oriented Programming with Java", "Java How to Program: Early Objects Version" and "Java, Object-Oriented Problem Solving", and administered by the faculty member who is currently teaching the OO course. However, the design is mainly concerned with the basic concept of building classes; it does not focus on the high level of the OO concepts in this course, such as abstract concepts and polymorphism. These concepts were not included because the faculty did not address them in the mid-term exam. Additionally, the complexity of these concepts would have added to the model if they were included, making it difficult to demonstrate the problems faced by the students while learning OO concepts. However, these concepts could be included in a separate conceptual model to diagnose the difficulties of high level OO concepts.

The approach and model steps to diagnose the problems followed Chu et al.'s model [10]. The approach constructed three levels, which are presented in the following details.

3.1 Constructing a Concept-Effect Propagation Table

A concept-effect propagation table (CEPT) is used to model the concept-effect propagation relationships among concepts. The CEPT records all concepts that may be influenced by another concept in the student learning process. For example, in Table 1, the concepts affected by C_1 are C_1, C_4…C_{15}. Also, the concepts affected by C_{10} are C_{12}, C_{13}, C_{14} and C_{15}. In general, C_j represents the concepts affected by C_i. If CEPT(C_i,C_j)=1,

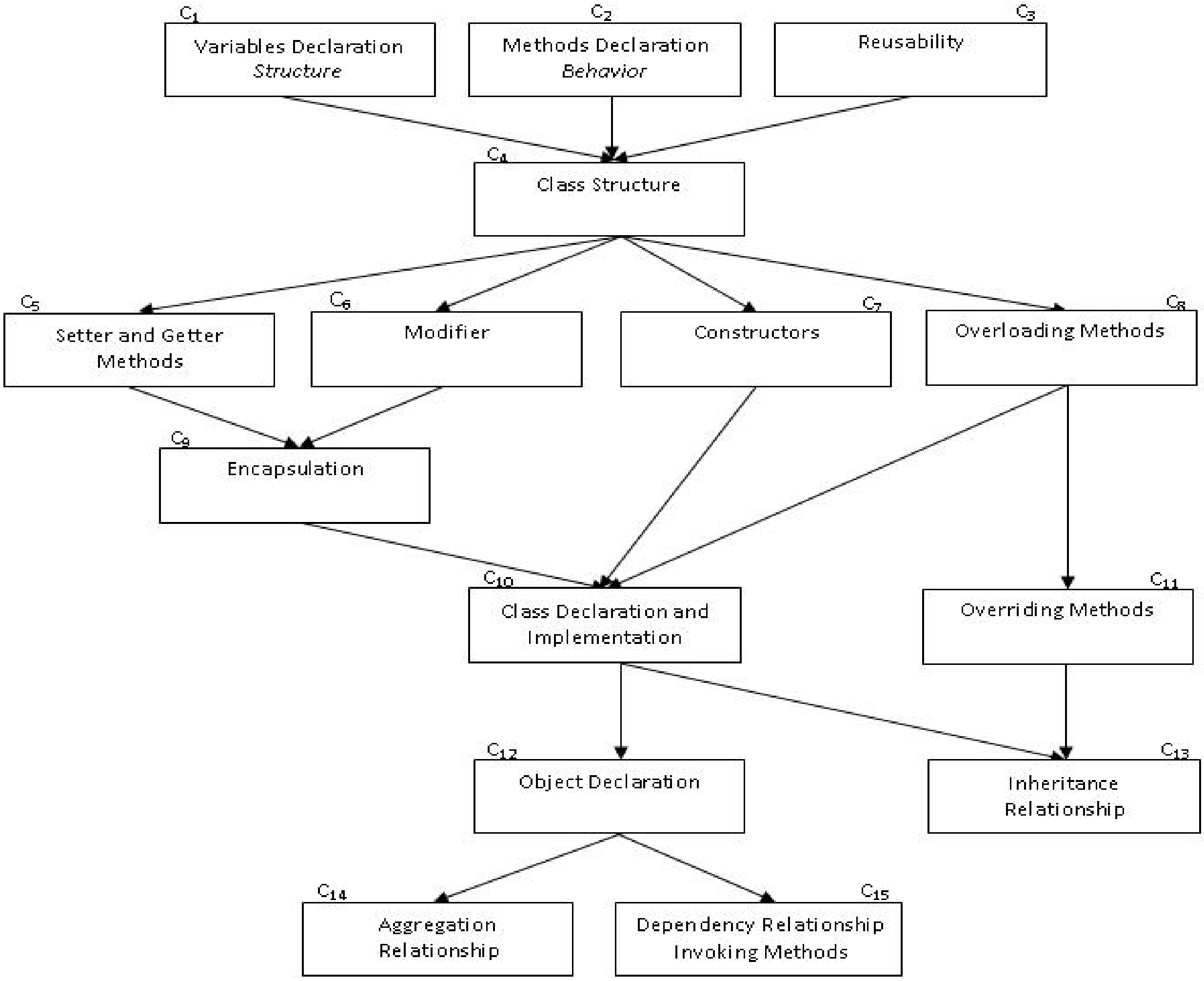


Fig. 1 Object oriented concept-effect relationship diagram.

Table 1 Object oriented concept-effect propagation table (CEPT).

		Propagated effect concept C_j														
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
Prerequisite concept C_i	C_1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	C_2	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	C_3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	C_4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	C_5	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1
	C_6	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
	C_7	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1
	C_8	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1
	C_9	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1
	C_{10}	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1
	C_{11}	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
	C_{12}	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
	C_{13}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	C_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	C_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

that means "C_j is one of the concepts affected by C_i during the student learning process", and if CEPT(C_i,C_j)=0, that means "C_i is not a prerequisite concept of C_j", and student does not need C_i to learn C_j.

3.2 Constructing Test Item-Concept Relationships

A midterm exam was given to 26 computing and information technology students on the 27th of December 2010, as three students were absent. The exam consisted of three parts that examined the students' basic logical thinking in OO concepts.

Part 1 (Q_1) asks students to trace a given code to generate the output. It examines the students' understanding of code flow (arrays and loops) with the inheritance structure, and it notes how they pass parameters to the constructors and methods in the parent and child classes. However, the arrays and loops marks were not included in the test item model, as the model handles only OO concepts. The total possible marks for part 1 was 5.

Parts 2 and 3 cover most of the course concepts, which required the students to write and design complete code according to the questions' requirements. If a student answered these two parts correctly, then she understood most concepts in OOP. Moreover, part 2 was divided into two items (Q_2 and Q_3). Q_2 examines how students design the given hierarchy model, and Q_3examines how they declared each child class. This division helped to demonstrate the students' difficulties with specific concepts. The total possible marks for part 2 is 10, divided into 3.5 for Q_2 and 6.5 for Q_3.

Part 3 is also divided into two items (Q_4 and Q_5). Q_4 asks students to illustrate an aggregation relationship and to declare a class with its behavior and structure. Q_5 is about how students would structure an application and declare objects to perform the requirements. The total marks for part 3 is 10, divided into 5 for Q_2 and 3.5 for Q_3. The remaining 1.5 marks that assess the loop and array structures were not included in the test item model, because structural programming was out of the present study's scope.

Table 2 presents the test item relationship table (TIRT). Each TIRT(Q_n, C_i) entry represents the degree of association between test item Q_n and concept C_i, which was calculated according to each item's (Q_n) mark. Each part of the solution on each item was related to specific concepts. The total mark for each concept in Q_n is represented in the TIRT(Q_n, C_i) entry. For example, Q_3 was associated with the inheritance concept by the degree 2 out of 6.5. However, since each Q_n has a different mark, the degree of the association range must be unified and calculated to be from 0 to r, where 0 indicates that the concept is not associated with the item and r was valued to be 1. This range was based on Chu et al.'s [10] proposed model. For example, as a result, Q_3 was associated with the inheritance concept by the degree 0.308 out of 1.

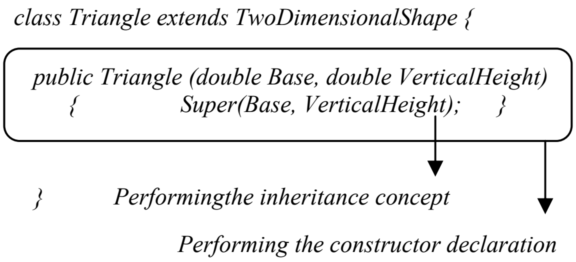
Furthermore, it is important to recognize that the total TIRT entries of all concepts in each item (Q_n) is not essentially equal, as addressed in Chu et al.'s model [10]. Each TIRT(Q_n, C_i) entry is valued according to each statement "code" that the students wrote while answering the questions. It could be that one statement is related to two or more concepts, which means the statement's mark must be assigned to both concepts. For example, Fig. 2 shows that in Q_3, when students wrote the constructor of an inherited class correctly, they demonstrated that they understood the inheritance structure by forming the super identifier and the constructor declaration concepts. Therefore, the mark for this statement is equally associated with the two mentioned concepts.

3.3 Diagnosing Student Learning Problems with Matrix Composition

This matrix represents the students' answers in an answer sheet table (AST). Table 3 illustrates the marks of 23 students who answered the midterm exam, where each entry AST(S_k, Q_n) is a value ranging from 0 to 1; 0 indicates that student S_k answered test item Q_n correctly, 1 indicates that S_k failed to answer Q_n correctly, and a

Table 2 Object oriented test item relationship table (TIRT).

		Concept C_j														
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
	Q_1	0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0.3
Test	Q_2	0.357	0	0.857	0.286	0	0.214	0.357	0.071	0.143	0	0.071	0	0.214	0	0
item	Q_3	0	0.462	0	0	0.231	0	0.462	0	0.231	1	0.231	0	0.308	0	0
Q_n	Q_4	0.15	0	0	0	0.3	0	0.25	0	0	0	0	0	0	0.3	0
	Q_5	0	0	0	0	0	0	0	0	0	0	0	0.571	0	0	0.429



```
class Triangle extends TwoDimensionalShape {
    public Triangle (double Base, double VerticalHeight)
    {        Super(Base, VerticalHeight);    }
}
```

Performingthe inheritance concept
Performing the constructor declaration

Fig. 2 A constructor of an inherited class.

Table 3 Object oriented answer sheet table (AST).

		Test item	Q_n			
		Q_1	Q_2	Q_3	Q_4	Q_5
	S_1	0.050	0.500	0.923	0.450	0.071
	S_2	1.000	0.857	1.000	1.000	1.000
	S_3	0.300	0.571	0.462	0.400	0.786
	S_4	0.300	0.571	0.923	0.400	0.500
	S_5	0.000	0.000	0.231	0.300	0.000
	S_6	0.900	0.929	0.923	1.000	1.000
	S_7	0.050	0.286	0.231	0.300	0.000
	S_8	0.000	0.214	0.038	0.500	0.000
	S_9	0.000	0.357	0.500	0.300	0.357
	S_{10}	0.050	0.286	1.000	0.400	0.000
The	S_{11}	0.200	0.929	0.231	0.700	0.071
students	S_{12}	0.000	0.500	0.654	0.400	0.643
S_k	S_{13}	0.400	0.786	0.231	0.300	1.000
	S_{14}	0.050	0.714	0.000	0.850	0.857
	S_{15}	0.300	0.571	0.923	0.350	1.000
	S_{16}	0.000	0.357	0.308	0.150	0.000
	S_{17}	0.000	0.714	0.115	0.450	0.357
	S_{18}	0.900	0.929	0.923	0.600	0.714
	S_{19}	0.000	0.357	0.231	0.300	0.000
	S_{20}	0.000	0.214	0.231	0.300	0.000
	S_{21}	0.000	0.000	0.269	0.300	0.000
	S_{22}	0.250	0.143	0.231	0.300	0.000
	S_{23}	0.050	0.500	0.500	0.300	0.000

value between 0 and 1 indicates a partially correct answer. The answered values were calculated to be within a unified range, as shown in the Eq. (1), because each item has a different mark. The values have been subtracted from 1 to be equivalent with the 0 and 1's indication. Therefore, the value of AST(S_k, Q_n) ranged from 0 to 1, as addressed in Chu et al.'s model [10]:

$$AST(Sk, Q_n) = 1 - ((Sk, Q_n)\ mark\ /the\ Q_n\ total\ mark) \quad (1)$$

4. Results and Discussion

After generating the three tables for AST, TIRT and CEPT, we must establish a relationship between the tables in order to diagnose each learning problem individually. Therefore, following Chu et al.'s study [10], we used a Max-Min composition method.

To illustrate the Max-Min composition, let

$$R1=\{(x,y)|(x,y)\in X\times Y\}\ and\ R2=\{(y,z)|(y,z)\in Y\times Z\} \quad (2)$$

and the Max-Min composition will be

$$R1\ o\ R2 = \{\ (x,z)|(x,z) = Max\{Min\ \{\mu R1(x,y),\ \mu R2(y,z)\}\}\ for\ x\in X,\ y\in Y\ \ and\ z\in Z\} \quad (3)$$

Therefore, applying the above method to the given tables can derive the error degree for each student's S_k regarding each concept:

$$Error\_Degree(S_k,C_j) = AST(S_k,\ Q_n)\ o\ TIRT(Q_n,C_i)\ o\ CEPT(C_i,\ C_j) \quad (4)$$

The following is an example to illustrate the relationship:

$$AST\ o\ TIRT\ (S_1,C_1) = MAX\{MIN((S_1,Q_1),\ (Q_1,C_1));\ MIN((S_1,Q_2),\ (Q_2,C_1));\ MIN((S_1,Q_3),\ (Q_3,C_1));\ MIN((S_1,Q_4),\ (Q_4,C_1));\ MIN((S_1,Q_5),\ (Q_5,C_1))\} \quad (5)$$

Table 4 shows the error degree for each student with each concept after performing the above Eq. (4) of *Error_Degree(S_k,C_j)*.

However, Chu et al.'s[10] model used a fuzzy inference to generate learning guidance for each student. In their work, the reasons why specific membership functions were used to generate the learning guidance was not made sufficiently clear, and we are not confident in the validity of their use. Therefore, in our case study, a statistical summary has been used to identify the students' lack of understanding within these particular concepts in OOP.

Table 4 The error degree table.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
S_1	0.357	0.462	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.923	0.500	0.923	0.923	0.923	0.923
S_2	0.357	0.462	0.857	0.857	0.857	0.857	0.857	0.857	0.857	1.000	0.857	1.000	1.000	1.000	1.000
S_3	0.357	0.462	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.571
S_4	0.357	0.462	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.923	0.571	0.923	0.923	0.923	0.923
S_5	0.150	0.231	0.000	0.231	0.300	0.231	0.250	0.231	0.300	0.300	0.231	0.300	0.300	0.300	0.300
S_6	0.357	0.462	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.923	0.857	0.923	0.923	0.923	0.923
S_7	0.286	0.231	0.286	0.286	0.300	0.286	0.286	0.286	0.300	0.300	0.286	0.300	0.300	0.300	0.300
S_8	0.214	0.038	0.214	0.214	0.300	0.214	0.250	0.214	0.300	0.300	0.214	0.300	0.300	0.300	0.300
S_9	0.357	0.462	0.357	0.462	0.462	0.462	0.462	0.462	0.462	0.500	0.462	0.500	0.500	0.500	0.500
S_{10}	0.286	0.462	0.286	0.462	0.462	0.462	0.462	0.462	0.462	1.000	0.462	1.000	1.000	1.000	1.000
S_{11}	0.357	0.231	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.857
S_{12}	0.357	0.462	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.654	0.500	0.654	0.654	0.654	0.654
S_{13}	0.357	0.231	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786
S_{14}	0.357	0.000	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714
S_{15}	0.357	0.462	0.571	0.571	0.571	0.571	0.571	0.571	0.571	0.923	0.571	0.923	0.923	0.923	0.923
S_{16}	0.357	0.308	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357
S_{17}	0.357	0.115	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714
S_{18}	0.357	0.462	0.857	0.857	0.857	0.857	0.857	0.857	0.857	0.923	0.857	0.923	0.923	0.923	0.923
S_{19}	0.357	0.231	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357	0.357
S_{20}	0.214	0.231	0.214	0.231	0.300	0.231	0.250	0.231	0.300	0.300	0.231	0.300	0.300	0.300	0.300
S_{21}	0.150	0.269	0.000	0.269	0.300	0.269	0.269	0.269	0.300	0.300	0.269	0.300	0.300	0.300	0.300
S_{22}	0.150	0.231	0.143	0.231	0.300	0.231	0.250	0.231	0.300	0.300	0.231	0.300	0.300	0.300	0.300
S_{23}	0.357	0.462	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500

By generating the mean, median and the standard deviation of the error degrees for each concept, we obtained, with significant confidence, results that demonstrate in which particular concepts students are facing problems while learning. Table 5 shows that most of the students have problems with class declaration and implementation, object declaration, inheritance relationships, aggregation relationships and dependency relationship concepts. This means students might understand the prerequisite concepts of the declaration of classes, but they are experiencing difficulty getting the big picture, learning how to demonstrate the relationship between classes in depth and learning how to structure a high level application that performs particular requirements.

5. Conclusions

Diagnosing students' learning problems with a specific course is an important research topic in adaptive learning systems. In this paper, we have presented a model for identifying the learning problems of students who are learning OOP concepts in KAU. The result illustrates that students do not demonstrate a strong ability to use effectively the relationships between classes to design and structure a high level application that meets particular requirements with an OO environment. However, more work needs to be done to clearly diagnose these problems, because the students may have run into problems with the exam structure, not grasping the abstract concepts or the instructors' teaching style, which may not be matching the students' needs.

We believe identifying the key concepts and their relations in different fields to create a conceptual map model might be useful. It may help the instructors to design a particular exam that might effectively diagnose the students' learning problems with any given concept. Through the identification of such problems, instructors can adjust their instruction to the students to improve their learning process.

Table 5 The statistical summary of the error degrees of each concept.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
Mean	0.31139	0.32300	0.48126	0.51978	0.53448	0.51978	0.52390	0.51978	0.53448	0.62717	0.51978	0.62717	0.62717	0.62717	0.62717
Median	0.35700	0.30800	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000	0.65400	0.50000	0.65400	0.65400	0.65400	0.65400
Std. deviation	0.077367	0.151318	0.272548	0.226749	0.208321	0.226749	0.221391	0.226749	0.208321	0.272460	0.226749	0.272460	0.272460	0.272460	0.272460

In addition, we have illustrated our approach with a small sample of students, which leaves us with low confidence in the results' consistency. Our next step will be to repeat the same experiment during the next semester with a richer statistical and assessment method supported by a larger number of students. These statistical assessment methods will measure the students' performance, skills, knowledge and their ability to comprehend the same particular concepts. The methods will also identify whether these students understand the OO relationship concepts or not. Moreover, we know that it will be challenging to assess the students, because we are aware of the challenges in the debate and the case studies for engaging visualization tools with educational aspects, such as the advantages of using visual tools and the educators' confidence andpersistence with using these tools while teaching OOP concepts. After statistically defining the students' comprehension barriers with the OO relationship concepts, we will also find it challenging to assess, by using a quantitative analysis, another group of students engaged with the proposed teaching methods (visualization tools) within particular concepts (inheritance, aggregation and dependency) and to assess whether that change will make a difference in the students' learning outcomes.

References

[1] E. Lahtinen, K.A.-Mutka, H.-M. Järvinen, A study of the difficulties of novice programmers, SIGCSE Bulletin 37 (2005) 14-18.

[2] T. Jenkins, On the difficulty of learning to program, in: Proc. 3rd LTSN for Information and Computer Science Conference, UK, 2002, pp. 65-71.

[3] F.P. Deek, H. Kimmel, J.A. McHugh, Pedagogical changes in the delivery of the first-course in computer science: problem solving, then programming, Journal of Engineering Education 87 (1998) 313-320.

[4] R. Lister, E.S. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, R. McCartneyet, J.E. Moström, K. Sanders, O. Seppälä, B. Simon, L. Thomas, A multi-national study of reading and tracing skills in novice programmers, SIGCSE Bulletin 36 (2004) 119-150.

[5] V.H. Allan, M.V. Kolesar, Teaching computer science: a problem solving approach that works, SIGCUE Outlook 25 (1997) 2-10.

[6] A.S. Al-Salman, J. Adeniyi, Computer science education in a Saudi Arabian university: a comparative study of its B.Sc. program, SIGCSE Bull. 32 (2000) 34-39.

[7] A. Allinjawi, The use of ALICE as a 3D interactive programming environment to teach novice programmers concepts in OOP, Master, Computer Science Department, King Abdulaziz University, Jeddah, 2007.

[8] G.-J. Hwang, A conceptual map model for developing intelligent tutoring systems, Computers & Education 40 (2003) 217-235.

[9] S.-M. Chen, S.-M. Bai, Learning barriers diagnosis based on fuzzy rules for adaptive learning systems, Expert Syst. Appl. 36 (2009) 11211-11220.

[10] H.-C. Chu, G. Hwang, J. C. R. Tseng, G. Hwang, A computerized approach to diagnosing student learning problems in health education, Asian Journal of Health and Information Sciences 1 (2006) 43-60.

Computer Technology and Application 2 (2011) 866-883

Computing All Pairs Shortest Paths on Sparse Graphs with Articulation Points

Carlos Roberto Arias[1, 2] and Von-Wun Soo[1, 3]

1. Institute of Information Systems and Applications, National TsingHua University, Hsinchu, Taiwan

2. Facultad de Ingenierías, Universidad Tecnológica Centroamericana, Tegucigalpa, Honduras

3. Computer Science Department, National TsingHua University, Hsinchu, Taiwan

Received: September 06, 2011 / Accepted: September 29, 2011 / Published: November 25, 2011.

Abstract: In most network analysis tools the computation of the shortest paths between all pairs of nodes is a fundamental step to the discovery of other properties. Among other properties is the computation of closeness centrality, a measure of the nodes that shows how central a vertex is on a given network. In this paper, the authors present a method to compute the All Pairs Shortest Paths on graphs that present two characteristics: abundance of nodes with degree value one, and existence of articulation points along the graph. These characteristics are present in many real life networks especially in networks that show a power law degree distribution as is the case of biological networks. The authors' method compacts the single nodes to their source, and then by using the network articulation points it disconnects the network and computes the shortest paths in the biconnected components. At the final step the authors proposed methods merges the results to provide the whole network shortest paths. The authors' method achieves remarkable speedup compared to state of the art methods to compute the shortest paths, as much as 7 fold speed up in artificial graphs and 3.25 fold speed up in real application graphs. The authors' performance improvement is unlike previous research as it does not involve elaborated setups since the authors algorithm can process significant instances on a popular workstation.

Key words: Graph algorithms, all pairs shortest paths, articulation points.

1. Introduction

The All Pairs Shortest Paths problem has been studied for a long time, and is one of the classical problems in combinatorial network optimization in Graph Theory [1-2]. But more than a theoretical problem it has profound importance in many practical areas: routing of vehicles and supplies, drug design, etc.. Traditional solutions were presented by Dijkstra [3], and Floyd [4], both solving the problem of finding the all pairs shortest paths in a graph with $O(n^3)$ time performance. Since then several algorithms have been presented that improve this bound [1, 5], taking advantage of data structures that were being introduced, and the specific topology of real world graphs. Parallel algorithms have been also presented in Refs. [6-8], these take advantage of external hardware (FPGA, Field Programmable Gate Array), powerful computers in their implementation, or the similarity of the computation of shortest paths to matrix multiplication.

In this paper, we present a method that shows improvement in time performance to the all pairs shortest path (APSP) problem when the articulation points are given and that can be executed using popular workstations. Our method benefits on the fact that many real world application graphs are sparse, and that graphs from biological networks, utilities networks, routing networks and others have many articulation points and nodes that we call singles, i.e., nodes with only one incoming connection. Taking advantage of these characteristics we compact the single nodes of the

Corresponding author: Carlos Roberto Arias, Ph.D. candidate, research fields: bioinformatics, disease gene prioritization. E-mail: carlos.r.arias@gmail.com.

network and then disconnect the components of the graph, achieving a time performance improvement. If an application computes the articulation points beforehand, then our method shows remarkable improvement in the computation of the shortest paths as much as a seven fold on some instances and at least it shows not much worse than state of the art methods to compute them as we show in the results section of this paper.

The development of this method is motivated by an increasing need of the all pairs shortest paths computation as a step to calculate other graph properties, such as: diameter, closeness centrality and betweenness centrality [9]. These properties, also known as graph invariants, are extensively used in several network analysis tools [10-11], and are fundamental for methods currently being developed for disease gene prioritization [12].

This paper is organized as follows: Section 2 presents the necessary background and definitions needed for the rest of the paper, and some previous and related work as well; section 3 makes a formal and detailed description of our method; section 4 shows our results on different types of graphs; finally, in section 5 we present our conclusions.

2. Background and Previous Work

In this section we present definitions and conventions used throughout the paper. We start with definition of graphs and the all pairs shortest path problem, followed by the definition of articulation points and single vertices. Then we discuss about previous solutions and approaches to this problem.

2.1 Graphs

A graph is a data structure that represents a set of relationships between elements or objects. Formally a graph G is a pair defined by $G = (V, E)$, where V is a set of elements that represent the nodes or vertices of the graph, the vertices may or may not hold information, for the purpose of this paper it is indifferent that information is held in the vertices of the graph, so we will hold natural numbers on them, thus $V = \{x|x\in N\}$. And E is the set of edges, where each edge represent a relation between two vertices, an edge is defined by $E = \{(u, v)|u, v\in V\}$, this edges may hold additional information as weight. The edges may represent direction, where $(u, v) \neq (v, u)$, in which case the graph is called directed graph, and when direction is not important, the graph is called undirected graph. This paper deals with undirected graphs, but the method can be easily adapted to directed graphs. We denote that n is the number of vertices in the graph, formally $n = |V|$, and e is the number of edges, $e = |E|$.

2.2 All Pairs Shortest Paths

The purpose of this problem is to build a matrix holding the shortest path value from every vertex to every other vertex, that is to find, for each pair of vertices $u, v\in V$, a shortest (least weight) path from u to v, where this shortest path is the sum of the weights of the shortest route from u to v [13]. The weights are assumed to be real numbers according to $\{weight(u, v)|weight(u, v)\in Re$ and $weight(u, v)\geq 0\}$, if weights were allowed to be negative, different approaches would be needed. This can be solved by computing the Single Source Shortest Path (SSSP) algorithm, like Dijkstra or its variations, for each of the vertices of the graph, or can be solved directly by using other methods like Floyd-Warshall or Johnson algorithm.

It has been shown that good Dijkstra’s algorithm implementations on SSSP, that is an implementation using the min-priority queue with a Fibonacci heap, can run on $O(n\log(n)+e)$ [1, 13], and if we are dealing with sparse graphs where $e \ll n^2$ thus having $e = O(n)$, then we can have a much better bound: $O(n\log(n)+n) \approx O(n\log(n))$, so if we run this for each vertex on the network we get $O(n^2log(n))$ for sparse graphs. In our experiments we use a min-priority queue implemented with a Min-heap, we call this ImprovedDijkstra algorithm, this one has a performance of $O((n+e)\log(n))$, and taking the assumption of sparse graphs we get the bound close to $O(nlog(n))$ for each vertex of the graph, for a total of $O(n^2log(n))$.

2.3 Articulation Points and Single Vertices

An articulation point, also referred as Cut Vertex, is a vertex v such that when it is eliminated from the graph along with all its incident edges $V' = V - \{v\}$ and $E' = E - \{(v, u) | u \in V\}$ a new graph $G' = (V', E')$ is created, and this new graph is divided in two or more connected components [14]. If all articulation points of a graph are found, then we would have found all the maximal biconnected components of the graph, which would form a tree whose elements are all the articulation points and the biconnected components. A single vertex is a vertex with only one incident edge, thus having degree equal to one.

2.4 Previous and Related Work

A comprehensive study on both theoretical and empirical evaluation of the Shortest Path problem was made by Cherbassky et al. [2], where they study several algorithms on an assorted set of networks, they conclude that for networks without negative arcs, as is our case, Dijkstra is the best choice. Our study approaches the problem in a serial fashion. There are others who have taken a parallel approach to the All Pairs Shortest Path (APSP) problem, using external and accessory hardware (FPGA) [6] and blocking techniques to achieve a speedup of 22 over the execution without using FPGA. Dey and Srimani [8] propose a solution that solves the APSP problem in $O(n^2log(n))$ using a massive amount of $O(n^3)$ processors. The previous solutions present good proposals for the solution of our problem, but they require the use of very special hardware, which may not be available for everyone interested in an application of the APSP problem. Usually APSP is a part of a system, so the solution should be easily integrated with these systems. This integration is seen in Tang et al. [7] for a system that deals with road networks, where they partitioned the graph in order to run the APSP algorithm on each part of the networks; and managed to achieve a 15 fold speed up with their method, but this was done over an IBM cluster.

3. Method Description

The method we propose in this paper takes advantage of three characteristics of real world networks:

- Sparseness, where the graphs, even though they are connected, have an amount of edges that is closer to be linearly related to the number of vertices, than to be quadratic on them;
- Abundance of Single Nodes, this property occurs in web connectivity networks and in biological networks as well to mention some;
- Existence of Articulation Points, it is common for real world networks to possess vertices that separate the network into biconnected components.

Due to these characteristics we developed a method that improves the time performance of the computation of the All Pairs Shortest Paths.

3.1 Input Assumptions

The method assumes that the graphs that are fed as input have the following properties:

- The graph is connected, or the first vertex is connected to the major component. In biological graphs there are many vertices that are disconnected from the main component, so we try to deal only with the biggest component;
- There are no negative weights.

As it can be seen the method does not make an assumption on the values of the edges or on the directedness of the graph. Therefore the method works on edges with positive weights, and with minor adaptations on directed graphs.

3.2 Input Preparation

This stage takes care of reading the graph from a data file and prepares it to the rest of the process and the result is a representation of the graph in an

Adjacency List. This stage could be avoided if the input graphs ensure the following additional properties:

- The graph has only one connected component;
- The vertices do not have “self-edges”, that is edges of the form $(u, u)|u \in V$;
- The labels of the vertices are Natural Numbers.

For this reason, this stage is not considered in the time performance measurement, but it does not add much to the overall bound since this stage takes only $O(n)$ time, since the three steps it does are linearly proportional to the number of vertices.

Additionally, a Distance Matrix is created based on the structure of the prepared graph representation. This Distance Matrix is implemented as an upper triangular, after all, the method deals with undirected graphs and by using this representation substantial memory savings can be achieved. A position $Matrix[i][j]$ represents the distance between vertex i and vertex j. If these two vertices are not connected, the value of ∞ is stored in that position. We are going to illustrate using the graph (Fig. 1) to show step by step how the method works on each stage. Notice that this graph complies with the mentioned assumptions.

3.3 Preprocessing

On this phase of the method the graph is prepared in such a way that the method can do a remarkable time improvement. The second part of this stage can be avoided if the system where this method is being implemented already computed the articulation points of the graph, thus saving a lot of computation time as we show in the results section of the paper.

3.3.1 Compact Singles

Taking advantage of this characteristic, the first step is to compact the singles so they are not considered in the APSP or on the Disconnection Phase, therefore gaining some performance time. Although single nodes detection is not intrinsically a part of the articulation points’ computation, it must be done before the cut vertex identification to prevent the formation of many single node components. In this step we traverse the Adjacency List to find all the single nodes, and relate them to the node they are connected to; this way at a posterior stage we can expand them again. This is done by keeping a separate stack that stores each of the nodes that were compacted, the node that it was compacted to and the distance between them. This process is done iteratively until there are no more singles. The worst case scenario for this process is that the graph forms a tree structure, in which case it would take $O(n^2)$ to process, $O(n)$ since it has to traverse the Adjacency List in search of singles, and repeat this as much as $O(n)$ time as this would be the maximum height of the degenerate tree case. If it was a balanced tree we get a better bound $O(n\log(n))$ since the height of the tree is at most $O(\log(n))$. It is key to see that if we run into the worst case on this phase, the whole graph would be compacted into one node, making all the posterior phases unnecessary, thus the performance of this phase, added to the Expansion Phase would be the total performance of computing all pairs shortest paths. The result of the compaction stage is shown in Fig. 2 where it can be seen that nodes 08, 10 and 12 in Fig. 1 are compacted into node 06’; then after the first compaction, node 11’ (compacted from node 11 in Fig. 1) becomes a single and needs to be compacted again, as Fig. 2 shows into 07’. The stack is called the “operation stack” and it will help expand the single nodes on the expansion stage of the method.

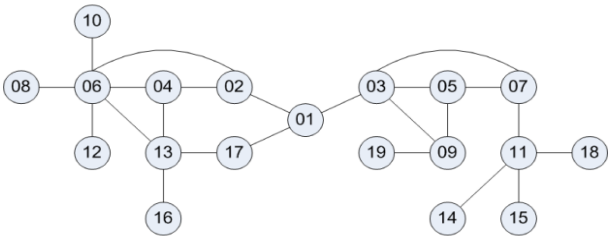


Fig. 1 A graph sample that follows the sparseness assumptions.

3.3.2 Disconnect Graph

On this step we compute the articulation points and biconnected components using an adaptation of the implementation of Hyung-Joon Kim[1], this algorithm performance bound is $O(n+e)$, which by our sparseness

[1] http://www.ibluemojo.com/school/articul_algorithm.html, retrieved: December 2010.

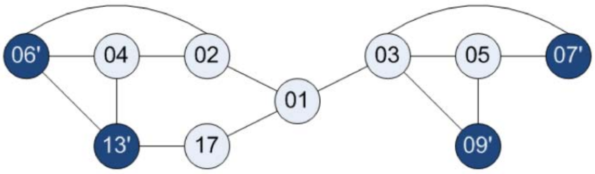


Fig. 2 Compacting singles.

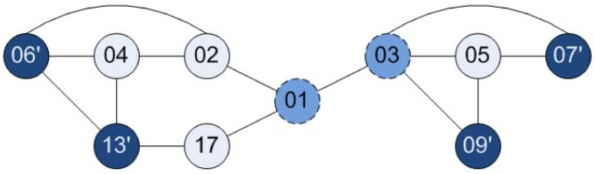


Fig. 3 Finding articulation points.

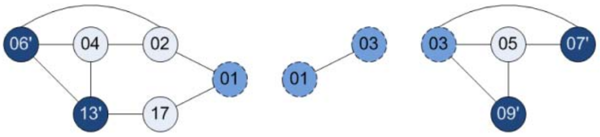


Fig. 4 Disconnect the graph on the articulation points.

assumption comes to be $O(n)$. This stage, as it will be shown on the results section, proved to add too much time to the performance of the method, this is due to the intrinsic nature of the computation of the articulation points. For this reason the method is of more use and provides exceptional time improvement when the articulation point computation is needed also for the system where the APSP method is going to be used in and they are provided to the method. The articulation points are detected as denoted in light blue in Fig. 3 and after the graph is partitioned at nodes 01 and 03 three separate components are obtained as shown in Fig. 4.

3.4 Compute APSP

Once the graph is successfully disconnected, it is time to compute the APSP on each of the components. This process is done using the same distance matrix that we have on the whole graph, this way we manage to save memory on this computation. Each of the components will store their results on their corresponding segment of the distance matrix, where they will be used on the next stage of the method.

3.5 Fold Distance Matrix

At this moment we have the APSP of all of the components of the graph, now we need to add them up to finish filling all the distance matrix. To fill the matrix we use Algorithm 1.

Line 3 and 5 avoid computation of elements that are not yet computed on that row and column respectively. Line 6 verifies if a position has not been defined before, it avoids computing the sum unnecessarily. Line 7 is the reason why this step is called matrix folding; because it works as we were folding the matrix to add the values together using the articulation point row and column as pivots to the folding process.

```
input  : List of Articulation Points APList
output: Updated Distance Matrix

1 foreach Articulation Point AP in APList do
2     for I ← 0 to N do
3         if distance(I, AP) <> ∞ then
4             for J ← 0 to N do
5                 if distance(J, AP) <> ∞ then
6                     if distance(I,J) = ∞ then
7                         distance(I,J) ←
                          distance(I,AP) +
                          distance(AP,J)
```

Algorithm 1 Fold distance matrix.

When the biconnected component with vertices {06’, 04, 13’, 02, 17, 01} is being merged back with biconnected component with vertices {01, 03, 05, 09’, 07’} on the articulation point 01, the values on row 01 and column 01 are being used as pivot. The cells that remain empty on the distance matrix indicate cells for which the APSP has not been computed yet, or we currently do not care about, for instance nodes {08, 10, 12, 16, 14, 15, 18, 19, 11} are singles that are going to be taken care at the expand singles stage of the process.

3.6 Expand Singles

This is the last stage of the process, and it uses the operation stack mentioned in the Compact Singles stage. On this stage each of the compacted single

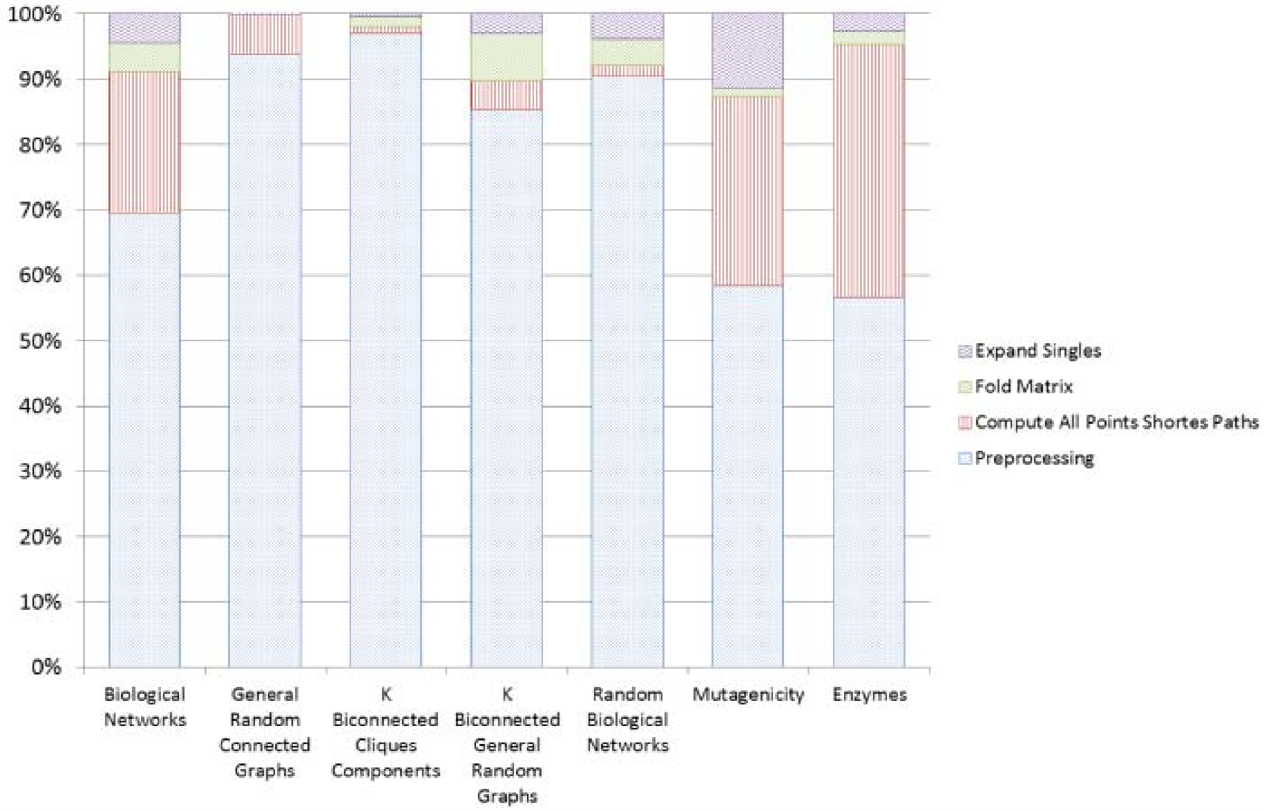


Fig. 5 Time consumption proportion on each of the method steps.

vertices is extracted from the operation stack, due to the nature of the stack it is guaranteed that the graph is going to be correctly reconstructed. For each vertex that is extracted from the stack the distance between it and the vertex it is connected to is updated, then the distance from the extracted vertex and every other vertex is updated. This process is described in Algorithm 2.

4. Results

This section offers the results obtained on our experiments, using the aforementioned method. We made experiments on several kinds of networks; Figs. 7-14, show the results where we compared the time performance of our method versus the inverse density of the graphs, the number of components and the relative size of the biggest component. The inverse density of a graph is given by the following definition:

$$InverseDensity(G)=\left(\frac{2e}{n(n-1)}\right)^{-1} \quad (1)$$

```
input  : Operation Stack Operation Stack
output: Updated Distance Matrix

while Operation Stack is not Empty do
    Pop an Operation from the stack, place it in curOp
    distance(curOp.father, curOp.single) ← curOp.distance
    foreach Vertex v, where v <> curOp.father AND v <> curOp.single do
        distance(v,curOp.single) ← distance(curOp.father, v) + curOp.distance
```

Algorithm 2 Expand singles.

We decided to use the inverse density because it combines nodes and edges together, therefore having a better representation of size of the graph and an idea of the topology of it as well. As this value grows bigger it means the graph is less dense, consequently the number of nodes dominating the relation.

Different charts are presented for each set of experiments, these are:

(a) Comparison between the ratio of *Improved Dijkstra* versus *KC-APSP* and the ratio of *Improved Dijkstra* versus *KC-APSP* with preprocessing where the Improved Dijkstra refers to the state of the art APSP algorithm implementation, that is an implementation of Dijkstra using a Min-heap; the *KC-APSP* curve refers to our implementation of the method presented on this paper, this curve assumes that the articulation points are provided in advance, unlike the *KC-APSP* with preprocessing that includes the time to find the articulation points;

(b) Presents the time fold of our method in terms of the number of components;

(c) Also presents the time fold but in this instance in terms of the relative size of the biggest component on the graph, this is relative to the total number of the nodes. In some instances (Figs. 7-10 and 12), this chart shows also the number of components on each point.

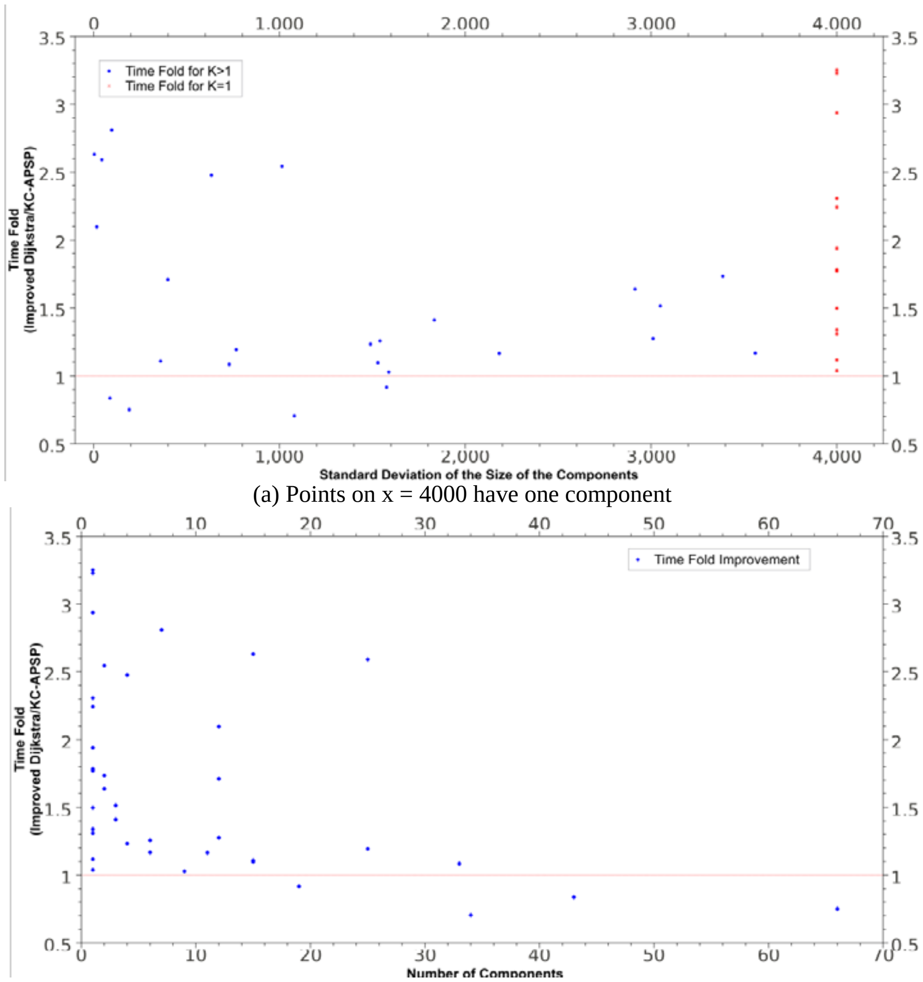
The networks on which we made our experiments are:

- Real Biological Graphs: This is a set of different protein-protein interaction (PPI) networks: H. sapiens, H. pilori, E. coli, S. cerevisae, C. elegans, D. melanogaster, Human Prostate Cancer PPI, and some specific PPI we have been working with, such as CASP3 related networks. See Fig. 7. On this set of graphs we can see that our method outperforms in 90% of the instances the Improved Dijkstra implementation, and that we get as much as 3.25 time fold on certain instances. In full PPI networks we noticed that, after the disconnection phase, most graphs were left with few components, one of them had only six components where the biggest one represented more than 80% of the graph's vertices, for this instance we got a time fold close to 1.2. On the other hand, there was a network that produced 66 components and resulted in one of the poorest time fold, which happened because most of the components where small ones of size between two and six, and one big component of 44% of the size of the network. This also can be seen in Fig. 6a where the "$k > 1$" (more than one component) curve shows that as the standard deviation of the size of the components increases the time fold decreases. These cases occur due mostly to the scale-free nature of PPI networks, so the method gained more time because of the singles merger than for the computation of APSP on smaller components. However for other networks we got several components of varied sizes, hence surpassing the performance of the Improved Dijkstra Algorithm. Additionally, in Fig. 6a on the "$k = 1$" curve is shown that even when there is only one component there are time gains using our algorithm, this is mostly due to the network compaction during the singles merger phase, this can be seen in Fig. 6b where we compare the percentage of compaction to the time fold, these percentages represent the proportion of nodes after compaction, for instance the higher time folds are achieved thanks to a more than 75% compaction.

- General Random Connected Graph: This is a collection of 50 random generated graphs that follow the $G(n, p)$ model presented by Mitzenmaker [15], with the addition that the graphs are assured to be connected (See Fig. 8). As expected this set of graphs show no significant improvement over the conventional algorithm, this is due to the complete lack of articulation points and almost no single nodes (see Figs. 8b-8c) where all graphs are marked to have only one component, and that this component represents the 100% of the graph.

- K Biconnected Cliques Components: This is a set of 50 graphs of cliques interconnected to each other in a random fashion that forces a high probability of articulation points (See Fig. 9). Clearly seen on this set of graphs, our method is better than the conventional one, and uniformly improving the time fold as more components are present on the graph.

- K Biconnected General Random Graphs: This is a set of 29 random generated graphs that are similar to the K-Connected Cliques on the general structure, but the components are not cliques but general random connected graphs (See Fig. 10). These results are



(a) Points on x = 4000 have one component

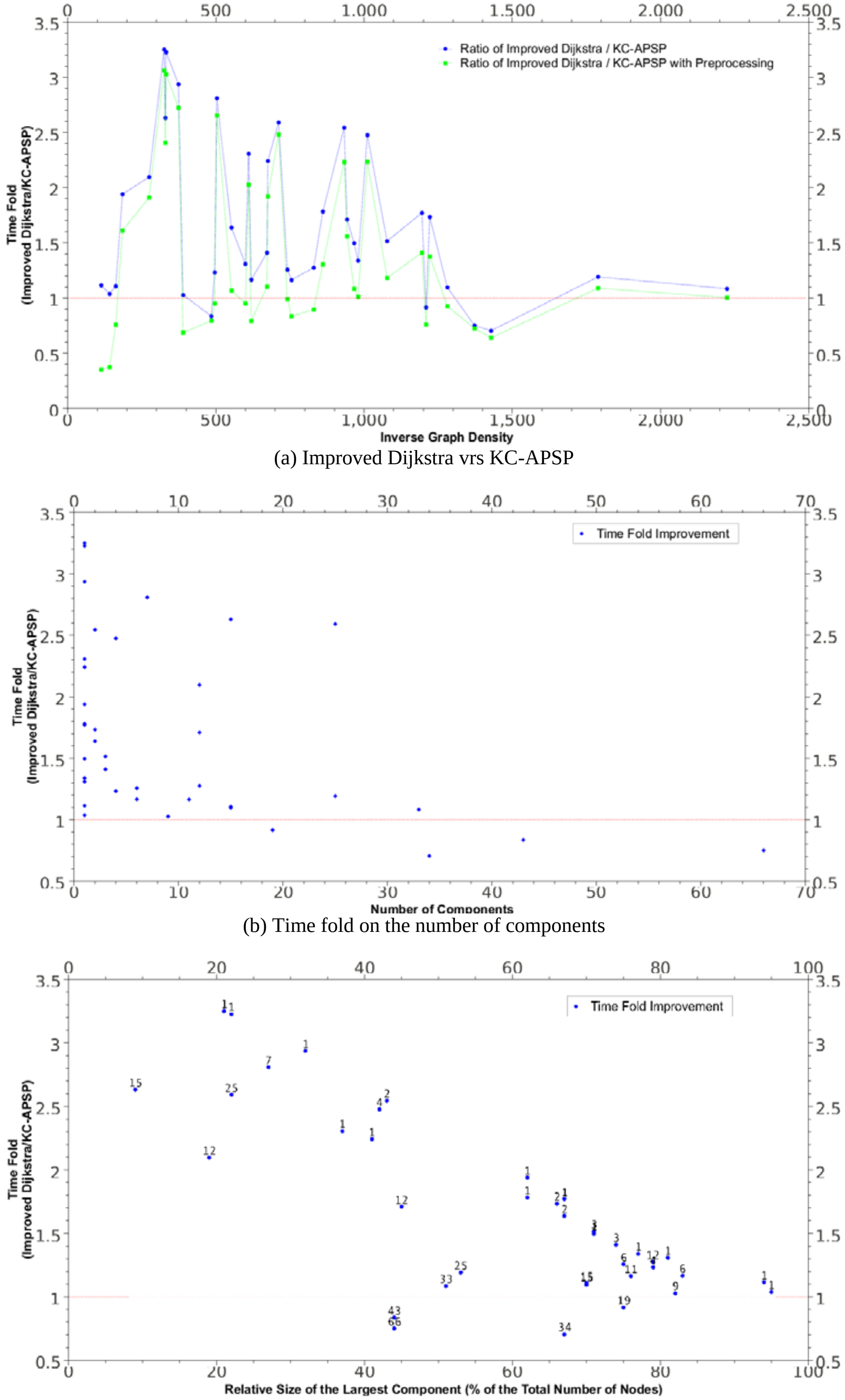
(b)

Fig. 6 Time fold on biological networks.

consistent with the previous set of graphs, differences are found as a result of the random structure of the components.

- K Biconnected Random Connected Graphs with same Sized Components: This is a set of 177 networks of assorted sizes, where the components of each network have similar sizes. In Fig. 11a, it can be seen that our method outperforms the Improved Dijkstra implementation in all instances as long as the articulation points are calculated in advance. Part (b) shows that as the number of components increases the time fold increases, but not as steadily as expected, closer inspection revealed that the other factor influencing the time fold is the size of the graph, so in (b) smaller graphs are lower in the chart than bigger graphs on each component size set. In (c) it is shown that the smaller components in terms of the size of the network have better performance than bigger components.

- Random Biological Networks: This is a set of 25 random generated graphs that follow a power law degree distribution, and with nodes with high probability of being single nodes. These networks vary in size from 172 nodes to a little over 4,800 nodes (See Fig. 12). This set of graphs shows an average 2.5 time fold over the Improved Dijsktra Method, this is thanks to the abundance of components and single nodes. These graphs share behavior with the Real Biological Graphs, where the amount of singles and the variability on the size of the components play a crucial role on the performance of our method, but, just like the Real Biological Graphs, always outperforming the straightforward implementation.

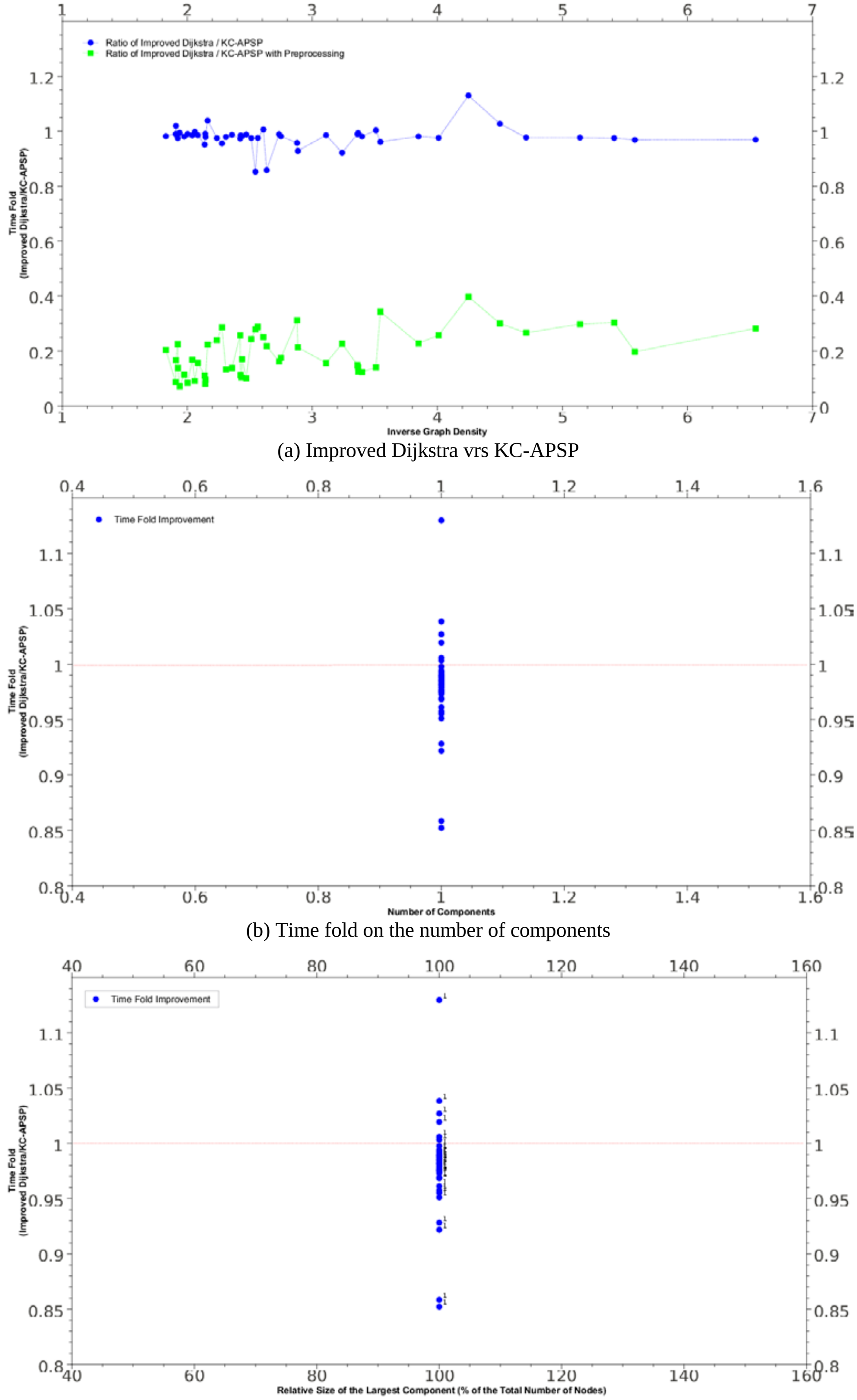


(a) Improved Dijkstra vrs KC-APSP

(b) Time fold on the number of components

(c) Time fold on the relative size of the largest component

Fig. 7 Biological networks experimental results.

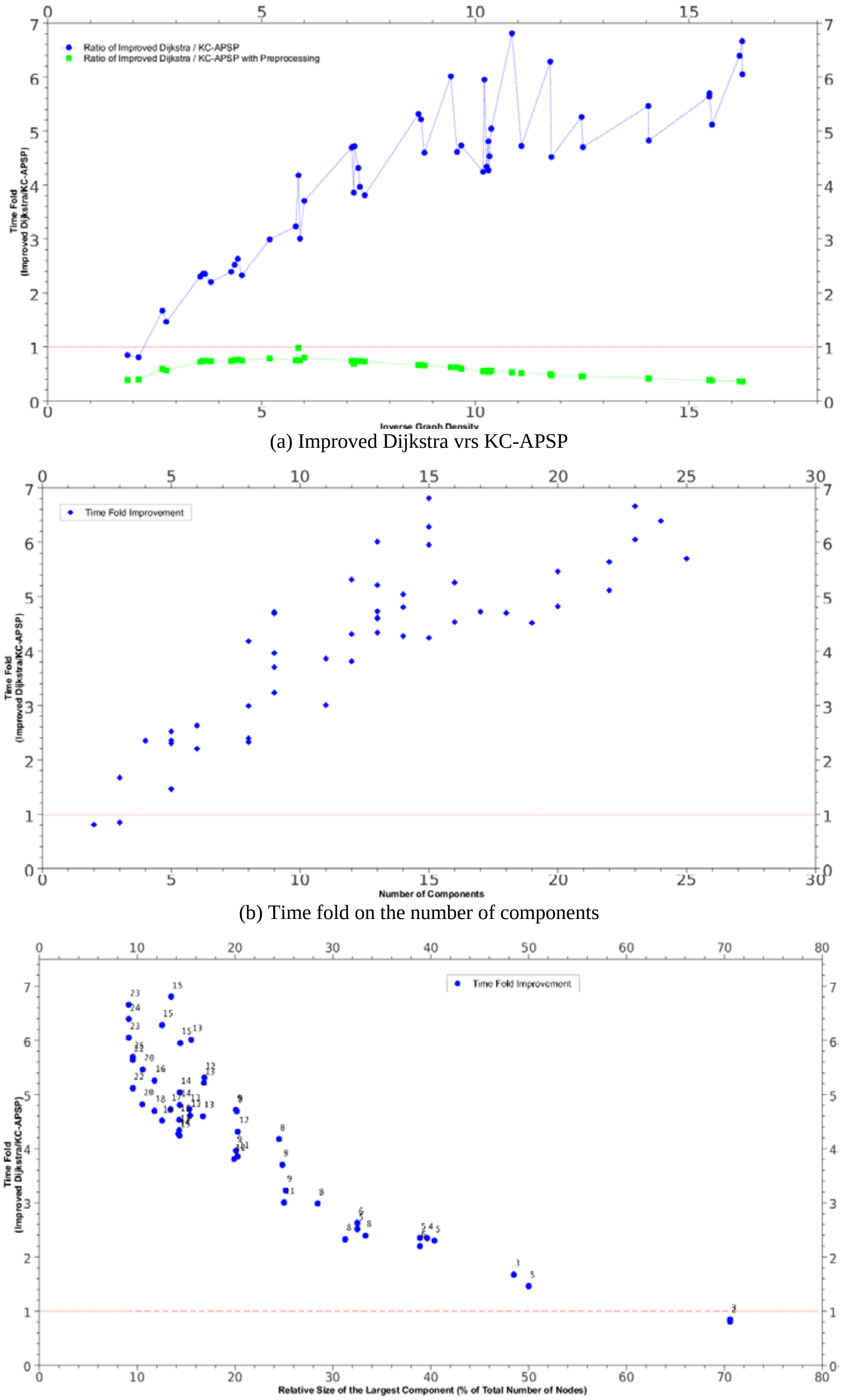


(a) Improved Dijkstra vrs KC-APSP

(b) Time fold on the number of components

(c) Time fold on the relative size of the largest component

Fig. 8 Regular random connected graphs experimental results.

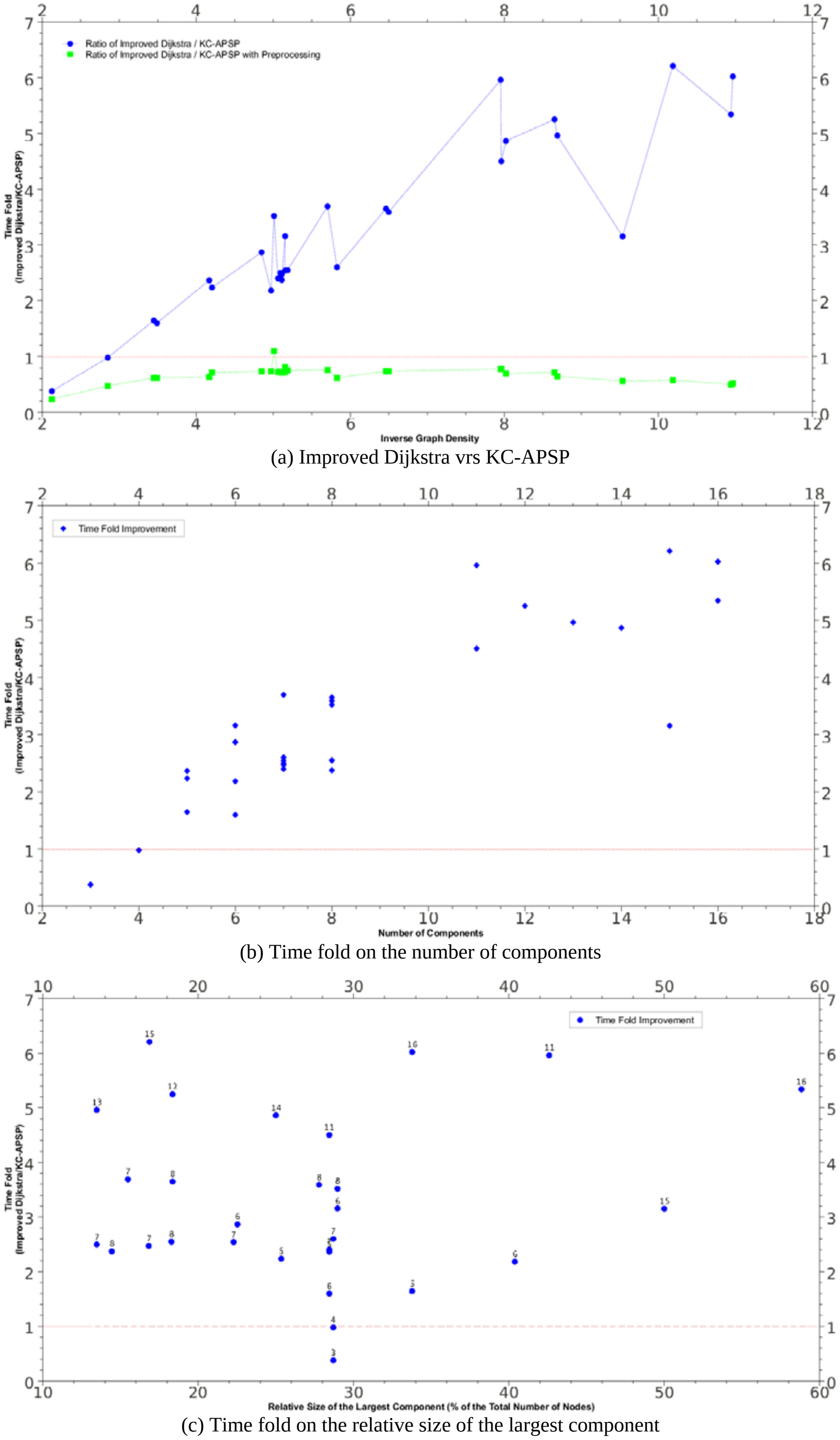


(a) Improved Dijkstra vrs KC-APSP

(b) Time fold on the number of components

(c) Time fold on the relative size of the largest component

Fig. 9 K biconnected cliques experimental results.

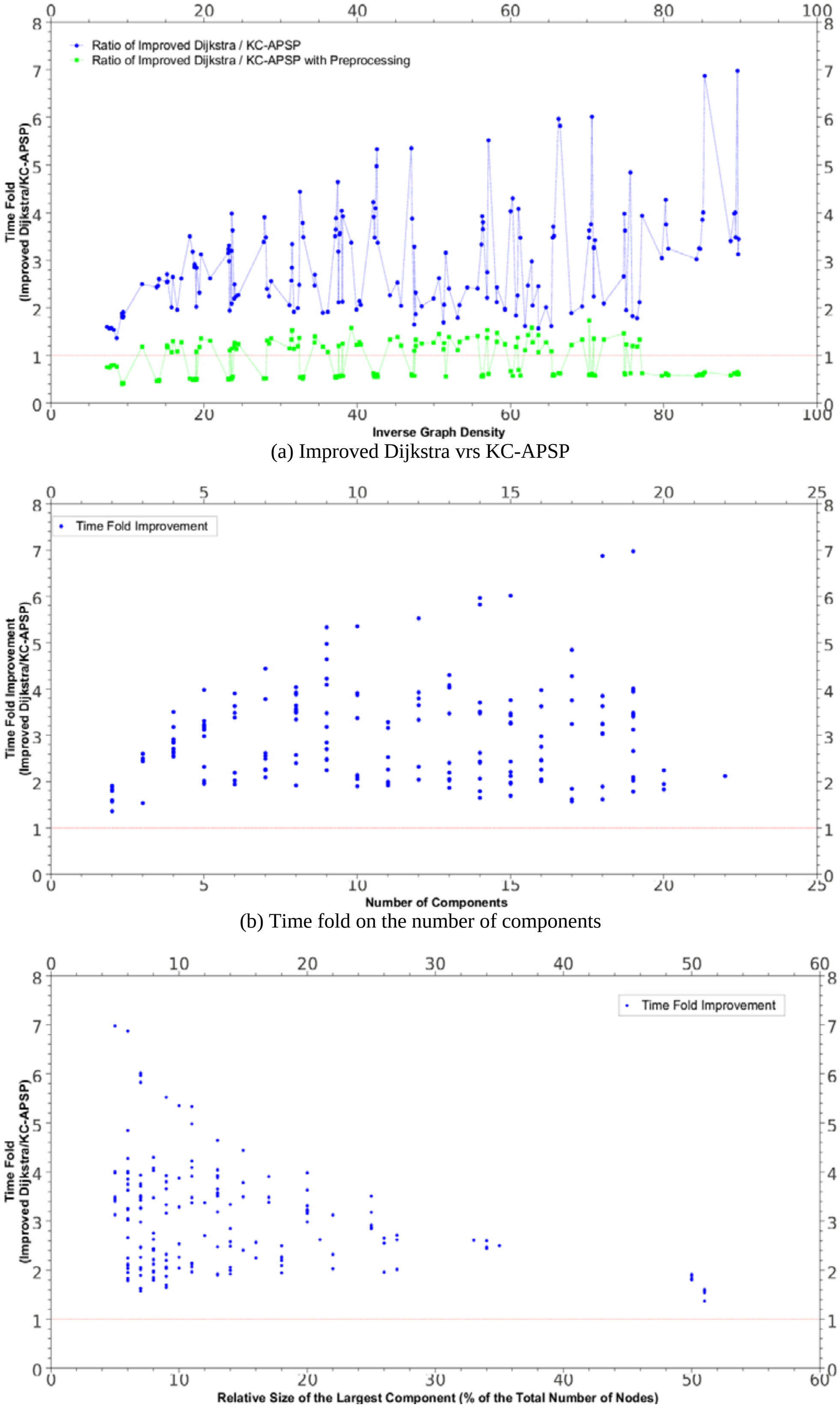


(a) Improved Dijkstra vrs KC-APSP

(b) Time fold on the number of components

(c) Time fold on the relative size of the largest component

Fig. 10 K biconnected random connected graphs experimental results.

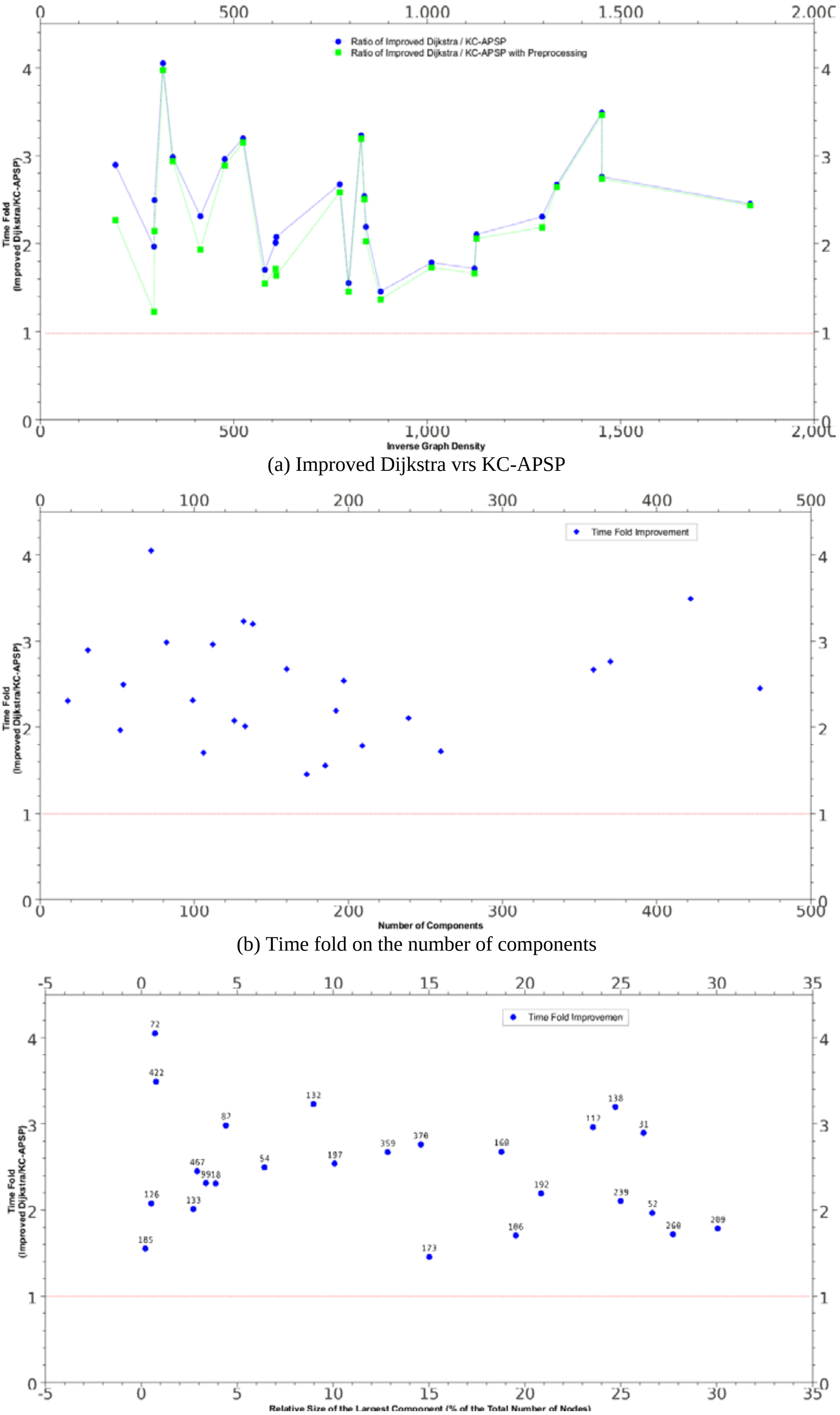


(a) Improved Dijkstra vrs KC-APSP

(b) Time fold on the number of components

(c) Time fold on the relative size of the largest component

Fig. 11 K biconnected random connected graphs with same sized components experimental results.

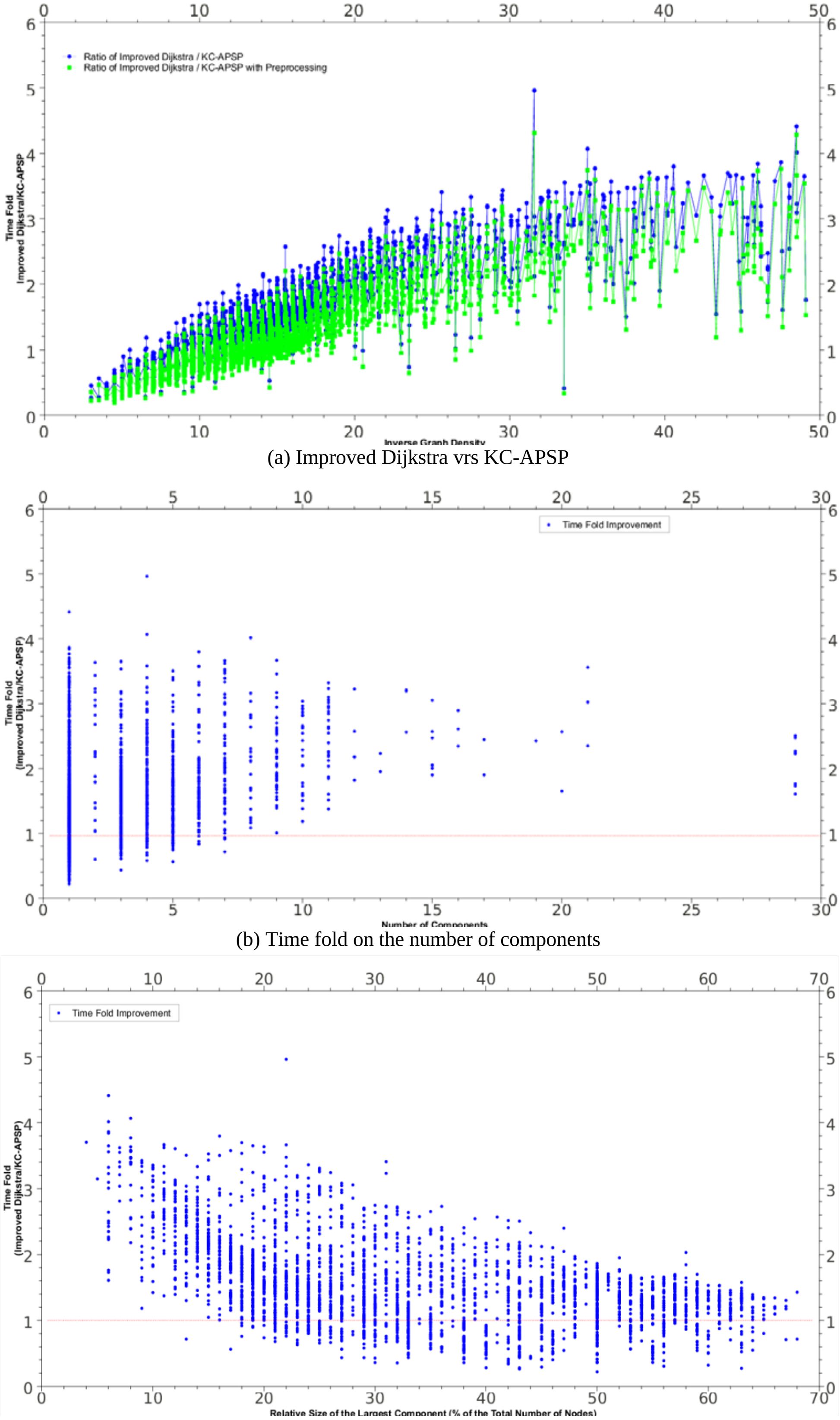


(a) Improved Dijkstra vrs KC-APSP

(b) Time fold on the number of components

(c) Time fold on the relative size of the largest component

Fig. 12 K random biological networks experimental results.

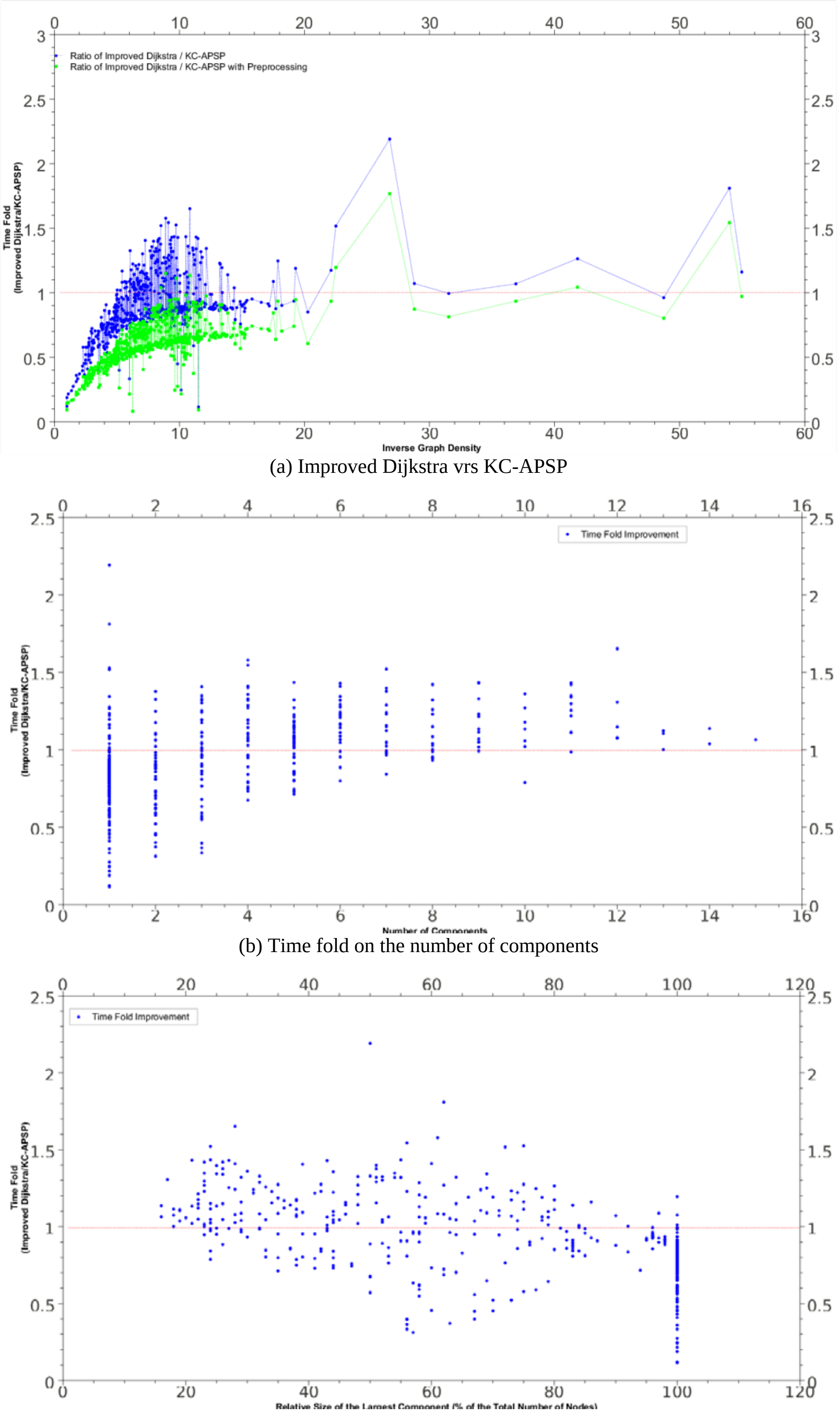


(a) Improved Dijkstra vrs KC-APSP

(b) Time fold on the number of components

(c) Time fold on the relative size of the largest component

Fig. 13 Mutagenicity molecules graphs experimental results.



(a) Improved Dijkstra vrs KC-APSP

(b) Time fold on the number of components

(c) Time fold on the relative size of the largest component

Fig. 14 Enzymes molecules experimental results.

- Mutagenicity Graphs [16]: This is a set of over 3,000 graphs that represent molecules that generate mutations on living organisms (See Fig. 13). On this set of graphs our method also outperforms the conventional method, in many cases by at least two times, as Fig. 13a shows this time fold increases systematically as the graphs have greater inverse graph density, and on (b) can be seen that these molecules graphs topology on terms of components holds little correlation with the actual time fold, hence gaining time actually because of the singles merger phase.
- Enzymes Graphs [17]: This is a collection of over 500 graphs that represent molecules of enzymes (See Fig. 14). The Enzymes Graphs behave more like the General Random Connected Graphs, showing very little or no improvement over the Improved Dijkstra Method, this because of the scarcity of components and single nodes on this set that has on average two components on each graph.

Fig. 5 shows the time proportion that each of the steps of our method takes for each of the types of networks in our experiments. It is clear, as it was stated before, that the Preprocessing step is the one that take most of the time. Notice also that on the Mutagenicity and Enzyme molecules the APSP time computation phase takes a lot of time (31% and 37% of the time respectively), this is due to the scarcity of components, on average the Mutagenicity and Enzymes sample decompose in two biconnected components. Evidently the topology of the graphs has a direct impact on the performance of our method, graphs with only few components (Mutagenicity, Enzymes) or with no components at all (General Random Connected Graphs) would have poor performance, but other graphs like Biological Networks take great advantage of the method, as the Time Fold figures show that most the points have more than one time fold over the Improved Dijkstra algorithm, specifically for graphs with lots of components (K-Biconnected Cliques or K-Biconnected Regular Random Graphs) or graphs that follow power law distribution and have a lot of single nodes and articulation points (Real Biological Networks, Random Biological Networks), where time is saved either by graph compaction or by graph disconnection.

The experiments were performed on a computer running Ubuntu Linux 9.10 with an Intel Core i7 920 Processor running at 2.66 GHz and with 4GB of available memory. The program was fully implemented in Java.

5. Conclusions

There are many areas where graphs can be used to model the problems the scientists in diverse disciplines wants to solve, and most of these areas, like bioinformatics, or other engineering disciplines, deal with graphs that have many single nodes and articulation points. Furthermore the applications of these areas usually compute the articulation points of the network as part of their normal process. As our results show and for all these previous reasons our method brings a definite advantage over the regular implementation of the All-Pairs Shortest Path algorithm, for it saves some more time in the computation performance, time that can be useful for other complicated computations; the time saving is even more astounding when the components sizes are similar or at least not too different. In our particular research in bioinformatics, this method improves the performance of the computation of additional information, information like Closeness Centrality measures that helps us distinguish proteins and genes that have central roles in the networks [18], and we can take full advantage of this method because we need to compute the articulation points as these also represent important nodes on the biological networks. Another advantage that this method clearly presents is that it does not need any complicated setup, as it can be directly implemented on a regular computer, and it has the flexibility that can also be supported by parallel computation given the case is needed.

Acknowledgments

This research is supported by the Taiwan Scholarship Program managed by the Democratic Pacific Union, the National TsingHua University of Taiwan ROC, and the Universidad Tecnológica Centroamericana of Honduras.

References

[1] D.B. Johnson, Efficient algorithms for shortest paths in sparse networks, J. ACM 24 (1) (1977) 1-13.

[2] B.V. Cherkassky, A.V. Goldberg, T. Radzik, Shortest paths algorithms: theory and experimental evaluation, Mathematical Programming 73 (1993) 129-174.

[3] E.W. Dijkstra, A note on two problems in connection with graphs, Numerische Mathematik 1 (1959) 269-271.

[4] R.W. Floyd, Algorithm 97: shortest path, Commun. ACM 5 (6) (1962) 345.

[5] S. Mondal, M. Pal, T. Pal, An optimal algorithm to solve the all-pairs shortest paths problem on permutation graphs, Journal of Mathematical Modelling and Algorithms 2 (2003) 57-65.

[6] U. Bondhugula, A. Devulapalli, J. Fernando, P. Wyckoff, P. Sadayappan, Parallel FPGA-based all-pairs shortest-paths in a directed graph, in: 20th International Parallel and Distributed Processing Symposium, Rhodes Island, Greece, 2006, p. 10.

[7] Y. Tang, Y. Zhang, H. Chen, A parallel shortest path algorithm based on graph-partitioning and iterative correcting, in: Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, Dalian, China, 2008, pp. 155-161.

[8] S. Dey, P. Srimani, Fast parallel algorithm for all-pairs shortest path problem and its VLSI implementation, in: IEEE Proceedings on Computers and Digital Techniques, 1989, Vol. 136, pp. 85-89.

[9] B.H. Junker, F. Schreiber, Analysis of Biological Networks, Wiley Series on Bioinformatics: Computational Techniques and Engineering, Hoboken, John Wiley & Sons, New Jersey, USA, 2008.

[10] B.H. Junker, D. Koschützki, F. Schreiber, Exploration of biological network centralities with CentiBin, BMC Bioinformatics 7 (2006) 219.

[11] S. Brohee, K. Faust, G. Lima-Mendez, O. Sand, R. Janky, G. Vanderstocken, Y. Deville, J. van Helden, Neat: a toolbox for the analysis of biological networks, clusters, classes and pathways, Nucl. Acids Res. 36 (2008) W444-W451.

[12] C.R. Arias, H.-Y. Yeh, V.-W. Soo, Disease Gene Prioritization, Bioinformatics, Rijeka, Croatia, INTECH, 2011.

[13] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd ed., MIT Press, MA, USA, 2001.

[14] A.V. Aho, J.E. Hopcroft, J. D. Ullman, Data Structures and Algorithms, 1st ed., Addison-Wesley, Massachusetts, USA, 1983.

[15] M. Mitzenmacher, E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, New York, USA, 2005.

[16] J. Kazius, R. McGuire, R. Bursi, Derivation and validation of toxicophores for mutagenicity prediction, Journal of Medicinal Chemistry 48 (1) (2005) 312-320.

[17] K. Borgwardt, C. Ong, S. Schönauer, S. Vishwanathan, A. Smola, H.-P. Kriegel, Protein function prediction via graph kernels, Bioinformatics 21 (1) (2005) 47-56.

[18] H.W. Ma, A.P. Zeng, The connectivity structure, giant strong component and centrality of metabolic networks, Bioinformatics 19 (2003) 1423-1430.

Computer Technology and Application 2 (2011) 884-892

Direct Adaptive Fuzzy Based Backstepping Control of Semi Strict Feedback Nonlinear Systems

Raoudha Ben Khaled[1, 4], Chaouki Mnasri[2, 4] and Moncef Gasmi[3, 4]

1. Department of Electrical Engineering, ISI Ariana, Tunisia

2. Department of Electronics Engineering, ISSAT Mateur, Tunisia

3. Department of Physics and Instrumentation, INSAT Tunis, Tunisia

4. Unité de Recherche en Automatique et Informatique Industrielle (URAII), INSAT, Université de Carthage, Tunisia

Received: May 26, 2011 / Accepted: June 13, 2011 / Published: November 25, 2011.

Abstract: In this paper, a direct adaptive fuzzy tracking control is proposed for a class of uncertain single-input single-output nonlinear semi-strict feedback systems. Based on Takagi-Sugeno type fuzzy systems, a direct adaptive fuzzy tracking controller is developed by using the backstepping approach. The main advantage of the developed method is that for an n-th order system, only one parameter is needed to be adjusted online. It is proven that, under the appropriate assumptions, the developed scheme can achieve that the output system converges to a small neighborhood of the reference signal and all the signals in the closed-loop system remain bounded. The efficacy of the proposed algorithm is investigated by an illustrative simulation example of one link robot.

Key words: Direct adaptive fuzzy control, perturbed nonlinear systems, strict feedback form, backstepping design, output tracking.

1. Introduction

Adaptive control of nonlinear systems has been an important area of active research for over five decades now, due to great demands from industrial application. Adaptive backstepping design, a recursive Lyapunov-based scheme, was proposed in the beginning of 1990s [1]. An important advantage of the backstepping approach is that it provides a systematic procedure to design adaptive controller to systems in triangular form, not necessarily, satisfying matching conditions. In the early stage of research on adaptive control of nonlinear systems, nonlinearities are assumed to be linearly parameterized [2-4]. Later, such restrictions have been removed by using fuzzy systems and neural networks [5-7]. In the aforementioned literatures, Fuzzy Logic Systems (FLS) and neural networks are employed to approximate the unknown nonlinearities and the backstepping technique is implemented to construct controllers. Recently, for the direct adaptive control, the FLS are used to directly approximate the desired control input signal [8-9]. In these control methods, the number of adjustable parameter depends on the order of the system. If the order of system is added online or taken offline, the number of adjustable parameter will be increased, and the online computation burden may be very heavy. Whereas, a direct adaptive fuzzy controller based on only one parameter estimation, has been proposed for a class of strict feedback system [10]. The main advantage of this approach is that, the number of adjustable parameter does not rely on the order of the system to be controlled.

The contribution of this paper is to extend the approach presented in Ref. [10] for a more general class of nonlinear systems such as semi-strict feedback system, containing both parametric uncertainties and unknown nonlinearities.

Corresponding author: Raoudha Ben Khaled, Ph.D., research fields: adaptive control, backstepping technique, nonlinear systems. E-mail: ben_khaled_raoudha@yahoo.fr.

The outline of this paper is as follows: The problem formulation and preliminaries are given in section 2; section 3 presents an adaptive fuzzy tracking control scheme for a class of semi strict feedback nonlinear systems by using the backstepping approach; in section 4, an example of one link robot manipulator is given to demonstrate the effectiveness of the proposed method.

2. Problem Formulation and Preliminaries

2.1 Problem Formulation

In this paper, we consider a class of SISO semi-strict feedback nonlinear system, which can be represented in the following form:

$$\begin{cases} \dot{x}_i = f_i(\overline{x_i}) + g_i(\overline{x_i})x_{i+1} + \Delta_i(t,x), \quad i = 1...n\text{-}1 \\ \dot{x}_n = f_n(\overline{x}) + g_n(\overline{x})u + \Delta_n(t,x) \\ y = x_1 \end{cases} \tag{1}$$

where $\overline{x} = [x_1,...,x_n]^T \in R^n$ is the state vector of the system, $u \in R$ and $y \in R$ are the control input and output signal, respectively; $\overline{x}_i = [x_1,...,x_i]^T \in R^i$, $f_i(.)$ and $g_i(.)$ are unknown nonlinear smooth functions, $\Delta_i(t,x), i = 1,...n$ are the disturbance uncertain nonlinearities of the system.

The control objective is to design an adaptive fuzzy controller such that: (i) The closed-loop system must be stable in the sense that all the variables in the closed-loop system must be bounded; (ii) The tracking error converges to a small neighborhood of zero. To design a controller satisfying the above control objectives, the following assumptions are made:

Assumption 1 [10]: The sign of $g_i(\overline{x_i})$ does not change and there exist positive constants b_m and b_M such that for $i = 1,...,n$,

$$0 < b_m \le \left| g_i(\overline{x_i}) \right| \le b_M \tag{2}$$

Remark 1: Assumption 1 implies that the unknown smooth functions $g_i(\overline{x_i}),\ i = 1,2,...,n$ are strictly either negative or positive. Without loss of generality, we assume that $g_i(\overline{x_i}) > 0$. It should be emphasized that the constants b_m and b_M are introduced only for the purpose of analysis and are not used to design controllers. Therefore, their values are not necessarily known.

Assumption 2 [10]: The reference signal $y_d(t)$ and its time derivatives up to the n-th order are continuous and bounded.

Assumption 3 [11]: For $1 \le i \le n$, there exist unknown positive smooth functions $\eta_i(.)$ such that

$$\left| \Delta_i(t,x) \right| \le \eta_i(\overline{x_i}), \forall (t,x) \in R_+ \times R^n, i = 1,...,n \tag{3}$$

2.2 Function Approximation with Fuzzy Logic Systems

It has been proved that FLS can approximate an arbitrarily continuous function to a given accuracy [12]. For instance, we adopt a zero-order Takagi-Sugeno FLS with singleton fuzzification method, product-type inference, and center-average defuzzifier to deduce a collection of IF-THEN rules. The *i*-th fuzzy rule is described by

R_l : *IF* X_1 *is* F_1^l and X_2 *is* F_2^l ..., and X_n *is* F_n^l
THEN y is B^l

with $l = 1,...,N$; N is the number of rules in the fuzzy rule base, $X = [X_1,...,X_n]^T \in R^n$ and $y \in R$ are the inputs and the output of the FLS, respectively; $F_i^l, i = 1,...,n$ and B^l are fuzzy sets in R, characterized by fuzzy membership functions $\mu_{B^l}(X_i)$ and $\mu_{F_i^l}(X_i)$, respectively. The FLS can be represented as

$$y(X) = \frac{\sum_{l=1}^{N} \overline{\phi}_l \prod_{i=1}^{n} \mu_{F_i^l}(X_i)}{\sum_{l=1}^{N} [\prod_{i=1}^{n} \mu_{F_i^l}(X_i)]} \tag{4}$$

where $\overline{\phi}_l$ is the point at which $\mu_{B^l}(\overline{\phi}_l) = 1$. Eq. (4) can be rewritten as

$$y(X) = \phi^T P(X) \tag{5}$$

where $\phi = [\overline{\phi}_1,...,\overline{\phi}_N]^T$ is a vector of adjustable parameters and $P(X) = [P_1(X),...,P_N(X)]^T$, with $P_l(X)$ is a fuzzy basis function described by

$$P_l(X) = \frac{\prod_{i=1}^{n} \mu_{F_i^l}(X_i)}{\sum_{l=1}^{N} [\prod_{i=1}^{n} \mu_{F_i^l}(X_i)]}$$

If all memberships are chosen as Gaussian function, the following lemma holds.

Lemma 1 [12]: Let $f(X)$ be a continuous function defined on a compact set U and τ an arbitrary positive constant. Then, there exist the FLS (5), such that $\sup_{X\in U}|f(X)-y(X)|<\tau$.

Using the above universal approximation theorem, any given continuous function $f(X)$ can be optimally approximated such that

$$f(X)=\phi^T P(X)+\delta(X) \tag{6}$$

where $\delta(X)$ is the approximation error and ϕ is an optimal parameter vector required for analytical purpose.

Assumption 4: There exists the constant $\varepsilon>0$ such that the approximation error is bounded, i.e., $|\delta(X)|\leq\varepsilon$.

3. Main Result

In this paper, the proposed controller contains only one adaptive parameter that needs to be updated online, which is defined as follows:

$$\theta=\max\{\frac{1}{b_m}\|\phi_i\|^2 : i=1,2,..,n\} \tag{7}$$

Theorem 1:

For the class of systems given by (1) and satisfying assumptions 1-3, the following direct adaptive fuzzy control law

$$u=-\frac{1}{2a_n^2}e_n\hat{\theta}P_n^T(X_n)P_n(X_n) \tag{8}$$

with $\hat{\theta}$ is the unique adaptive law, given by

$$\dot{\hat{\theta}}=\sum_{i=1}^{n}\frac{r}{2a_i^2}e_i^2P_i^T(X_i)P_i(X_i)-\sigma_0\hat{\theta} \tag{9}$$

where $\hat{\theta}$ is the estimate of the constant parameter θ; r, σ_0 and a_i, $i=1,...,n$ are positive design parameters; $X_i=[\overline{x}_i^T,\hat{\theta},\overline{y}_d^{(i)T}]^T$ where $\overline{y}_d^{(i)}=[y_d,y_d^{(1)},...,y_d^{(i)}]^T$; e_i is the *i*-th tracking error variable given as $e_i=x_i-\alpha_{i-1}$, such that $\alpha_0=y_d$ and α_i is given as follows:

$$\alpha_i(X_i)=-\frac{1}{2a_i^2}e_i\hat{\theta}P_i^T(X_i)P_i(X_i),1\leq i\leq n-1 \tag{10}$$

guarantees that the tracking error converges to a small neighborhood of zero and all the signals in the closed loop system are bounded.

The proof of theorem 1 consists of two steps. First, a direct adaptive fuzzy control design is presented based on the backstepping approach. Then, the stability analysis of the closed-loop system is carried out.

3.1 Adaptive Fuzzy Controller Design

The backstepping design procedure contains n steps. In each step, a virtual control function $\hat{\alpha}_i$ should be developed using an appropriate Lyapunov function V_i. The backstepping design procedure for system (1) will be given as follows:

Step 1: The first Lyapunov function candidate is chosen as

$$V_1=\frac{1}{2}e_1^2+\frac{b_m}{2r}\tilde{\theta}^2 \tag{11}$$

where $\tilde{\theta}=\theta-\hat{\theta}$. The derivative of V_1 is given as follows:

$$\dot{V}_1=e_1(f_1+g_1x_2-\dot{y}_d+\Delta_1(t,x))-\frac{b_m}{r}\tilde{\theta}\dot{\hat{\theta}} \tag{12}$$

According to assumption 3, and using the completion of squares, the following inequality holds

$$e_1\Delta_1(t,x)\leq|e_1|\eta_1(x_1)\leq\frac{\eta_1^2(x_1)}{2\rho^2}e_1^2+\frac{\rho^2}{2} \tag{13}$$

with ρ is a positive constant. Now, substituting (13) and $\overline{f}_1(X_1)=f_1+\frac{1}{2}\rho^{-2}e_1\eta_1^2+\frac{1}{2}g_1^2e_1-\dot{y}_d$ into (12), we have

$$\dot{V}_1\leq e_1(\overline{f}_1+g_1x_2)+\frac{\rho^2}{2}-\frac{1}{2}g_1^2e_1^2-\frac{b_m}{r}\tilde{\theta}\dot{\hat{\theta}} \tag{14}$$

To stabilize this system, an estimate of the intermediate control signal is selected as $\hat{\alpha}_1(X_1)=-g_1^{-1}\{k_1e_1+\overline{f}_1\}$ with k_1 a positive constant. Further, by adding and subtracting $g_1\hat{\alpha}_1$ in the bracket in (14), $\dot{V}_1$ can be verified as

$$\dot{V}_1\leq -k_1e_1^2+e_1g_1(x_2-\hat{\alpha}_1)+\frac{\rho^2}{2}-\frac{1}{2}g_1^2e_1^2-\frac{b_m}{r}\tilde{\theta}\dot{\hat{\theta}} \tag{15}$$

However, since $\hat{\alpha}_1$ consists of unknown functions f_1, η_1 and g_1, it cannot be implemented in practice.

According to Lemma 1, the FLS (5) can approximate the unknown virtual control $\hat{\alpha}_1$ to a given accuracy, as follows:

$$\begin{aligned}&\hat{\alpha}_1 = \phi_1^T P_1(X_1) + \delta_1(X_1),\\ &\left|\delta_1(X_1)\right| \le \varepsilon_1, \quad \forall \varepsilon_1 > 0\end{aligned} \tag{16}$$

Furthermore, it follows from (6), (16) and the completion of squares that

$$\begin{aligned}-e_1 g_1 \hat{\alpha}_1 &= -e_1 g_1 \phi_1^T P_1 - e_1 g_1 \delta_1\\ &\le \frac{b_m}{2a_1^2} e_1^2 \theta P_1^T P_1 + \frac{b_M}{2} a_1^2 + \frac{1}{2}\varepsilon_1^2 + \frac{1}{2} g_1^2 e_1^2\end{aligned} \tag{17}$$

Remark 2 [11]: From (9), it can be seen that the function $S(t) = \sum_{i=1}^{n} \frac{r}{2a_i^2} e_i^2 P_i^T(X_i) P_i(X_i)$ is nonnegative. This implies that if $\hat{\theta}(t) \le S(t)/\sigma_0$, then $\dot{\hat{\theta}}(t) \ge 0$. Consequently, $\hat{\theta}(t)$ increases until $\hat{\theta}(t) = S(t)/\sigma_0$. So, for a given initial condition $\hat{\theta}(0) \ge 0$, $\hat{\theta}(t) \ge 0$ holds for all $t \ge 0$.

Hence, by use of Remarks 1-2 and (10), the following inequality can be verified easily:

$$e_1 g_1 \alpha_1 \le -\frac{b_m}{2a_1^2} e_1^2 \hat{\theta} P_1^T P_1 \tag{18}$$

Combining (15) with (17) and (18) gives

$$\begin{aligned}\dot{V}_1 \le& -k_1 e_1^2 + \frac{b_M}{2} a_1^2 + \frac{1}{2}\varepsilon_1^2 + e_1 g_1 (x_2 - \alpha_1)\\ &+ \frac{1}{2}\rho^2 + \frac{b_m}{r}\tilde{\theta}\left(\frac{r}{2a_1^2} e_1^2 P_1^T P_1 - \dot{\hat{\theta}}\right)\end{aligned} \tag{19}$$

Step k: $2 \le k \le n-1$

Assume that after step $k-1$, for the Lyapunov function candidate $V_{k-1} = V_{k-2} + \frac{1}{2} e_{k-1}^2$, the following inequality holds

$$\dot{V}_{k-1} \le \psi(k-1) + e_{k-1} g_{k-1} e_k \tag{20}$$

where

$$\begin{aligned}\psi(k) =& -\sum_{i=1}^{k} k_i e_i^2 + \sum_{i=1}^{k} \frac{1}{2}(b_M a_i^2 + \varepsilon_i^2)\\ &+ \sum_{i=1}^{k} \frac{\rho^2}{2}(k-i+1) + \sum_{i=2}^{k} e_i(\varphi_i(X_i)\\ &- \frac{\partial \alpha_{i-1}}{\partial \hat{\theta}}\dot{\hat{\theta}}) + \frac{b_m}{r}\tilde{\theta}\left(\sum_{i=1}^{k} \frac{r}{2a_i^2} e_i^2 P_i^T P_i - \dot{\hat{\theta}}\right)\end{aligned} \tag{21}$$

At the step k, consider the Lyapunov function candidate as

$$V_k = V_{k-1} + \frac{1}{2} e_k^2 \tag{22}$$

Then, the derivative of V_k is given by

$$\dot{V}_k = \dot{V}_{k-1} + e_k (f_k + g_k x_{k+1} - \dot{\alpha}_{k-1} + \Delta_k(x,t)) \tag{23}$$

where

$$\begin{aligned}-e_k \dot{\alpha}_{k-1} =& -e_k \sum_{i=1}^{k-1} \frac{\partial \alpha_{k-1}}{\partial x_i}(f_i + g_i x_{i+1} + \Delta_i)\\ &- e_k \sum_{i=0}^{k-1} \frac{\partial \alpha_{k-1}}{\partial y_d^{(i)}} y_d^{(i+1)} - e_k \frac{\partial \alpha_{k-1}}{\partial \hat{\theta}}\dot{\hat{\theta}}\end{aligned} \tag{24}$$

Using the completion of squares to the term $-e_k \frac{\partial \alpha_{k-1}}{\partial x_i}\Delta_i$, we have

$$-e_k \frac{\partial \alpha_{k-1}}{\partial x_i}\Delta_i \le \frac{1}{2\rho^2}\eta_i^2 e_k^2 \left[\frac{\partial \alpha_{k-1}}{\partial x_i}\right]^2 + \frac{\rho^2}{2} \tag{25}$$

In order to proceed in a similar way as in the first step, we define the function $\bar{f}_k(X_k)$ in such a way as

$$\begin{aligned}\bar{f}_k(X_k) =& f_k + g_{k-1} e_{k-1} - \varphi_k - \sum_{i=0}^{k-1} \frac{\partial \alpha_{k-1}}{\partial y_d^{(i)}} y_d^{(i+1)}\\ &+ \frac{1}{2\rho^2} e_k \sum_{i=1}^{k-1}\left[\frac{\partial \alpha_{k-1}}{\partial x_i}\right]^2 \eta_i^2 - \sum_{i=1}^{k-1} \frac{\partial \alpha_{k-1}}{\partial x_i}(f_i\\ &+ g_i x_{i+1}) + \frac{1}{2}(g_k^2 + \eta_k^2 \rho^{-2}) e_k\end{aligned} \quad 26)$$

It is worth noting that $\dot{\hat{\theta}}$ is a function of all state variables of the system, so that φ_k is introduced in order to compensate the term $\frac{\partial \alpha_{k-1}}{\partial \hat{\theta}}\dot{\hat{\theta}}$, and will be specified later.

Now, combining (20-26) to have

$$\begin{aligned}\dot{V}_k \le& \psi(k-1) + \sum_{i=1}^{k} \frac{\rho^2}{2} - \frac{1}{2} g_k^2 e_k^2\\ &+ e_k\left(\varphi_k - \frac{\partial \alpha_{k-1}}{\partial \hat{\theta}}\dot{\hat{\theta}}\right) + e_k(\bar{f}_k + g_k x_{k+1})\end{aligned} \tag{27}$$

Similarly to step 1, choosing $\hat\alpha_k(X_k) = -g_k^{-1}\{k_k e_k + \overline{f}_k\}$ to obtain the following inequality:

$$\begin{aligned} \dot V_k \le \psi(k-1) - k_k e_k^2 + \sum_{i=1}^{k} \frac{\rho^2}{2} + e_k(\varphi_k - \\ \frac{\partial\alpha_{k-1}}{\partial\hat\theta}\dot{\hat\theta}) - \frac{1}{2} g_k^2 e_k^2 + e_k g_k (x_{k+1} - \hat\alpha_k) \end{aligned} \tag{28}$$

By Lemma 1, the FLS (5) is again utilized to approximate the unknown $\hat\alpha_k$ such that for any given constant $\varepsilon_k > 0$:

$$\begin{aligned} &\hat\alpha_k = \phi_k^T P_k(X_k) + \delta_k(X_k), \\ &|\delta_k(X_k)| \le \varepsilon_k \end{aligned} \tag{29}$$

Using the similar way to (17) and (18) in the first step, we obtain

$$-e_k g_k \hat\alpha_k \le \frac{b_m}{2a_k^2} e_k^2 \theta P_k^T P_k + \frac{1}{2} g_k^2 e_k^2 + \frac{1}{2}(\varepsilon_k^2 + b_M a_k^2) \tag{30}$$

$$e_k g_k \alpha_k \le -\frac{b_m}{2a_k^2} e_k^2 \hat\theta P_k^T P_k \tag{31}$$

Taking (30) and (31) into account, the derivative of V_k as expressed in (28), can be rewritten in such a way that

$$\dot V_k \le \psi(k) + e_k g_k e_{k+1} \tag{32}$$

Step n: Taking the following Lyapunov function candidate:

$$V_n = V_{n-1} + \frac{1}{2} e_n^2 \tag{33}$$

Proceeding as in step *k* from (22) until (28), we can verify that the derivative of V_n satisfy the following inequality:

$$\begin{aligned} \dot V_n \le \psi(n-1) - k_n e_n^2 + \sum_{i=1}^{n} \frac{\rho^2}{2} + e_n(\varphi_n - \\ \frac{\partial\alpha_{n-1}}{\partial\hat\theta}\dot{\hat\theta}) - \frac{1}{2} g_n^2 e_n^2 + e_n g_n (x_{n+1} - \hat u) \end{aligned} \tag{34}$$

where $\hat u(X_n) = -g_n^{-1}\{k_n e_n + \overline{f}_n\}$, $\psi(n-1)$ and $\overline{f}_n$ are as expressed respectively, in (21) and (26), for $k = n$.

Similarly, for any given positive constant ε_n , the FLS (5) is utilized to approximate the unknown control $\hat u$.

$$\begin{aligned} &\hat u = \phi_n^T P_n(X_n) + \delta_n(X_n), \\ &|\delta_n(X_n)| \le \varepsilon_n \end{aligned} \tag{35}$$

Now, following a similar line as used in the procedure from (16) to (18) yields:

$$-e_n g_n \hat u \le \frac{b_m}{2a_n^2} e_n^2 \theta P_n^T P_n + \frac{1}{2} g_n^2 e_n^2 + \frac{1}{2}(\varepsilon_n^2 + b_M a_n^2) \tag{36}$$

$$e_n g_n u \le -\frac{b_m}{2a_n^2} e_n^2 \hat\theta P_n^T P_n \tag{37}$$

Hence, combining (34) with (36) and (37) gives

$$\dot V_n \le \psi(n) \tag{38}$$

with $\psi(n)$ is defined as in (21) for $k = n$.

3.2 Analysis of Stability

For the stability analysis of the closed loop system, take the Lyapunov function candidate as $V = V_n$. Then, from (38) and (9), the derivative of V satisfy the following inequality:

$$\begin{aligned} \dot V \le -\sum_{i=1}^{n} k_i e_i^2 + \sum_{i=1}^{n} \frac{1}{2}(b_M a_i^2 + \varepsilon_i^2) + \sum_{i=1}^{n} \frac{\rho^2}{2}(n-i+1) \\ + \frac{b_m}{r}\sigma_0 \tilde\theta\hat\theta + \sum_{i=2}^{n} e_i(\varphi_i - \frac{\partial\alpha_{i-1}}{\partial\hat\theta}\dot{\hat\theta}) \end{aligned} \tag{39}$$

To handle the term $\frac{b_m}{r}\sigma_0\tilde\theta\hat\theta$, we use the completion of squares and the fact that $\hat\theta = \theta - \tilde\theta$, to have

$$\frac{b_m}{r}\sigma_0\tilde\theta\hat\theta \le -\frac{b_m}{2r}\sigma_0\tilde\theta^2 + \frac{b_m}{2r}\sigma_0\theta^2 \tag{40}$$

At the present stage, φ_i will be determined so that

$$\sum_{i=2}^{n} e_i(\varphi_i - \frac{\partial\alpha_{i-1}}{\partial\hat\theta}\dot{\hat\theta}) \le 0 \tag{41}$$

Replacing $\dot{\hat\theta}$ by its expression in (9), the following holds:

$$-\sum_{i=2}^{n} e_i \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\dot{\hat{\theta}} = -\sum_{i=2}^{n} e_i \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\sum_{j=1}^{n}\frac{r}{2a_j^2}e_j^2P_j^TP_j + \sum_{i=2}^{n} e_i \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\sigma_0\hat{\theta} \qquad (42)$$

The first term on the right hand side of (42) can be rewritten as

$$-\sum_{i=2}^{n} e_i \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\sum_{j=1}^{n}\frac{r}{2a_j^2}e_j^2P_j^TP_j = -\sum_{i=2}^{n} e_i \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\sum_{j=1}^{i-1}\frac{r}{2a_j^2}e_j^2P_j^TP_j - \sum_{i=2}^{n} e_i \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\sum_{j=i}^{n}\frac{r}{2a_j^2}e_j^2P_j^TP_j \qquad (43)$$

Noting that $\left\|P_j(X_j)\right\| \le 1$, the last term in (43) is expressed as

$$-\sum_{i=2}^{n} e_i \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\sum_{j=i}^{n}\frac{r}{2a_j^2}e_j^2P_j^T(X_j)P_j(X_j) \le \sum_{i=2}^{n}\left|\frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}e_i\right|\left(\sum_{j=i}^{n}\frac{r}{2a_j^2}e_j^2\right) = \sum_{i=2}^{n}\frac{r}{2a_i^2}e_i^2\left(\sum_{j=2}^{i}\left|\frac{\partial\alpha_{j-1}}{\partial\hat{\theta}}e_j\right|\right) \qquad (44)$$

Substituting (43) and (44) into (42) yields:

$$-\sum_{i=2}^{n} e_i \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\dot{\hat{\theta}} = \sum_{i=2}^{n} e_i\Big[-\sum_{j=1}^{i-1}\frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\frac{r}{2a_j^2}e_j^2P_j^TP_j + \frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\sigma_0\hat{\theta} + \frac{r}{2a_i^2}e_i\Big(\sum_{j=2}^{i}\left|\frac{\partial\alpha_{j-1}}{\partial\hat{\theta}}e_j\right|\Big)\Big]$$

At the present stage, we choose φ_i such that:

$$\varphi_i = -\sigma_0\hat{\theta}\frac{\partial\alpha_{i-1}}{\partial\hat{\theta}} - \sum_{j=2}^{i} e_i\frac{r}{2a_i^2}\left|e_j\frac{\partial\alpha_{j-1}}{\partial\hat{\theta}}\right| + \sum_{j=1}^{i-1}\frac{\partial\alpha_{i-1}}{\partial\hat{\theta}}\frac{r}{2a_j^2}e_j^2P_j^TP_j$$

in order to verify the inequality (41). Now, taking into account (40) and (41), the derivative of V can be expressed as

$$\dot{V} \le -\sum_{i=1}^{n}k_ie_i^2 + \sum_{i=1}^{n}\frac{1}{2}(b_Ma_i^2+\varepsilon_i^2) + \sum_{i=1}^{n}\frac{\rho^2}{2}(n-i+1) + \frac{b_m}{2r}\sigma_0\theta^2 \qquad (45)$$

Furthermore, let $a_0 = \min\{2k_i, \sigma_0, i = 1, 2, ..., n\}$ and $b_0 = \frac{b_m}{2r}\sigma_0\theta^2 + \sum_{i=1}^{n}\frac{1}{2}(b_Ma_i^2+\varepsilon_i^2) + \sum_{i=1}^{n}\frac{\rho^2}{2}(n-i+1).$

From (45) and (33) we have

$$\dot{V}(t) \le -a_0V(t) + b_0 \qquad (46)$$

Now we can verify that

$$\frac{d}{dt}(V(t)e^{a_0t}) \le b_0e^{a_0t} \qquad (47)$$

Integrating inequality (47) over $[0,\ t]$, we have

$$V(t) \le (V(0) - \frac{b_0}{a_0})e^{-a_0t} + \frac{b_0}{a_0}, \qquad t \ge 0 \qquad (48)$$

which implies that $e = [e_1, ..., e_n, \tilde{\theta}]^T$ belongs to the compact set $\Omega = \{e / V(e(t)) \le V(0) + \frac{b_0}{a_0}\}$. Therefore, $e_i (i = 1, 2, ..., n)$ and $\tilde{\theta}$ are bounded. Since θ is constant, thus $\hat{\theta}$ is bounded. Consequently, from (10), α_i is also bounded because $P_i^TP_i \le 1$. Hence, we conclude that all the state variables of the system x_i are bounded. In addition from (48) we have

$$e_1^2 \le 2(V(0) - \frac{b_0}{a_0})e^{-a_0t} + 2\frac{b_0}{a_0} \qquad (49)$$

which implies that $\lim_{t\to\infty} e_1^2 \le 2b_0 / a_0$.

Remark 3: The above analysis shows that the tracking error e_1 depends on both a_0 and b_0. Because θ^*, b_m and b_M are unknown; an explicit estimation of the tracking error is impossible. It is worth noting that reducing a_i and σ_0, meanwhile increasing r will diminish the tracking error. It is clear that the parameters k_i and ε_i are not used in the controller design, they are introduced just for stability analysis.

4. Simulation Study

In this section, an example will be used to test the effectiveness of the proposed controller. Consider the one link robot with the inclusion of motor dynamics proposed in Refs. [8, 13],

$$\begin{aligned} D\ddot{q} + B\dot{q} + N\sin(q) &= \tau + \tau_d \\ M\dot{\tau} + H\tau &= u - K_m\dot{q} \end{aligned} \tag{50}$$

where $q, \dot{q}, \ddot{q}$ denote the link position, velocity and acceleration, respectively; τ and $\dot{\tau}$ are the motor shaft angle and velocity; τ_d represents the torque disturbance and u is the control input used to represent the motor torque. By setting $x_1 = q, x_2 = \dot{q}, x_3 = \tau$, the system (50) can be rewritten in the following semi strict feedback form:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{B}{D}x_2 - \dfrac{N}{D}\sin(x_1) + \dfrac{1}{D}x_3 + \dfrac{\tau_d}{D} \\ \dot{x}_3 = -\dfrac{H}{M}x_3 - \dfrac{K_m}{M}x_2 + \dfrac{u}{M} \\ y = x_1 \end{cases}$$

The parameter values with appropriate units are $D = 1, M = 0.05, B = 1, K_m = 10, H = 0.5, N = 10.$ The torque disturbance τ_d and the reference signal y_d are chosen, respectively, as: $\tau_d = x_2\sin(x_3)$ and $y_d = 0.5(\sin(t) + \sin(0.5t)).$ For the fuzzy control, seven fuzzy sets are characterized by the following membership functions:

$$\mu_{F_i^1} = \exp[\frac{-(x+1.5)^2}{2}], \mu_{F_i^2} = \exp[\frac{-(x+1)^2}{2}],$$

$$\mu_{F_i^3} = \exp[\frac{-(x+0.5)^2}{2}], \mu_{F_i^4} = \exp[\frac{-(x+0)^2}{2}],$$

$$\mu_{F_i^5} = \exp[\frac{-(x-0.5)^2}{2}], \mu_{F_i^6} = \exp[\frac{-(x-1)^2}{2}],$$

$$\mu_{F_i^7} = \exp[\frac{-(x-1.5)^2}{2}]$$

Noting that the first subsystem is a linear differential equation, the adaptive fuzzy controller and adaptive law can be constructed as follows:

$$\alpha_1 = -k(x_1 - y_d) + \dot{y}_d$$

$$\alpha_2 = -\frac{1}{2a_2^2}(x_2 - \alpha_1)\hat{\theta}P_2^T(X_2)P_2(X_2)$$

$$u = -\frac{1}{2a_3^2}(x_3 - \alpha_2)\hat{\theta}P_3^T(X_3)P_3(X_3)$$

$$\dot{\hat{\theta}} = \sum_{i=2}^{3}\frac{r}{2a_i^2}e_i^2P_i^T(X_i)P_i(X_i) - \sigma_0\hat{\theta}$$

where the design parameters are taken as $k = 6, a_2 = a_3 = 0.35, r = 15, \sigma_0 = 0.05.$ The simulation is run under the initial conditions $x(0) = [0, 0, 0]^T$ and $\hat{\theta}(0) = 0$.

Figs. 1-5 display the simulation results, it can be seen that this system with only one adaptation law achieves the required performances. The closed-loop system is stable in the sense that all the variables in the closed-loop system are bounded. Moreover, the tracking error $e_1(t)$ converges to a small neighborhood of zero.

5. Conclusions

In this paper, the problem of output tracking control has been considered for a class of uncertain nonlinear system in semi-strict-feedback form. A direct adaptive fuzzy tracking control scheme has been proposed by means of the backstepping approach. The proposed controller ensures that all the signals of the resulting closed-loop system are bounded and the tracking error

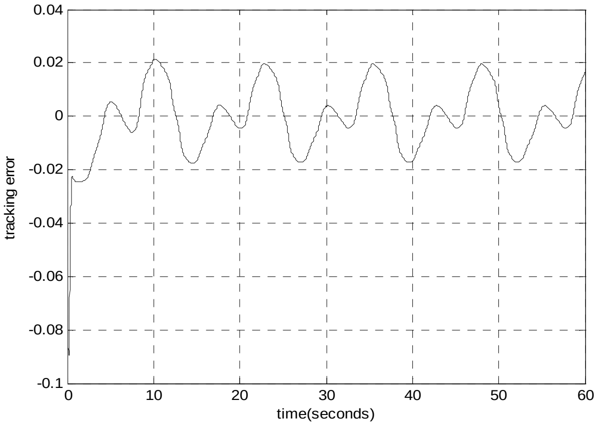


Fig. 1 Tracking error *e₁(t)*.

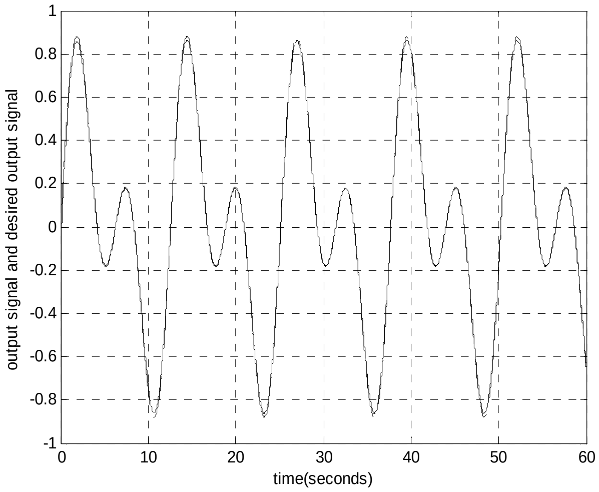


Fig. 2 Output signal y (solid line) and desired output signal y_d (dashed line).

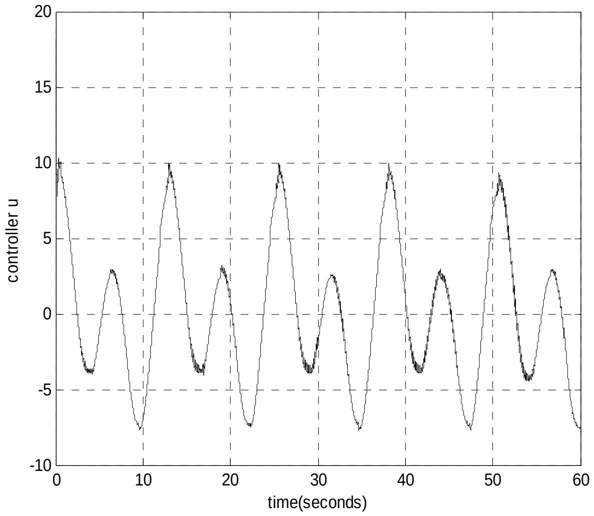


Fig. 3 Control signal input u.

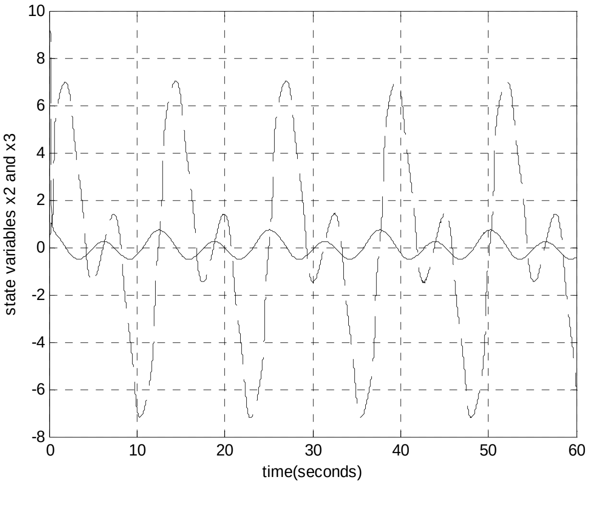


Fig. 4 State variables x_2 (solid line) and x_3 (dashed line).

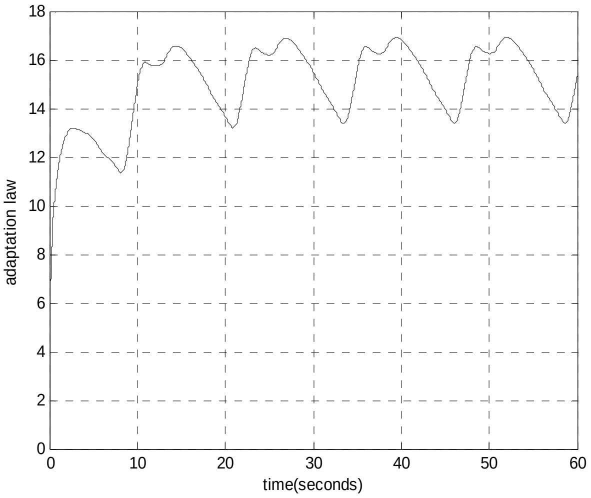


Fig. 5 Adaptation parameter $\hat{\theta}$.

converges to a small neighborhood of the origin. In addition, only one parameter is needed to be adjusted. The efficacy and the validity of the proposed approach are illustrated through an example of one link robot manipulator.

References

[1] M. Krstic, I. Kanellakopoulos, P.V. Kokotovic, Nonlinear and Adaptive Control Design, John Wiley and Sons, New York, 1995.

[2] R.B. Khaled, C. Mnasri, M. Gasmi, A combined adaptive backstepping-sliding mode control of a class of uncertain nonlinear systems, IREACO 3 (2010) 443-451.

[3] R.B. Khaled, C. Mnasri, M. Gasmi, Adaptive backstepping control for uncertain strict-feedback nonlinear systems, in: JTEA'2010, Tunisie.

[4] M. Krstic, I. Kanellakopoulos, P.V. Kokotovic, Adaptive nonlinear control without over parametrization, Systems and Control Letters 19 (1992) 177-185.

[5] S. Jagannathan, F.L. Lewis, Robust backstepping control of a class of nonlinear systems using fuzzy logic, Information Sciences 123 (2000) 223-240.

[6] Y.G. Leu, J.Y. Lin, Adaptive backstepping fuzzy control for a class of nonlinear systems, Lecture Notes in Computer Science 5552 (2009) 1123-1129.

[7] S. Tong, Y. Li, Observer-based fuzzy adaptive control for strict-feedback nonlinear systems, Fuzzy Sets and Systems 160 (2009) 1749-1764.

[8] B. Chen, X. Liu, K. Liu, P. Shi, C. Lin, Direct adaptive fuzzy control for nonlinear systems with time-varying delays, Information Sciences 180 (2010) 776-792.

[9] M. Wang, B. Chen, S.L. Dai, Direct adaptive fuzzy tracking control for a class of perturbed strict-feedback

nonlinear systems, Fuzzy Sets and Systems 158 (2007) 2655-2670.

[10] B. Chen, X. Liu, K. Liu, C. Lin, Direct adaptive fuzzy control of nonlinear strict-feedback systems, Automatica 45 (2009) 1530-1535.

[11] M. Wang, B. Chen, X. Liu, P. Shi, Adaptive fuzzy tracking control for a class of perturbed strict-feedback nonlinear time-delay systems, Fuzzy Sets and Systems 159 (2008) 949-967.

[12] L.X. Wang, J.M. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, IEEE Transactions on Neural Networks 3 (1992) 807-814.

[13] Y. Yang, G. Feng, A combined backstepping and small-gain approach to robust adaptive fuzzy control for strict-feedback nonlinear systems, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Human 34 (2004) 674-692.

Computer Technology and Application 2 (2011) 893-905

The NP-Complete Face of Information-Theoretic Security

Stefan Rass and Peter Schartner

System Security Research Group, Institute of Informatics, Alpen-Adria Universität Klagenfurt, Klagenfurt 9020, Austria

Received: September 20, 2011 / Accepted: October 11, 2011 / Published: November 25, 2011.

Abstract: The problem of perfectly secure communication has enjoyed considerable theoretical treatment over the last decades. Results in this area include the identification of multipath transmission as a necessary ingredient, as well as quantum key distribution (QKD), which can perfectly protect direct lines. Combining the advantages of the quantum and multipath transmission paradigm, as well as rigorously analyzing the security of such combined techniques, is possible by virtue of game-theory. Based on a game-theoretic measure of channel vulnerability, the authors prove the problem of setting up infrastructures for QKD-based multipath transmission to be NP-complete. The authors consider the problem in two flavors, both being computationally hard. Remarkably, the authors' results indicate that the P-vs-NP-question is only of minor effect for confidentiality, because either nowadays public-key cryptosystems remain secure (in case that $\mathrm{P} \neq \mathrm{NP}$) or infrastructures facilitating perfectly confidential communication can be constructed efficiently (in case that $\mathrm{P} = \mathrm{NP}$).

Key words: Risk management, cryptography, complexity theory, NP-completeness, network security.

1. Introduction

In his seminal work, Shannon [1] gave rigorous necessary arguments for perfectly secure communication, and proved the one-time pad as an encryption that is theoretically unbreakable. Recent progress in the field of quantum key distribution [2] demonstrated the concept to be technically feasible on point-to-point lines. However, it has been shown [3-5] that perfectly secure end-to-end communication calls for multipath transmission regimes and is impossible over a single line if no pre-shares secrets exist. So, if one wants to escape the risks induced by computational infeasibility assumptions, multipath transmission is the way to go. However, nowadays infrastructures, with overwhelming probability will not meet the demand of multiple node-disjoint paths between any two nodes in the network. Extending the network to meet these needs is closely related to the graph augmentation problem which is known to be NP-complete (among the hardest nondeterministic-polynomially (NP) solvable problems) in certain cases [6]. Nevertheless, pure graph augmentation is most likely overkill, if we have to create a maximum security environment at minimal costs. In this work, we formalize the problem of finding network topologies that are optimal in terms of security or costs, with desired lower or upper bounds on the other, respectively. In particular, we recall two optimization problems, both of which are known to be NP-hard:

- Find a network of minimum vulnerability that is still feasible from the cost point of view, and
- Find a minimum cost extension to a network that provides some minimum required attack resilience.

The yet abstract concept of vulnerability has rigorously been formalized, and will be briefly introduced in the next section.

Our contribution: The technical contribution of this article is proving the problem of optimized security to be NP-complete in certain cases. Hence, we come to the remarkable conclusion that the P-vs-NP-question hasrather little impact on the problem of secure

Peter Schartner, Dr., research fields: system security, key-management and security token.

Corresponding author: Stefan Rass, Dr., research fields: system security, key-management and information-theoretic security. E-mail: stefan.rass@uni-klu.ac.at.

communication, because creating an infrastructure that facilitates effective information-theoretic security is by itself NP-complete (notice the irony!). So as long as building such infrastructures is infeasible, we are well protected by nowadays public-key infrastructures. Once the technology permits solving NP-complete problems efficiently, we can set out to construct communication networks with information-theoretic security at feasible effort.

Our work in particular offers an answer to the problem that Goldreich and Goldwasser discussed in Ref. [7]. In this reference, the authors provide a very interesting treatment, yet conclude that the question remains open. Until now, we are unaware of any affirmative solution along these lines of research. Our main contribution in this work is thus providing one possible answer.

In the remainder of this work, we shallformalize the following questions, along with a complexity-theoretic classification for each of them:

- What is the minimum vulnerability (in terms of confidentiality) that can be achieved in a given network under given budget-constraints?
- What is the cheapest way of achieving a given vulnerability threshold (i.e., a desired attack resilience)?

Once formalized properly (subject of section 2), both problems are known to be NP-hard [8]. The contribution of this work is to prove them NP-complete, or even harder than this in some special cases of the second question. We tackle both problems separately in section 3 and section 4. The concluding big picture is drawn in section 5, with conclusions following in section 6.

Related work: Finding optimal behavior in a given security infrastructure is often tied to a thorough analysis of vulnerability. Topological vulnerability analysis [9-10] using network scanners like Nessus (http://www.nessus.org/) in connection with decision support tools like Cauldron [11] greatly facilitate the decision process when it comes to security enhancements. Our goal is to investigate the computational efforts tied to such a process, assuming that it relies on the above mentioned approaches.

A vast amount of work, related to optimized security configurations, exists in the literature. The authors of Ref. [12] describe a (heuristic) game-theoretic network vulnerability assessment. Specifically for quantum networks, the problem of optimized topologies has been tackled in Ref. [13]. Our analysis is done on a more general basis, since it is in no way restricted to quantum networks (as we do not exploit any specific features of the quantum paradigm), nor is it restricted to the treatment of confidentiality (see Ref. [14] for an application of the framework to secure authentication).

The authors of Refs. [15-16] present ideas on how to optimize security policies using (bi-) matrix games as communication model. We shall take a conceptually close route, however, with the different target of infrastructure design.

The authors of Ref. [17] as well as references therein describe a large number of applications of game theory in communication theory, but mention only few specific applications to system security. Among them is the work of Ref. [18], describing an intrusion-detection system based on game-theory. Regarding ad hoc networks, the authors of Refs. [19-20] describe, how to find optimal strategies, to defend against malicious nodes in ad hoc networks. This is achieved either through packet forwarding games or direct cooperation among the nodes to identify a malicious intruder. The author of Ref. [21] models the fight for getting secret information as a game, but as well assumes detailed knowledge about the adversary. The authors of Ref. [22] consider stochastic games for network intrusion. Whilst all of this is useful in topological vulnerability analysis, it does not directly facilitate network construction from a cost point of view.

Finally, it is interesting to notice the work of Ref. [23], because these authors too investigate the protection of information flow and arrive at a measure that they call quality of protection. An application to optimized network design has not been demonstrated yet.

2. The Concept of Vulnerability

Recalling the known results about multipath transmission (cited above), the existence of a certain number of node-disjoint paths has been recognized as a necessary (and occasionally also sufficient) condition for perfectly concealed information exchange [5]. Information-theoretic security calls for the probability of a plain text remaining unchanged by conditioning on the ciphertext (Theorem 6 in Ref. [1]). A related notion on middle ground, given in Ref. [5] and prior to this used in Ref. [24], classifies a communication as ε-*private*, if two transcripts arising from different plaintexts M_1, M_2, where the same coin-flips are made for encryption, have probability distributions differing at most by ε in the 1-norm. Similarly, a transmission is δ-*reliable*, if the probability of failure is upper-bounded by δ. The rough intuition behind this is that information acquired by eavesdropping will be ε-close to basically any possible plaintext. Obviously, setting $\delta = \varepsilon = 0$ is equivalent to demanding perfect reliability and information-theoretic security. The latter fact is deduced from the distribution of the transcript, emerging independently of the plain text [24].

If Alice and Bob enjoy some degrees of freedom in choosing the transmission paths for multipath communication, then the problem of path choice can be made into a game, with the adversary's strategies being the subsets that can be conquered. This perfectly matches the standard model of adversary structures used in generalized secret sharing [25]. In the light of nowadays attack strategies on the Internet, an adversary structure is a natural model capturing the possibility of certain subsets of network nodes being compromised. The adversary structure is the collection of all these sets. As an example, assume that every node in a network is equipped with a selection of software from some pool. Different security bug exploits may yield to different sets of machines that can become compromised. Similarly, virus scanners or virus definitions of some machines may be outdated, turning these into candidates for viral infection. Any such scenario can yield a different set of machines being compromised and a further entry in the adversary structure.

Summarizing this, the degrees of freedom for multipath transmission, along with the adversary structure, permit setting up a classical matrix-game in the following manner: let i range over n possibilities of multipath transmission (i.e., selections of non-intersecting paths), and let j be the index of an attack-scenario, assuming m such scenarios exist. A pure strategy for either player in this game is any chosen transmission regime, or any attack scenario. More specifically for the multipath transmission setting, a pure strategy for Alice and Bob is a selection of node-disjoint paths, and a pure strategy for the adversary is a set of chosen nodes to conquer.

In the game-matrix A, we set $a_{ij} = 1$ for a successful transmission in the network, and note $a_{ij} = 0$, if the attack has been successful. A Nash-equilibrium of the game would be a simultaneously optimal probability distribution for the honest parties as well as the adversary to draw their actions from. In other words, if p^*, q^* is a Nash-equilibrium of the game induced by the matrix A, then Alice and Bob should choose the i-th transmission strategy with probability p_i^* in order to maximize the chance of secret communication, whereas the adversary should use strategy j with probability q_j^*, in order to maximize his chance of disclosing the secret content. Notice that this zero-sum game assumption may be totally inaccurate in reflecting the adversary's true intentions and behavior, but it can be shown to be a valid worst-case scenario from Alice and Bob's point of view [26-27].

The average outcome of the game, i.e., the mean utility gained through infinitely many transmissions, is the probability of success. Its difference to the optimum 1 has been assigned the name *vulnerability*, as it measures how much can be achieved on average, compared to how much would be possible in absence of the adversary. Formally, let $A \in T^{n\times m}$ denote the matrix of a game set up as above, where any compact set $T \subset \mathbb{R}^+$ (a taxonomy) is permissible (in the above paragraphs, we used $T = \{0,1\}$). Furthermore, let $v(A)$ denote the saddle-point value of the function $(x,y) \mapsto x^T Ay$, where x, y are discrete probability distributions over the sets $\{0,1,\dots,n\}$ and $\{0,1,\dots,m\}$, respectively. The quantity $v(A)$ is known as the value of the zero-sum game, and is interpreted as the average outcome through infinitely many repetitions of the game. When a Nash-equilibrium strategy is played, then the *vulnerability* of the overall system is defined as

$$\rho(A) = \max T - v(A)$$

The vulnerability enjoys useful theoretical properties. If the game is set up as above, then

- $\rho(A)$ is the maximum probability of a secret message becoming disclosed (Theorem 5.3.19 in Ref. [28]).
- If C denotes the transcript of a secret transmission of the plain text M, then the mutual information $I(M;C)$ between M and C, is bounded as $I(M;C) \leq \rho(A) \cdot H(M)$, where $H(M)$ denotes the entropy of the plain text's source (Corollary 5.3.20 in Ref. [28]).
- When A is referring to success or failure of a transmission, the system is ρ-reliable, and accordingly 2ρ-private, if the game is about secrecy (Theorem 5.3.21 in Ref. [28]).

For other choices of T, the vulnerability matches the usual definition of decision-theoretic risk as expected loss (also known as regret function in a statistical context). However, most interesting for us is the following vulnerability-based characterization of the possibility of arbitrarily secure communication. This is Theorem 1 ([28]): Let Alice and Bob set up their game matrix with binary entries $\mathrm{a_{ij}} \in \{0,1\}$, where $\mathrm{a_{ij}} = 1$ if and only if a message can securely be delivered by choosing the i-th pure strategy, and the adversary uses his j-th pure strategy for attacking. Then $\rho(\mathrm{A}) \in [0,1]$, and

- For any $\varepsilon > 0$, if $\rho(\mathrm{A}) < 1$, then a protocol exists so that Alice and Bob can communicate with an eavesdropping probability of at most ε;
- If $\rho(\mathrm{A}) = 1$, then the probability of the message becoming extracted by the adversary is 1.

Theorem 1 justifies restricting attention to games set up over the binary set $T = \{0,1\}$, whereas our results do not entirely hinge on this.

It is computationally efficient and in fact widely automatable, to set up the game and determine its value: We start by modeling the physical network topology as an undirected graph. Multiple paths and according degrees of freedom for Alice and Bob can be determined by using an integral min-cut-max-flow approach. Calculating the game's value amounts to solving a linear program, which in turn is doable in polynomial time (polynomial in the size of the network graph). Listing the (pure) attack strategies for the adversary is straightforward by enumerating all fixed cardinality subsets in the network. If the adversary's threshold is fixed, this is accomplishable in polynomial time. However, even this effort can further be cut down, if an adversary structure based on common vulnerabilities in the network nodes is employed (as discussed at the beginning of this section). Topological vulnerability analysis [10] deals with exactly this problem.

A natural further step is required if the network turns out vulnerable to a non-acceptable extent. If the vulnerability is unacceptably high, then suitable extensions are required that increase attack resilience. Speaking in terms of graph topology, we seek a network layout that is maximal resilient against attacks. For the sake of simplicity, we shall confine ourselves to adding further edges to increase the network connectivity, whilst leaving the set of nodes untouched. In fact, the problem remains equally complex if we add nodes and edges, or only nodes. Both variants can be cast into equivalent problem instances where only edges are added.

3. Minimum Vulnerability Is NP-Complete

Suppose that we are given a graph $G = G(V,E)$ with each direct link capable of unconditionally secure message delivery (we will use the notation $G(V,E)$ to explicitly emphasize the pair (V, E) constituting a graph). Any available quantum key distribution facility is a candidate for implementing such a link. Furthermore, assume that a list E' of candidate extension links exists, which comes with several implied impact factors, such as construction costs, running costs (operating staff and maintenance), and others. Each of these may itself be subject to constraints, such as environmental requirements for the link to work, or other conditions which may prevent one technology from working, while another may perform perfectly. Reliability affects maintenance costs, and is as such to be considered when setting up the cost overview for possible enhancements of a network topology. We can model these costs as a (possibly vector-valued) function $c_l: E' \to \mathbb{R}^d$, with $d \geq 1$, which gives the costs associated with a single link in E'. For a set of links, suppose that we have an extended version of c_l, defined as $c: 2^{E'} \to \mathbb{R}^d$, which returns the total costs for a given set of links. Simply assuming the costs to be additive (i.e., setting $c(E) := \sum_{e \in E} c_l(e)$) might be misleading, since maintenance staff may be on duty for several links, or infrastructure (such as cable tunnels) may identically be re-used for a set of links, rather than only for a single link (we remark that the additive cost functional has been used in Ref. [8] to establish Lemma 7 below).

Let $K \in \mathbb{R}^d$ be the constraint vector on all these costs, i.e., assume that the only feasible extensions $\tilde{E} \subseteq E'$ to our graph are such that $c(\tilde{E}) \leq K$ component-wise. The set U denotes the set of instances that are able to actively communicate (access-points to the back-bone for instance). We seek to maximize security under this setting. The goal function of choice is therefore

$$R(U, G(V, E)) := \max_{\substack{s,t \in U,\\ s \neq t}} \rho(A(s,t)) \qquad (1)$$

giving the maximum vulnerability experienced by each pair $s, t \in U$. Notice that the game-matrix is specific for each pair in this case, indicated by the explicit dependence on the pair s, t. The dependency of $A(s,t)$ on the network topology is not explicitly stated any further, but should be kept in mind in all that follows.

From the goal function (1), we construct the following optimization problem for finding a maximum security extension at affordable cost:

$$\begin{array}{lrcl} & R\left(U, G\left(V, E \cup \tilde{E}\right)\right) & \to & \min_{\tilde{E} \subseteq E'} \\ \text{s.t.} & c\left(\tilde{E}\right) & \leq & K \end{array} \qquad (2)$$

and $G(V, E \cup \tilde{E})$ is the graph $G(V, E)$ being extended by the additional links in $\tilde{E}$. Notice that the above cost functional can also be set up to account for additional nodes.

It has been shown that problem (2) is NP-hard [29], and we ought to prove its membership in NP to conclude NP-completeness. We work our way through a series of lemmata, sketching the idea first: it is widely known that, if the goal function maps into a finite set of no more than exponential cardinality, then a bisective search through all values of the goal function permits reducing the optimization problem to the decision problem. The converse reduction is trivial. We shall take the same route here, proving that no two instantiations of the game matrix A, can yield vulnerability values closer than some exponentially small threshold. In other words, we explicitly determine a lower bound on the improvement achievable through any extension, which will be the stopping criterion for the bisective search. If, by a bisective search, the bound can be reached within a polynomial number of steps, and the decision problem is in NP, then we are done. The bound is obtained in Lemma 3, and Lemma 4 asserts the decision problem, corresponding to (2), to be a member of NP.

Let us get started with some number-theoretic groundwork: *Farey-sequences* are (roughly spoken) the set of all fractions $\frac{p}{q} \in [0,1]$ whose denominator q obeys a prescribed bound $q \leq m$. This value m is called the *order* of the Farey-sequence.

Lemma 1: ([30], Thm. 28) Let $\frac{\mathrm{h}}{\mathrm{k}}$ and $\frac{\mathrm{h}'}{\mathrm{k}'}$ denote two consecutive elements of a Farey-sequence of order m.

Then

$$kh' - hk' = 1 \tag{3}$$

Dividing Eq. (3) by kk' and using $kk' \leq m^2$, we obtain

Lemma 2: Any two elements $\frac{\mathrm{h}}{\mathrm{k}}$ and $\frac{\mathrm{h}'}{\mathrm{k}'}$ in a Farey-sequence of order m satisfy

$$\left|\frac{h}{k} - \frac{h'}{k'}\right| \geq \frac{1}{m^2}.$$

In the following, let $poly(n) := \bigcup_{k\in\mathbb{N}} \mathcal{O}(n^k)$ denote the set of all functions being bounded by some univariate polynomial. Hence, the notation $f(n) \in poly(n)$ indicates a function of no more than polynomial growth in n. From the theory of linear optimization, we require

Theorem 1: ([31], Thm.3.4]) A linear function $\mathrm{c}^\mathrm{T}\mathrm{x}$ attains its maximum over a bounded domain $D = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$, i.e., a convex polytope, in an extremal point (edge point) of D. If D is unbounded and if a maximum of $\mathrm{c}^\mathrm{T}\mathrm{x}$ exists, then it is as well attained at an edge point of D.

Equipped with these two tools, we can state a general result about matrix-games (illustrated in Fig. 1).

Lemma 3: Let $\mathrm{A} \in \{0,1\}^{\mathrm{n}_1 \times \mathrm{m}_2}, \mathrm{B} \in \{0,1\}^{\mathrm{n}_2 \times \mathrm{m}_2}$ be two binary matrices. Call $\mathrm{v(A)}, \mathrm{v(B)}$ the (saddle-point) values of the two zero-sum games induced by the matrices A, B. Then

$$|\mathrm{v(A)} - \mathrm{v(B)}| \geq 4^{-\mathrm{k}\log_2 \mathrm{k}} \tag{4}$$

where $\mathrm{k} = 2 + \max\{\mathrm{m}_1 + \mathrm{n}_1, \mathrm{m}_2 + \mathrm{n}_2\}$.

Proof: We start by looking at the matrix A, keeping in mind that an identical line of arguments applies to the matrix B.

Denote the entries of $A \in \{0,1\}^{n_1 \times m_1}$ as a_{ij}. We ought to prove that the saddle-point value $v(A)$ can take on only a finite set of rational values. It is known [32] that the value v can be obtained through a linear optimization problem, where $v = v(A)$,

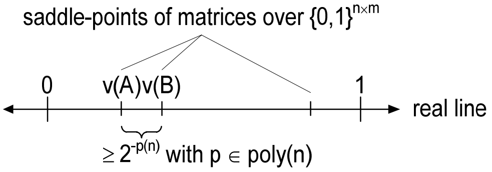

saddle-points of matrices over {0,1}^{n×m}
0
v(A)v(B)
1
real line
≥ 2^{-p(n)} with p ∈ poly(n)

Fig. 1 Distribution of saddle-point values.

$$\begin{array}{rcl} v & \rightarrow & \max \\ \text{s.t. } \sum_{j=1}^{n_1} a_{ij} y_j & \geq & v \ \ \forall i = 1, \dots m_1, \\ \sum_{j=1}^{n_1} y_j & = & 1, \\ y_j & \geq & 0 \ \ \forall j = 1, \dots, n_1. \end{array} \tag{5}$$

The general version of such linear programming problems takes the form,

$$\begin{array}{rcl} c^T x & \rightarrow & \max\limits_{x} \\ \text{s.t. } Mx & \leq & b, \end{array}$$

where the equality constraint $\sum_{j=1}^{n_1} y_j = 1$ can easily be decomposed into two inequality constraints. Problem (5) can be cast into its general version above, setting $x = (y_1, \dots, y_{n_1}, v), c = (0, \dots, 0, 1)$ and by virtue of the block-matrix M (where the subscripts indicate the shapes of the blocks),

$$\underbrace{\begin{pmatrix} -A^T & \mathbf{1}_{(n_1\times 1)} \\ \mathbf{1}_{(1\times n_1)} & 0 \\ -\mathbf{1}_{(1\times n_1)} & 0 \\ -I_{(n_1\times m_1)} & \mathbf{0}_{(n_1\times 1)} \end{pmatrix}}_{=M} \begin{pmatrix} x_1 \\ \vdots \\ x_{n_1} \\ v \end{pmatrix} \leq \underbrace{\begin{pmatrix} \mathbf{0}_{(n_1\times 1)} \\ 1 \\ -1 \\ \mathbf{0}_{(n_1\times 1)} \end{pmatrix}}_{=b}$$

where $1_{(n\times m)}, 0_{(n\times m)}$ denote $(n \times m)$-matrices of all 1's or 0's respectively. M is rectangular with shape $(m_1 + n_1 + 2) \times (n_1 + 1)$. Put $k_1 := m_1 + n_1 + 2$ in the following. Consider the simplex method [31] for solving such a linear program. Geometrically, we seek the optimal solution along the edges of the polyhedron defined by the linear inequality constraints $Mx \leq b$. A base-solution is obtained by intersecting a number of hyperplanes from the set of constraints, and looking for the maximal outcome of the target functional. Maximizing the target functional's value therefore boils down to finding solutions of linear systems of the (general) form $Cx = d$, where $C \in \{-1,0,1\}^{k' \times k'}$ is some invertible (sub-)matrix of M, and $d \in \{0,1\}^{k'}$ is the vector of corresponding elements of the right hand side b. The Simplex-algorithm iterates through multiple such base-solutions (systems $Cx = d$) in an attempt to increase the value of the target functional in each step.

We apply Cramer's rule to obtain a solution of (an arbitrary sub-system) $Cx = d$. Notice that C has integer elements, and $\det(C) \neq 0$. Furthermore, call C_i the matrix arising from exchanging the i-th column in C with the right-hand side d (as prescribed by Cramer's rule). As before, observe that $\det(C_i) \in \mathbb{Z}$, because d has only integer entries. Therefore, the final solution is $x_i = \frac{\det(C_i)}{\det(C)} \in \mathbb{Q}$.

Both, C and C_i are $(k_1^{'} \times k_1^{'})$ -matrices (with $k_1^{'} \leq \min\{m_1 + n_1 + 2, n_1 + 1\}$) over the set $\{-1,0,1\}$, so their determinants can be upper-bounded using Hadamard's inequality ([33], Cor. 9.24), giving $1 \leq |\det(\mathrm{C})| \leq {\mathrm{k}_1^{'}}^{\mathrm{k}_1^{'}/2} \leq 2^{\mathrm{k}_1 \log_2 \mathrm{k}_1}$. The matrix C_i does not need to be invertible, but the same upper bound applies to it, i.e., $0 \leq |\det(C_i)| \leq 2^{k_1 \log_2 k_1}$. This provides us with a neat characterization of possible values for the solution variables (observe that, by Theorem 2, the optimal saddle-point value v from (5) must be among them, and that the constraints of (5) imply non-negativity of any involved variable): $x_i, v(A) \in S_A$ for all $i = 1,2, \dots, n_1$ with

$$S_A := \left\{ \frac{p}{q} \middle| p, q \in \mathbb{N};\ p, q \leq\ 2^{k_1 \log_2 k_1};\ q \geq 1 \right\}.$$

In the very same manner, we conclude that the saddle-point value v_B must lie in a set S_B defined analogously as S_A, except for using the bound $k_2 = m_2 + n_2 + 2$ instead of k_1. Hence both values $v(A)$ and $v(B)$ must lie in the union

$$\begin{aligned} S &:= S_A \cup S_B \\ &= \left\{ \frac{p}{q} \middle| p, q \in \mathbb{N};\ p, q \leq\ 2^{k \log_2 k};\ q \geq 1 \right\} \end{aligned}$$

with $k = \max\{k_1, k_2\}$.

With this in mind, we return to the original problem: find the smallest nonzero increment $|v(B) - v(A)|$ for any two matrices $A \in \{0,1\}^{n_1 \times m_1}, B \in \{0,1\}^{n_2 \times m_2}$. Furthermore, as the game-matrices are over $\{0,1\}$, we have $0 \leq v(A), v(B) \leq 1$. By sorting S in ascending order, we obtain a Farey-sequence of order $2^{k \log_2 k}$. From this, knowing that $v(A)$ and $v(B)$ are elements of the same Farey-sequence S, Lemma 2 tells that $|v(B) - v(A)| \geq\ 2^{-2k \log k}$. Substituting the value $k = \max\{m_1 + n_1 + 2, m_2 + n_2 + 2\}$ into the above formula completes the proof.

Lemma 4: Let G be a graph, and assume a fixed multipath encoding and adversary structure, where the model matrix is set up over the set $\{0,1\}$ with 1 indicating success. Let $\rho_0 \in [0,1]$. The language

$$L = \{(G, \rho_0) | G \text{ has vulnerability} \leq \rho_0\} \qquad (6)$$

is in NP. Moreover, the numbers of rows and columns of the model matrix for the derivation of G is polynomial in the order and size of G.

Proof: Let G be a graph with n nodes. A theorem of H. Whitney ([34], Thm.5.17) tells that the number of node-disjoint paths between any two nodes equals the vertex connectivity number $\kappa(G)$, and is thus trivially bounded as $\kappa(G) \leq\ n$. Because the encoding is fixed, at most k paths will be chosen for multipath routing, where k is a constant. So, the number of strategies is upper-bounded by $\sum_{i=1}^{k} \binom{\kappa(G)}{i} \leq \sum_{i=1}^{k} \binom{n}{i} \in poly(n)$. Similarly, as the adversary structure is fixed, we have no more than k' nodes compromised, where k' is as well a constant. As before, the number of adversarial pure strategies is no more than $\sum_{i=1}^{k^{'}} \binom{n}{i} \in poly(n)$.

To decide L, a Turing-machine can first set up the model matrix (game-matrix) A. According to the above considerations, this can be done in polynomial time. Next, it (nondeterministically) guesses a solution x, y, v, and (deterministically) verifies that x, y are discrete probability distributions. Observe that we do not need to check whether (x, y) is a saddle-point, since, if the vulnerability threshold ρ_0 can be undercut, then our nondeterministic Turing-machine will always make the right guess (i.e., produce a saddle-point). The verification of the vulnerability constraint proceeds in polynomial time by evaluating and checking whether $1 - x^T A y \leq \rho_0$.

Calling U the set of nodes in G that can communicate, there are only $\binom{|U|}{2}$ (i.e., polynomially many) such pairs that can communicate. Combining all these polynomial efforts, we end up concluding that $L \in \mathrm{NP}$ and completing the proof.

Now we can show that the optimization problem (2) is in NP by exhibiting a polynomial time reduction to the language (6).

Lemma 5: If the cost-function c in (2) can be evaluated in polynomial time, then the optimization problem (2) is in NP.

Proof: Consider the decision version of problem (2): for a given $\rho_0 \in [0,1]$ cost limit $K \in \mathbb{R}$ and edge augmentation set $E^{'} \subseteq (V \times V) \setminus E$:

$$\exists \tilde{E} \subseteq E^{'}\colon \left[R\left(U, G\left(V, E \cup \tilde{E}\right)\right) \leq \rho_0\right] \wedge \left[c\left(\tilde{E}\right) K\right]?(7)$$

It is easy to construct a Turing-machine that decides question (7), by nondeterministically guessing some set $\tilde{E}$ and later on checking the constraint on the vulnerability and the cost. Taking advantage of Lemma 4, the effort for this is polynomial (assuming that the cost-function can be evaluated in polynomial time to check the cost-constraint $c\left(\tilde{E}\right) \leq K$). Call this nondeterministic Turing-Machine $\mathcal{M}_{\mathrm{D}}$, having hard-coded parameters K and E' and taking ρ_0 as its input. We denote a call to $\mathcal{M}_{\mathrm{D}}$ resulting in a yes/no decision and an encoding of $\tilde{E}$ on its output tape, as a "function call" $\mathcal{M}_{\mathrm{D}}(\rho_0)$ returning "yes" and $\tilde{E}$ or "no" without further output. A yes-decision indicates that the network can (within the budget limit K) be augmented to have less vulnerability than ρ_0.

Now, take two different extensions $E_1, E_2 \subseteq E'$ and let A, B denote the model matrices referring to the graphs $G(V, E \cup E_1)$ and $G(V, E \cup E_2)$. Without loss of generality, we can assume $A \neq B$ (in case that $A = B$, there would be no point in distinguishing between E_1 and E_2 except for different costs perhaps). By Lemma 3, we know that the saddle-point values $v(A), v(B)$ cannot be arbitrarily close to each other; in fact

$$\begin{aligned} 2^{-k} &\leq |v(A) - v(B)| \\ &= \left|\left(1 - v(A)\right) - \left(1 - v(B)\right)\right| \\ &= \left|R\left(U, G(V, E \cup E_1)\right) - R\left(U, G(V, E \cup E_2)\right)\right| \end{aligned}$$

where k is polynomial in the shapes of A and B (see Lemma 3). To obtain the value of k, we set up the matrix $C \in \{0,1\}^{n \times m}$ for the fully augmented graph $G(V, E \cup E')$ and put $k = m + n + 2$, noticing that no smaller (in terms of edges) graph can ever yield a larger model matrix.

Since edge augmentation preserves existing transmission strategies (for obvious reasons) and the vulnerability R ranges between 0 and 1, we can run a bisective search on [0,1], calling $\mathcal{M}_{\mathrm{D}}$ in each step to decide whether to search the left or right half of the search space:

```
Algorithm 1 (BisectiveSearch):
  $a \leftarrow 0, b \leftarrow 1$
  Set up the model matrix $C \in \{0,1\}^{n \times m}$ for $G(V, E \cup E')$
  $k \leftarrow m + n + 2$
  while$|b - a| > 2^{-k}$do
    $r \leftarrow \frac{a+b}{2}$
    if$\mathcal{M}_D(r)$ returns "yes" then
      $b \leftarrow r$
    else
      $a \leftarrow r$
  endwhile
  Let $\mathcal{M}_D$ run on input $b$ to guess a solution extension$\tilde{E}$
  return $R\left(U, G\left(V, E \cup \tilde{E}\right)\right)$
```

The condition in the while-loop is justified by Lemma 3, and the loop obviously terminates after polynomially many steps at most. Observe that the final step in which $\mathcal{M}_{\mathrm{D}}$ is used to guess a solution within the range $[\mathrm{a}, \mathrm{b}]$, it will necessarily terminate with "yes", leaving some (encoding of) $\tilde{E}$on its output tape. This must be the optimal solution, since the search range $[a, b]$ for the vulnerability has been narrowed down to finally contain only a single extension $\tilde{E}$ that can yield a vulnerability $R\left(U, G\left(V, E \cup \tilde{E}\right)\right) \in [a, b]$ (and no extension can do better than offering us vulnerability a).

Hence, the whole algorithm runs in nondeterministic polynomial time (since $\mathcal{M}_{\mathrm{D}}$ is nondeterministic) and solves the optimization problem (2). Thus, the problem is in NP, and the proof is complete.

By a reduction from the 0-1-integer programming problem, one can prove

Lemma 6 [28]: The optimization problem (2) is NP-hard.

Lemma 5 and Lemma 6 now directly imply Theorem 3: The optimization problem (2) is NP-complete.

For later reference, let us denote the set of all instances of the optimization problem (2) as MINIMUMVULNERABILITY. Then Theorem 3 is rephrased into saying that (the decision version of) MINIMUMVULNERABILITY is NP-complete. We shall come back to this, when we draw the overall picture in section 5.

4. Minimal Costs for Desired Security

Let us look at the converse problem, obtained by minimizing the costs for a desired level of security. We shall use the same symbols as before, i.e., a graph $G(V,E)$ along with a set E' of possible edge-extensions. We avoid ambiguities regarding the definition of "optimum", by assuming a scalar-valued cost functional $c: V \times V \to \mathbb{R}^+$. The optimization problem is

$$\begin{array}{rcl} c(\tilde{E}) & \to & \min_{\tilde{E}\subseteq E'} \\ \text{s.t.}\quad R\left(U, G(V, E \cup \tilde{E})\right) & \leq & M \end{array} \tag{8}$$

where $\mathrm{M} \in \mathbb{R}$ is the maximum allowed vulnerability. So we seek to identify the minimum cost extension to the graph G that provides maximal attack resilience. The problem has already been classified:

Lemma 7 [8]: The optimization problem (8) is NP-hard.

The decision version of the optimization problem (8) would, for a fixed graph $G(V,E)$, a fixed edge-extension E' and a fixed vulnerability threshold M, be stated as

$$\exists \tilde{E} \subseteq E' \colon \left[R\left(U, G(V, E \cup \tilde{E})\right) \leq M\right] \wedge [c(\tilde{E}) \leq c_0]? \tag{9}$$

We can construct a nondeterministic Turing-machine deciding question (9) in an obvious way: Given c_0, it (nondeterministically) guesses $\tilde{E}$ and then checks whether the conditions of (9) hold, all of which can be done in time being polynomial in the size of the problem, provided that the cost functional c can be evaluated in polynomial time. On the other hand, proving (9) to be NP-complete requires a more careful argument, since the technique used in Section 3 relies on an optimization target functional that attains only discrete values. Since the cost functional c in (8) was arbitrary (i.e., possibly continuous), we need to suitably constrain it. Fortunately, this restriction turns out rather mild, as we will discuss later.

In the following, let PSPACE denote the set of all languages that are decidable with polynomial space demand, and let EXPTIME be the set of languages that can be decided in exponential time. It turns out that the problem (8) can be instantiated to be NP-complete as well as to become unsolvable within polynomial space:

Theorem 4: Let L_c be the set of instances of problem (8). There are subsets $L_1 \subseteq L_c$ and $L_2 \subseteq L_c$ such that

- L_1 is NP-complete;
- $L_2 \in$ EXPTIME \ PSPACE.

Proof: We give an explicit construction of L_1 and L_2 by specifying the cost functionals that either yield to an NP-complete problem, or to a problem, being unsolvable with less than exponential effort.

Construction of L_1: Let $n = |E'|$ and define the cost functional $c: \{0,1\}^n \to \mathbb{R}^+$ as an m-degree polynomial $c(x) := \sum_{|\mu|\leq m} \alpha_\mu x^\mu$, where μ is a multiindex, and each $\alpha_\mu = \frac{a_\mu}{b_\mu}$ with $a_\mu \in \mathbb{Z}, b_\mu \in \mathbb{N}^+$is a rational number (i.e., the cost-functional operates on the indicator variables for each edge in E'). To occasionally ease notation, let us fix some enumeration of the coefficients with a unique association between a multi-index μ and its (integer) representative index i, so let us keep in mind that $\{\alpha_\mu | 0 \leq |\mu| \leq m\} \simeq \left\{\alpha_0, \alpha_1, \ldots, \alpha_i = \frac{a_i}{b_i}, \ldots, \alpha_d\right\}$ for some number d. Assume that c can be encoded with $p(n)$ bits, where p is some polynomial (see Fig. 2 for an illustration.

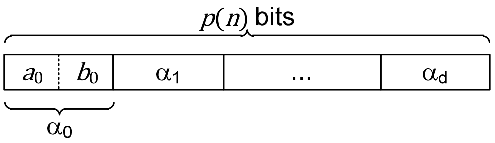


Fig. 2 Encoding of a polynomial with rational coefficients.

A suitable encoding is readily constructed, since each coefficient requires only a constant number of bits, and there are linearly many of them).

We will show that the language L_1 that comprises all instances of problem (8) with the described cost functional is NP-complete.

The encoding of c within a polynomial number $p(n)$ bit implies $d+1 \le p(n)$ (as there cannot be more coefficients than bits) and furthermore

$$\mathbb{Q} \ni |\alpha_i| = \frac{|a_i|}{b_i} \le 2^{p(n)}$$

for all $i = 0,1,\dots,d$ (the bound would be attained for $d = 0$, in which case the problem is trivial). The encoding also implies that $|a_i|, b_i \le 2^{p(n)}$, and we may use $q = \operatorname{lcm}(b_0,\dots,b_d) \le \left[2^{p(n)}\right]^{d+1} \le 2^{p(n)^2}$ as a common denominator, so that $\alpha_i = \frac{p_i}{q}$ for all $i = 0,1,\dots,d$. From $|\alpha_i| \le 2^{p(n)}$ we find

$$|p_i| \le 2^{p(n)} q \le 2^{p(n)+p(n)^2} \qquad (10)$$

which will become useful later. For some fixed $x_0 \in \{0,1\}^n$, the value $c(x_0)$ is simply the sum over a subset of coefficients of c, i.e.,

$$c(x_0) = \sum_{\mu \in S(x_0)} \alpha_\mu$$

where $S(x_0) = \{\mu | x_0^\mu \neq 0, |\mu| \le m\}$. Consequently, we have

$$|c(x_0) - c(y_0)| = \left| \sum_{\mu \in S(x_0)} \frac{p_\mu}{q} - \sum_{\mu \in S(y_0)} \frac{p_\mu}{q} \right|$$

$$= \frac{1}{q} \left| \sum_{\mu \in S(x_0)} p_\mu - \sum_{\mu \in S(y_0)} p_\mu \right| \in \frac{1}{q}\mathbb{N}$$

A trivial bound to c is given by the sum of all absolute coefficients, so

$$|c(x)| \le \textstyle\sum_{i=0}^{d} |\alpha_i| = \frac{1}{q} \sum_{i=0}^{d} |p_i| =: C \qquad (11)$$

for every x, and we can perform a bisective search as in Algorithm 1: we subdivide the initial cost interval $[a,b] = [0,C]$ until $|b-a| < \frac{1}{q}$ with $q = 2^{-p(n)^2}$ (remember that p is a polynomial). As in the proof of Lemma 5, let the respective nondeterministic machine be denoted as $\mathcal{M}_{\mathrm{D}}$. Assume that this machine upon terminating leaves a "yes/no" decision on its output tape, and in case of a "yes"-answer, prints a solution $\tilde{E}$ in addition. The decision whether to take the left or right subinterval can be made by virtue of calling $\mathcal{M}_{\mathrm{D}}$ with input $r = \frac{a+b}{2}$, where a "yes"-answer indicates that we continue searching in the right half $[a,r]$, and otherwise we go on searching $[r,b]$. The algorithm is almost identical to Algorithm 1, so we refrain from restating it.

Once the bisective search has stopped, we can use $\mathcal{M}_{\mathrm{D}}$ to guess a solution within the remaining interval $[a,b]$. At this stage, we can be sure that $\mathcal{M}_{\mathrm{D}}$ will give us a solution $\tilde{E}$, and that this solution is the only one within $[a,b]$. Hence, it must be optimal.

We count the steps until the search space has become sufficiently narrow to contain at most one element (which by then must be the solution): we perform k iterations of the binary search to shrink the interval down to $\frac{C}{2^k} < \frac{1}{q}$, which happens after (using inequality $d+1 \le p(n)$ from above),

$$\mathcal{O}(\log(qC)) = \mathcal{O}\left(\log \sum_{i=0}^{d} |p_i|\right)$$

$$\subseteq \mathcal{O}\big(\log(p(n)) + p(n) + p(n)^2\big)$$

iterations. The overall workload is thus polynomial and $L_1 \in$ NP follows. Lemma 7 classifies the problem as NP-hard regardless of the cost-functional, and therefore proves L_1 NP-complete.

Construction of L_2 : Let the cost functional $c: \{0,1\}^n \to \mathbb{R}^+$ be an unordered list, and let L_2be the collection of all instances of problem (8), where $M \ge \max T$.

By construction, we have the risk bounded as $R\left(U, G\left(V, E \cup \tilde{E}\right)\right) \leq \max T \leq M$ for every $\tilde{E} \subseteq E'$, so the constraints do not impose any restriction on the set of feasible solutions. So we are left with the search for the minimum over a list with 2^n elements, requiring $\Omega(2^n)$ bits of space (to store the list), and at least exponential time to find the minimum element. The proof is complete.

The alert reader will instantly notice the possibility to substantially simplify the proof by using a cost function polynomial with integer rather than rational coefficients. The advantage of using the latter, however, is the fact that rationals can be used to approximate reals (cf. Dirichlet's approximation theorems), and that the vector-space of polynomials over some compact set V is (topologically) dense in the set of continuous functions over V, by Weierstra ß' approximation theorem. Hence, all problems with a continuous cost functional (e.g., polynomials, exponential costs, logarithmic costs, etc.) are as well "close" to some neighboring problem instance that belongs to an NP-complete language. This makes the restriction to polynomial cost functions rather mild and still covers a large class of potentially interesting cost functionals.

5. The Overall Picture

Leaving the cost-functional in optimization problem (8) as a degree of freedom gives rise to a variety of languages, which we can collect in some (auxiliary) complexity class MCS (i.e., minimum cost security). Theorem 4 can now be rephrased into the statement that MCS has a non-empty intersection with the class NP-complete (NPC), as well as the exterior of PSPACE, and is contained in EXPTIME. This conclusion is depicted in Fig. 3.

An interesting yet open problem is whether problem (8) can be instantiated to be in any complexity class in-between the NP-complete problems and the outside of PSPACE. This unknown region is displayed as a dashed ellipsis.

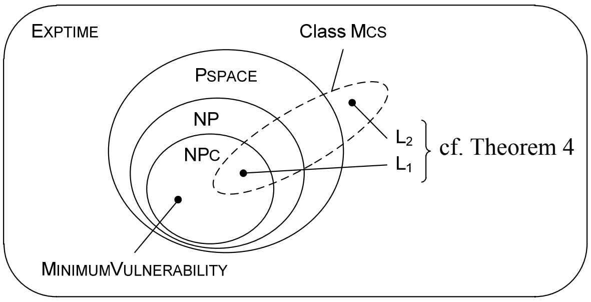


Fig. 3 Problems of maximal security in the landscape of prominent complexity classes.

6. Conclusions

Multipath transmission is a known necessity for perfectly secure communication, even in the presence of quantum key distribution. Based on a game-theoretic measure of security, we proved that the problem of extending a network towards maximal attack resilience is NP-complete. The related problem of finding a minimum cost extension to undercut a given vulnerability threshold is as well NP-complete in certain cases. Unfortunately, this problem can be instantiated to yield a class of languages even outside PSpace, and hence beyond anything computationally feasible nowadays. Problems falling into the latter class are of minor interest, but might be observed if the initial graph already satisfies the imposed security requirements and thus no extension is needed at all. Thus, the problem either diminishes, or the envisioned extension allows for better security than yet imagined. In the latter case, the bound on the vulnerability can be tightened, thus making the problem nontrivial again.

The conclusion to be drawn from all this is still a positive one: it appears that the problem of secret (confidential) communication is hardly affected by the way in which the P-vs-NP-question is settled.

If P ≠ NP, then nowadays public-key cryptography will continue providing a sufficient level of confidentiality. For example, McElice-encryption [35] is based on the decoding problem for linear codes, which is known to be NP-complete [36]. If P ≠ NP, then NP is strictly separated from P, implying that there really is no polynomial time algorithm to break McElice's encryption.

Otherwise, if P = NP, then both problems exhibited here have computationally feasible solutions and the development of infrastructures facilitating arbitrarily secure communication is efficiently possible: If P = NP, then the enumeration of paths and attack strategies (or general graph-theory based topological vulnerability analysis) can be done with only polynomial efforts. The complexity of solving the subsequently arising optimization problems is then polynomial as well. So on the theoretical side, we can efficiently set up an infrastructure that permits perfectly confidential communication at feasible (monetary) expenditures (of course, from a practical point of view, the polynomial's degree must be feasibly low, which we can reasonably assume from working through the proof's arguments with a focus on that issue).

Rephrasing the lessons that we learnt from our analysis, the death of computational infeasibility assumptions (and as such the collapse of many nowadays public-key cryptosystems) will be the birth of information-theoretically secure network infrastructures, giving rise to a new era of secure communication.

References

[1] C. Shannon, Communication theory of secrecy systems, Bell System Technical Journal 28 (1949) 656-715.

[2] A. Poppe, M. Peev, O. Maurhart, Outline of the SECOQC quantum-key-distribution network in Vienna, International Journal of Quantum Information 6 (2) (2008) 209-218.

[3] M.A. Kumar, P.R. Goundan, K. Srinathan, C.P. Rangan, On perfectly secure communication over arbitrary networks, in: Proceedings of the Twenty-First Annual Symposium on Principles of Distributed Computing, New York, USA, 2002, pp. 193-202.

[4] A. Patra, B. Shankar, A. Choudhary, K. Srinathan, C.P. Rangan, Perfectly secure message transmission in directed networks tolerating threshold and non-threshold adversary, Lecture Notes in Computer Science 4856 (2007) 80-101.

[5] Y. Wang, Y. Desmedt, Perfectly secure message transmission revisited, IEEE Transactions on Information Theory 54 (6) (2008) 2582-2595.

[6] T.-S. Hsu, Graph augmentation and related problems: theory and practice, Ph.D. Thesis, University of Texas at Austin, December 1993.

[7] O. Goldreich, S. Goldwasser, On the possibility of basing cryptography on the assumption that P = NP, Technical Report 005, Cryptology ePrint Archive, February1998.

[8] S. Rass, A.Wiegele, P. Schartner. Building a quantum network: how to optimize security and expenses, Journal of Network and Systems Management 18 (3) (2010) 283-299.

[9] S. Hariri, G. Qu, T. Dharmagadda, M. Ramkishore, C.S. Raghavendra, Impact analysis of faults and attacks in large-scale networks, IEEE Security and Privacy 1 (5) (2003) 49-54.

[10] S. Jajodia, S. Noel, B. O'Berry, Massive computing, in: Topological Analysis of Network Attack Vulnerability, Springer, US, 2005, pp. 247-266.

[11] Combinatorial Analysis Utilizing Logical Dependencies Residing on Networks (CAULDRON), 2008, available online at: http://csis.gmu.edu/TVA/, last access: October 12, 2011.

[12] W. He, C. Xia, C. Zhang, Y. Ji, X. Ma, A network security risk assessment framework based on game theory, Future Generation Communication and Networking 2 (2008) 249-253.

[13] R. Alleaume, F. Roueff, E. Diamanti, N. Lütkenhaus, Topological optimization of quantum key distribution networks, New Journal of Physics 11 (2009) 075002.

[14] S. Rass, P. Schartner, Multipath authentication without shared secrets and with applications in quantum networks, in: Proceedings of the International Conference on Security and Management (SAM), July 12-15, 2010, Vol. 1, pp. 111-115.

[15] J. Shi, Y. Lu, L. Xie, Game theory based optimization of security configuration, in: International Conference on Computational Intelligence and Security, 2007, pp. 799-803.

[16] Z. Ying, H. Hanping, G. Wenxuan, Network security transmission based on bimatrix game theory, Wuhan University Journal of Natural Sciences 11 (3) (2006) 617-620.

[17] E. Altman, T. Bolougne, R. El-Azouzi, T. Jiménez, L. Wynter, A survey on networking games in telecommunications, Elsevier Journal on Computers & Operations Research 33 (2006) 286-311.

[18] T. Lakshman, K. Murali, Detecting network intrusions via sampling: a game theoretic approach, in: IEEE INFOCOM, San Francisco, California, USA, 2003.

[19] P. Michiardi, R. Molva, Game theoretic analysis of security in mobile ad hoc networks, Technical Report Research, Report No. RR-02-070, Institut Eurécom, 2229 Route des Crêtes-BP 193, 06904 Sophia-Antipolis, France, April 2002.

[20] W. Yu, K.J.R. Liu, Game theoretic analysis of cooperation stimulation and security in autonomous mobile ad hoc networks, IEEE Transactions on Mobile Computing 6 (5)

(2007) 507-521.

[21] D.A. Burke, Towards a game theory model of information warfare, Master's Thesis, Faculty of the Graduate School of Engineering and Management of the Air Force Institute of Technology, Air University, Air Education and Training Command, 1999.

[22] K.-W. Lye, J.M. Wing, Game strategies in network security, International Journal of Information Security 4 (2005) 71-86.

[23] S.N. Foley, S. Bistarelli, B. O'Sullivan, J. Herbert, G. Swart, Quality of protection, in: Multilevel Security and Quality of Protection, Springer, US, 2006, pp. 93-105.

[24] M. Franklin, R. Wright, Secure communication in minimal connectivity models, Journal of Cryptology 13 (1) (2000) 9-30.

[25] J.C. Benaloh, J. Leichter, Generalized secret sharing and monotone functions, in: Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology, 1988, pp. 27-35.

[26] T. Alpcan, T. Başar, Network Security: a Decision and Game Theoretic Approach, Cambridge University Press, 2010.

[27] S. Rass, P. Schartner, Game-theoretic security analysis of quantum networks, in: Proceedings of the Third International Conference on Quantum, Nano and Micro Technologies, February 2009, pp. 20-25.

[28] S. Rass, On information-theoretic security: contemporary problems and solutions, Ph.D. Thesis, Klagenfurt University, Institute of Applied Informatics, June 2009.

[29] S. Rass, P. Schartner, Security in quantum networks as an optimization problem, in: Proceedings of the International Conference on Availability, Reliability and Security, 2009, pp. 493-498.

[30] G. Hardy, E. Wright, An Introduction to the Theory of Numbers, Oxford Science Publications, 5th ed., 1984.

[31] A. Koop, H. Moock, Lineare Optimierung: Eine Anwendungsorientierte Einführung in Operations Research, Spektrum Akademischer Verlag, 2008.

[32] W. Schlee, Einführung in Die Spieltheorie, Springer, 2004.

[33] H. Dym, Linear Algebra in Action, American Mathematical Society, 2007.

[34] G. Chartrand, P. Zhang, Introduction to Graph Theory, Higher Education, McGraw-Hill, Boston, 2005.

[35] A. Menezes, P.C.V. Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press LLC, 1997.

[36] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, New York, 1979.

Computer Technology and Application 2 (2011) 906-911

Games-Based Learning Framework

Antonio Garcia-Cabot, Eva García, Roberto Barchino, Luis de-Marcos, José-María Gutiérrez, José-Antonio Gutiérrez, José-Javier Martínez, Salvador Otón and José-Ramón Hilera

Computer Science Department, University of Alcala, Alcala de Henares, Madrid 28871, Spain

Received: September 16, 2011 / Accepted: September 29, 2011 / Published: November 25, 2011.

Abstract: This work presents the authors' experience in the field of mobile technologies, from which several initiatives have emerged. As result of this, a games-based framework for learning has been developed in these last years. This framework is composed by a competition called Mobigame, which has as main aim to stimulate the participation of the students. By participating in this competition participants learn to develop for mobile devices. A game to practice Japanese is also presented in this article, which was presented in the above mentioned competition. This game has been developed for mobile phones or PDAs (Personal Digital Assistants) based on the JME (Java Mobile Edition) technology. Finally, another initiative is also presented: A free download platform of digital contents for mobile devices based on info-educational games.

Key words: Game-based learning, mobile, framework.

1. Introduction

Our department, and our research group in particular, has been working more than ten years in the field of e-learning systems. One of its lines of research is the application of mobile devices in the learning process, which is known as m-learning [1]. These devices can be used for many other tasks than those for which they were designed a priori [2-3]. Thus, extending their functionality to other fields such as m-learning is one of the objectives of the TIFYC (Information technologies for the training and knowledge) research group. As a result of this work, a Technical Workshop on Developing for mobile devices has been consolidated since the 2000 year, and it is celebrated each year as a prelude of the Mobigame Competition [4]. This technical workshop offers to the students of Computer Science a basic knowledge of developing for mobiles technologies, such as JME (Java Mobile Edition), Android and Windows Phone. A few months later of this workshop the Mobigame Competition takes place and all students are welcome to participate developing a game or an application using mobile technologies. The jury is composed by some professors of computer science, experts in mobile devices and company representatives. In the competition of the 2010 year the Kanatest Mobile game was presented, which is an application for mobile phones developed using JME technology. The main aim of this game-based learning is to practice and memorize easily the Kana. The Japanese Kana includes symbols used to write the Japanese alphabet. There are three kinds of symbols sets: Katana, usually used to denote foreign words; Hiragana, used for Japanese words; and Kanji. However, Kanji has not phonetic correspondences, but each symbol represents a different word, thus this game works only with the first two (Katana and Hiragana).

Finally, a Technological Platform has been developed to share info-educational contents [5]. This platform was developed with the objective of being a free and open platform for the exchange of educational contents (games, applications, etc.) for mobile devices. The three main aims of the project were the following:

- Developing contents (games and applications)

Corresponding author: Antonio Garcia-Cabot, M.Sc., master, research fields: mobile devices, mobile learning, artificial intelligent. E-mail: a.garciac@uah.es.

compatible with the highest possible number of devices, analyzing the characteristics of the devices themselves. It was also considered the possibility of creating specific contents that consider the most of the special characteristics of some devices, raising the usability or the level of interactivity (e.g., by using pointing devices);

- Creating a technological architecture to promote the access and use of the developed contents mentioned above;
- Providing the educational contents and games with the required accessibility properties, so they can be used by people with physical limitations.

The following sections present a description of Mobigame competition, Kanatest mobile and the technological platform. And finally, conclusions are drawn.

2. Mobigame

The event Mobigame has been growing since the last years, being now the most important event organized by our Computer Science Department. This event is focused on the creation and development of educational applications, using different mobile technologies such as JME, Android, Windows Phone or iOS (iPhone Operating System). It also promotes team working because participants must work in groups, usually from two to five students. Once the groups are built, they start working on their applications.

From the point of view of the members of the research group, it also allows us to observe common patterns in the participants. The following items are part of these patterns:

- Most of the students asked for help concerning to the same topics. This implies that professors have had to help the students repeating over and over again the same solutions;
- Most of the questions have a technical nature.

It seems a good idea for us to organize some seminars or workshops where we could solve the same problems to all participants at once. In this way, a few months before the event, a set of seminars or technical workshops are organized each year.

The seminars are designed to manage these problems. The following section describes the organization of these seminars.

2.1 Technical Seminars

As mentioned above, some months before the Mobigame event, a series of technical seminars are given to the students. The seminars usually take place in laboratories and consist of technical demonstrations on the functions most commonly used, for example, working with canvas, scrolls, sprites, and so on. These seminars are presented in different mobile technologies, thus the students can choose the technology that they prefer.

The seminars are distributed in several days with duration of around five hours in total. We found enough this time to explain the most important aspects. Moreover, in the last years, as the event has been opened to more development platforms, the seminars have been expanded to cover these new platforms as well. Once the students have participated in the seminars, they are more prepared to complete their work. Some small questions may also appear afterwards, during the development process, but they can be easily solved by the members of our research group. Finally, and after a reasonable period of time, enough for the students to complete their applications, the event Mobigame begins.

2.2 Methodology

Mobigame event spans over three days. The first one, a jury composed of professors of the department evaluates all the proposals. This step is necessary due to the large number of applications submitted by the students. We are forced to choose a selection of applications among all of them, otherwise there would not be enough time for presenting all of them in the remaining days. It also ensures that all the applications that are kept to the second day have a minimum level of quality. Those that are selected the first day are headed into the second day, they are presented in one of the assembly halls of the school, and authors must talk about their work in front of a real audience, comprising students and members of the department. During the second day, students are asked to prepare a small presentation summarizing the main points of their applications. These points include the reasons that led them to choose a specific platform, how the application is used; or in the case of a game, how it is played; and how they developed the work. Finally, they are also asked to provide a real demonstration of the work with real mobile devices. Based on those points, the best proposals are selected to participate in the final, the following day. It is important to mention that while the first and second day, the jury is composed of members of the department and the research group, the jury of the final day is composed of a group of experts of the IT (Information Technologies) field and companies representatives. During all the years while the event has been taking place, the university has signed agreements with companies related to the mobile communications area, as well as some other important international companies, e.g., Microsoft, Java, Apple, etc. It is also worth mentioning that thanks to this collaboration between the University and the companies, many students have found a job as a result of their work in the event. This is also a great motivation for the students.

3. Kanatest Mobile

3.1 Introduction

Kanatest Mobile is an application for mobile phones, developed in Java using JME technology, which allows practicing and memorizing, through the completion of tests, the Japanese Kanas. One of main development challenges of this application has been the creation of an attractive graphical interface for the user. That is the reason why it has included multimedia content.

There were many possible technologies to develop this game, e.g., Java, Dot Net, etc., but finally Java has been used mainly because this technology is taught at the University of Alcala and in addition this technology can be implemented in a large number of mobile devices because it is widely used. It can be executed in mobile devices with a virtual machine of Java, regardless of operating system.

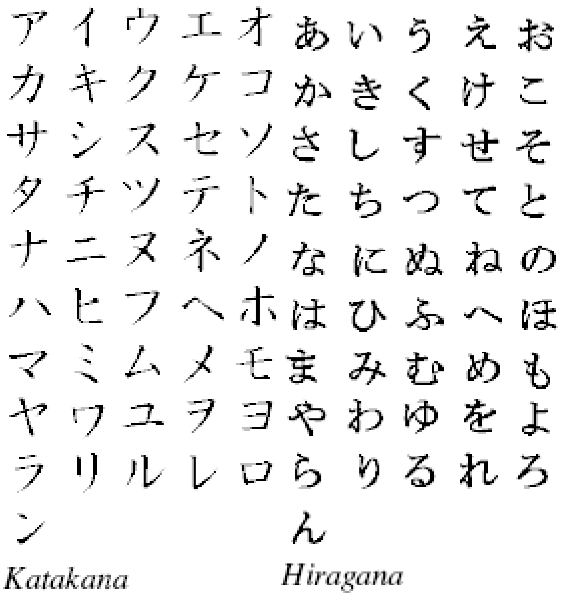
As it was mentioned above, the application uses Katakana and Hiragana (Fig. 1).

Each of these symbols has a phonetic correspondence, usually a syllable. The application will help the user to learn and recognize these symbols and their phonetic correspondences. Moreover, Kanatest Mobile has the possibility to store the test results, as well as to use different user profiles so each user can check his/her history of performed tests, and the improvement recognizing kanas.

3.2 Game Play

The tools used to develop the game have been:

- NetBeans IDE (Integrated Development Environment). For developing the code and its compilation;
- GIMP 2.6.6 and Inkscape 0.46. For creating and editing graphics;
- Melody Raiser. For creating and editing music;
- Sony Ericsson 760i. For tests and demos.



ア	イ	ウ	エ	オ	あ	い	う	え	お
カ	キ	ク	ケ	コ	か	き	く	け	こ
サ	シ	ス	セ	ソ	さ	し	す	せ	そ
タ	チ	ツ	テ	ト	た	ち	つ	て	と
ナ	ニ	ヌ	ネ	ノ	な	に	ぬ	ね	の
ハ	ヒ	フ	ヘ	ホ	は	ひ	ふ	へ	ほ
マ	ミ	ム	メ	モ	ま	み	む	め	も
ヤ	ワ	ユ	ヲ	ヨ	や	わ	ゆ	を	よ
ラ	リ	ル	レ	ロ	ら	り	る	れ	ろ
ン					ん				
Katakana					*Hiragana*				

Fig. 1 Katakana and Hiragana alphabet.

The game operation is really simple. Once the user is logged into the game, he or she only has to choose which of the two available kanas wants to practice. Once chosen, the Konichiwa professor welcomes us as we see in Fig. 2.

- Do test: To practice the chosen kana. Reviewing the table. To study the phonetic symbols of the chosen kana.
- Personal statistics: All our results until now.
- Help: This shows a help message.
- Exit: This option allows leaving KanaTest Mobile.

The main menu has the above options (Fig. 3).

If the user chooses the option “Do Test”, once he or she has enough knowledge about the chosen kana, the application will ask some questions to determine his/her Japanese language learning level. A test example is shown in Fig. 4.

4. Technological Platform

The technological platform is named Dmovil and it allows the creation of an exchange point to share educational contents; games and applications with an educational objective. The main aim was to create a platform where users could upload their own developments and where other users may also download and score them. To deal with this objective, the platform is built using the latest Java technologies, represented by the Struts framework, which is designed to create web applications with an excellent maintainability and easily extensible. As the system would be accessed from mobile devices and desktop computer web browsers, it supports web protocols and languages as well as WAP (Wireless Application Protocol). The WAP version (Fig. 5) of this platform is optimized for small-screen devices, like those that PDAs (Personal Digital Assistants) and mobile phones have.

The project was based on the following ideas:

- To provide a place where students could share ideas related to their work in the Mobigame event;

Fig. 2 Konichiwa professor.

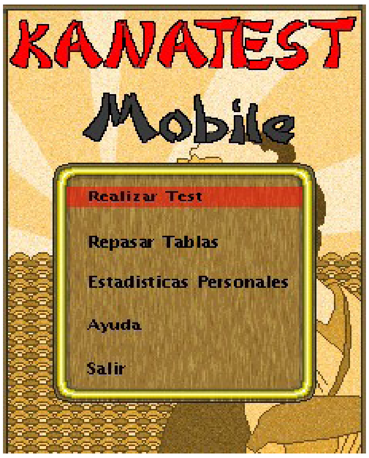


Fig. 3 Main menu (in Spanish).

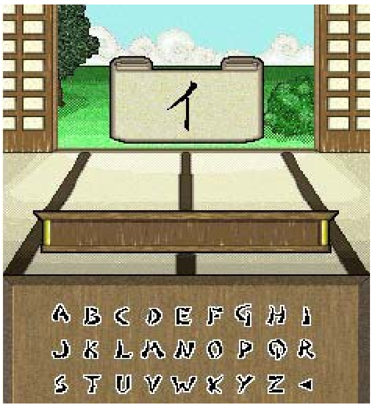
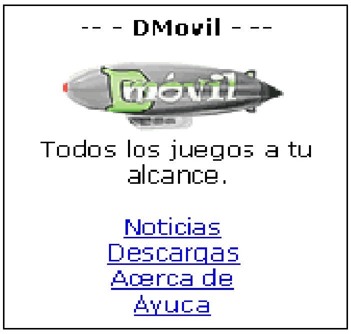


Fig. 4 Test example.



-- - DMovil - --

Todos los juegos a tu alcance.

Fig. 5 Main screen WAP version (in Spanish).

- To gather all the information and examples from the technical seminars;
- To offer a place where the developments could be uploaded, and thus be shared with other students.

These ideas were kept in our minds for developing a viable solution with the objective of providing answers to the students' needs. This project aims to create a platform to host and distribute learning content. More specifically the project is focused on the following points:

- Developing specific contents and dealing with the required actions to make the platform more accessible, so they can be used by the widest range of potential users;
- Developing contents (games and applications) compatible with as many devices as possible, analyzing the actual characteristics of the devices (screen size, input method, operating system, etc.). It is also being considered the possibility of creating specific contents that fit the most of the special characteristics of some devices;
- Making use of the connectivity features of the devices, e.g., bluetooth, UMTS (Universal Mobile Telecommunications System), etc., as well as the connective communication technologies, currently used in applications. These technologies would be used to create and promote collaborative games with the required technological architecture;
- Providing the educational contents and games with the required accessibility properties, so they can be used by disabled people.

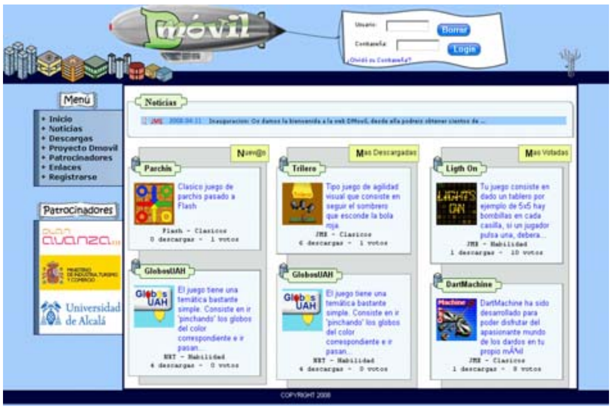
Currently, games and applications for mobile devices can be found on the Internet. However, there are not specialized systems created to distribute such contents, and those which really exist are cumbersome while using and they are not free. Our system fills an important gap in this area, because it is intended to cover both of these aspects, and can be considered as an integral m-learning platform [6].

Our platform was conceived from the beginning to be free of use, and also to be as easy to work with as possible. The platform was created and intended to be used by students so, after all, the main idea was to create a place where the students could share their knowledge and get and provide support to and from others. The main screen of Dmovil platform is shown in Fig. 6.

The platform is a portal for distributing educational content implemented on a Web environment. We decided to create a web platform because it was the easiest way to reach all the intended audience (students), much better than distributing an application on a digital format as Compact Disc or a memory stick [7] or forcing the users to download a local application.

Another advantage of a Web application is that updates in the content or the structure of the application are transparent to the users. Changes can be made at any time, and the user will not need to make any changes himself. The application meets the following technological components:

- Java programming language: We decided to choose Java because it was more appropriated for ours needs than other solutions. This platform has a lot of support from the developers' community and, as a result, many libraries and frameworks have been developed. All these pieces of code could be easily put together and adapted to create the platform we had in mind. On the other hand, the application was indented to run on various platforms, so we needed a multiplatform, and Java fits perfectly to this requirement;



JME 2008-04-11 Inauguracion: Os damos la bienvenida a la web DMovil, desde ella podreis obtener cientos de ...

Clasico juego de parchis pasado a Flash
Flash - Clasicos
0 descargas - 1 votos

Tipo juego de agilidad visual que consiste en seguir el sombrero que esconde la bola roja
JME - Clasicos
6 descargas - 1 votos

Tu juego consiste en dado un tablero por ejemplo de 5x5 hay bombillas en cada casilla, si un jugador pulsa una, debera...
JME - Habilidad
1 descargas - 10 votos

El juego tiene una temática bastante simple. Consiste en ir 'pinchando' los globos del color correspondiente e ir pasan...
NET - Habilidad
4 descargas - 0 votos

El juego tiene una temática bastante simple. Consiste en ir 'pinchando' los globos del color correspondiente e ir pasan...
NET - Habilidad
6 descargas - 0 votos

DartMachine ha sido desarrollado para poder disfrutar del apasionante mundo de los dardos en tu propio mÃ³vil
JME - Clasicos
1 descargas - 8 votos

Fig. 6 Main screen web version.

- Struts framework: The Struts framework is a project of the Apache Foundation and also an open source project. Since it was introduced in 2004, it became a facto standard for web applications, and is now the most used java web framework. The use of this framework can significantly reduce the costs of the development process and the maintaining tasks;
- Web Server—Apache Tomcat: The Apache Tomcat server is another project from the Apache Foundation. It is also an open source application, and could easily deploy written applications using the Struts framework;
- Database Server—any SQL (Structured Query Language) standard compatible: The system has been operating with success using the most popular database servers, i.e., PostgreSQL, MySQL, Oracle and SQL Server. We focused our efforts on creating a system that could be used with the most known database systems.

It is worth mentioning the predominant use of open source software, including multiplatform, which allows its use on Windows and Linux systems. This contributes to our goal of creating a free-to-use system. Another important point of using open-source systems is that defects are found and fixed more quickly [8], which makes our work easier.

5. Results and Conclusions

Mobigame event has been organized over the last years and it has allowed us to discover new ideas around the world of mobile devices and the possibility to exchange perspectives and experiences with some IT professionals. Moreover, the competition motivates students because many of them find their first job when participating with a good project. The Kanatest Mobile game has not been tested yet in a real case but people who have tried it have been pleasantly surprised because it is easy of use, interactive and with the ability to practice easily a complex language like Japanese. The web platform during its first year of life has received multiple visits, downloads of info-educational content (some of these more than 40 downloads) and ratings of the downloads, in addition some users have contributed with more than thirty developments.

References

[1] T. Georgiev, E. Georgieva, A. Smrikarov, M-learning, a new stage of e-learning, in: International Conference on Computer Systems and Technologies, 2004.

[2] J.M. Gutiérrez, S. Otón, M.L. Jiménez, R. Barchino, M-learning enhancement using 3D worlds, International Journal of Engineering Education 24 (1) (2008) 56-61.

[3] R. Barchino, M.L. Jiménez, S. Otón, J.M. Gutiérrez, Experiences in appling mobile technologies in an e-learning environment, International Journal of Engineering Education 23 (3) (2007) 454-459.

[4] Mobigame Event, available online at: http://www.mobigame.uah.es/, accessed: February 2010.

[5] Dmovil Platform, available online at: http:// dmovil.cc.uah.es/DMovil/, accessed: February 2010.

[6] K. Nyiri, Towards a philosophy of m-learning, in: Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02), Teleborg Campus, Växjö University, Växjö, Sweden, 2002, pp. 121-124.

[7] P. Atzeni, A. Gupta, S. Sarawagi, Design and maintenance of data-intensive web-sites, LNCS 1377 (1998) 436-450.

[8] J.W. Paulson, G. Succi, A. Eberlein, An empirical study of open-source and closed-source software products, IEEE Transactions on Software Engineering 30 (4) (2004) 246-256.

Computer Technology and Application 2 (2011) 912-925

An Integrated Approach for Decision Support through Multi-objective Optimization with Application to an Ill-posed Design Problem

Yoshiaki Shimizu, Yasumasa Kato and Takeshi Kariyahara

Mechanical Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan

Received: August 31, 2011 / Accepted: September 09, 2011 / Published: November 25, 2011.

Abstract: In the current scenario of global competition and short product life cycles, customer-defined satisfaction has attracted interest in artifact design. Accordingly, intelligent decision-making through multi-objective optimization has been proposed as an efficient method for human-centered manufacturing. However, previous vast researches on optimization have been mainly focused on optimization theory and optimization techniques and paid little interests on the process of problem formulation itself. In this paper, therefore, the authors present a total framework for supporting multi-objective decision making. Then, the authors try to solve the formulated multi-objective optimization problem that involves both qualitative and quantitative performance measures as a general consequence from the above procedure. Taking especially quality as a qualitative measure, the authors gave a new idea to evaluate the quality quantitatively. Additionally, to facilitate the portability of the proposed method in multidisciplinary decision-making environments, the authors implement the proposal algorithm in an Excel spreadsheet and validate the effectiveness of the approach through a case study.

Key words: Integrated systems approach, optimization engineering, multi-objective optimization, meta-modeling, value system design.

Nomenclature

a_{ij}	value evaluated by pair-wise comparison	[-]
a_i	event in the grey theory	
b_j	game in the grey theory	
k	number of trial solutions	
f	objective function vector	
N	number of objective functions	[-]
p, $\underline{p}$	upper/lower number of selection for decision variables	
P	constant in Eq. (3)	[m]
Q	quality matrix	[-]
q	index corresponding to the Kano classifier	[-]
s_i	intensity of softness of i-th item	
S	situation in the gray set theory	
t, $\underline{t}$	upper/lower number of selection for objective functions	
u_i	intensity of realization of i-th item	
V_{RBF}	value function identified by RBF network	[-]
v	velocity of rolling machine	[m/min]
w	weighing vector	[-]
x	decision variable vector	
X	feasible region in decision variable space	
y	decision variable vector	
z	attribute value	[-]
μ_i	contribution merit of i-th row	
ν_i	contribution merit of j-th column	
ω_{ij}	binary score of relationship between i and j	
Ω_{ij}	score of relationship between i and j	

Superscript

b	lower basis
B	upper basis
R	reference or target

Subscript

max	maximum value
min	minimum value

Yasumasa Kato, post-graduate student, research field: production systems engineering.

Takeshi Kariyahara, under-graduate student, research field: production systems engineering.

Corresponding author: Yoshiaki Shimizu, professor, Dr. Eng., research field: optimization engineering. E-mail: shimizu@me.tut.ac.jp.

1. Introduction

Previous vast researches on optimization have been mainly focused on theories and solution techniques and paid little interests on problem formulation and interpretation of the result from optimization. In other words, there exist many studies on “how to interpret the problem generally” and “how to solve the problem effectively” while few on “how to formulate the problem tangibly” or “how to support decision making in terms of the result of optimization”. Not to stay at a level of concept and just a guideline but to play a key role for the decision making, studies on optimization must turn its attention and make different effort from the conventional aspects. Such attempt can be viewed as a task for establishing a new paradigm referred to optimization engineering. Eventually, its accomplishment is essential for developing an adaptive and human-centered decision support system toward customer-defined satisfaction.

As a challenge for such task, in this paper, we will propose a total framework for supporting multi-objective decision making in an ill-posed environment. This basic idea is mentioned as follows [1-2]: Using appropriate software of systems approach and artificial intelligence in a collaborative manner, we define and formulate the optimization problem as a prior stage of optimization. Then, in the optimization stage, we cope with ill-defined and ill-posed circumstances by employing meta-modeling techniques and subjective decision making approaches. Thereat, we apply a multi-objective optimization method referred to as MOON^{2R} (Multi-Objective Optimizer with value function identified by Neural Network of Radial basis function) [3]. Being robust to unsteady and unstable subjective human judgments peculiar to multi-objective optimization (MOP) problems, MOON^{2R} is amenable to MOPs that rely on a particular meta-model and involve a qualitative evaluation measure such as quality for example. Eventually, to facilitate the widespread and easy application to multidisciplinary decision-making, we implement the solution algorithm in an Excel spreadsheet. Finally, we carry out a case study taking a plastic-sheet-rolling machine and validate the effectiveness.

The rest of this paper is organized as follows: Section 2 briefly outlines the element methods employed in this study; in section 3, we will explain concretely the proposed procedures for problem formulation associated with a case study; section 4 explains the multi-objective decision making in an ill-posed environment; finally, we summarize our conclusions in section 5.

2. Outline of the Employed Element Technologies

2.1 IDEF0

IDEF0 (Integrated DEFinition Method Zero) [4] is a structural modeling method and developed to standardize the specification for placing an order to multiple manufacturers. Today, it has become a basis of suite of software known as IDEF family. In Europe and America, it is widely used in the field such as CIM (Computer Integrated Manufacturing), CALS (Continuous Acquisition and Life-cycle Support), TQM (Total Quality Management), CE (Concurrent Engineering), etc.. In these decision environments, many experts with different disciplines need to collaborate mutually before accomplishing assigned tasks. Hence, in order to entirely understand complex business flows associated with procurement of resources, product development, production, quality control and assurance, and sales practice without misunderstanding and share them among the members, it is of special importance to describe definitely business requirements between organizations in understandable manners. For such requirement, IDEF0 provides a hierarchical structural modeling technique using a basic component composed of one box and 4 arrows that represent activity of business and specific roles of information, respectively (Refer to Fig. 1). In a word, IDEF0 can describe, in a universal

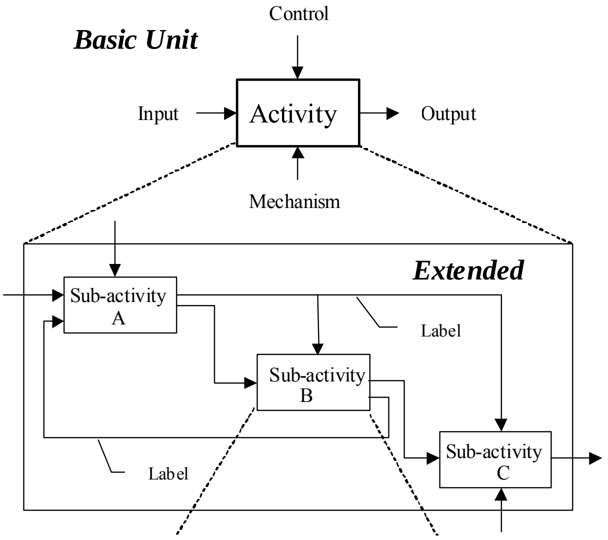


Fig. 1 Basic structure of IDEF0 model and its extended diagram.

way, various activities appearing in the organization by using a simple structure that is easy for everyone to understand.

2.2 Mind Map

Mind map [5] is developed to record the emerging ideas illustratively. By depicting related information radially (hierarchically), according to the deployment of idea, from the theme put in the center of paper, it aims at "association" and "emphasis" of information at the same time. It can be used both to expand the conception and reversely to concentrate the ideas. It always requires us to use only "image" or "keyword" and to avoid not writing too much. This comes from such a fact that human brain is possible to trace the memory from incomplete information like keywords by virtue of order (sequence pattern) and self association. Hence, simplicity is much more weighed than conciseness. Consequently, leveraging this brain function becomes a major point to draw a Mind map.

2.3 TRIZ

TRIZ [6] stands for the initial letters of Russian language that means "Theory of Inventive Problem Solving" in English. It was developed by G. Altshuller through the analysis of almost 40 million patents, under such a proposition that there certainly exist some techniques or rules for leading to a creative invention. It can be viewed as a method for invention and creative problem-solving. Today, many important principles and know-how are known for TRIZ. Among them, (1) 40 principles of invention; (2) 76 standard solutions of invention; (3) trends of technical system evolution; (4) contradiction matrix; (5) ARIZ (algorithm of inventive problem-solving) are popularly known, for examples.

TRIZ has successfully derived a specific guideline for dismissing technical contradictions based on such an assertion that overcoming the contradictions can bring about the evolution of technologies and systems. Its key process is to apply a separation principle for physical contradictions after discovering them embedded as root cause of the problem (a situation where there exists a certain system request in forward and reverse directions at the same time).

2.4 Kano Method

Kano method [7] makes it possible to define readily a customer-defined quality and derive how to improve product development and commerce. This function is realized by contrasting an objective evaluation for every attribute of product quality decided in terms of physical achievement to a subjective satisfaction for each component of quality. For this purpose, Kano method defines three categories of customer needs, namely, *must-have*, *linear satisfier* and *delighter* as shown in Fig. 2. The *must-have* requirement represents product features that customers expect to receive as a matter of course. When this *must-have* is not adequately addressed, the customer experiences dissatisfaction. The second type of need, *linear satisfier* (better), is characterized by such a relationship that customer satisfaction changes proportionally with the increase in product functionality. The third type of need, *delighter*, is not expected at first by the customer, but its fulfillment brings about great delight. Through revealing these differences imbedded in the quality,

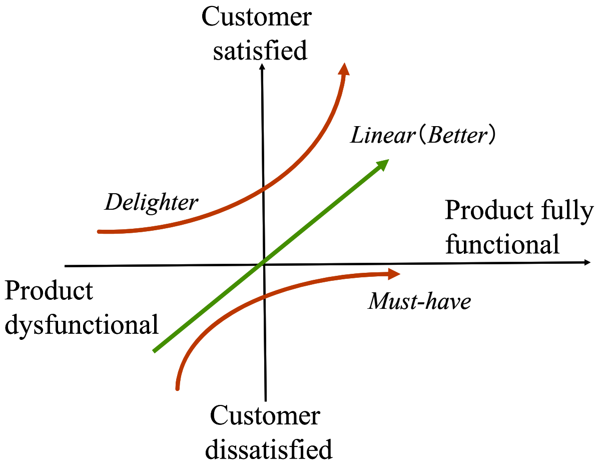


Fig. 2 Classification of quality attribute by Kano method (Kano diagram).

concepts in new product development should be finally decided.

2.5 Quality Function Deployment (QFD)

QFD [8] is a problem-solving method through deploying features of quality from abstract to concrete ones. It is particularly useful for fuzzy tasks often encountered at the initial stage like design and product planning. Using a matrix called quality request deployment table, whose rows show the target quality (required quality), whose columns the directly manageable elements (quality component), and whose cross element intensity between their mutual relationship, it can point out the elements with higher priority for the problem-solving. In other words, it can tell us what we must control first at the design stage.

2.6 Multi-objective Optimization Associated with Meta-Modeling

Generally, MOP is described by the triplet (x, f, X) as follows:

(p. 1) Min $f(x) = \{f_1(x), f_2(x), \ldots, f_N(x)\}$ subject to $x \in X$

where x is a decision variable vector; X a feasible region; and f an objective function vector, certain elements of which are in conflict and incommensurable with each other. In addition to the conventional physical-model-based approach, simulation-based approaches are being increasingly used to deal with complicated artifact designs. Such approach makes it possible to replace the time-consuming and labor-intensive tasks performed iteratively in a cycle of prototyping, inspection, evaluation, and improvement with integrated and structured computer-assisted procedures.

By applying such an idea to MOON^{2R}, we can restate (p. 1) as follows:

$$\text{(p. 2)} \quad \begin{array}{l} \text{Max} \quad V_{\text{RBF}}[\boldsymbol{f}(\boldsymbol{x}), \text{Meta\_}\boldsymbol{f}(\boldsymbol{x}); \boldsymbol{f}^{\text{R}})] \\ \quad\quad \text{subject to} \quad \boldsymbol{x} \in \{X, \text{Meta\_}X\} \end{array}$$

where X and Meta_X are feasible regions that can be described by a physical model and meta-model, respectively. V_{RBF} is an overall value function that integrates each objective function involving meta-model objective function Meta_$f(x)$, and f^{R} is an appropriate reference vector of the objective function that provides a basis for evaluation. In MOON^{2R}, V_{RBF} is modeled using a radial basis function (RBF) network (Orr [9]). A set of training data of the RBF network is collected by pair-wise comparison, which requires the decision maker (DM) to judge which pair is preferred and the extent to which the pair is preferred, by using linguistic statements, as in AHP (analytic hierarchy process) [10].

Eventually, (p. 2) refers to a typical single-objective problem (Refer to Fig. 3), to which a variety of conventional optimization methods can be applied. However, since the solution may be often unsatisfactory to the DM, mainly because of the low accuracy of the meta-models, an iterative procedure should be adopted to obtain the final solution. For this purpose, the meta-model can be updated by adding new data around the tentative solution and deleting the old data far from it. After rebuilding the system meta-model (Meta_X), the value function V_{RBF} can be updated by rebuilding the objective function meta-model (Meta_$f(x)$). As long as re-modeling is carried out in a consecutive and cooperative manner, such MOP can be solved not only effectively but also satisfactorily [11].

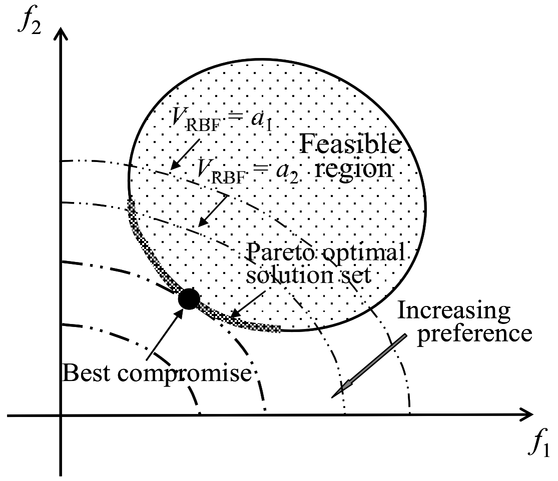


Fig. 3 Procedure for obtaining the solution in MOP.

3. Procedures from Problem Definition to Problem Formulation through a Case Study

Plastic sheets are widely used in chemical products and shipped in rolls from the factory. These rolls are shaped using a rolling machine such as that shown in Fig. 4. To increase the productivity while maintaining product quality, it is required to find the operating conditions through multi-objective optimization.

Taking this rolling process as a case study, the proposed steps will be explained concretely in the following. As presume of the following steps, an IDEF0 model (see Fig. 5) should be referred to grasp the necessary information. Looking at whole activities appeared there, the members can know the extent of the task and flow of the associated tasks, and eventually understand the intents regarding who, why, how and what of the present problem-solving associated with the correlated activities.

Step 1: Brainstorming to pick up elements

In terms of the foregoing triplet expression, we know the formulation of the optimal problem is referred to the selection of suitable objective functions, constraints and decision variables. Hence, through brainstorming, for example, list up the candidates of soft constraints (objective function) and (hard) constraints and decision variables. If it is unnecessary to distinguish the soft and hard constraints explicitly, we will simply call them as constraints hereinafter.

Fig. 4 Photograph of rolling machine.

In the present case, elements recalled from product value (customer claim) and technical problems (in house request) are described by using the Mind map.

Step 2: Complement from TRIZ

Just from the brainstorming among the engineers and/or customers, we might happen to miss certain fundamental elements that are substantial from technical reasons. To complement this incompleteness, we used an approach from TRIZ. As the intersection of row and column of the contradiction matrix, TRIZ indicates a referred item available for removing the contradiction from "40 invention principles". In the present plastic film rolling process, improvement of quality (especially shape) is of special interest. Production efficiency (reduction of unit production time) becomes a barrier to achieve this goal (contradiction between two goals). Then, by applying the invention principle against the contradiction between "12: Shape" and "32: Manufacturability", we finally selected "1: Segmentation" to remove the technical conflict between driving speed (efficiency) and wrinkle (quality). Narrowing the width of the film, we can avoid the wrinkle and control the roll temper easily (separation by space). From this, the following correlation was selected, respectively:

- Constraints: Wrinkle, side gap, roll temper;
- Decision variables: Driving speed, acceleration.

In addition, regarding the contradiction between shape and speed, we applied the rule from "35: Physical or chemical properties" and "18: Mechanical

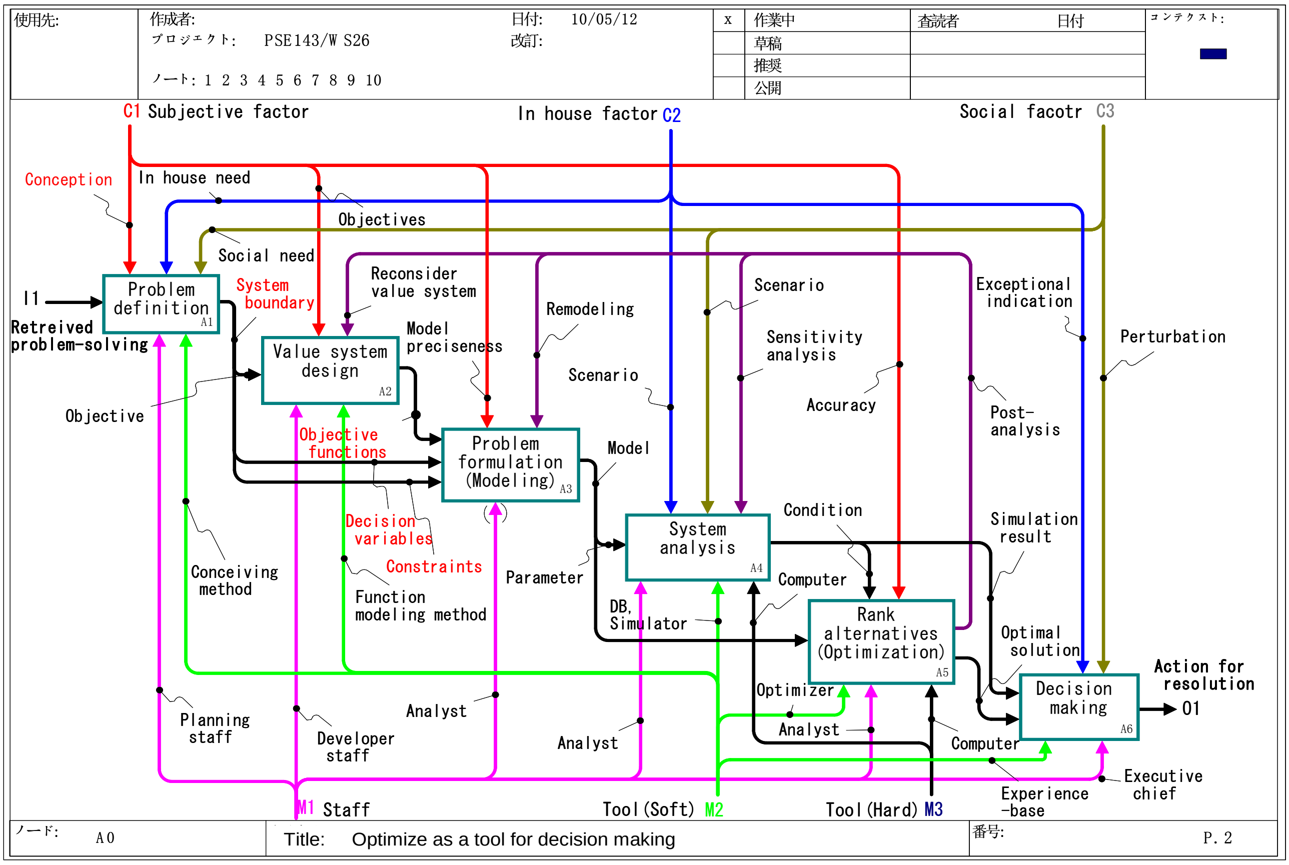


Fig. 5 IDEF0 model for general problem-solving.

vibration". Then, we imagined that spreading detachment of electrostatic charge at the film surface enables us to realize high speed driving without slip by virtue of surface antistatic effect (separation by situation). Hence, we chose below for the candidates of the constraints and decision variables, respectively:

- Constraints: Electrostatic charge, humidity;
- Decision variables: Driving speed, acceleration.

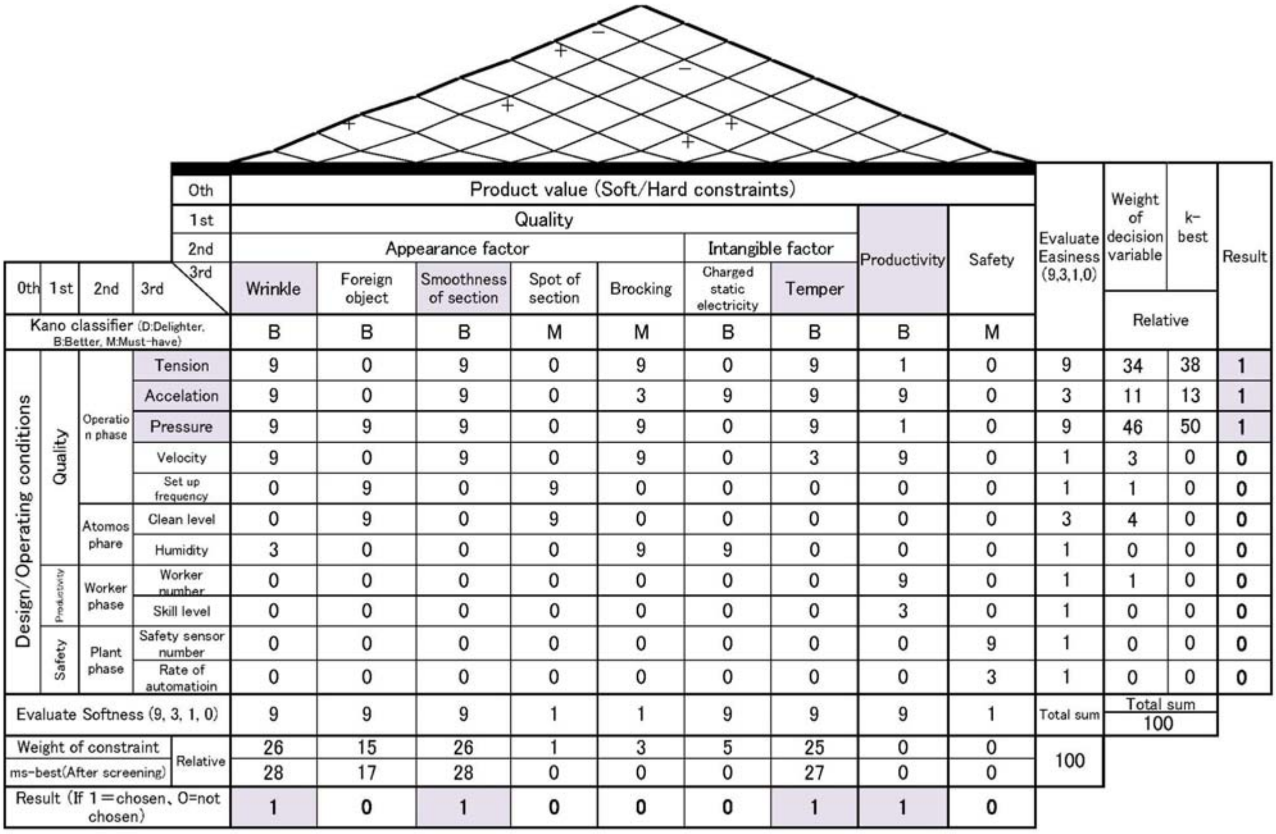
Step 3: QFD-alike procedure to estimate the strength of correlation

We move to the construction of correlation matrix similar to one used in QFD as shown in Fig. 6. From so far consideration, technical problems and product value (zero-th order factor) are concretely deployed both in row and column directions (from first to third). Following the similar procedure to QFD, strength of the relationship between the deployed product value (constraints) say *i*, and the deployed technical problems (decision variables), say *j*, Ω_{ij} is required to fill out numerically. In practice, it is represented for each pair (the intersection of rows and columns of the matrix) by using four scale metrics like {0: None, 1: A little, 3: Considerably, 9: Strongly}.

Step 4: Evaluate the two conditions regarding the constraint items

(1) According to the classifiers of KANO method, mark the characteristic of each constraints (M: Must-have, D: Delighter, B: Linear) in the row deciding degree of satisfaction.

(2) Evaluate the mutual relationship between each element of product value and mark it at the roof of the matrix using the following notation. That is, notation +/- means the case where there is trade-off between the two + proportional relation or - inverse proportional relation, respectively. Use the notations ++/--, similarly if there is a synergy effect (dramatic effect or deterioration when occurred at the same time).



			0th	Product value (Soft/Hard constraints)									Evaluate Easiness (9,3,1,0)	Weight of decision variable	k-best	Result
			1st	Quality							Productivity	Safety				
			2nd	Appearance factor					Intangible factor							
0th	1st	2nd	3rd	Wrinkle	Foreign object	Smoothness of section	Spot of section	Brocking	Charged static electricity	Temper						
Kano classifier (D:Delighter, B:Better, M:Must-have)				B	B	B	M	M	B	B	B	M		Relative		
Design/Operating conditions	Quality	Operation phase	Tension	9	0	9	0	9	0	9	1	0	9	34	38	1
			Accelation	9	0	9	0	3	9	9	9	0	3	11	13	1
			Pressure	9	9	9	0	9	0	9	1	0	9	46	50	1
			Velocity	9	0	9	0	9	0	3	9	0	1	3	0	0
			Set up frequency	0	9	0	9	0	0	0	0	0	1	1	0	0
		Atomos phare	Clean level	0	9	0	9	0	0	0	0	0	3	4	0	0
			Humidity	3	0	0	0	9	9	0	0	0	1	0	0	0
	Productivity	Worker phase	Worker number	0	0	0	0	0	0	0	9	0	1	1	0	0
			Skill level	0	0	0	0	0	0	0	3	0	1	0	0	0
	Safety	Plant phase	Safety sensor number	0	0	0	0	0	0	0	0	9	1	0	0	0
			Rate of automatioin	0	0	0	0	0	0	0	0	3	1	0	0	0
Evaluate Softness (9, 3, 1, 0)				9	9	9	1	1	9	9	9	1	Total sum	Total sum 100		
Weight of constraint			Relative	26	15	26	1	3	5	25	0	0	100			
ms-best(After screening)				28	17	28	0	0	0	27	0	0				
Result (If 1=chosen、0=not chosen)				1	0	1	0	0	0	1	1	0				

Fig. 6 An example of EXCEL sheet applied QFD-like technique.

Step 5: Identify the properties of constraint items

According to the standard if it has a {1: Reference value; 3: Aspiration level; 9: Maximization or minimization criteria}, fill out one of these metrics in the softness row for each element of constraints. Consequently, the constraint with larger value of this softness s_i is viewed more suitable as the objective function.

Step 6: Determine the degree of realization

Depending on the easiness of realization in the real production site, input the merit by using 4 stage scale, i.e., {0: Infeasible; 1: Difficult; 3: Possible, 9: Easy}. Larger values of u_i mean less troublesome application at actual operation.

Step 7: Calculate the contribution level

From so far steps, calculate the weights of the developed attributes based on the input values. Being different a bit from the original QFD, these calculations are taken place by weighting the feasibility index u_i for the decision variables and softness index s_i for the constraints. That is, when we denote contribution merit of the i-th decision variable by μ_i, it is calculated by $\mu_i' = u_i\sum_j\omega_{ij}s_j$ as letting ω_{ij} be the binary value decided from Eq. (1). Then, it is normalized as $\mu_i = \mu_i/\sum_i\mu_i$. On the other hand, contribution merit of the j-th constraint v_j is obtained from $v_j = v_j'/\sum_j v_j'$ where $v_j' = s_j\sum_i\omega_{ij}u_i$.

$$\omega_{ij} = \begin{cases} 1 & if \quad \Omega_{ij} = \{9,3,1\} \\ 0 & if \quad \Omega_{ij} = 0 \end{cases} \tag{1}$$

Step 8: Decide the candidates of decision variables and constraints

- Easy application implemented on Excel sheet

(1) No limitation on number of selection

Simply, we can choose k-bests of the candidate in terms of thus calculated merit for each attribute.

(2) There is bounds on selection number and when objective functions are decided first.

Step 1: From the t-bests of the softness index, select those by more than $\underline{t}$.

Step 2: Recalculate the feasibility index after deleting every column that were not selected at Step 1.

Step 3: Among the p-bests of the feasibility index, select those by more than $\underline{p}$.

Here ($\underline{t}$, t) and ($\underline{p}$, p) denote a pair of upper and lower bounds of selection for the objective functions and decision variables, respectively.

Similarly, we can imagine the procedure when the decision variables come first. In the developed Excel sheet, we realized software available for dealing with this opposite case where the decision variables are decided first. In addition, we select the remaining elements among the candidates as the constraints if they have the softness indexes above a certain threshold.

- Rigid application by solving IP (Integer Programming problem)

When we want to decide these more rigidly, we can solve the following optimization problems (0-1 programming problem).

$$\text{(p. 3)} \quad Max\ f(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} \mu_i \nu_j \,/\, (\sum_{i=1}^{m} x_i \sum_{j=1}^{n} y_j)$$

$$\text{subject to} \quad \begin{cases} \mu_i = u_i \sum_{i=1}^{n} \omega_{ik} x_i y_k s_k \quad (i = 1, \cdots, m) \\ \nu_j = s_j \sum_{k=1}^{m} \omega_{kj} x_k y_j u_k \quad (j = 1, \cdots, n) \\ \underline{p} \leqq \sum_{i=1}^{m} x_i \le p \\ \underline{t} \leqq \sum_{j=1}^{n} y_j \leqq t \\ x_i, y_i \in \{0,1\} \end{cases}$$

where x_i is a variable that takes one if it is selected as the decision variable, and otherwise zero. Similarly, y_j takes one if it is selected as the constraint, and otherwise zero. Here, the objective function of this problem represents an average strength of correlation weighted by the feasibility and softness indexes over the constraints and decision variables to be selected.

We solved (p. 3) by using commercial software called LINGO [12]. The same results as shown in Table 1 were obtained both from Excel software and IP solution. There, if someone belonging to the "*must-have*" in KANO classifier is selected or someone with "*linear*" was not chosen as one of the objective functions, we must reconsider the result as supposed easily. The Mind map shown in Fig. 7 is also available to share the final result among the members of the team.

At the next step of this study, multi-objective optimization problem was solved by MOON^{2R}. Among the four objective functions selected by the proposed procedure, three of them are integrated as a quality of product as mentioned below and the rest remained as the productivity. On the other hand, the decision variables are those of the results, i.e., tension, acceleration and pushing pressure.

4. Decision Making through MOP regarding Plastic-Sheet-Rolling Machine Problem

4.1 Method for Evaluating Quality

Although quality is an important factor in the evaluation of the design and operation of artifacts, it has scarcely been evaluated in a definite manner. This is partly because the evaluation of quality involves qualitative and fuzzy attributes related to appearance, emotion, and aesthetics, i.e., attributes related to Kansei. To cope with this problem, we propose the use of a decision making method based on the grey theory (especially, the grey situation decision model) together with the classifiers of the Kano method. The grey theory [13] is designed to deal with the uncertainty in a system and is considered to be more general than the fuzzy set theory. On the other hand, the Kano method

Table 1 Result of selection.

Objective function	Decision variable	Constraint
Wrinkle	Tension	Blocking
Temper	Acceleration	Foreign object
Smoothness	Pushing pressure	Charged static electricity
Productivity		Spot of section

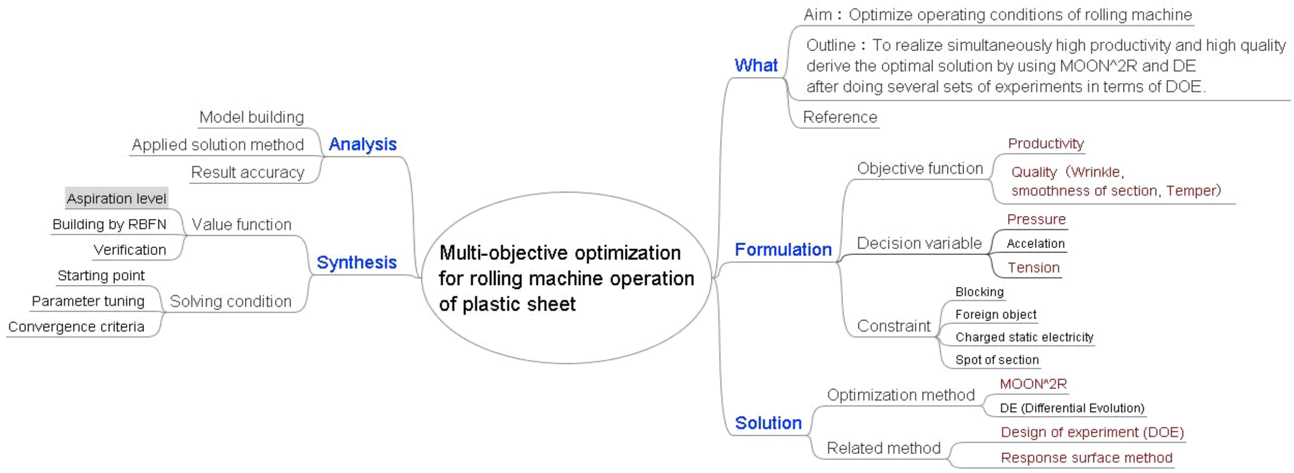

Multi-objective optimization for rolling machine operation of plastic sheet
What
Aim：Optimize operating conditions of rolling machine
Outline：To realize simultaneously high productivity and high quality derive the optimal solution by using MOON^2R and DE after doing several sets of experiments in terms of DOE.
Reference
Formulation
Objective function
Productivity
Quality（Wrinkle, smoothness of section, Temper)
Decision variable
Pressure
Accelation
Tension
Constraint
Blocking
Foreign object
Charged static electricity
Spot of section
Solution
Optimization method
MOON^2R
DE (Differential Evolution)
Related method
Design of experiment (DOE)
Response surface method
Analysis
Model building
Applied solution method
Result accuracy
Synthesis
Value function
Aspiration level
Building by RBFN
Verification
Solving condition
Starting point
Parameter tuning
Convergence criteria

Fig. 7 Mind map depicted to share the final result.

can be used to define customer-defined quality and to identify the manner in which product development and commerce can be improved.

The basis of the grey situation decision model classifies an attribute into an upper effect measure, a lower effect measure, and a medium effect measure. Then, the effectiveness of the attribute is evaluated as $z_i/z_{\max}$, $z_{\min}/z_i$, and $z_i^{\mathrm{R}}/(z_i^{\mathrm{R}}+|z_i - z_i^{\mathrm{R}}|)$, for the upper, lower, and medium effect measures, respectively. Here, $z_{\max}$, $z_{\min}$, and z_i^{R} denote the maximum, minimum, and target effects of the i-th attribute, respectively.

Moreover, for quality evaluation, we must consider the existence of the threshold (z_i^{b}, z_i^{B}), beyond which the product becomes irregular or does not meet the design specifications. In addition, it is suitable to take into account the classifiers of the Kano method for quality evaluation. After modifying the grey situation decision model to take into account the classifiers of the Kano method, we define a white function or the level of quality for each attribute as follows:

For the medium effect measure,

$$Q_i = \begin{cases} \left(\dfrac{z_i - z_i^{\mathrm{b}}}{z_i^{\mathrm{R}} - z_i^{\mathrm{b}}} \right)^q, \text{if} \quad z_i^{\mathrm{b}} \le z_i \le z_i^{\mathrm{R}} \\ \left(\dfrac{z_i - z_i^{\mathrm{B}}}{z_i^{\mathrm{R}} - z_i^{\mathrm{B}}} \right)^q, \text{if} \quad z_i^{\mathrm{R}} \le z_i \le z_i^{\mathrm{B}} \\ 0, \qquad\qquad \text{if} \quad z_i \le z_i^{\mathrm{b}},\ z_i \ge z_i^{\mathrm{B}} \end{cases} \tag{2}$$

For the upper and lower effect measures,

(I) Upper:

$$Q_i = \begin{cases} 1, & \text{if} \quad z_i \ge z_i^{\mathrm{R}} \\ \left(\dfrac{z_i - z_i^{\mathrm{B}}}{z_i^{\mathrm{R}} - z_i^{\mathrm{B}}} \right)^q, & \text{if} \quad z_i^{\mathrm{B}} \le z_i \le z_i^{\mathrm{R}} \\ 0, & \text{if} \quad z_i \le z_i^{\mathrm{B}} \end{cases} \tag{3-a}$$

(II) Lower effect:

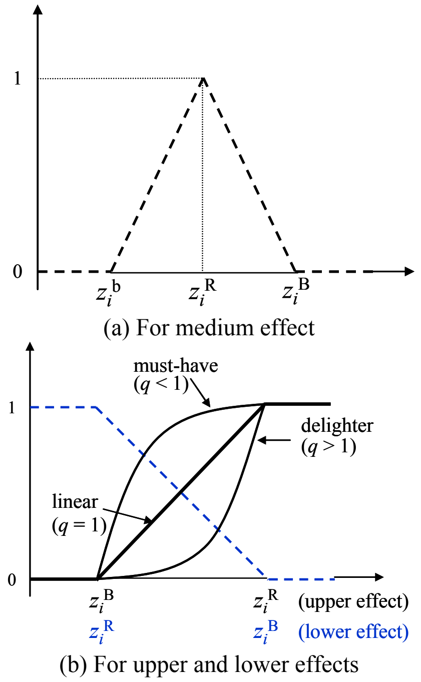
$$Q_i = \begin{cases} 1, & \text{if} \quad z_i \le z_i^{\mathrm{R}} \\ \left(\dfrac{z_i - z_i^{\mathrm{B}}}{z_i^{\mathrm{R}} - z_i^{\mathrm{B}}} \right)^q, & \text{if} \quad z_i^{\mathrm{R}} \le z_i \le z_i^{\mathrm{B}} \\ 0, & \text{if} \quad z_i \ge z_i^{\mathrm{B}} \end{cases} \tag{3-b}$$

Here, the parameter q is greater than 1 for delighter, equal to 1 for linear satisfier, and less than 1 for must-have. Only for the upper effect, we illustrate the modified white functions by imposing them to the original ones (q=1; broken lines) in Fig. 8 (b).

Eventually, we evaluate quality as a weighted sum of these measures over the attributes, i.e., $Q = \sum_{i=1}^{N} w_i Q_i$, where w_i denotes the weight representing the relative importance of attribute i. We can decide the weights by applying AHP, for an example.

4.2 Software Implementation

To facilitate easy and wide application of the MOP to a manufacturing process, we propose the implementation of the MOON$^{2\mathrm{R}}$ algorithm in Microsoft Excel [14]. The spreadsheet involves the



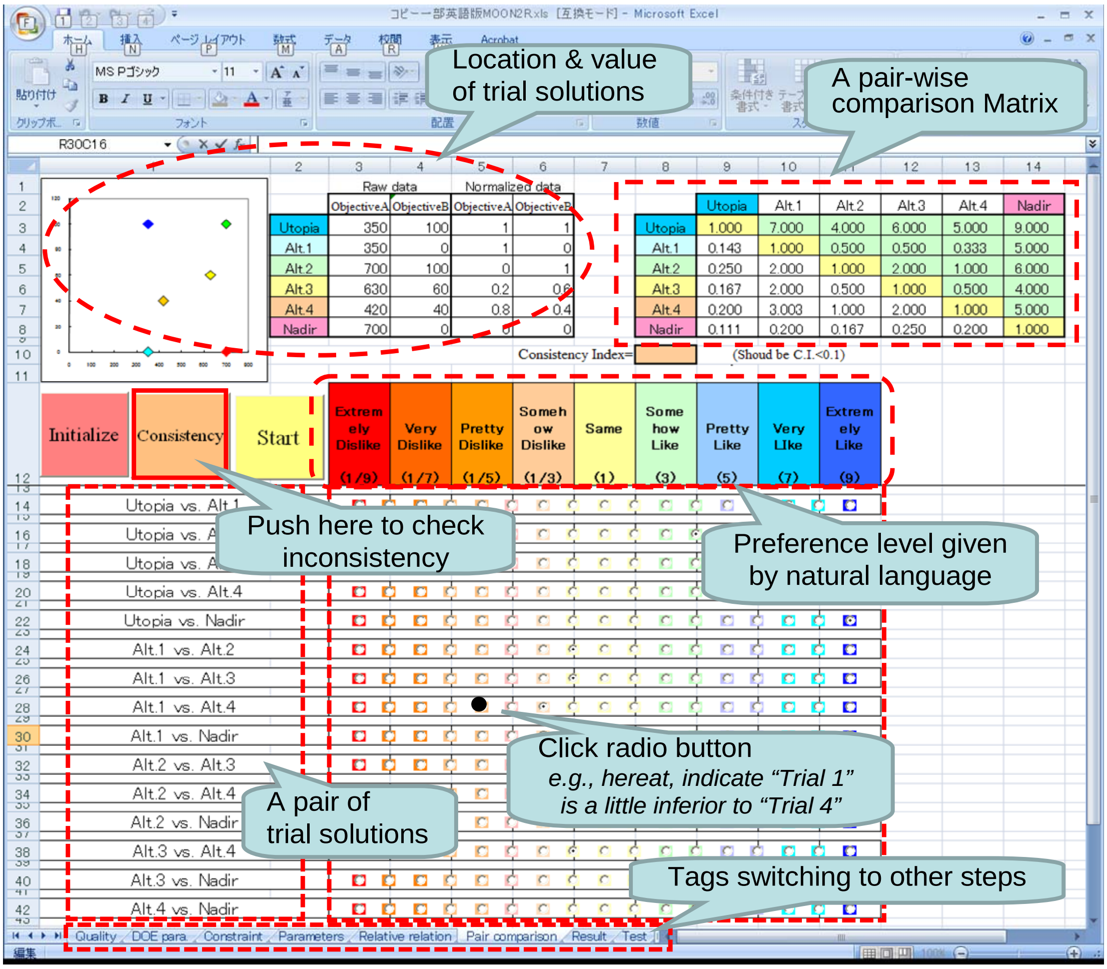
(a) For medium effect

(b) For upper and lower effects

Fig. 8 Original and modified white functions.

problem definition phase, meta-modeling phase, pair-wise comparison phase, optimization phase, result phase, and supplementary phase to evaluate a qualitative objective such as quality. The algorithm of each phase is programmed by using VA (Visual Basic) and is interconnected by spreadsheet macros.

In Fig. 9, we show the pair-wise comparison phase of the sheet. Here, the DM can provide his/her answers to the trial solutions, whose locations and their real values are shown in the upper left corner (within the broken-line ellipse), by clicking the radio buttons depending on the preference level described by natural language (indicated by the dashed rectangle). After the preference levels are all set, the pair-wise comparison matrix shown in the upper right corner is derived. The consistency of such pair-wise comparisons can be examined by pressing the button provided in the same



	Raw data		Normalized data	
	ObjectiveA	ObjectiveB	ObjectiveA	ObjectiveB
Utopia	350	100	1	1
Alt.1	350	0	1	0
Alt.2	700	100	0	1
Alt.3	630	60	0.2	0.6
Alt.4	420	40	0.8	0.4
Nadir	700	0	0	0

	Utopia	Alt.1	Alt.2	Alt.3	Alt.4	Nadir
Utopia	1.000	7.000	4.000	6.000	5.000	9.000
Alt.1	0.143	1.000	0.500	0.500	0.333	5.000
Alt.2	0.250	2.000	1.000	2.000	1.000	6.000
Alt.3	0.167	2.000	0.500	1.000	0.500	4.000
Alt.4	0.200	3.003	1.000	2.000	1.000	5.000
Nadir	0.111	0.200	0.167	0.250	0.200	1.000

Consistency Index= (Shoud be C.I.<0.1)

Utopia vs. Alt.1
Utopia vs. A
Utopia vs. A
Utopia vs. Alt.4
Utopia vs. Nadir
Alt.1 vs. Alt.2
Alt.1 vs. Alt.3
Alt.1 vs. Alt.4
Alt.1 vs. Nadir
Alt.2 vs. Alt.3
Alt.2 vs. Alt.4
Alt.2 vs. Nadir
Alt.3 vs. Alt.4
Alt.3 vs. Nadir
Alt.4 vs. Nadir

Location & value of trial solutions

A pair-wise comparison Matrix

Push here to check inconsistency

Preference level given by natural language

Click radio button *e.g., hereat, indicate "Trial 1" is a little inferior to "Trial 4"*

A pair of trial solutions

Tags switching to other steps

Fig. 9 Sheet for the pair-wise comparison phase of the spread sheet.

sheet (indicated by the solid square). In the case of poor consistency, the system suggests that the DM revise the pair-wise comparison.

Freely navigating over different sheets depending on the requirements of their task, users can successfully accomplish their decision making. Hence, users can readily begin with the decision making process by using the MOP and repeat the procedure until he/she is satisfied with the result.

Implementation of the software in Excel will ensure portability of the tool as well as facilitate the easy application of the spreadsheet to other problems after slight modifications.

4.3 Numerical Experiment and Results

From so far consideration, present concern is described as the following bi-objective optimization problem.

(p. 3) Maximize $\{f_1(x)$: Quality, $f_2(x)$: Productivity$\}$

subject to $\underline{x_i} \le x_i \le \overline{x_i}, (i = 1,2,3)$

with respect to $\{x_1$: extension force of rolling [N/m], x_2: acceleration of rolling speed [m/min^2], x_3: press pressure of rolling machine [N/m^2]$\}$, where $\underline{x_i}$ and $\overline{x_i}$ denote the lower and upper bounds for each operating condition, respectively. Moreover, productivity is expressed as the reciprocal of the unit production rate defined by

$$f_2 = \left(\frac{P}{v} + \frac{v}{x_2} \right)^{-1} \tag{4}$$

where v denotes the velocity, and P is a constant corresponding to the length of the sheet to be rolled.

Quality is evaluated on the basis of three attributes of the roll, namely, the wrinkle, temper, and smoothness (refer to Fig. 10). However, since direct evaluation of the quality from these attributes is difficult, we apply the procedure described above. We assume that each property associated with the grey theory and the Kano classifiers is described by the following pairs: (lower effect, must-have) for the wrinkle length, (medium effect, linear satisfier) for the temper of the rolled sheet, and (lower effect, must-have) for the smoothness of the section; accordingly, we adopt the parameters listed in Table 2.

To obtain a response surface model [15] of the quality, we set three levels and define a set of experimental conditions, by applying the $L_9(3^4)$ orthogonal array of DOE (design of experiment). Here L indicates the Latin square, and 3 stands for the number of levels. Then, the number of design points or number of rows is given as $3^2 = 9$. Moreover, the column is calculated as (9 - 1)/2 = 4 (See Futami and Nishi [16] for details). Table 3 presents the results of this experiment. Using these results, we derive the response surface model or meta-model of quality (f_1) as an RBF network model. In the last column of the table, we also show the subjective value supplied separately by the expert operator (DM). Here, we should note that the results of both evaluations are similar except in the first run. Even for this first run, the DM admitted the score was too high when reviewing all the evaluations in the post-analysis.

In Fig. 11, we show an example of RSM (Response Surface Model) for quality when the press pressure is set at 200 N/m.

In the value-function-modeling phase, after setting the utopia F^u and nadir F^n, the system randomly

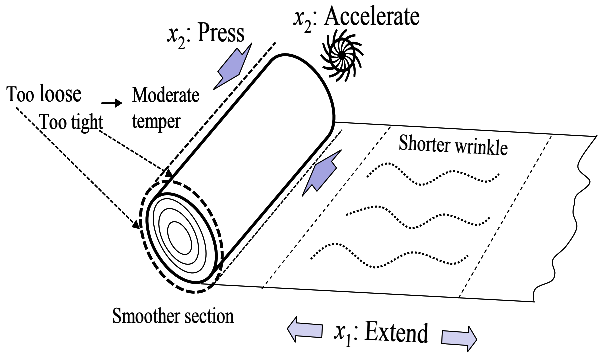


Fig. 10 Schematic explanation of the formulated problem.

Table 2 Parameters used for evaluating Q_i.

Attribute	x_i^b	x_i^R	x_i^B	q
Wrinkle (low, must)	-	0	80	2
Temper (med, linear)	580	680	780	1
Smoothness (low, must)	-	0	40	2

Table 3 Experimental results and evaluation of quality.

	Factor of DOE (real value)			Observed attribute value			Evaluated quality (f_1)	
Run #	Extension force [N/m^2]	Acceleration [m/min^2]	Press pressure [N/m^2]	Wrinkle [m]	Temper [-]	Smoothness [mm]	Metric(Q)	Subjective (DM)
1	-1 (40)	-1 (100)	-1 (100)	95	592	1	28	40
2	-1	0 (350)	0 (200)	0	642	0	88	80
3	-1	1 (600)	1 (300)	50	685	0	72	75
4	0 (60)	-1	0	70	742	0	57	65
5	0	0	1	0	729	0	84	75
6	0	1	-1	0	671	1	83	75
7	1 (80)	-1	1	100	761	0	43	55
8	1	0	-1	120	705	0	59	55
9	1	1	0	50	732	0	58	60

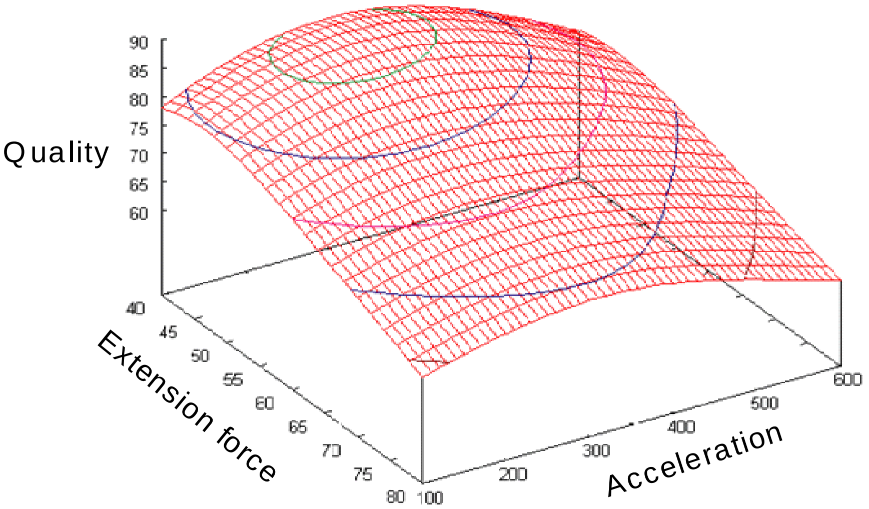


Fig. 11 An example of the response surface model of quality.

generates four trial solutions within the objective search space enclosed in a rectangle defined by these two points, as shown in Fig. 12. Then, the system asks the DM to reply his/her preference for every pair of trial solutions through pair-wise comparison. Eventually, we obtain a pair-wise comparison matrix like Table 4 into which the DM's preferred information is integrated. For example, the *i-j* element of the matrix, say, $a_{ij} = 9$ implies that F^i is extremely preferable to F^j (for details, refer to the original paper [3]). Here, we need address the problems related to inconsistencies in the pair-wise comparison, if necessary. Using the results thus obtained, we can train another RBF network to identify the value function as V_{RBF} (Meta_f_1, f_2; f^R), where Meta_f_1 represents the quality meta-model derived from the abovementioned procedure.

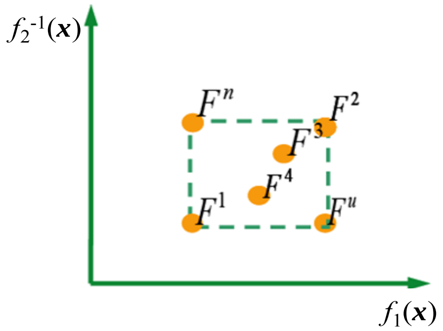


Fig. 12 Location of trial solutions.

Now, we can restate the above problem (p. 3) as follows[1]:

[1] Presently, f^R is first given as a middle point in the normalized search space, i.e., (0.5, 0.5). Then, it is replaced with the foregoing result, i.e., $f^R = f^k$, until the condition $\| f^{k+1} - f^k \| \le \varepsilon$ is satisfied. Here, k denotes the number of iterations, and ε is a small positive value.

Table 4 Pair-wise comparison matrix.

	F^U	F^1	F^2	F^3	F^4	F^N
F^U	1	8	7	5	2	9
F^1	1/8	1	1/2	1/5	1/7	2
F^2	1/7	2	1	1/3	1/7	3
F^3	1/5	5	3	1	1/4	5
F^4	1/2	7	7	4	1	8
F^N	1/9	1/2	1/3	1/5	1/8	1

(p. 4) $Max\ V_{RBF}(\text{Meta}\_f_1(\boldsymbol{x}), f_2(\boldsymbol{x}); \boldsymbol{f}^R)$

subject to $\underline{x}_i \le x_i \le \overline{x}_i,\ (i = 1,2,3)$

Since every value function can be evaluated for the arbitrary decision variable, we adopt DE (differential evolution) in the optimization phase of the implemented spreadsheet. DE, developed by Storn and Price [17], is a powerful real number coding version of GA (genetic algorithm).

When the procedures embedded in the Excel spreadsheet are adopted, the MOP can be readily solved only by setting a few tuning parameters and the initial conditions for optimization. Table 5 presents the results obtained by using the software. Since the DM feels that the first result can be further improved, a second round of the procedure is carried out to improve the solution; in this case, the meta-model is modified in accordance with the first solution. This is accomplished by adding five experimental points and deleting six experimental points, as shown in Table 6, by following a previously reported method [14]. By selecting an appropriate Excel sheet for this task, users can easily revise the tentative solution. Notably, the second solution in Table 5 shows the improvement made to the first solution as well as the conventional decision based on the experience and intuition of the operator.

5. Conclusions

In conventional optimization studies, major interests have been paid at theories and solution techniques. In contrast, in this paper, first, we have proposed a new attempt that tries to comprehensively consider optimization problem itself. Applying various methods for recognition and conception that have been used separately so far, we give a novel procedure to formulate optimization problems itself.

Since such efforts will result in solving the multi-objective optimization problem (MOP) in an ill-posed circumstance, then we focused on an MOP that is amenable even for such case. Being especially interested in a qualitative measure such as quality, we developed a method for numerically expressing measures of the quality of a product. For this purpose, we borrowed the tenets of the grey theory in combination with the classifiers of the Kano method. Moreover, we implemented the solution algorithm for an MOP in a Microsoft Excel spreadsheet in order to facilitate the practical application of the spreadsheet to real-world manufacturing.

To demonstrate the idea, we carried out a case study on a plastic-sheet rolling machine. Numerical experiments revealed that the proposed procedure can

Table 5 Comparison of compromise solution with the conventional solution.

Decision		f_1 [-]	f_2^{-1}[min]	V_{RBF}	x_1: Extension force [N/m^2]	x_2: Acceleration [m/min^2]	x_3: Pressure [N/m^2]
Conventional		85	7.42	-	70	400	250
This work	1st	99	6.64	0.16	44	244	140
	2nd	96	5.89	0.33	42	447	172

Table 6 Added and deleted points for meta-model updating.

	Extension force [N/m^2]	Acceleration [m/min^2]	Press pressure [N/m^2]
1 Added	43	279	153
2 Added	75	429	105
3 Added	75	249	177
4 Added	40	429	177
5 Added	40	249	105
1 Deleted	40 (-1)*	100 (-1)	100 (-1)
2 Deleted	40 (-1)	600 (1)	300 (1)
3 Deleted	60 (0)	100 (-1)	200 (0)
4 Deleted	60 (0)	350 (0)	300 (1)
5 Deleted	80 (1)	600 (1)	200 (0)
6 Deleted	80 (1)	600 (1)	200 (0)

*Real value (factor)

be used for problem solving by mutual collaboration between parties with different qualifications, for example, engineers and field operators. Finally, from the remarkable advances made to computer simulation techniques, we state the proposed meta-modeling-based approach is useful for solving flexible MOPs in multidisciplinary decision-making environments.

Acknowledgments

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Exploratory Research, 23656067, 2011-2012, Japan.

References

[1] Y. Shimizu, Y. Kato, T. Kariyahara, Development of decision support system through multi-objective optimization for ill-posed problems, Computer Aided Chemical Engineering 28 (2010) 841-846.

[2] Y. Shimizu, An integrated systems approach for formulating engineering optimization problems, in: Proc. of the 2nd International Conference on Engineering Optimization, Lisbon, Portugal, 2010, 01083.

[3] Y. Shimizu, A. Kawada, Multi-objective optimization in terms of soft computing, Trans. Society Instrument and Control Engineers 38 (2002) 974-980.

[4] D.A. Marca, C.L. McGowan, IDEF0/SADT, Eclectic Sol. Corp., San Diego, USA, 1988.

[5] T. Buzan, K. Chang, B. Buzan, Mind Map, Diamond Company, 2005.

[6] R. Fey, V.E.I. Rivin, The science of innovation, in: Y. Hatamura, et al. (Eds., Trans.), Nikkann Kogyo Shinbunsya, 1997.

[7] N. Kano, N. Seraku, F. Takahashi, S. Tsuji, Attractive quality and must-be quality, Journal of the Japanese Society for Quality Control (in Japanese) 31 (4) (1984) 147-156.

[8] S. Mizuno, Y. Akao, Quality Function Deployment, Nikka Giren Publisher, 1978.

[9] M. Orr, Introduction to radial basis function networks, Time 302 (1996) 1-67.

[10] T.L. Saaty, The Analytic Hierarchy Process, McGraw-Hill, 1980.

[11] Y. Shimizu, K. Miura, J-K. Yoo, Y. Tanaka, A progressive approach for multi-objective design through inter-related modeling of value system and meta-model, JSME (in Japanese) C-71 (2005) 296-303.

[12] LINGO, User's Guide, LINDO System Inc., Chicago, USA, 1995.

[13] J. Deng, Introduction to grey system theory, Journal of Grey System 1 (1) (1989) 1-24.

[14] Y. Shimizu, T. Nomachi, Integrated product design through multi-objective optimization incorporated with meta-modeling technique, J. Chem. Eng. Japan 41 (2008) 1068-1074.

[15] R.H. Myers, D.C. Montgomery, Response Surface Methodology, John Wiley & Sons, 2002.

[16] R. Futami, T. Nishi, Design of Experiment for Problem Solving (Kadaikaiketsu notameno Zikkenkeikakuhou) (in Japanese), JUSE Press, 2002.

[17] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. of Global Optim. 11 (1997) 341-359.

Computer Technology and Application 2 (2011) 926-929

Effective Channel Length Degradation under Hot-Carrier Stressing

Anucha Ruangphanit[1], Kunagone Kiddee[2], Rangson Muanghlua[2], Surasak Niemcharoen[2] and Amporn Poyai[1]

1. Thai Microelectronics Center (TMEC), National Electronics and Computer Technology Center (NECTEC), Thailand

2. Department of Electronics Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand

Received: August 22, 2011 / Accepted: September 02, 2011 / Published: November 25, 2011.

Abstract: This article describes the effective channel length degradation under hot carrier stressing. The extraction is based on the I_{DS}-V_{GS} characteristics by maximum transconductance (maximum slope of I_{DS} & V_{GS}) in the linear region. The transconductance characteristics are determine for the several devices of difference drawn channel length. The effective channel length of submicron LDD (Lightly Doped Drain) NMOSFETs (Metal Oxide Semiconductor Field Effect Transistor) under hot carrier stressing was measured at the stress time varying from zero to 10,000 seconds. It is shown that the effective channel length was increased with time. This is caused by charges trapping in the oxide during stress. The increased of effective channel length (ΔL_{eff}) is seem to be increased sharply as the gate channel length is decrease.

Key words: NMOSFETs (metal oxide semiconductor field effect transistor), effective channel length, hot carrier stressing.

1. Introduction

The effective channel length of Metal Oxide Semiconductor Field Effect Transistor (MOSFET) is a very useful parameter for process monitoring, circuit design and circuit simulation as well as for the technology fabrication. The methodology proposed here for extracting the effective channel length is based on the *I-V* characteristics by maximum transconductance (maximum slope of I_{DS} & V_{GS}). The procedure for extracting for effective channel length is very simple, reliable and independent of some parameter such as threshold voltage and series resistance. When the device feature size scaling down, hot carrier induced degradation has become a serious problem. Due to the hot carrier injection into the gate oxide, threshold voltage (V_{TH}) and transconductance (g_m) of the device can shift substantially. Depending upon the degree of the V_{TH} and g_m shift, the speed of device will degrade and finally the device will not function. An origin of degradation is the high electric field induced in the channel region. In order to reduce this electric field, LDD (Lightly Doped Drain) technology is used in transistor structures which reduce the amount of damage, and consequently increase devices life time [1-3]. In this paper, we study the effect of hot carrier stressing on the effective channel length of NMOSFETs by changing the stress time from zero to 10,000 seconds. The effective channel length before and after hot carrier degradation were measured and investigated to the understanding well the behaviors nature of hot carrier degradation in channel length degradation of LDD NMOSFETs. The effects of hot carrier on effective mobility and parasitic series resistance will discuss later on.

2. Backgrounds

The hot carrier injection creates new interface states and leads to the trapping of carriers in the gate or the

Corresponding author: Anucha Ruangphanit, electronics engineering, master degree in engineering, research fields: test chip design, sub micron CMOS process design and development, device measurement and extraction. E-mail: cmoslee_6@hotmail.com.

sidewall oxide. Fig. 1 shows the hot-carrier generation and components which result from such generation. The hot-carrier effect damages the gate oxide and/or the Si-SiO$_2$ interface. In our model, L_{eff} is defined as the metallurgical junction spacing between the source and drain junctions and is thus gate bias independent. This is consistent with and inherently simplifies the use of MOSFET's models in circuit and reliability simulation. The linear drain current is modeled as

$$I_{DS} = \mu_{eff} C_{ox} \frac{W_{eff}}{L_{eff}} \left(V_{GS} - V_{TH} - \frac{V_{DS}}{2} \right) V_{DS} \tag{1}$$

The transconductance define by $g_m = \frac{\partial I_{DS}}{V_{GS}}$ at constant V_{DS}.

$$g_m = \mu_{eff} C_{ox} \frac{W_{eff}}{L_{eff}} V_{DS} = K_{lin} V_{DS} \tag{2}$$

$$K_{lin} = \mu_{eff} C_{ox} \frac{W_{eff}}{L_{eff}} \tag{3}$$

where K_{lin} is a device transconductance in linear region.

$$\mu_{eff} = \frac{\mu_s}{1 + \frac{\mu_s \cdot V_{DS,sat}}{v_{\max} \cdot L_{eff}}} \tag{4}$$

$$\mu_s = \frac{\mu_0}{1 + \theta (V_{GS} - V_{TH})} \tag{5}$$

$$W_{eff} = W - \Delta W = W - 2WD \tag{6}$$

$$L_{eff} = L - \Delta L = L - 2LD \tag{7}$$

where μ_{eff} is the effective mobility, μ_o is the zero bias low field mobility, μ_s is the surface mobility due to vertical field, $V_{DS,sat}$ is the saturation drain voltage ($V_{DS,sat}$=V_{GS}-V_{TH}), v_{max} is the maximum carrier velocity, θ (theta) is the gate field induced mobility degradation parameter, W_{eff} is the effective channel gate width, L_{eff} is the effective channel gate length, LD and WD are parameters describe the reduction of the channel length and channel width of the device due to Source/Drain diffuse in to a channel. C_{ox} is the gate oxide capacitance per unit area. The transconductance can be easily extracted by taking the numerical derivative of the drain current with respect to the gate bias using the measured I_{DS} & V_{GS} characteristics. The transconductance is usually taken as the maximum value of this derivative and sometimes referred to as the maximum transconductance.

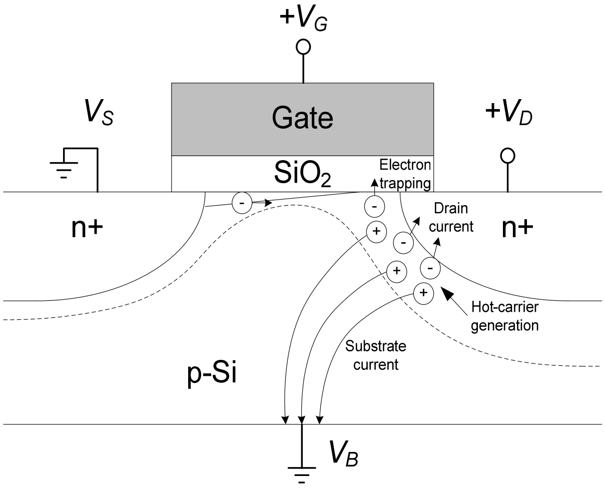


Fig. 1 Hot-carrier generation and components in NMOSFETs.

3. Experimental Details

The devices were measured by using a semiconductor parameter analyzer HP4156B, with a PC personal computer as the central controller. The measurement accuracy is 0.02% for voltage and 0.06% for current within the measurement range. The NMOSFETs with the gate width of 40 μm and the gate length (L_g) of 0.8 μm had the effective channel length (L_{eff}) of 0.58 μm (lateral diffusion = 0.11 μm/side).The hot carrier stressing was performed on a testing NMOSFETs having channel lengths of 0.6, 0.8, 1.0 and 1.2 μm respectively and a channel width of 40 μm, gate oxide thickness were 25 nm. The testing devices were stressed with the bias condition of drain voltage V_{DS} = 5 V and gate voltage at maximum substrate current V_{GS} = 2.2 V to induce maximum hot carrier degradation [4]. After the initial device characterization, measurements were made at 100, 500, 1,000, 5,000 and 10,000 seconds of hot carrier stress. The effective channel lengths were measured before and after the hot carrier stress. By plotting the L_{mask} versus 1/K_{lin} of a testing devices. The intercept of L_{mask} axis is ΔL. The channel length reduction ΔL has evaluate by through the fitting curve of a straight line of the plot of L_{mask} versus $1/K_{lin}$, the intercept of L_{mask} axis is ΔL. where the ΔL extracted for several device with difference the channel length, as shown in Fig. 2. Fig. 3 shows the I_{DS} & V_{GS} curves of testing device with various stress time. Fig. 4 shows transconductance reduction of devices with before and after stress time.

4. Results and Discussion

The effective channel lengths (L_{eff}) before the hot carrier stress of the testing device having channel length of 0.6, 0.8, 1.0, 1.2 and 3.0 um were 0.37, 0.57, 0.77, 0.97 and 2.77 μm respectively. Fig. 5 shows the effective channel length versus the hot carrier stress time with various different channel gate lengths. After the hot carrier stress, the effective channel length (L_{eff}) were increased as the hot carrier stress time increased. The increase of the effective channel length of NMOSFETs by hot carrier degradation can be explained as follows [3]. Because of the barrier height for the injection of electron is much lower than that for the injection of hole. Then, electrons are injected into the gate oxide and leaved a positive charge in lightly doped drain region. These charges invert the lightly doped drain region of n-type into p-type. Which causes an increase of a parasitic series resistance and the mobility degradation also make an increase of the effective channel length as shown in Fig. 6. In Fig. 7, as

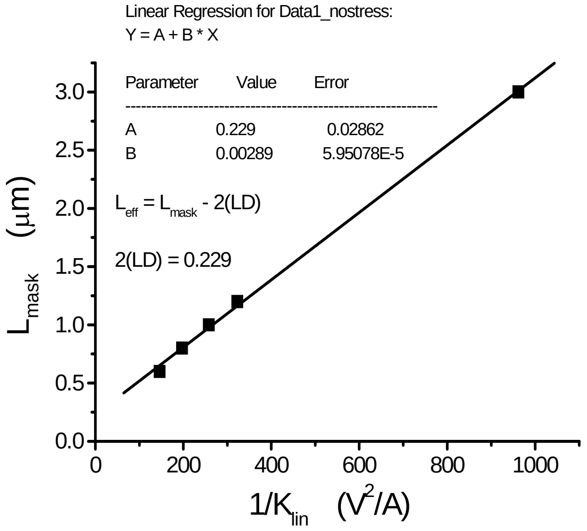


Fig. 2 L_{mask} **versus** $1/K_{lin}$ **with various devices channel length at no stress condition.**

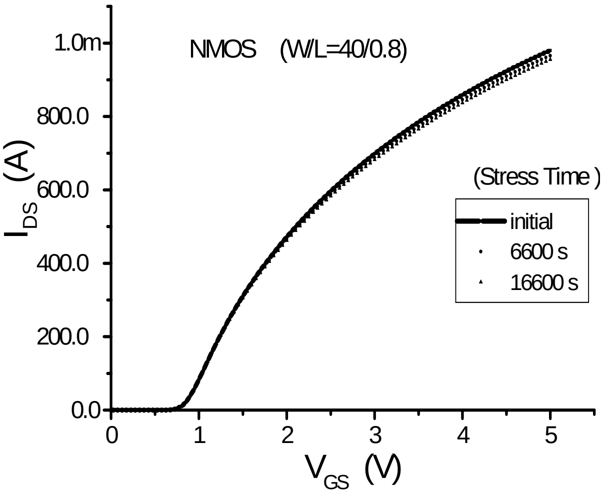


Fig. 3 I_{DS} **&** V_{GS} **curves of with various stress time.**

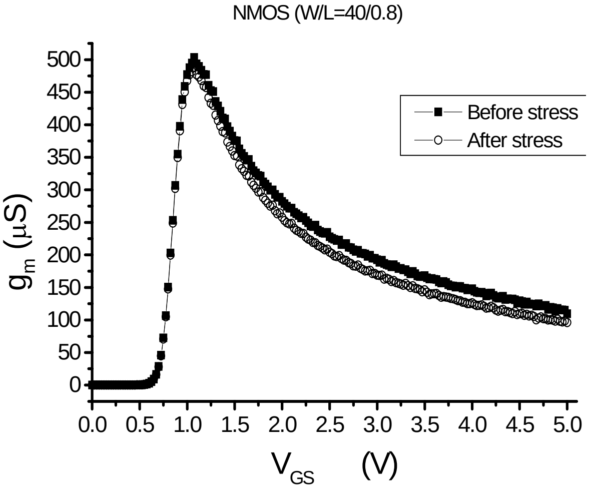


Fig. 4 g_m **&**V_{GS} **curves with before and after stress time.**

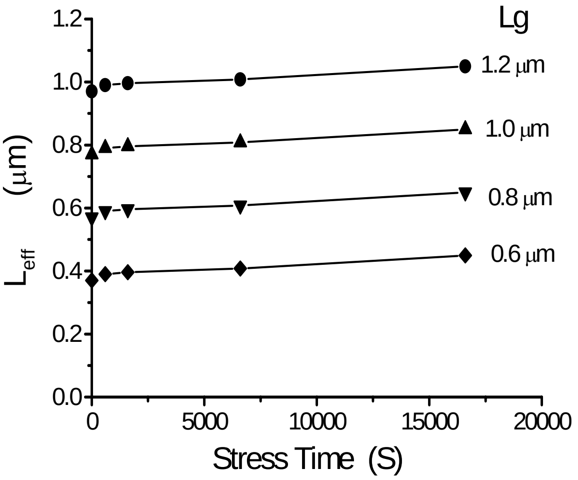


Fig. 5 Effective channel length versus the hot carrier stress time with various channel gate length (*Lg*).

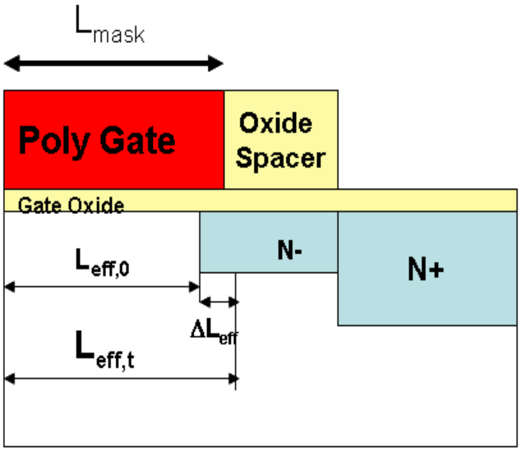


Fig. 6 Effective channel length before and after stress of NMOSFETs.

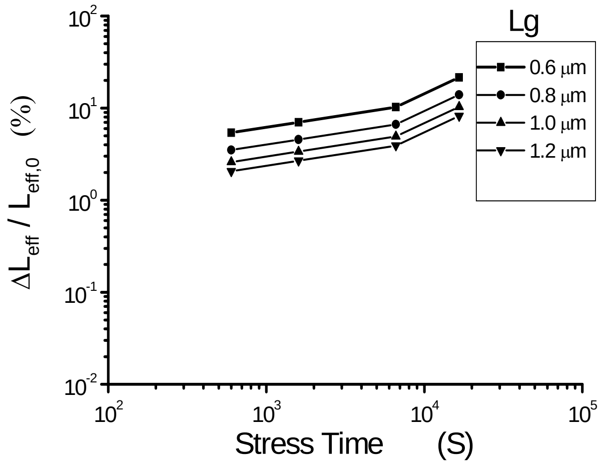


Fig. 7 Change of $\Delta L_{eff}/L_{eff,0}$ versus hot carrier stress time with various channel gate length (*Lg*).

you have seen that the slope of curves has a difference value at the time of 10^4 seconds. It is seemed to be that the trap distribution is increased toward overlapped region between gate and lightly doped drain region.

5. Conclusions

The effective channel lengths of LDD NMOSFETs before and after hot carrier stress time were measured and investigated. The extraction of the effective channel length is based on the I_{DS}-V_{GS} characteristics by the maximum transconductance (maximum slope of I_{DS} & V_{GS}) in the linear region. After the hot carrier stress time, the effective channel lengths were increased as the hot carrier time increase. And the channel conduction was decreased as the hot carrier time increase. It can be explained by the increased of effective channel length. The effects of hot carrier on effective mobility and parasitic series resistance will discuss later on.

References

[1] N. Phongphanchanthra, A. Ruangphanit, N. Klungien, W. Yamwong, S. Niemcharoen, Comparison of conventional and LDD NMOSFETs hot carrier degradation in 0.8 um CMOS technology, in: ECTI-CON 2008, Vol. 2, pp. 825-828.

[2] C. Hu, S.C. Tam, F.-C. Hsu, P.K. Ko, T.-Y. Chan, K.W. Terrill, Hot-electron-induced MOSFET degradation-model, monitor and improvement, IEEE Trans. Electr. Dev. 32 (1985) 375-385.

[3] J.-Y. Kim, M.-S. Kang, Y.-S. Koo, C. An, Saturation of effective channel length increased due to hot carrier degradation in submicron LDD nMOSFETs, in: Proceedings of Conference on Optoelectronic and Microelectronic Materials and Devices, 1996, pp. 215-218.

[4] JEDEC JESD28-A-2001, Procedure for Measuring N-Channel MOSFET Hot Carrier-Induced Degradation under DC Stress.

Computer Technology and Application
Volume 2, Number 11, November 2011

David Publishing Company
1840 Industrial Drive, Suite 160, Libertyville, IL 60048
Tel: 1-847-281-9862; Fax: 1-847-281-9855
http://www.davidpublishing.org
computer@davidpublishing.org

ISSN 1934-7332
11