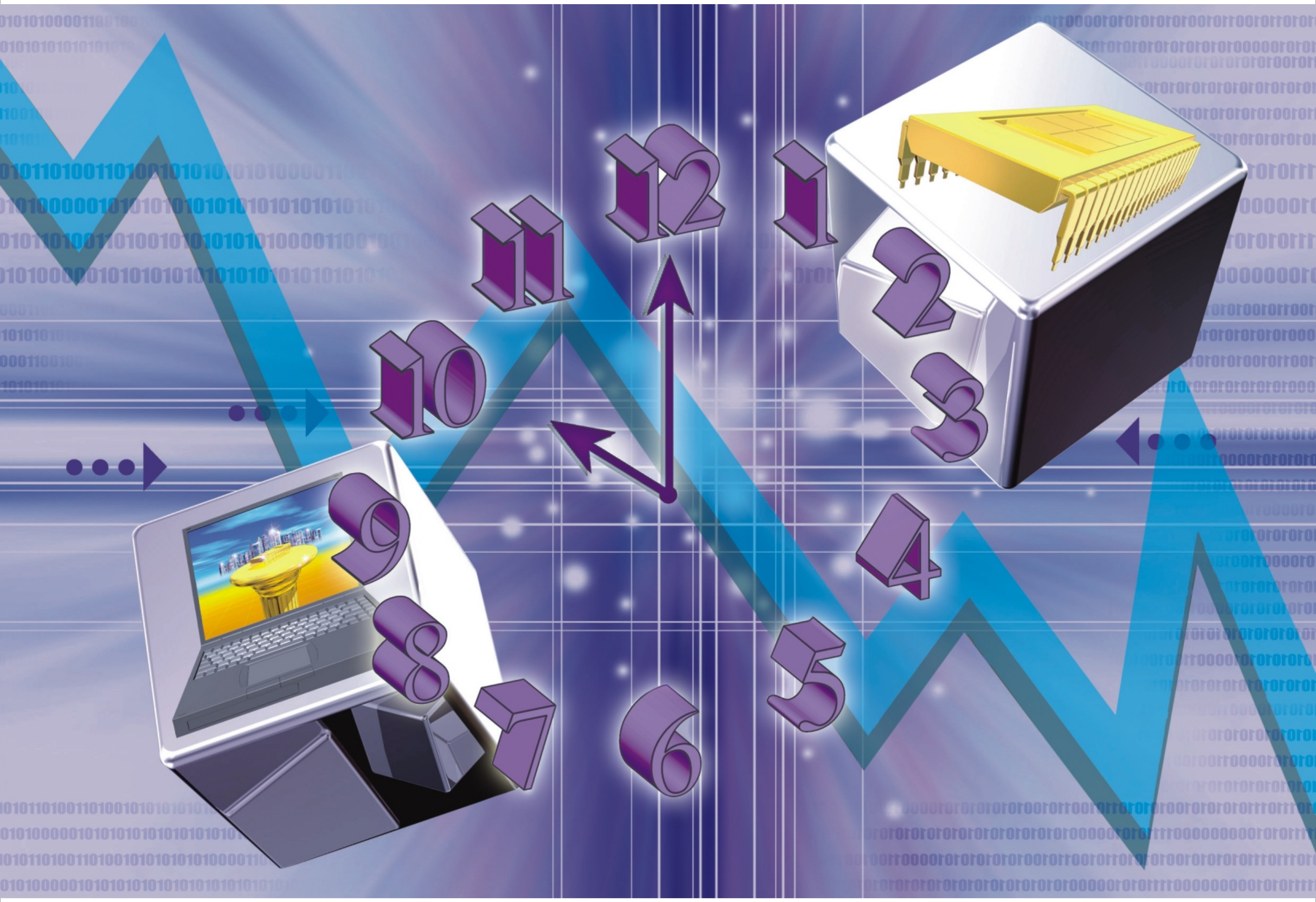
ISSN: 1945-3116 Vol. 3, No. 8, August 2010

Scientific
Research

Journal of Software Engineering and Applications



ISSN: 1945-3116

9 771945 311001 08

www.scirp.org/journal/jsea

Journal Editorial Board

http://www.scirp.org/journal/jsea

Editor-in-Chief

Dr. Ruben Prieto-Diaz	Universidad Carlos III de Madrid, Spain

Executive Editor in Chief

Prof. Yanxiang He	Wuhan University, China

Editorial Advisory Board

Prof. Guoliang Chen	University of Science and Technology of China, China
Prof. Hengda Cheng	Utah State University, USA
Dr. Mengchi Liu	Carleton University, Canada
Prof. Yi Pan	Georgia State University, USA
Prof. Shi Ying	Wuhan University, China

Editorial Board (According to Alphabet)

Dr. Jiannong Cao	Hong Kong Polytechnic University, China
Dr. Raymond Choo	Australian Institute of Criminology, Australia
Dr. Zonghua Gu	Hong Kong University of Science and Technology, China
Dr.Nabil Hameurlain	University of Pau, France
Dr. Keqing He	Wuhan University, China
Dr. Wolfgang Herzner	Austrian Research Centers GmbH-ARC, Austria
Dr. Vassilios (Bill) Karakostas	City University, London, UK
Dr. Chang-Hwan Lee	DongGuk University, Korea (South)
Dr. Hua-Fu Li	Kainan University, Taiwan (China)
Dr. Weiping Li	Peking University, China
Dr. Mingzhi Mao	Sun Yat-Sen University, China
Dr. Kasi Periyasamy	University of Wisconsin-La Crosse, USA
Dr. Michael Ryan	Dublin City University, Ireland
Dr. Juergen Rilling	Concordia University, Canada
Dr. Midthe V. Srinath	Anna University,India
Dr. Jian Wang	Chinese Academy of Sciences, China
Dr. Mark A. Yoder	Electrical and Computer Engineering, USA
Dr. Mao Zheng	University of Wisconsin-La Crosse, USA

Editorial Assistant

Tian Huang	Scientific Research Publishing, USA

J. Software Engineering & Applications, **2010, 3, 737-826**
Published Online August 2010 in SciRes (http://www.SciRP.org/journal/jsea/)

TABLE OF CONTENTS

Volume 3 Number 8 August 2010

Copyright © 2010 SciRes.

Journal of Software Engineering and Applications (JSEA)

Journal Information

SUBSCRIPTIONS

The *Journal of Software Engineering and Applications* (Online at Scientific Research Publishing, www.SciRP.org) is published monthly by Scientific Research Publishing, Inc., USA.

Subscription rates:

Print: $50 per issue.
To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements
Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)
Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: sub@scirp.org

COPYRIGHT

Copyright©2010 Scientific Research Publishing, Inc.

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as described below, without the permission in writing of the Publisher.

Copying of articles is not permitted except for personal and internal use, to the extent permitted by national copyright law, or under the terms of a license issued by the national Reproduction Rights Organization.

Requests for permission for other kinds of copying, such as copying for general distribution, for advertising or promotional purposes, for creating new collective works or for resale, and other enquiries should be addressed to the Publisher.

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assumes no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: jsea@scirp.org

J. Software Engineering & Applications, **2010, 3, 737-745**
doi:10.4236/jsea.2010.38085 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

A New Tool: Solution Boxes of Inequality

Ferenc Kálovics

Analysis Department, University of Miskolc, Miskolc-Egyetemváros, Hungary.
Email: matkf@uni-miskolc.hu

Received May 4th 2010; revised June 19th 2010; accepted June 23rd 2010.

ABSTRACT

Many numerical methods reduce the solving of a complicated problem to a set of elementary problems. In this way, the author reduced the finding of solution boxes of a system of inequalities, the computation of integral values with error bounds, the approximation of global maxima to computing solution boxes of one inequality. These previous papers presented the mathematical aspects of the problems first of all. In this paper the computational handling of solution boxes of an inequality (as a new tool of computer methods) is discussed in detail.

Keywords: *Inequality, Solution Box, Coded Form of Expression*

1. Introduction

Let $g : D \subset \mathrm{R}^m \to \mathrm{R}$ be a continuous multivariate real function, where $D = \left((\underline{x}_1, \overline{x}_1), (\underline{x}_2, \overline{x}_2), \ldots, (\underline{x}_m, \overline{x}_m) \right)$ is an open box. Define the box $B[g, c, \alpha] \subset D$, where $c \in D, \alpha \in \mathrm{R}$, as an open box around c, in which the relation is the same as between $g(c)$ and α (it is supposed that $g(c) \neq \alpha$). Thus, if $g(c) < \alpha$, then $g(x) < \alpha$ for all $x \in B[g, c, \alpha]$, if $g(c) > \alpha$, then $g(x) > \alpha$ for all $x \in B[g, c, \alpha]$. Now consider a particular case. Let $g : D \subset \mathrm{R}^2 \to \mathrm{R}, g(x_1, x_2) = \sin(x_1 x_2) + \dfrac{x_1^2}{\ln x_2}$, where $D = \left((0, \pi), (1, 4) \right)$ is an open box. Try to give the box $B\left[\sin(x_1 x_2) + x_1^2 / \ln x_2, (1,2), 1 \right]$.

Since $g(c) \approx 2.35 > 1$, an open box around point $(1,2)$ in which $g(x) > 1$ is searched. Use the decomposition

$$B\left[\sin(x_1 x_2) + x_1^2 / \ln x_2, (1,2), 1 \right] =$$
$$B\left[\sin(x_1 x_2), (1,2), \beta \right] \cap B\left[x_1^2 / \ln x_2, (1,2), \gamma \right].$$

The spontaneous choice $\beta = \gamma = 0.5$ is suitable because $\sin(1 \cdot 2) > 0.5$ and $1^2 / \ln 2 > 0.5$, hence the boxes $B[\sin(x_1 x_2), (1,2), 0.5]$ and $B\left[x_1^2 / \ln x_2, (1,2), 0.5 \right]$, around c, satisfy the properties $\sin(x_1 x_2) > 0.5$ and $x_1^2 / \ln x_2 > 0.5$, respectively. Since the decompositions

$$B\left[\sin(x_1 x_2), (1,2), 0.5 \right] =$$
$$B\left[x_1 x_2, (1,2), \pi / 6 \right] \cap B\left[x_1 x_2, (1,2), 5\pi / 6 \right] =$$
$$B\left[x_1, (1,2), \pi / 6 \right] \cap B\left[x_2, (1,2), 1 \right] \cap$$
$$B\left[x_1, (1,2), \sqrt{5\pi / 12} \right] \cap B\left[x_2, (1,2), \sqrt{5\pi / 3} \right],$$

where the first row shows that $\pi / 6 < x_1 x_2 < 5\pi / 6$, the second row shows that $x_1 > \pi / 6$, $x_2 > 1$, $x_1 < \sqrt{5\pi / 12}$, $x_2 < \sqrt{5\pi / 3}$ in the box $B[\sin(x_1 x_2), (1,2), 0.5]$, and the decomposition

$$B\left[x_1^2 / \ln x_2, (1,2), 0.5 \right] =$$
$$B\left[x_1^2, (1,2), 0.5 \right] \cap B\left[\ln x_2, (1,2), 1 \right]$$

where the formula shows that $x_1^2 > 0.5, \ln x_2 < 1$ in the box $B\left[x_1^2 / \ln x_2, (1,2), 0.5 \right]$, are also fortunate, therefore

$$B\left[\sin(x_1 x_2) + x_1^2 / \ln x_2, (1,2), 1 \right] =$$
$$\left((\pi / 6, \pi), (1,4) \right) \cap \left((0, \pi), (1,4) \right) \cap$$
$$\left(\left(0, \sqrt{5\pi / 12}\right), (1,4) \right) \cap \left((0, \pi), \left(1, \sqrt{5\pi / 3}\right) \right) \cap$$
$$\left(\left(\sqrt{0.5}, \pi\right), (1,4) \right) \cap \left((0, \pi), (1, e) \right) \approx \left((0.71, 1.14), (1, 2.28) \right).$$

To get the result, spontaneous rules were used for sum (+), product (·), quotient (/) and composit function (∘). In the last step the boxes were computed to some elementary functions. Naturally, the decomposition can break down with any spontaneous rule, e.g. the choice

$\beta = 0.95, \gamma = 0.05$ does not work in the first step, although $\beta + \gamma = 1$. If the guaranteed rules of Theorem 1 of [1] are used, then

$$B\left[\sin(x_1 x_2) + x_1^2 / \ln x_2, (1,2), 1\right] \approx \left((0.84, 1.21), (1, 2.41)\right).$$

Our aim is to make a C++ program which can follow the above decomposition process. Suppose that the continuous multivariate real function $g : D \subset \mathrm{R}^m \to \mathrm{R}$, where $D = \left((\underline{x}_1, \overline{x}_1), (\underline{x}_2, \overline{x}_2), \ldots, (\underline{x}_m, \overline{x}_m)\right)$, is built of the well-known (univariate real) elementary functions:

$$x^r,\ e^x,\ \ln x,\ |x|,\ \sin x,\ \cos x,\ \tan x, \cot x,$$
$$\arcsin x,\ \arccos x,\ \arctan x,\ \operatorname{arc}\cot x$$

by the function operations $+, *, /, \circ$. (It is supposed that r of x^r is a positive integer or negative integer or reciprocal of a positive integer. This is only a formal restriction because of $x^{k/n} = \left(x^k\right)^{1/n}, \forall k, n \in \mathrm{N}^+$.) The functions $a^x, \log_a x, \sinh x,$ etc. can be used by the connections $a^x = e^{x \ln a}$, $\log_a x = \ln x / \ln a$, $\sinh x = (e^x - e^{-x})/2$, etc., respectively, the operation of difference can be used in form $a - b = a + (-1)b$. Naturally, the program can be enlarged by further elementary functions, but we would like to make a program in a publishable length. The notations, names, definitions and discussions are simpler and clearer than they were in the former papers of the author (experience has slowly become knowledge). Using our new notations, the rules of Theorem 1 of [1] are as follows. (It is supposed that the functions p, q satisfy the restrictions belonging to the function g, the function e is an allowed elementary function, furthermore p/q and $e \circ q$ are well-defined on D.)

Rule of constant addition:

$B[p + \lambda, c, \alpha] = B[p, c, \beta]$, where $\lambda \in \mathrm{R}, \beta = \alpha - \lambda$.

Rule of constant multiplication:

$B[\lambda p, c, \alpha] = B[p, c, \beta]$, where $0 \neq \lambda \in \mathrm{R}, \beta = \alpha / \lambda$.

Rule of sum:

$$B[p + q, c, \alpha] = B[p, c, \beta] \cap B[q, c, \gamma],$$
where $\beta = (\alpha + p(c) - q(c))/2, \gamma = \alpha - \beta$.

Rule of quotient:

$$B[p/q, c, \alpha] = B[p, c, 0] \text{ if } \alpha = 0;$$
$$B[p/q, c, \alpha] = B[p, c, \beta] \cap B[q, c, \gamma], \text{ where}$$
$$\gamma = (\alpha p(c) + q(c)) / (1 + \alpha^2), \beta = \alpha\gamma, \text{ if } \alpha \neq 0.$$

Rule of product:

$$B[pq, c, \alpha] = B[p, c, 0] \cap B[q, c, 0]$$
$$\text{if } \alpha = 0 \text{ or } \alpha p(c) q(c) < 0;$$
$$B[pq, c, \alpha] = B[p, c, \beta] \cap B[p, c, \gamma] \cap B[q, c, \beta] \cap B[q, c, \gamma],$$

where $\beta = \sqrt{|\alpha|}, \gamma = -\beta$, if $\alpha \neq 0, p(c) = 0, q(c) = 0$;

$$B[pq, c, \alpha] = B[p, c, \beta] \cap B[p, c, 0] \cap B[q, c, \gamma],$$

where $\beta = 2p(c), \gamma = \alpha / \beta$, if $\alpha \neq 0, p(c) \neq 0, q(c) = 0$;

$$B[pq, c, \alpha] = B[p, c, \beta] \cap B[q, c, \gamma] \cap B[q, c, 0],$$

where $\gamma = 2q(c), \beta = \alpha / \gamma$, if $\alpha \neq 0, p(c) = 0, q(c) \neq 0$;

$$B[pq, c, \alpha] = B[p, c, \beta] \cap B[p, c, 0] \cap B[q, c, \gamma],$$

where $\beta = \operatorname{sgn} p(c) \sqrt{\alpha p(c) / q(c)}$,

$\gamma = \operatorname{sgn} q(c) \sqrt{\alpha q(c) / p(c)}$, if $\alpha p(c) q(c) > 0$.

Rule of composite function:

$$B[e \circ q, c, \alpha] = D \cap B\left[q, c, \beta_l^c\right] \cap B\left[q, c, \beta_r^c\right],$$

where β_l^c is the maximum value for which $e(\beta_l^c) = \alpha$, $\beta_l^c < q(c)$ and β_r^c is the minimum value for which $e(\beta_r^c) = \alpha$, $\beta_r^c > q(c)$. If β_l^c or β_r^c does not exist, then the actual member is omitted.

After the decomposition process the task is to evaluate the solution boxes belonging to elementary functions. The processes for the 12 cases (which are published now for the first time) can be obtained easily enough by the graphs of these functions. (The starting box is $D = \left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right)$ in every case.)

Power function:

$$B\left[x_i^r, c, \alpha\right] =$$
$$\left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right), b_1 = \underline{x}_i, b_2 = \overline{x}_i,$$

if $r \in \mathrm{Z}^+$ is odd, then $b_1 = \alpha^{1/r}$;

if $r \in \mathrm{Z}^+$ is even and $\alpha \geq 0$, then $b_1 = \alpha^{1/r}$, $b_2 = -b_1$;

if $r \in \mathrm{Z}^-$ is odd and $\alpha \neq 0$, then $b_1 = \alpha^{1/r}$;

if $r \in \mathrm{Z}^-$ is even and $\alpha > 0$, then $b_1 = \alpha^{1/r}$, $b_2 = -b_1$;

if $r \in (0,1), 1/r \in \mathrm{Z}^+$ and $\alpha \geq 0$, then $b_1 = \alpha^{1/r}$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$,

if $b_2 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_2$, if $b_2 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_2$.

Copyright © 2010 SciRes.

Exponential function:

$B\left[e^{x_i}, c, \alpha\right] = \left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right)$, $b_1 = \underline{x}_i$,

if $\alpha > 0$, then $b_1 = \ln\alpha$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$.

Logarithm function:

$B[\ln x_i, c, \alpha] = \left((\underline{x}_1, \overline{x}_1), (\underline{x}_2, \overline{x}_2), \ldots, (\underline{x}_m, \overline{x}_m)\right)$, $b_1 = e^{\alpha}$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$.

Absolute value function:

$B\left[|x_i|, c, \alpha\right] =$

$\left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right), b_1 = \underline{x}_i, b_2 = \overline{x}_i$,

if $\alpha \geq 0$, then $b_1 = \alpha$, $b_2 = -\alpha$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$,

if $b_2 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_2$, if $b_2 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_2$.

Sinus function:

$B[\sin x_i, c, \alpha] =$

$\left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right), b_1 = \underline{x}_i$, $b_2 = \overline{x}_i$,

$x = \arcsin|\alpha|, per = 2\pi \cdot$ int $(c_i / 2\pi)$, if $c_i < 0$, then $per = per - 2\pi$,

if $\alpha \in [0,1]$, then $b_1 = x + per$, $b_2 = \pi - x + per$;

if $\alpha \in [-1,0)$, then $b_1 = \pi + x + per$, $b_2 = 2\pi - x + per$;

if $b_2 < c_i$, then $y = b_1, b_1 = b_2, b_2 = y + 2\pi$;

if $b_1 > c_i$, then $y = b_2, b_2 = b_1, b_1 = y - 2\pi$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$,

if $b_2 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_2$, if $b_2 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_2$.

Cosinus function:

$B[\cos x_i, c, \alpha] =$

$\left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right)$, $b_1 = \underline{x}_i$, $b_2 = \overline{x}_i$,

$x = \arccos|\alpha|$, $per = 2\pi \cdot$ int $(c_i / 2\pi)$, if $c_i < 0$, then $per = per - 2\pi$,

if $\alpha \in [0,1]$, then $b_1 = x + per$, $b_2 = 2\pi - x + per$;

if $\alpha \in [-1,0)$, then $b_1 = \pi - x + per$, $b_2 = \pi + x + per$;

if $b_2 < c_i$, then $y = b_1, b_1 = b_2, b_2 = y + 2\pi$;

if $b_1 > c_i$, then $y = b_2, b_2 = b_1, b_1 = y - 2\pi$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$,

if $b_2 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_2$, if $b_2 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_2$.

Tangent function:

$B[\tan x_i, c, \alpha] =$

$\left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right)$, $b_1 = \underline{x}_i$, $b_2 = \overline{x}_i$,

$x = \arctan|\alpha|$, $per = 2\pi \cdot$int $(c_i / 2\pi)$, if $c_i < 0$, then $per = per - 2\pi$,

if $\alpha \geq 0$, then $b_1 = x + per, b_2 = \pi + x + per$;

if $\alpha < 0$, then $b_1 = \pi - x + per, b_2 = 2\pi - x + per$;

if $b_2 < c_i$, then $b_1 = b_2$, $b_2 = b_1 + \pi$;

if $b_1 > c_i$, then $b_2 = b_1$, $b_1 = b_2 - \pi$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$,

if $b_2 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_2$, if $b_2 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_2$.

Cotangent function:

$B[\cot x_i, c, \alpha] =$

$\left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right)$, $b_1 = \underline{x}_i$, $b_2 = \overline{x}_i$,

$x = \operatorname{arccot}|\alpha|$, $per = 2\pi \cdot$int $(c_i / 2\pi)$, if $c_i < 0$, then $per = per - 2\pi$,

if $\alpha \geq 0$, then $b_1 = x + per$, $b_2 = \pi + x + per$;

if $\alpha < 0$, then $b_1 = \pi - x + per$, $b_2 = 2\pi - x + per$;

if $b_2 < c_i$, then $b_1 = b_2$, $b_2 = b_1 + \pi$;

if $b_1 > c_i$, then $b_2 = b_1$, $b_1 = b_2 - \pi$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$,

if $b_2 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_2$, if $b_2 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_2$.

Arcus sinus function:

$B[\arcsin x_i, c, \alpha] =$

$\left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right)$, $b_1 = \underline{x}_i$,

if $\alpha \in [-\pi/2, \pi/2]$, then $b_1 = \sin\alpha$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$.

Arcus cosinus function:

$B[\arccos x_i, c, \alpha] =$

$\left((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)\right)$, $b_1 = \underline{x}_i$,

if $\alpha \in [0, \pi]$, then $b_1 = \cos\alpha$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$.

Arcus tangent function:

$$B[\arctan x_i, c, \alpha] =$$

$$((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)),\ b_1 = \underline{x}_i,$$

if $\alpha \in (-\pi/2, \pi/2)$, then $b_1 = \tan\alpha$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$.

Arcus cotangent function:

$$B[\operatorname{arc}\cot x_i, c, \alpha] =$$

$$((\underline{x}_1, \overline{x}_1), \ldots, (\underline{x}_i, \overline{x}_i), \ldots, (\underline{x}_m, \overline{x}_m)),\ b_1 = \underline{x}_i,$$

if $\alpha \in (0, \pi)$, then $b_1 = \cot\alpha$;

if $b_1 \in (\underline{x}_i, c_i)$, then $\underline{x}_i = b_1$, if $b_1 \in (c_i, \overline{x}_i)$, then $\overline{x}_i = b_1$.

Now we have 6 rules for decomposition and 12 processes for elementary boxes. Since the application of the 6th rule (rule of composite function) requires the same investigation as the evaluation of elementary boxes, therefore the decomposition part of the program contains 17 different cases, and the evaluation part contains 12 cases. At the end of this section, let us emphasize some facts about solution boxes. 1) If $g(c) < \alpha$, then $x \in B(g, c, \alpha)$ implies $-\infty < g(x) < \alpha$. Consequently, the box $B(g, c, \alpha)$ of domain D is assigned to the interval $(-\infty,\ \alpha)$ of function values. Similarly, if $g(c) > \alpha$, then the box $B(g, c, \alpha)$ of domain D is assigned to the interval $(\alpha,\ \infty)$ of function values. The so-called interval extension functions used in interval methods (see e.g. in [2]) are inverse type functions, they assign intervals of function values to boxes of domain. The handling and application of these two tools require a highly different mathematical and computational background. 2) The decomposition of the expression $g(x)$, the determination of the elementary boxes, uses only the structure of the expression $g(x)$ and supposes only the continuity of g on *D*. 3) The box $B(g, c, \alpha)$ is not a symmetrical box around c. Often it has a large volume, although $g(c) < \alpha$ or $g(c) > \alpha$ is only just satisfied.

2. Numerically Coded Form of Expression

When the author introduced the above decomposition of expressions, the process was followed by a Maple program. This code computed a box $B(g, c, \alpha)$ very similarly as it is done in our numerical example. The program utilizes strongly the ability of Maple to give the type of an expression by the function $type(e,t)$ where e is an expression and t is a type name, furthermore to give operands from an expression by the function $op(i,e)$ where i is the position index and e is an expression. The use of our Maple program is very convenient because we must only give the function g (the expression $g(x)$) in a customary form. Unfortunately, the program is fairly slow. The conventional program languages (e.g. C++, Fortran) have no similar functions, but they can follow the decomposition, if a numerically coded form of the expression $g(x)$ is used. This form is built of triples of integer or real numbers. The first number in a triple is the so-called operation code (using the order of rules and elementary functions):

const. add. $\leftrightarrow 1$, const. mult. $\leftrightarrow 2$, $+ \leftrightarrow 3$, $/ \leftrightarrow 4$, $* \leftrightarrow 5$, power func. $\leftrightarrow 6$, exp $\leftrightarrow 7$, ln $\leftrightarrow 8$, abs. func. $\leftrightarrow 9$, sin $\leftrightarrow 10$, cos $\leftrightarrow 11$, tan $\leftrightarrow 12$, cot $\leftrightarrow 13$, arcsin $\leftrightarrow 14$, arccos $\leftrightarrow 15$, arctan $\leftrightarrow 16$, arccot $\leftrightarrow 17$.

The second number in the triples is minus i if an elementary function uses the argument x_i (the minus sign identifies the elementary funcions). If the operation uses a former triple as the first operand, then the ordinal number of this triple is the second number in the actual triple. The third number in the triples is the given constant or the exponent if the constant addition, constant multiplication or the power function is used, respectively. If the operation uses a former triple as a second operand (cases of $+, /, *$), then the ordinal number of this triple is the third number in the actual triple. If the operation is $\exp, \ln, \ldots, \operatorname{arccot}$ (the first number in the triple is between 7 and 17), then the third number in the actual triple is 0. To make the description of expressions by triples unique, we use the left-right rule in every choice. Exercise the description on the expression $g(x_1, x_2) = \sin(x_1 x_2) + x_1^2 / \ln x_2$. The four elementary functions used in the build-up of $g(x_1, x_2)$ (the elementary boxes belong to these functions after the decomposition) are $x_1 = (x_1)^1, x_2 = (x_2)^1, x_1^2, \ln x_2$. Their descriptions are:

$$x_1 = (6,-1,1),\quad x_2 = (6,-2,1),$$

$$x_1^2 = (6,-1,2),\quad \ln x_2 = (8,-2,0).$$

Now we have the sequence of triples:

$$(6,-1,1), (6,-2,1), (6,-1,2), (8,-2,0).$$

Using the elements of this sequence, the functions $x_1 x_2, \sin(x_1 x_2)$ and $x_1^2 / \ln x_2$ can be obtained in the form:

$$x_1x_2 = (5,1,2),\quad \sin(x_1x_2) = (10,5,0),$$
$$x_1^2 / \ln x_2 = (4,3,4).$$

Since the last operation can be written in form $(3,6,7)$, the complete description of $g(x_1, x_2)$ is:

$$\{(6,-1,1),\ (6,-2,1),\ (6,-1,2),\ (8,-2,0),$$
$$(5,1,2),\ (10,5,0),\ (4,3,4),\ (3,6,7)\}.$$

It can be seen that the preparation of numerically coded forms of expressions is easy enough, nevertheless the author has a short Maple program also for this one-off work. (But a numerically coded form can be used a thousand or a million times in an application.) The construction of our Maple program can be illustrated (on the former expression) by the **Table 1**.

Table 1. Creation of numerically coded form

i	Compo-nents		Triples	⇓↑	ii	New triples	Result
1	$\sin(x_1x_2)+$ $x_1^2 / \ln x_2$	$\Rightarrow$	$(3,2,3)$	$\rightarrow$	8	$(3,7,6)$	$(6,-1,2)$
2	$\sin(x_1x_2)$	$\Rightarrow$	$(10,4,0)$	$\rightarrow$	7	$(10,5,0)$	$(8,-2,0)$
3	$x_1^2 / \ln x_2$	$\Rightarrow$	$(4,5,6)$	$\rightarrow$	6	$(4,1,2)$	$(6,-1,1)$
4	x_1x_2	$\Rightarrow$	$(5,7,8)$	$\rightarrow$	5	$(5,3,4)$	$(6,-2,1)$
5	x_1^2	$\Rightarrow$	$(6,-1,2)$	$\Rightarrow$	1	√	$(5,3,4)$
6	$\ln x_2$	$\Rightarrow$	$(8,-2,0)$	$\Rightarrow$	2	√	$(4,1,2)$
7	x_1	$\Rightarrow$	$(6,-1,1)$	$\Rightarrow$	3	√	$(10,5,0)$
8	x_2	$\Rightarrow$	$(6,-2,1)$	$\Rightarrow$	4	√	$(3,7,6)$

The 1st column shows the index (serial number) of the expression appearing in the 2nd column. The 4th column contains the temporary triples which use the indexes of the 1st column for identification. Going downwards the temporary triples, the elementary tripes receive their new indexes, and going backwards, the other triples receive their new indexes (6th column). Using these new indexes the final triples (7th column) are obtained. Ordering them by the indexes of the 6th column the result is ready (8th column). This result differs from our previous one formally (here the left-right rule is omitted), but they are equivalent in essence. The arrays of Maple program x, com, tri, ii, res belong to vector x components of $g(x)$, triples, new indexes and result, respectively. The array v is an auxiliary vector. The complete code is as follows.

```
DECLARATIONS
 > x:=array(1..10): com:=array(1..10): tri:=array(1..10,1..3):
 > ii:=array(1..10): res:=array(1..10,1..3): v:=array(1..3):
INITIAL VALUES
 > com[1]:=sin(x[1]*x[2])+x[1]^2 / ln(x[2]):
 > i:=0: imax:=1:
COMPONENTS and TRIPLES
 > while i <= imax do
 >        i:=i+1: g:=com[i]:
 >        if type(g,name) then v:=[0,op(1,g),0]: fi:
 >        if type(g,`+`) then
 >              if type(op(1,g),numeric) then v:=[1,g-op(1,g),op(1,g)]:
 >              elif type(g-op(1,g),numeric) then v:=[1,op(1,g),g-op(1,g)]:
 >              else v:=[3,op(1,g),g-op(1,g)]: fi: fi:
 >        if type(g,`*`) then
 >              if type(op(1,g),numeric) then v:=[2,g/op(1,g),op(1,g)]:
 >              elif type(g/op(1,g),numeric) then v:=[2,op(1,g),u/op(1,g)]:
 >              elif type(op(2,g),`^`) and op(2,op(2,g)) < 0 then v:=[4,op(1,g),op(2,g)^(-1)]:
 >              else v:=[5,op(1,g),g/op(1,g)]: fi: fi:
 >       if type(g,`^`) then v:=[6,op(1,g),op(2,g)]: fi:
 >       if type(g,function) then
 >              if op(0,g)=exp then v:=[7,op(1,g),0]: fi: if op(0,g)=ln then v:=[8,op(1,g),0]: fi:
 >              if op(0,g)=abs then v:=[9,op(1,g),0]: fi: if op(0,g)=sin then v:=[10,op(1,g),0]: fi:
 >              if op(0,g)=cos then v:=[11,op(1,g),0]: fi: if op(0,g)=tan then v:=[12,op(1,g),0]: fi:
 >              if op(0,g)=cot then v:=[13,op(1,g),0]: fi: if op(0,g)=arcsin then v:=[14,op(1,g),0]: fi:
 >              if op(0,g)=arccos then v:=[15,op(1,g),0]: fi:if op(0,g)=arctan then v:=[16,op(1,g),0]: fi:
 >              if op(0,g)=arccot then v:=[17,op(1,g),0]: fi: fi:
 >       if v[1]=0 then tri[i,1]:=6: tri[i,2]:=-v[2]: tri[i,3]:=1: fi:
 >       if v[1] > = 6 and v[1] < = 17 then
 >              if type(v[2],name) then    tri[i,1]:=v[1]: tri[i,2]:=-op(1,v[2]): tri[i,3]:=v[3]:
 >              else imax:=imax+1: com[imax]:=v[2]: tri[i,1]:=v[1]: tri[i,2]:=imax: tri[i,3]:=v[3]: fi: fi:
 >       if v[1]=1 or v[1]=2 then
 >              imax:=imax+1: com[imax]:=v[2]: tri[i,1]:=v[1]: tri[i,2]:=imax: tri[i,3]:=v[3]: fi:
 >       if v[1] > = 3 and v[1] < = 5 then
 >              imax:=imax+1: com[imax]:=v[2]: imax:=imax+1: com[imax]:=v[3]:
 >              tri[i,1]:=v[1]: tri[i,2]:=imax-1: tri[i,3]:=imax: fi:
```

```
 > od:
NEW INDEXES
 > ind:=1:
 > for i from 1 to imax do if tri[i,2] < 0 then ii[i]:=ind: ind:=ind+1: fi: od:
 > for i from imax by -1 to 1 do if tri[i,2] > 0 then ii[i]:=ind: ind:=ind+1: fi: od:
RESULT
 > for ind from 1 to imax do
 >      for i from 1 to imax do
 >              if ii[i]=ind then res[ind,1]:=tri[i,1]:
 >                      if tri[i,2] < 0 then res[ind,2]:=tri[i,2]: res[ind,3]:=tri[i,3]: fi:
 >                      if tri[i,2] > 0 then res[ind,2]:=ii[tri[i,2]]:
 >                              if tri[i,1]=3 or tri[i,1]=4 or tri[i,1]=5 then res[ind,3]:=ii[tri[i,3]]: else res[ind,3]:=tri[i,3]:
 >              fi: fi: fi:
 > od: od:
 > print(`The result`, res):
```

Now we have a suitable form of expressions to make a fast program for our original problem.

3. A Complete C++ Program for Computation of Solution Boxes

Here the function segment *solbox* is made for the computation of solution box $B[g,c,\alpha]$ where

$$g\ :\ D\subset \mathrm{R}^m \to \mathrm{R}, D = \left(\left(\underline{x}_1,\overline{x}_1\right),...,\left(\underline{x}_m,\overline{x}_m\right)\right), c\in D, \alpha\in \mathrm{R}$$

have the former definitions. The headline of this segment contains the input parameters $D\leftrightarrow D, G\leftrightarrow g$ (in triple form), $c\leftrightarrow c, alp\leftrightarrow \alpha, m\leftrightarrow m, nt\leftrightarrow$ number of triples. The result $B\left[g,c,\alpha\right]$ (as output parameter) is placed into the array B. The segment contains 3 essential parts (great loops). In the first one the function values of components of $g(x)$ at point c are computed by using the triples in succession. These function values are placed into the array fv and will be used in the decomposition steps. In the second loop the decomposition (making elementary boxes) goes on. In the normal case ($nt>1$), first the last triple (expression $g(x)$) and α are placed into the array of non-elementary expressions (ne). At this time the number of elements in this array is 1 ($ine=1$). Applying the suitable rule for this element, new elements of the array ne or (and) those of the array of elementary expressions (el) are created. The values β,γ, etc. appearing in the rules are denoted by be, ga, de, ep, and s shows the number of new boxes. The indexes ine and iel show the numbers of elements in the arrays ne and el. The decomposition procedure is finished when there is no element for examination in the array ne, that is the necessary elementary expressions are all in the array el. In the third great loop of segment *solbox* the (iel pieces of) elementary boxes are evaluated and the array B is made. As it was mentioned earlier, the application of the composite function rule requires the same investigation as the evaluation of elementary boxes. Therefore the second and third loops of the segment contain similar lines (only some parameters are different) between case 6 and case 17.

```
#include <iostream.h>
#include <math.h>
double B[10][3];
/*DECLARATIONS*/
void solbox(double D[][3],double G[][4],double c[],double alp,int m,int nt)
{double fv[100],ne[100][5],el[100][5],al,be,ga,de,ep,x,y,w,alf,per;
int i,j,k,l,ii,jj,kk,ine,iel,s; const double Pi=3.14159265;
/*FUNCTION VALUES TO COMPONENTS OF EXPRESSION g(x)*/
for (i=1; i<=nt; i++)
        {j=(int)G[i][1]; k=(int)G[i][2]; l=(int)G[i][3]; w=G[i][3];
        if (k<0) x=c[-k]; else x=fv[k]; if (j>=3 && j<=5) y=fv[l];
        switch (j)
                {case 1:fv[i]=x+w;break; case 2:fv[i]=x*w;break; case 3:fv[i]=x+y;break;
                case 4:fv[i]=x/y;break; case 5:fv[i]=x*y;break; case 6:fv[i]=pow(x,w);break;
                case 7:fv[i]=exp(x);break; case 8:fv[i]=log(x);break; case 9:fv[i]=fabs(x);break;
                case 10:fv[i]=sin(x);break; case 11:fv[i]=cos(x);break; case 12:fv[i]=tan(x);break;
                case 13:fv[i]=1/tan(x);break; case 14:fv[i]=asin(x);break; case 15:fv[i]=acos(x);break;
                case 16:fv[i]=atan(x);break; case 17:fv[i]=Pi/2-atan(x);break;}}
/*DECOMPOSITION BY USING THE 6 RULES*/
        for (i=1; i<=3; i++) ne[1][i]=G[nt][i]; ne[1][4]=alp; ine=1; iel=0; i=0;
        while (i<ine && nt>1)
                {i++; j=(int)ne[i][1]; k=(int)ne[i][2]; l=(int)ne[i][3]; w=ne[i][3]; al=ne[i][4]; s=0;
        switch (j)
            {case 1: be=al-w;s=1;break;
            case 2: be=al/w;s=1;break;
            case 3: be=(al+fv[k]-fv[l])/2;ga=al-be;s=2;break;
            case 4: if (al==0) {be=0;s=1;}; if (al !=0) {ga=(al*fv[k]+fv[l])/(1+al*al);be=al*ga;s=2;};break;
            case 5: if (al==0 || fv[k]*fv[l]*al < 0) {be=0;ga=0;s=2;}
                if (al!=0 && fv[k]==0 && fv[l]==0) {be=sqrt(fabs(al));ga=be;de=-be;ep=de;s=4;}
                if (al!=0 && fv[k] !=0 && fv[l]==0) {be=2*fv[k];ga=al/be;de=0;s=3;}
                if (al!=0 && fv[k]==0 && fv[l]!=0) {ga=2*fv[l];be=al/ga;de=be;ep=0;s=4;}
                if (fv[k]*fv[l]*al>0) {be=fv[k]/fabs(fv[k])*sqrt(al*fv[k]/fv[l]);
                    ga=fv[l]/fabs(fv[l])*sqrt(al*fv[l]/fv[k]);de=0;s=3;};break;
            case 6: if (w>0 && (l+1)/2*2==l+1) {be=pow(al,1/w); s=1;}
                if (w>0 && l/2*2==l && al>=0) {be=pow(al,1/w);ga=-be;s=2;}
                if (w<0 && (l-1)/2*2==l-1 && al !=0) {be=pow(al,1/w);s=1;}
                if (w<0 && l/2*2==l && al>0) {be=pow(al,1/w);ga=-be;s=2;}
                if (w>0 && w<1 && al>=0) {be=pow(al,1/w);s=1;};break;
            case 7: if (al > 0) {be=log(al);s=1;};break;
            case 8: be=exp(al);s=1;break;
            case 9: if (al >= 0) {be=al;ga=-al;s=2;};break;
            case 10:if (fabs(al)<=1)
                {x=asin(fabs(al));per=2*Pi*(int)(fv[k]/2/Pi); if(fv[k]<0) per=per-2*Pi;
                if (al>=0) {be=x+per;ga=Pi-x+per;}; if (al<0) {be=Pi+x+per;ga=2*Pi-x+per;}
                if (ga<fv[k]) {y=be;be=ga;ga=y+2*Pi;}; if (be>fv[k]) {y=ga;ga=be;be=y-2*Pi;}; s=2;};break;
            case 11:if (fabs(al)<=1)
                {x=acos(fabs(al));per=2*Pi*(int)(fv[k]/2/Pi); if(fv[k]<0) per=per-2*Pi;
                if (al>=0) {be=x+per;ga=2*Pi-x+per;}; if (al<0) {be=Pi-x+per;ga=Pi+x+per;}
                if (ga<fv[k]) {y=be;be=ga;ga=y+2*Pi;}; if (be>fv[k]) {y=ga;ga=be;be=y-2*Pi;}; s=2;};break;
            case 12:x=atan(fabs(al));per=2*Pi*(int)(fv[k]/2/Pi); if(fv[k]<0) per=per-2*Pi;
                if (al>=0) {be=x+per;ga=Pi+x+per;}; if (al<0) {be=Pi-x+per;ga=2*Pi-x+per;}
                if (ga<fv[k]) {be=ga;ga=be+Pi;}; if (be>fv[k]) {ga=be;be=ga-Pi;};s=2;break;
            case 13:x=Pi/2-atan(fabs(al));per=2*Pi*(int)(fv[k]/2/Pi); if(fv[k]<0) per=per-2*Pi;
                if (al>=0) {be=x+per;ga=Pi+x+per;}; if (al<0) {be=Pi-x+per;ga=2*Pi-x+per;}
                if (ga<fv[k]) {be=ga;ga=be+Pi;}; if (be>fv[k]) {ga=be;be=ga-Pi;};s=2;break;
            case 14:if (fabs(al)<=Pi/2) {be=sin(al);s=1;};break;
            case 15:if (al>=0 && al<=Pi) {be=cos(al);s=1;};break;
            case 16:if (fabs(al)<Pi/2) {be=tan(al);s=1;};break;
            case 17:if (al>0 && al<Pi) {be=1/tan(al);s=1;};break;}
        for (ii=1; ii<=s; ii++)
            {switch (ii)
                {case 1:alf=be;kk=k;break; case 2:alf=ga;if (j>=3 && j<=5) kk=l;break;
                case 3:alf=de;kk=k;break; case 4:alf=ep;if (j>=3 && j<=5) kk=l;break;}
            if (G[kk][2]<0) {iel++; for (jj=1; jj<=3; jj++) el[iel][jj]=G[kk][jj];el[iel][4]=alf;}
            if (G[kk][2]>0) {ine++; for (jj=1; jj<=3; jj++) ne[ine][jj]=G[kk][jj];ne[ine][4]=alf;}}}
    if (nt==1) {for (ii=1; ii<=4; ii++) el[1][ii]=ne[1][ii];iel=1;}
/*EVALUATING THE ELEMENTARY BOXES BY USING THE 12 PROCESSES*/
    for (i=1; i<=m; i++) {B[i][1]=D[i][1]; B[i][2]=D[i][2];}
    for (i=1; i<=iel; i++)
        {j=(int)el[i][1]; k=(int)-el[i][2]; l=(int)el[i][3]; w=el[i][3]; al=el[i][4]; s=0;
        switch (j)
            {case 6: if (w>0 && (l+1)/2*2==l+1) {be=pow(al,1/w);s=1;}
                if (w>0 && l/2*2==l && al>=0) {be=pow(al,1/w);ga=-be;s=2;}
                if (w<0 && (l-1)/2*2==l-1 && al!=0) {be=pow(al,1/w);s=1;}
                if (w<0 && l/2*2==l && al>0) {be=pow(al,1/w);ga=-be;s=2;}
                if (w>0 && w<1 && al>=0) {be=pow(al,1/w);s=1;};break;
            case 7: if (al > 0) {be=log(al);s=1;};break;
            case 8: be=exp(al);s=1;break;
            case 9: if (al >= 0) {be=al;ga=-al;s=2;};break;
            case 10:if (fabs(al)<=1)
                {x=asin(fabs(al));per=2*Pi*(int)(c[k]/2/Pi);if(c[k]<0) per=per-2*Pi;
                if (al>=0) {be=x+per;ga=Pi-x+per;}; if (al<0) {be=Pi+x+per;ga=2*Pi-x+per;}
                if (ga<c[k]) {y=be;be=ga;ga=y+2*Pi;}; if (be>c[k]) {y=ga;ga=be;be=y-2*Pi;}; s=2;};break;
            case 11:if (fabs(al)<=1)
                {x=acos(fabs(al)); per=2*Pi*(int)(c[k]/2/Pi);if(c[k]<0) per=per-2*Pi;
                if (al>=0) {be=x+per;ga=2*Pi-x+per;}; if (al<0) {be=Pi-x+per;ga=Pi+x+per;}
                if (ga<c[k]) {y=be;be=ga;ga=y+2*Pi;}; if (be>c[k]) {y=ga;ga=be;be=y-2*Pi;}; s=2;};break;
            case 12:x=atan(fabs(al));per=2*Pi*(int)(c[k]/2/Pi);if(c[k]<0) per=per-2*Pi;
                if (al>=0) {be=x+per;ga=Pi+x+per;}; if (al<0) {be=Pi-x+per;ga=2*Pi-x+per;}
                if (ga<c[k]) {be=ga;ga=be+Pi;}; if (be>c[k]) {ga=be;be=ga-Pi;};s=2;break;
            case 13:x=Pi/2-atan(fabs(al));per=2*Pi*(int)(c[k]/2/Pi);if(c[k]<0) per=per-2*Pi;
```

```
                    if (al>=0) {be=x+per;ga=Pi+x+per;}; if (al<0) {be=Pi-x+per;ga=2*Pi-x+per;}
                    if (ga<c[k]) {be=ga;ga=be+Pi;}; if (be>c[k]) {ga=be;be=ga-Pi;};s=2;break;
                case 14:if (fabs(al)<=Pi/2) {be=sin(al);s=1;};break;
                case 15:if (al>=0 && al<=Pi) {be=cos(al);s=1;};break;
                case 16:if (fabs(al)<Pi/2) {be=tan(al);s=1;};break;
                case 17:if (al>0 && al<Pi) {be=1/tan(al);s=1;};break;}
            for (ii=1; ii<=s; ii++) { alf=be; if (ii==2) alf=ga;
                if (alf < c[k] && alf > B[k][1]) B[k][1]=alf;
                if (alf > c[k] && alf < B[k][2]) B[k][2]=alf;}}}
/*THE CALLING SEGMENT WITH THE SAMPLE EXPRESSION*/
void main()
    {double D[10][3],G[100][4],c[10],alp; int m,nt,i,j;
    double g[][3]={{6,-1,1},{6,-2,1},{6,-1,2},{8,-2,0},{5,1,2},{10,5,0},{4,3,4},{3,6,7}};
    D[1][1]=0.;D[1][2]=3.141593;D[2][1]=1.;D[2][2]=4.; c[1]=1.;c[2]=2.; alp=1.; m=2; nt=8;
    for (i=1; i<=nt; i++) {for (j=1; j<=3; j++) G[i][j]=g[i-1][j-1];}; solbox(D,G,c,alp,m,nt);
    for (i=1; i<=m; i++) {cout << B[i][1] << " " << B[i][2] << endl;}}
```

The calling (main) segment uses a simple trick (push down of indexes) for the easy handling of triple forms. Using the form (produced by the Maple program)

$$\{(6,-1,2),(8,-2,0),(6,-1,1),(6,-2,1),$$
$$(5,3,4),(4,1,2),(10,5,0),(3,7,6)\}$$

and the form (produced by us)

$$\{(6,-1,1),(6,-2,1),(6,-1,2),(8,-2,0),$$
$$(5,1,2),(10,5,0),(4,3,4),(3,6,7)\}$$

of our sample expression $g(x_1,x_2)$, the result is the same:

$$B\left[\sin(x_1x_2)+x_1^2/\ln x_2,(1,2),1\right]=$$
$$\big((0.839584,1.20543),(1,2.41086)\big),$$

which was given (with rounded values) in the first section.

4. A numerical Example for Illustration

The surfaces given by formulas

$$x_3=2\sin(x_1x_2)+2,\quad x_3=x_1^2+x_2^2-2$$

have their intersection over the two dimensional interval $\big((-3,3),(-3,3)\big)$. Maple 11 can produce **Figure 1** with these data. Our aim is to give an approximating value for the volume of the section set S determined by the surfaces.

Since the set S is the solution set of the system of inequalities:

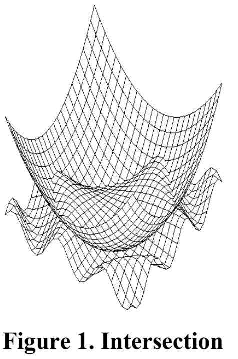


Figure 1. Intersection

$$2\sin(x_1x_2)-x_3+2\ge 0,\quad x_3-x_1^2-x_2^2+2\ge 0,$$

one of methods of [3] can be used. This method searches for disjunct open solution boxes of an inequality system and it is based on the following four principles. 1) If $B\left[f_1,c,0\right]$ is a solution box to the inequality $f_1(x)\ge 0$ and $B\left[f_2,c,0\right]$ is a solution box to the inequality $f_2(x)\ge 0$, then the box $B\left[f_1,c,0\right]\cap B\left[f_2,c,0\right]$ is a solution box to the system of the two inequalities. 2) If B_1 and B_2 are m-dimensional boxes, then the set B_1-B_2 can be divided into (at most) $2m$ boxes easily. In this way a sequence of solution boxes and another sequence of boxes waiting for examination can be generated. 3) A boundary region is used to prevent the appearance of too small boxes. Here this region is defined by the inequality $\|f(x)\|_\infty<10^{-3}$ and a cube with edges of 10^{-3} is excluded at a region point c. 4) The most promising box (the box with maximum volume) is chosen from boxes waiting for examination in every step. The essential input data are:

$$f_1(x)=2\sin(x_1x_2)-x_3+2=$$
$$\{(6,-1,1),\ (6,-2,1),...,(3,6,7),\ (1,8,2.)\},$$
$$f_2(x)=x_3-x_1^2-x_2^2+2=$$
$$\{(6,-1,2),\ (6,-2,2),...,(3,3,5),\ (1,6,2.)\},$$

the initial box $I=\big((-3,3),\ (-3,3),\ (-2,4)\big)$ which includes the set S, and the number N_b of solution boxes looked for, as stop criteria. For the values $N_b=10^4,\ 5\cdot 10^4,\ 10^5$ the results are

$$27.0008,\quad 28.1484,\quad 28.4231,$$

respectively. The running times are (with the original Lahey Fortran 90 version 4.5 code on a PC of two 2 GHz processor):

$$2\,\text{sec},\quad 26\,\text{sec},\quad 107\,\text{sec},$$

respectively. The fast increase of running time shows that neither the 3rd nor the 4th principle work well if N_b is

Copyright © 2010 SciRes.

large. (Only a few hundred solution boxes were searched for in [3].) Now consider another possibility. Since

$$vol(S)=\int\int\int_S 1\ dxdydz,$$

and [4,5] contain methods for computation of integrals, perhaps better results can be obtained. The method of [5] is based on the following three principles. 1) The value of a (simple, double, triple, etc.) integral can be given by the volume of a suitable set T. If the box I contains this set, then a good scanning of T gives an approximation of the integral value and the scanning of the set $I-T$ also facilitates computation of an error bound. Here the set T can be described by the system of inequalities

$$2\sin(x_1x_2)-x_3+2\geq 0,\ x_3-x_1^2-x_2^2+2\geq 0,\ 1-x_4\geq 0,$$

and the scanning works by computing solution boxes. The scanning of the set $I-T$ is based on the examination of "the worst inequality" at points $c\in I-T$. 2) If B_1 and B_2 are *m*-dimensional boxes, then the set B_1-B_2 can be divided into (at most) $2m$ boxes easily. 3) The too small boxes are filtered by the simple condition $vol(B)>\kappa$. Naturally, the value κ has a strong influence on the available error bound. The program is stopped when there is no more box waiting for examination. Now the essential input data are: the expressions $f_1(x),\ f_2(x),\ f_3(x)$ in triple form, the initial box $I=\left((-3,3),\ (-3,3),(-2,4),(0,1)\right)$ which includes the set T, and the filter parameter κ. For the values $\kappa=10^{-5},\ \ 10^{-6},\ \ 10^{-7}$, the results are

$$28.9953\pm 1.0169,\ 29.0499\pm 0.4840,\ 29.0705\pm 0.2294,$$

respectively. The running times are (with the original Visual C++ version 6.0 code on the former PC of two 2 GHz processor):

$$4\,\text{sec},\ \ 16\,\text{sec},\ \ 79\,\text{sec},$$

respectively. Since the difference between the third and first results is 0.0752 only, the third error bound suggests that all three of the results are good enough approximations. The extra work of the second and third cases was spent mostly on improving the error bound. Now make a goal-oriented transformation of the method of [3]. If the third principle (boundary region) and the fourth principle (most promising box) are omitted and principles of method of [5] belonging to error bound and filter parameter κ are used, then the new method will be competitive. For the values $\kappa=10^{-5},\ \ 10^{-6},\ \ 10^{-7}$, the results are

$$29.0862\pm 0.6765,\ 29.0829\pm 0.3124,\ 29.0826\pm 0.1447,$$

respectively. The running times are (with the new Lahey Fortran code):

$$1\,\text{sec},\ \ \ 5\,\text{sec},\ \ \ 24\,\text{sec},$$

respectively. These results are better from all points of view than the previous ones were. Since now the difference between the first and third results is 0.0036 only, the 3 values suggest much more accurate approximation than the guaranteed error bounds do. On the other hand, by the geometrical meaning, here the solution (section) set S (and its complementary set) was covered with cuboids, while the solution set T (and its complementary set) was covered with four-dimensional (abstract) boxes at the triple integral. It is not surprising that the computational effort increases with dimension. Finally some remarks are given.

1) The computation efforts (the evaluation times) belonging to $B(g,c,\alpha)$ and $g(c)$ can be characterized well enough by the formula: effort $\left(B(g,c,\alpha)\right)\approx 10*$ effort $\left(g(c)\right)$, with respect to all three programming languages mentioned and problems of [1,3-5].

2) The speeds belonging to the evaluation of $B(g,c,\alpha)$ satisfy the relations: speed(C++) ≈ $200*$ speed(Maple), speed(Fortran) ≈ $300*$ speed(Maple) for the examples of [1]. Nevertheless, the Fortran codes are hardly faster than the C++ ones (differences of only 0-10 percent in running time) for the examples of [5].

3) We always used floating point arithmetic for computing solution boxes (zone function values by former name). Since these boxes are computed by lower estimates (see proofs of [1]), we have never happened to obtain a faulty result because of the effect of rounding errors.

4) The author would gladly send the C++, Fortran, Maple codes mentioned to interested readers in e-mail as an attached file.

REFERENCES

[1] F. Kálovics, "Creating and Handling Box Valued Functions Used in Numerical Methods," *Journal of Computational and Applied Mathematics*, Vol. 147, No. 2, 2002, pp. 333-348.

[2] R. Hammer, M. Hocks, U. Kulisch and D. Ratz, "Numerical Toolbox for Verified Computing," Springer-Verlag, Berlin, 1993.

[3] F. Kálovics and G. Mészáros, "Box Valued Functions in Solving Systems of Equations and Inequalities," *Numerical Algorithms*, Vol. 36, No. 1, 2004, pp. 1-12.

[4] F. Kálovics, "Zones and Integrals," *Journal of Computational and Applied Mathematics*, Vol. 182, No. 2, 2005, pp. 243-251.

[5] F. Kálovics, "Two Improved Zone Methods," *Miskolc Mathematical Notes*, Vol. 8, No. 2, 2007, pp. 169-179.

Copyright © 2010 SciRes.

J. Software Engineering & Applications, **2010, 3, 746-755**
doi:10.4236/jsea.2010.38086 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

A Residual Time Based Scheduling: Performance Modeling in M/G/C Queueing Applications*

Sarah Tasneem[1], Lester Lipsky[2], Reda Ammar[2], Howard Sholl[2]

[1]Math and Computer Science, Eastern Connecticut State University, Willimantic, USA; [2]Computer Science and Engineering, University of Connecticut, Storrs, USA.
Email: tasneems@easternct.edu

Received June 3rd 2010; revised July 16th 2010; accepted July 20th 2010.

ABSTRACT

It is well known, in queueing theory, that the system performance is greatly influenced by scheduling policy. No universal optimum scheduling strategy exists in systems where individual customer service demands are not known a priori. However, if the distribution of job times is known, then the residual time (expected time remaining for a job), based on the service it has already received, can be calculated. Our particular research contribution is in exploring the use of this function to enhance system performance by increasing the probability that a job will meet its deadline. In a detailed discrete event simulation, we have tested many different distributions with a wide range of C^2 and shapes, as well as for single and dual processor system. Results of four distributions are reported here. We compare with RR and FCFS, and find that in all distributions studied our algorithm performs best. In the study of the use of two slow servers versus one fast server, we have discovered that they provide comparable performance, and in a few cases the double server system does better.

Keywords: *Simulation, Residual Time Scheduling, Coefficient of Variation, M/G/C Queue, Processor Sharing*

1. Introduction

There have been innumerable papers written on how to optimize performance in queueing systems. They cover any number of different goals and conditions. From our point of view they fall into two categories (with many possibilities in between): 1) The service demand of each customer is known, and; 2) The service demands of the individual customers are not known, but the probability distribution function (pdf) of those demands is presumed to be known. So, for instance, the mean and variance, (the expectation, or mean, of the deviation squared of the variable from its expected value or mean) over all customers might be known, even though the time needed by a particular customer is not known until that customer is finished.

Within each of these categories there are many possible goals. For instance, the goal might be to minimize the system time, as in job processing in computer systems, or minimize the waiting time, as in setting up telecommunication links. However, both of these would be useless in situations where there are critical deadlines, as would be the case in airport security lines, where it would be more important to maximize the fraction of passengers who get through in time to catch their flights, or in the case of web server where the millions of users surfing the Internet, are unwilling to spend more than few seconds waiting for a web page to be displayed. One of the major metrics to be considered for a high performance web server is the number of pages downloaded per second by a web server. In this paper we consider the second category with the goal of maximizing the number of jobs that meet their deadline. We compare several queueing disciplines for systems with Poisson customer arrivals to a service station, with a given service time distribution. The distributions we consider include: Uniform, Exponential, hyper exponential and hyper-Erlangian. The distributions were chosen to show a wide range of coefficient of variation of service times (C^2) which is the ratio of the standard deviation of the service time to its mean.

The numerous scheduling disciplines proposed for multiprogrammed systems, for the most part the evaluation has been based on workloads showing a relatively low variability in the service requirements of jobs. In other words when $C^2 \le 1$. However, reports from va-

* Partially funded by CSU-AAUP Summer Research Grant 2010-2011

Copyright © 2010 SciRes.

rious high performance computing centers show that the variability in service time demands can be quite high. In a detailed workload characterization study [1] in NASA Ames research center facility reports that the overall C^2 is 7.23. In another study the authors reported that C^2 observed on a weekly basis on the CM-5 at the university of Wisconsin is as high as 6 and some measurements in Cray YMP sites ranges C^2 from 30 to 70 [2].

Our particular research contribution is in using residual times as a criterion for selecting jobs for immediate service. For the purpose of comparisons the other service disciplines we consider are: First Come First Serve (FCFS), Round-Robin (RR). FCFS discipline is used in present day routers [3]. It is interesting that these disciplines have all been studied in the context of mean system time and have been shown to have the same mean as that for the M/M/1 queue for all service time distributions [4]. When $C^2 > 1$, RR outperforms FCFS. If $C^2 \leq 1$, one should stick to FCFS. The famous Pollaczek-Khinchin formula gives the mean system time for FCFS M/G/1, which has been tabulated in **Table 1**. Even though, different system time distributions may have the same mean and variance (and thus the same system time for jobs), they may have different qualities for meeting deadlines. As far as we know, this aspect has not been studied previously.

A job arrives with a service time and deadline requirement. As time evolves, both: 1) the residual service time, depending on the amount the job has already serviced, and 2) the time remaining until the deadline, change. Consequently, the state variable of such a dynamic queueing system is of unbounded dimension, which may make an exact analysis extremely complicated. Thus, we are motivated to approach with a simulation study.

The principle underlying the proposed dynamic scheduling discipline is based on the fact that, the variability of service demands both in uniprocessor and multiprocessor scheduling plays a significant role in determining the best scheduling policy especially when one does not have the exact knowledge of individual service time demands. In this paper we investigate how the various disciplines used in the present research behave under high variability in service demands and explore ways the proposed discipline can be adapted to better cope with this condition to be implemented in web server scheduling. We also compare the performance of single and double server systems with the same maximal capacity. The technique used to evaluate the scheduling disciplines is discrete event simulation. Useful analytic models are difficult to derive as the precise and subtle distinctions between several disciplines are complex to model. However, we made comparisons with analytic results that are available [4,5].

Table 1. Job turn-around time obtained from simulation, for RR policy with $\rho = 8$

Distributions	Turn-around times
uniform	4.99
exponential	4.985
Hyper exponential	4.96
Power tail	4.89

In this paper we have developed a dynamic scheduling algorithm for real time tasks using task residual execution time, $rm(t)$. By definition $rm(t)$ is the expected remaining execution time of a job after receiving a service time of t time units. In summary, our Residual Time Based (RTB) algorithm provides a higher percentage of deadline makers than any of the other disciplines considered. Our results also show that for some distributions, two slow processors are at least marginally better than one fast processor.

The structure of this paper is as follows. In Section 2 we discuss the related works. Section 3, provides with a mathematical description of residual times of the different distributions used. In Section 4 we explicitly illustrate our prototype scheduling system, together with our algorithm for using residual times. In Section 5 we explain the simulation experiments and upon which we base our conclusions. We also present the results of the simulations experiments we have at this time, and discuss their implications. Finally, in Section 6 we draw the conclusions of the paper.

2. Literature Review

2.1 The Task Scheduling Problem

Given the enormous amount of literature available on scheduling, any survey can only scratch the surface. Moreover, a large number of scheduling approaches are used upon radically different assumptions making their comparison on a unified basis a rather difficult task [6,7]. At the highest level the scheduling paradigm may be divided into two major categories: *real time* and *non real time.* Within each of the categories scheduling techniques may also vary depending on whether one has a precise knowledge of task execution time or the pdf.

2.1.1. Scheduling in Non Real Time Systems

In the non real time case the usual objective is to minimize the total execution time of a program and increase the overall throughput of the system. Topcouglo *et al.*, provides a classification of the proposed algorithms [8]. The algorithms are classified into a variety of categories such as, list-scheduling algorithms [9], clustering algorithms [10], duplication based algorithms [11] and guided random search methods. Genetic Algorithms (GAs) [12] are one of the widely studied guided random search techniques for the task scheduling problem. Although they provide a good quality of schedules, the execution time of Genetic Algorithms is significantly higher than the other alternatives. Several researchers have studied the problem of scheduling, which is known to be NP-complete in its general form, when the program is represented as a task graph [8]. The execution time of any task and the communication cost between any pair of processors is considered to be one time unit. Thus, they have considered only deterministic execution time. Adam *et al.* [13] studied both deterministic and stochastic models for task execution time and made a comparison of different list scheduling algorithms. For the latter, task execution is limited to the uniform distribution. Kwok and Ahmed [9] focused on taxonomy of DAG scheduling algorithms based on a static scheduling problem, and are therefore only partial. Some algorithms assume the computational cost of all the tasks to be uniform. This simplifying assumption may not hold in practical situations. It is not always realistic to assume that the task execution times are uniform. Because the amount of computations encapsulated in tasks are usually varied. Others do not specify any distribution for task execution time. They just assume any arbitrary value.

2.1.2. Scheduling in Real Time Systems

Scheduling algorithm in RT applications can be classified along many dimensions. For example: periodic tasks, pre-emptable non-pre-emptable. Some of them deal only with periodic tasks while others are intended only for a-periodic tasks. Another possible classification ranges from static to dynamic. Ramamaritham and Stankovic [14] discuss several task scheduling paradigms and identify several classes of scheduling algorithms. A majority of scheduling algorithms reported in the literature, perform static scheduling and hence have limited applicability since not all task characteristics are known a priori and further tasks arrive dynamically. Recently many scheduling algorithms [15] have been proposed to dynamically schedule a set of tasks with computation times, deadlines, and task requirements. Each task is characterized by its worst-case computation time deadline, arrival time and ready time.

Ramamritham in [16] discusses a static algorithm for allocating and scheduling subtasks of periodic tasks across sites in distributed systems. The computation times of subtasks represent the worst-case computation time and are considered to be uniformly distributed between a minimum (50 time unit) and maximum (100 time unit) value. It is assumed that the execution of each subtask cannot be preempted. This way of selecting the execution time range (from 50 to 100) reduces C^2 to a very small value (= 4/27), which the authors may not have realized. Chetto and Chetto [17] considered the problem of scheduling hard periodic real time tasks and hard sporadic tasks (tasks that arrive are required to be run just once [14]). In addition to computation time and period, each periodic task is characterized by its dynamic remaining execution time. But the authors have not mentioned how to obtain the remaining execution time in a dynamic manner. Liu and Layland [18] were perhaps the first to formally study priority driven algorithms. They focused on the problem of scheduling periodic tasks on a single processor and proposed two preemptive algorithms. The first one is called Rate-Monotonic (RM) algorithm, which assigns static priorities to periodic tasks based on their periods. The second one is called the Earliest Deadlines First (EDF), a dynamic priority assignment algorithm. The closer a tasks' deadline, the higher the priority. This again is an intuitive priority assignment policy. EDF and RM schedulers do not provide any quality of service (QoS) guarantee when the system is overloaded by overbooking. Since, EDF [18] does not make use of the execution time of tasks, it is difficult to mix non-real-time and real-time tasks effectively. Scheduling both real-time and non real-time tasks under load requires knowing how long a real-time task is going to run. As a result, some new/different scheduling approach that addresses these limitations is desirable. One can determine the latest time the task must start executing if one has the knowledge of the exact time of how long a task will run. Thus, the scheduler may delay a job to start executing in order to increase the number of jobs meeting deadlines. In this paper, we investigated this issue and developed a dynamic scheduling strategy, which includes both the deadline and the estimated execution time.

For dynamic scheduling with more than one processor, Mok and Destouzos [19] show that an optimal scheduling algorithm does not exist. These negative results point out the need for new approaches to solve scheduling problems in such systems. Those approaches may be different depending on the job service time distribution (job size variability). In this paper we have addressed this issue by running simulation experiments for several distributions for both single and multiple processor systems.

3. Terminology and Mathematical Background

In this section we first define the job execution model. Then we present the mathematical expressions for residual times for different job service time distributions. Let X be the job execution time which can be modeled either by deterministic or stochastic distribution. In the present paper, we consider the latter. Task execution time can follow any standard distribution, for example: uniform, exponential, hyper exponential, etc, or take discrete values with associated probabilities.

Before that we show the well-known Pollaczek-Khinchin formula, as $\bar{n} = \frac{\rho}{1-\rho} + \frac{\lambda^2}{1-\rho}\left(\frac{C^2-1}{2}\right)$ which states that the mean number of jobs in a system, $\bar{n}$ depends only on their arrival rate λ, the mean, $E[X]$ and variance σ^2 of the service time distribution. In the formula it is expressed in terms of the coefficient of variation $C^2 = \sigma^2 / (E[X])^2$.

3.1 Job Execution Model

We assume that job arrival follows a Poisson process. Tasks are independent and identically distributed and preemptable. Each job is characterized by the following: task arrival time A_i, task execution time E_i task deadline D_i, task start time, S_i. A job is ready to execute as soon as it arrives, regardless of processor availability. Tasks may have to wait for some time, W_i, before they start executing for the first time due to scheduling decision. As a result, we have $S_i = A_i + W_i$, where $W_i \geq 0$.

By definition, the Cumulative Distribution Function (CDF) is, $F(x) = P(X \leq x) = \int_0^{x+} f(x).dx$. The Reliability function, $R(x) = P(X > x) = \int_{x^+}^{\infty} f(x).dx$ where $R(x) + F(x) = 1$. The expected execution time, $E(x) = \int_0^0 xf(x).dx = \int_0^{\infty} R(x)dx$. The conditional reliability, $R_t(x)$, is the probability that the task executes for an additional interval of duration of x given that it has already executed for time t, $R_t(x) = R(t+x)/R(t)$. By definition, the task residual time [20] $rm(t)$ given that the task has already executed for t time units is; $rm(t) = \int_0^{\infty}(R(t+x)/R(t))dx$. Here we show different expressions for residual time for different types of distributions.

Uniform: Here the task execution time follows the uniform distribution with CDF, $F(x) = \frac{x}{2T}$. The reliability function, $R(x) = \frac{2T-x}{2T}$ and $C^2 = 1/3$. The residual time, after time t is $rm(t) = (2-t)/2$.

Exponential: Here the task execution time follows the exponential distribution with CDF $F(x) = 1 - e^{-\mu x}$. The reliability function, $R(x) = e^{-\mu x}$ .The expected value, $E(x) = \frac{1}{\mu}$ and $C^2 = 1$ Where, $\frac{1}{\mu}$ is referred to as the mean service time. The residual time at time t can be shown to be equal to $rm(t) = \frac{1}{\mu}$. Due to the memory-less property of exponential distributions the residual time, does not change with time.

Hyper Exponential: In this paper we have considered a two stage hyper-exponential distribution with CDF $F(x) = 1 - p_1 e^{-\mu_1 x} - p_2 e^{-\mu_2 x}$ and $p_2 = 1 - p_1$. The reliability function $R(x) = p_1 e^{-\mu_1 x} + p_2 e^{-\mu_2 x}$. There are three free parameters, namely p_1, μ_1 and μ_2, where $T_i = 1/\mu_i$. Now, we define the free parameters [5] $\gamma = \frac{(C^2-1)}{2}$ and $p_1 = \frac{2}{C^2-1}$, $T_1 = \bar{x}(1+\sqrt{p_2\gamma/p_1})$ and $T_2 = \bar{x}(1+\sqrt{p_1\gamma/p_2})$. Hyper exponential distribution can be used when it or desired that $C^2 > 1$. The residual time can be shown to be

$$rm(t) = \frac{p_1 T_1 e^{-t/T_1} + p_2 T_2 e^{-t/T_2}}{p_1 e^{-t/T_1} + p_2 e^{-t/T_2}}$$

Hyper Erlangian: We have considered a four stage Hyper Erlangian distribution with pdf, $f(x) = p_1[\mu_1 (\mu_1 x)e^{-\mu_1 x}] + p_2[\mu_2(\mu_2 x)e^{-\mu_2 x}]$ and CDF, $F(x) = 1 - p_1(1+\mu_1 x)e^{-\mu_1 x} - p_2(1+\mu_2 x)e^{-\mu_2 x}$.

The reliability function $R(x) = p_1(1+\mu_1 x)e^{-\mu_1 x} + p_2(1+\mu_2 x)e^{-\mu_2 x}$. There are three free parameters, namely p_1, μ_1 and μ_2, where $T_i = 1/\mu_i$. For the definition of those parameters we refer [5]. Hyper Erlangian distributions can be used when it is expected that $C^2 > 1$.

The residual time can be shown to be $rm(t) = \frac{p_1 T_1[2+t/T_1]e^{-t/T_1} + p_2 T_2[2+t/T_2]e^{-t/T_2}}{p_1(1+1/T_1)e^{-t/T_1} + p_2(1+1/T_2)e^{-t/T_2}}$ In this case the residual time at first decreases, then increases greatly and then decreases gradually to $\max\lceil T_1, T_2 \rceil$.

4. The Scheduling Model

Each process or task or job is associated with an execution time. **Figure 1** shows the schematic for the present

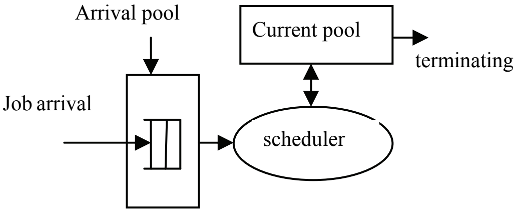


Figure 1. The Scheduling Model

scheduling model. Upon arrival, a task joins the arrival pool. The duration of those execution times of the processes are not known in advance. We assume they follow a distribution function, which can either be deterministic or non-deterministic. A task will have a deadline that represents a soft timing constraint on the completion of the task.

The scheduler will pick a process to execute from the arrival pool depending on the scheduling policy. A process joins the current pool as soon as it is picked by the scheduler to run. Note, the CPU/s is/are being shared by several concurrent processes/tasks. The current pool keeps the record of execution time already taken of the chosen tasks (concurrent tasks). Once picked, each task i will execute for a time slice called time quantum, q_i. At the end of time : q_i.

1) the executing task is either completed or partially executed. A partially executed task goes back to the current pool, whereas, a completed task leaves the current pool.

2) the scheduler then chooses the next task either from the arrival pool or the current pool.

We define a function called risk factor (*rf*). As soon as a process i starts executing, time starts running out before it reaches deadline. At clock time C_T the time left before deadline is $D_i - C_T$. Let $rm(t)$ be the remaining execution time required to finish the given process I, which has to fit in time interval to meet the deadline. Let us define the risk factor for process i as follows: $rf(t) \; = rm(t) / (D_i - C_T - A_i)$. In other words,

$$rf \; = \frac{residual \;\; time}{remaining \;\; time \;\; before \;\; deadline}$$

4.1 RTB Algorithm

We propose a dynamic scheduling algorithm called Residual Time Based (RTB) which incorporates the task residual time. Task risk factor is used as a measure to choose the next task to execute. Tasks in the task set are maintained in the order of decreasing risk factor.

```
The proposed Residual Time Based (RTB) algorithm:
1) As soon as a job arrives.
    a) compute estimated residual time for the job.
    b) compute risk factor (rf) for the job.
    c) append the job to the queue.
2) Select the job that has the highest rf and execute it for the next quantum.
    a) compute the end time if the job is complete at or before the end of quantum.
3) At the end of each quantum.
    a) compute estimated residual time for each job in the queue.
    b) compute rf for each job in the queue.
```

The risk factor is inversely proportional to the difference between task deadline and the clock time. The risk factor increases as the task approaches to its deadline. As soon as the clock time passes the deadline the *rf* becomes negative if the task has not yet been completed. The goal in the present research is to increase the number of tasks meeting the deadline. As the clock time passes the deadline the task is considered to be less important compared to tasks that are very close to deadline but have not yet crossed the deadline. Therefore, tasks with positive risk factor are given higher priority than tasks with negative risk factor.

4.2 Why does RTB Bring Enhancement

The Let us consider the P-K formula from Section 3. When $C^2 = 1$ $\bar{n} = \frac{\rho}{1-\rho}$. When $C^2 > 1$ $\bar{n} > \frac{\rho}{1-\rho}$. When $C^2 < 1$ $\bar{n} < \frac{\rho}{1-\rho}$ and FCFS gives the least waiting time. However, as C^2 tends to increase beyond 1, the number of jobs, $\bar{n}$ keeps on increasing over $\frac{\rho}{1-\rho}$. So it is unwise to use FCFS. Mean number of jobs in a queue, $\bar{n}$, for PS is $\frac{\rho}{1-\rho}$ which is the same as M/M/1 queue. In practice, PS is implemented by RR with finite time slicing. In RR jobs form a queue upon arrival and the server picks one after another from the queue in a RR fashion without addressing how much time left of the job to finish.

In RTB a function *rf* is devised to let the scheduler pick a job with minimum amount of residual time to finish. In other words shorter jobs are favored in RTB. We know when $C^2 > 1$ there are more short jobs than long ones. As a result, more jobs will be complete in RTB than in RR which is clearly demonstrated in our simulation results presented in the next section.

5. Simulation and Results

5.1 Simulation

The proposed Residual Time Based (RTB) algorithm has been evaluated through discrete event simulation under various task arrival rates and task service time distributions. A stochastic discrete event simulator was constructed to implement the operation of RTB, FCFS, and RR policy. The simulation code is developed in Microsoft visual C environment, but using Linux 48 bit random number generator. In this paper we have considered exponentially distributed inter arrival times (*i.e.*, a Poisson arrival process). As each arrival a new task is created with various attributes such as, arrival time, service time, deadline, residual time, run time, etc. The following cases of job service time distributions have been investigated:

- Uniform distribution with $C^2 = 1/3$
- Exponential distribution with. $C^2 = 1$
- A two stage hyper exponential distribute with $C^2 = 10$
- A four stage Hyper Erlangian with $C^2 = 10$

The distributions were selected to give a wide variety of coefficient of variation, C^2. This would also yield a wide variety of mean system time, but with the same deadline. In all cases the mean service time per job is 1.0, the deadline is 4, and $\rho = 0.8$ (*i.e.*, the processor is busy for 80% of the time).

Validation of the Simulator: Several test cases were run to validate the performance of the simulator. As a first set of tests, the simulator has been set to run a FCFS algorithm with exponential arrival time, and exponential service time. The output results were compared with the analytical results: process turn-around time calculated from the Pollaczek-Khinchin formula [5, 7] for M/M/1/FCFS and for $\rho = 0.8$ is $3\frac{2}{3}$. Our simulation result gives 3.66. Our RTB algorithm is a Processor-Sharing algorithm. So, we further verified the simulator by running the common Round Robin Processor-Sharing algorithm for M/M/1. Again, process turn-around time for $\rho = 0.8$ can be calculated from the Pollaczek-Khinchin formula to be 5 for M/M/1/RR. Our simulation yields the job turn-around for the same scenario as 4.985. For any M/G/1 queue RR should yield the same value (=5) for the system time as obtained from M/M/1 [7]. Table 1 shows the turn-around time obtained from the simulation.

5.2 Performance of RTB vs. FCFS and RR in a Single Server System

In this section we compare the performance of RR, FCFS and RTB algorithms. **Figure 2** depicts our simulation results for three different scheduling policies namely, FCFS, RR and RTB when the job size distribution is considered to be uniformly distributed. The graph shows the fraction (normalized) of the number of jobs that have been finished by time *t*, after they arrived. The x-axis represents the request turn-around time. The y-axis represents the normalized number of jobs completed on or before the corresponding turn-around time, shown by the x-axis. The cusp of the RTB graph indicates the request deadline, which is 4 times the mean service time. Here the mean service time is unity. For uniform distribution, there is not much variability within the request size. FCFS appears to be a better choice than RR in the sense that a higher fraction of number of jobs finishes earlier when they are serviced using FCFS policy as compared to when they are serviced using RR policy.

This again validates the fact that using FCFS is detrimental when $C^2 > 1$. Instead, a processor-sharing algorithm should be used to improve system performance, whether the performance parameter is the system time (non real time case) or the number of jobs meeting deadline (real time case). As one may observe in figure 2 that the plot for RTB is rising faster than that for FCFS and RR policy, indicating that more jobs (87.4%) can be processed before the deadline when they are scheduled by the RTB policy as opposed to either FCFS (66.6% jobs satisfied before deadline) or RR (58.8% jobs satisfied before deadline) policy. Moreover, the plot for FCFS rises above the plot for either RTB or RR as the turn-around time passes the deadline. This indicates that longer jobs (those that did not have an opportunity to meet deadline), can get a better chance to finish beyond the deadline when they are serviced by either RTB or RR. In the world of real time jobs one of the major goals is to increase the number of jobs meeting the deadline, which is accomplished better if the RTB scheduling policy is used instead FCFS or RR. **Figure 3** depicts the same as **Figure 2** but for exponentially distributed job service times. In exponential distribution, $C^2 = 1$, so the job size variability is higher than that in uniform distribution. The deadline is 4 time units as in the previous case. RTB algorithm outperforms both RR and FCFS algorithms in

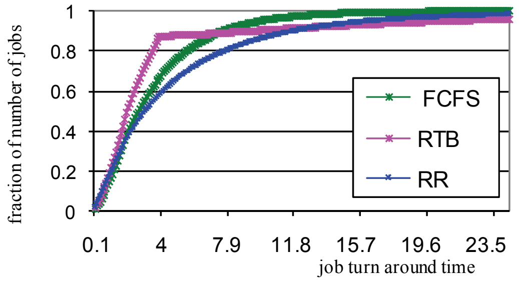


Figure 2. Fraction (normalized) of number of jobs vs. turn-around time, *t*, for uniform job service time distribution for a single processor case and for FCFS, RR, and RTB algorithms

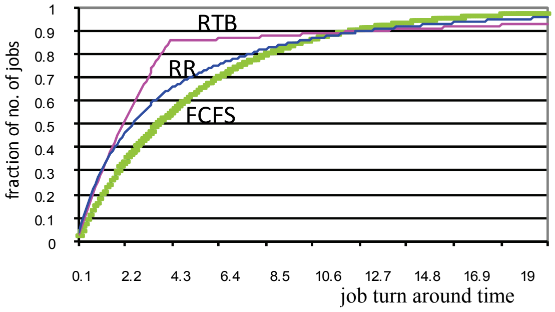


Figure 3. Fraction (normalized) of number of jobs vs. turn-around time, *t*, for exponential job service time distribution for a single processor case and for FCFS, RR, and RTB algorithms

Copyright © 2010 SciRes.

terms of number of jobs finishing before deadline. **Table 3** shows that when service time is exponentially distributed, 85.99% finishes before deadline if RTB is used (single processor case).

Whereas, using RR and FCFS only 65.4% and 55.4% of jobs meet their deadlines, respectively (single processor case). Note: job turnaround time for the 3 scheduling policies obtained from simulation are very close to each other and also close to the value of the. They are 4.95, 4.985, and 4.94 time units for FCFS, RR, and RTB policy, respectively (**Table 2**). Turn-around time for exponential distribution obtained from P-K formula is 5 and the value obtained from simulation is 4.985. Moreover, as the deadline passes, longer jobs tend to get finished earlier if they are serviced using FCFS instead of either RTB or RR policy.

5.3 Performance of RTB vs. FCFS and RR in a Dual Server System

In this section we present results which compare the performance of RTB and RR algorithm in the presence of single and double processors. **Figure 4** depicts our simulation results when the job size distribution is considered to be exponentially distributed. The graph shows the fraction (normalized) of the number of jobs that have been processed by time t, after they arrived. The x-axis represents the request turn-around time. The y-axis represents the normalized number of jobs completed on or before the corresponding turn-around time, shown by the x-axis. In **Figure 4** there are two pairs. The upper pair represents RTB and the lower pair represents RR algorithms.

Within each pair the lower one is for two processor case. The two processors run at half the speed of the single processor. Thus, each job needs twice as much processor time, but the maximum capacity is the same. This is true for both RTB and RR algorithms, that one double speed processor is able to meet deadline for more jobs than two half speed processors, when the service time is assumed to be exponentially distributed. **Figure 5** depicts three pairs of plots when job service times are hyper exponential, with $C^2 = 10$. The pair with knees represents

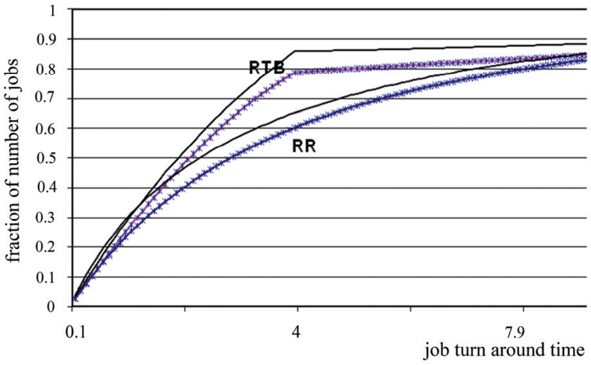


Figure 4. Fraction (normalized) of number of jobs vs. turn-around time, *t*, for exponential job service time distribution for a single and dual processor case and for RR and RTB algorithm

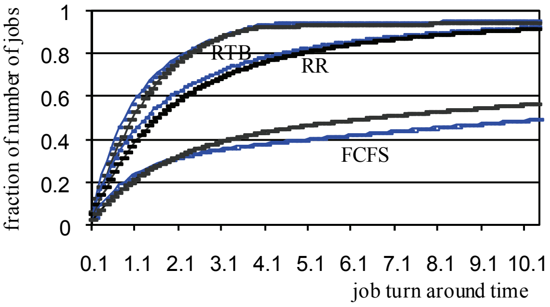


Figure 5. Fraction (normalized) of number of jobs vs. turn-around time, *t*, for hyper-exponential job service time distribution for a single and dual processor case and for FCFS, RR, and RTB algorithms

Table 2. Comparison of job turn-around time (for different job service time distributions) obtained from the P-K formula and the simulation results

Distribution Types	No. of processors	Turn-around time calculated from P-K formula	Turn-around time from simulation using		
			FCFS	RR	RTB
uniform	1	3.666	3.66	4.99	4.48
	2	not available	4.39	5.27	5.637
exponential	1	5	4.95	4.985	4.94
	2	5.55	5.51	5.38	5.51
hyper–exponential	1	23	22.23	4.92	4.098
	2	not available	19.71	5.48	5.078
Hyper Erlangian	1	23	23.3	4.97	4.05
	2	not available	20.9	5.54	4.24

RTB algorithm and the knee is at the deadline point, which in all cases is 4 time units. The lower pair is for FCFS algorithm and the middle pair is for RR algorithm. The jagged plots are for 2 processors. In this simulation p_1 and p_2 are considered to be 0.047733 and 0.952267, respectively, whereas, T_1 and T_2 are considered to be 10.4749 and 0.525063, respectively.

It is clear from the graph that our RTB algorithm performs better in satisfying the deadline. More jobs can meet their deadline when they are scheduled by RTB algorithm than when they are scheduled either by FCFS or RR algorithm. $C^2 = 10$, means job variability is higher than that in uniform or exponential distribution. More jobs can meet their deadlines if they are serviced using RR (a processor sharing algorithm) as opposed to being serviced by FCFS algorithm. When FCFS is used long jobs are occupying the CPU longer, thus short jobs do not get a chance to run. This again validates the fact that using FCFS is detrimental when, $C^2 > 1$. Instead, a processor sharing algorithm should be used to improve system performance, whether the performance parameter is the system time (non real time case) or the number of jobs meeting deadline (real time case). **Figure 6** depicts the relationship between the fraction of the number of jobs with turn-around time less than or equal to t with their corresponding turnaround time when job service time is represented by hyper-Erlangian distribution with $C^2 = 10$. The values for probabilities p_1 , p_2 and the corresponding times T_1 and T_2 are 0.0477, 0.952, 6.12026 and 0.218281, respectively. There are two pairs of plots shown in **Figure 6**. The lower pair represents RR and the upper sets represents RTB algorithm. In the case of RR the upper plot shows the turnaround time when there is only one processor is available, whereas the lower plot shows the same when there are two processors are available. Our RTB algorithm performs better than RR in satisfying deadline. More than 90% jobs can meet their deadline when they are scheduled by RTB algorithm either by a single or a double processor (also see **Table 3**). Whereas, only 80.7% and 79.1% jobs can meet their deadline if they are serviced using RR policy, by a single processor and a double processors, respectively. FCFS policy can meet the deadlines for less than or equal to 42.3% jobs only. In between the two plots for RTB the jagged one and the smoother one represent the two and one processor case, respectively. According to **Table 3** double and single processor can meet deadlines for 95% and 93.1% of jobs respectively. The 2 processors run at half the speed of the single processor. It is very important to note that two half-speed processors can meet deadlines of as many jobs as one processor with double the speed.

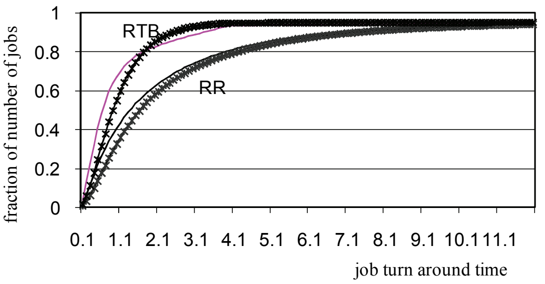


Figure 6. Fraction (normalized) of number of jobs vs. turn-around time, *t*, for hyper-Erlangian job service time distribution for a single and dual processor case and for RR, and RTB algorithms

6. Conclusions and Future Extensions

In this paper we have proposed a residual time based dynamic real time scheduling algorithm. We used discrete event simulation to evaluate the performance of the RTB policy and compare it with FCFS and RR policies. We have investigated several distributions, namely, uniform, exponential, hyper exponential, hyper Erlangian. The distributions were selected to give a wide variety of C^2 and shapes. This would also yield a wide variety of mean system time, but with the same deadline. In all cases the mean service time per job is 1.0, the deadline is 4, and $\lambda = 1.25$.

Table 3. Improvement of RTB over FCFS and RR

Distribution Types	No. of processors	% of jobs satisfying D in		
		FCFS	RR	RTB
uniform	1	66.6	45.2	87.4
	2	57.0	41.8	78.7
exponential	1	55.4	65.4	85.99
	2	49.5	60.3	78.9
hyper–exponential	1	37.9	78.0	93.7
	2	43.0	76.0	92.5
Hyper Erlangian	1	33.9	80.7	93.1
	2	42.3	79.1	95.0

Copyright © 2010 SciRes.

The simulation results demonstrated that for all service time distributions considered in the paper, RTB enables more jobs to meet their deadlines, which is one of the major goals in real time jobs. Our results show that when job size variability is higher ($C^2 = 10$) more than 93.7% of the jobs are able to meet their deadline when they are serviced using RTB algorithm, whereas, only 37.9% and 78% of the jobs can meet their deadline when they are serviced by FCFS and RR policy, respectively.

We have also presented results when there are two servers present where the total capacity is the same as the one server previously considered. There are some important implications in our study for dual server scheduling. For hyper exponential distributions two half-speed processors allow the same number of jobs to meet their deadlines as one double speed processor. For the hyper Erlangian distribution, two half-speed processors together do a little better than one double speed processors. As a side effect the mean system of RTB is better than RR when $C^2 = 10$ .We are not claiming that our residual time based algorithm, RTB, is the optimal one, but certainly using residual time as a criterion to select job could be very useful. The performance of RTB shows consistent improvements across the various distributions studied in the present project indicating the robustness of RTB.

As a future extension we plan to investigate RTB on web server scheduling especially for dynamic web requests when responses are created on the fly. Web file sizes have been shown to exhibit highly variable distributions, and our present research involved job size with $C^2 = 10$. Results reported here show RTB favors short jobs without penalizing long jobs too much, eventually suggesting it is worthy to investigate RTB in web server performance.

7. Acknowledgements

Thanks to Adam Chase for help in formatting the document.

REFERENCES

[1] D. G. Feitelson and B. Nitzberg, “Job Characteristics of a Production Parallel Scientific Workload on the NASA Ames iPSC/860,” In: D. G. Feitelson and L. Rudolph, Eds., *Job Scheduling Strategies for Parallel Processing*, IPPS’95 Workshop, Santa Barbara, *Lecture Notes in Computer Science*, Springer, Vol. 949, 1995, pp. 337-360.

[2] S.-H. Chiang, R. K. Mansharamani and M. K. Vernon, “Use of Application Characteristics and Limited Preemption for Run-To-Completion Parallel Processor Scheduling Policies,” *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, Nashville, 1994, pp. 33-44.

[3] R. A. Idris, U.-K. Guillaume and W. E. Biersack, “Analysis of LAS Scheduling for Job Size Distributions with High Variance,” *Proceedings of ACM SIGMETRICS*, San Diego, 10-12 June 2003, pp. 218-228.

[4] F. Baskett, K. M. Chandy, R. Muntz and F. G. Palacious, “Open, Close and Mixed Networks of Queues with Different Classes of Customers,” *Journal of the ACM*, Vol. 22, No. 2, 1975, pp. 248-260.

[5] L. Lipsky, “Queueing Theory: A linear Algebraic Approach (LAQT),” 2nd Edition, Springer Verlag, New York, 2008.

[6] T. L. Casavant and J. G. Kuhl, “A Taxonomy of Scheduling in General Purpose Distributed Computing Systems”, *IEEE Transactions on Software Engineering*, Vol. 14, No. 2, February 1988, pp. 141-145.

[7] K. G. Shin and P. Ramanathan, “Real Time Computing: A New Discipline of Computer Science and Engineering,” *Proceedings of the IEEE*, Vol. 82, No. 1, 1994, pp. 6-24.

[8] H. Topcuoglu, S. Hariri and M.-Y. Wu, “Performance Effective Low Complexity Task Scheduling for Heterogeneous Computing,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 3, March 2002, pp. 260-274.

[9] Y. Kwok and I. Ahmed, “Static Scheduling Algorithm for Allocating Directed Task Graphs to Multiprocessors,” *ACM Computing Surveys*, Vol. 31, No. 4, 1999, pp. 406-471.

[10] T. Yang and A. Gerasoulis, “DSC: Scheduling Parallel Task on an Unbounded Number of Processor,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 9, 1994, pp. 951-967.

[11] I. Ahmed and Y. Kwok, “A New Approach to Scheduling Parallel Programs Using Task Duplication,” *Proceedings of the International Conference on Parallel Processing* (*ICPP*), St. Charles, Vol. 3, August 1994, pp. (II)47-51.

[12] E. S. H. Hou, N. Ansari and H. Ren, “A Genetic Algorithm for Multiprocessor Scheduling,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 2, 1994, pp. 113-120.

[13] T. L. Adam, K. M. Chandy and J. R. Dickson, “A Comparison of List Schedules for Parallel Processing Systems,” *Communications of ACM*, Vol. 17, No. 12, 1974, pp. 685-690.

[14] K. Ramamritham and J. Stankovic, “Scheduling Algorithms and Operating Systems Support for Real Time Systems,” *Proceedings of the IEEE*, Vol. 82, No. 1, 1994, pp. 55-67.

[15] K. Ramamritham, I. Stankovic, *et al*., “Efficient Scheduling algorithms for Real time Multiprocessor Systems,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 1, No. 2 April 1990, pp. 184-194.

[16] K. Ramaritham, “Allocating and Scheduling of Precedence-Related Periodic Tasks,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 6. No. 4, April 1995, pp. 412-420.

[17] H. Chetto and M. Chetto, “Some results of the Earliest

Deadline Scheduling Algorithm," *IEEE Transactions on Software Engineering*, Vol. 15, No. 10, October 1989, pp. 1261-1269.

[18] C. L. Liu and J. W. Layland, "Scheduling Algorithm for Multiprogramming in a Hard Real Time Environment," *Journal of the ACM*, Vol. 20, No. 1, 1973, pp. 46-61.

[19] K. Mok and M. L. Dertouzos, "Multiprocessor Scheduling in a Hard Real Time Environments," *Proceedings of the 7th Texas Conference on Computing Systems*, Houston, 1978, pp. 5-12.

[20] S. K. Trivedi, "Probability and Statistics with Reliability, Queuing, and Computer Science Applications," Prentice Hall, Englewood Cliffs, 1982.

Copyright © 2010 SciRes.

J. Software Engineering & Applications, **2010, 3, 756-760**
doi:10.4236/jsea.2010.38087 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

A Novel Regression Based Model for Detecting Anemia Using Color Microscopic Blood Images

Saif AlZahir, Han Donker

University of N. British Columbia, Prince George, Canada.
Email: {zahirs, donker}@unbc.ca

Received May 7th 2010; revised June 27th 2010; accepted July 1st 2010.

ABSTRACT

Modeling human blood components and disorders is a complicated task. Few researchers have attempted to automate the process of detecting anemia in human blood. These attempts have produced satisfactory but not highly accurate results. In this paper, we present an efficient method to estimate hemoglobin value in human blood and detect anemia using microscopic color image data. We have developed a logit regression model using one thousand (1000) *blood samples that were collected from Prince George Hospital laboratory. The output results of our model are compared with the results of the same sample set using CELL-DYN* 3200 *System in Prince George Hospital laboratory, and found to be near identical. These results exceed those reported in the literature. Moreover, the proposed method can be implemented in hardware with minimal circuitry and nominal cost.*

Keywords: *Regression Models, Logit Regression Model, Anemia, Blood Components, CBC*

1. Introduction

Color image analysis techniques are employed in a wide range of medical applications including human blood testing. Blood testing refers to laboratory analysis of blood. A variety of blood tests are accessible to provide information about the condition and status of the human body. The most regular test is the complete blood count (CBC). CBC is a series of tests used to appraise the composition and concentration of the cellular components of blood [1]. For example, anemia is a disease caused by the reduction of red blood cells (RBC) count and/or the hemoglobin (Hgb) level in human blood [2].

In general, a microscopic color image is a multi-spectral image with one band for each of the three primary colors (red, green and blue), assuming that we are working with RGB color model. Based on this understanding, color images are produced by a weighted combination of these three primary colors for each pixel. The color of an image also depends on the light source illuminating the image object and on the color of the surrounding region of that object or simply the ambient. At present, color images are extensively used in a wide range of applications and are exploited by researchers and developers in almost every aspect of real life applications.

In this paper, we present a new method for estimating hemoglobin level in human blood to detecting anemia using microscopic color images.

The remaining part of this paper is organized as follows: In Section 2, we introduce the current methods for testing CBC and present previous work related to cell count, cell analysis, and hemoglobin estimation to provide the necessary information and appropriate background for this research. In Section 3 we present the proposed model, and finally, in Section 4 we provide our results and conclusions.

2. Previous Related Work

In this section, we briefly describe the main methods used to calculate or estimate human blood components. **Table 1** depicts male and female hemoglobin components and their normal ranges as analyzed by CELL-DYN 3200 System. The CELL-DYN 3200 system is popular and is being used by Prince George Regional Hospital (PGRH) Laboratory where we obtained the blood samples and with which we will compared our output results.

There are two main types of procedures to compute CBC and hemoglobin level: 1) manual; and 2) auto-

Table 1. Hgb reference intervals for CELL-DYN 3200 System

Parameter	Normal range
Hemoglobin	132–159 g/L for male * 123–151 g/L for female

* The content is taken from Reference [3]

Copyright © 2010 SciRes.

mated. The following are some examples on both procedures.

2.1 Manual Procedures for Determining Hemoglobin in Blood

A manual also called spectrometric procedure is straightforward and requires the use of spectrophotometer. The spectrophotometer is a device that measures the monochromatic light transmitted through a solution to determine the concentration of the light absorbing substance in that solution. The light from the lamp passes through the prism, which allows light of only a predetermined wavelength to pass through the cuvette. The transmitted light strikes a detector, where it is converted into electrical energy and presented to the readout device [4]. This procedure is satisfactory but requires the presence of an attendant and consumes a significant amount of time to perform. In short, this method is acceptable but not efficient nor economic.

2.2 Automated Procedures for Determining Hemoglobin in Blood

The CELL-DYN 3200 System is a modern automated CBC analyzer and hemoglobin estimator. It combines spectrophotometry and modified cyanmethemoglobin method for hemoglobin determination. In the cyanmethemoglobin method, the whole blood is mixed with a solution of potassium ferrycyanide to convert hemoglobin in the ferrous state, to methemoglobin in the ferric state, which then reacts with potassium cyanide to form cyanmethemoglobin. This final product is also called hemiglobincyanide (HiCN). HiCN is very stable and has a wide absorption maximum of about 540 nm. The absorption of the solution at 540 nm is directly proportional to the amount of Hemoglobin present in the blood [5]. The CELL-DYN 3200 system measures hemoglobin within a sample in a hemoglobin Flow Cell. The hemoglobin dilution that is analyzed in the Flow Cell is a mixture of the sample plus reagent. The system takes five reference readings. The lowest and highest readings are discarded, and the remaining three readings are averaged to produce the final hemoglobin reading. This is done to eliminate the extreme values. The hemoglobin dilution is analyzed at 555 nm in the hemoglobin Flow Cell; the system receives, and then saves the results [6].

At present, both CELL-DYN 3200 and CELL-DYN 4000 systems are being used in hospital laboratories in around the world. Both systems are considered to be reliable. As for the cost of the equipment, both systems are considered to be expensive and require attendants presence compared to the proposed method which is highly efficient and inexpensive.

3. The Proposed Model

In this section, we introduce the proposed method for detecting anemia. A flow chart of this method is shown in **Figure 1**. This method comprises of two main components. The first is concerned with the development of the logit regression model, (LRM) shown in the upper rectangle of the figure and the second component is concerned with testing of new blood samples, shown in the lower rectangle of the figure. The major elements of the first component are: 1) blood samples collection; 2) capturing microscopic images of the blood samples; 3) image preprocessing and color information gathering; and 4) the development of the logit regression model. The second component handles detecting anemia, if it exists, in new blood samples. The process of doing that are similar to those in the model development component except for the last step which is the employment of the regression, LRM, model. The following is a brief description of the proposed method:

3.1 Blood Samples Collection

To build our regression model, we have collected one thousand (1000) blood samples from Prince George Regional Hospital Laboratory, British Columbia. These samples are randomly chosen from the general public and we arranged them in a specific numbering scheme so as to protect their identity. The blood samples were smeared on glass slides by the hospital-automated system. At that point, we captured microscopic color images of the samples using a digital camera mounted on a microscope with 10 × magnification. Our original plan was to use the whole size of sample image for processing but as most of the sample images were distorted at the borders due to the staining process and the nature of blood smea-

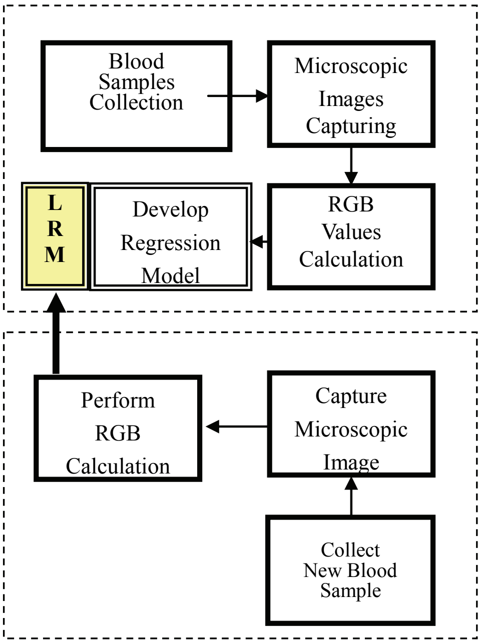


Figure 1. Flow diagram of the proposed scheme

ring, our choice was to select a segment that is uncorrupted of a window size (256 × 256 pixels) from each image. The choice of the segments was done randomly. The clipped images are saved in a separate file for further processing. **Figure 2** shows six (6) clipped images from the blood samples. The samples shown in the figure are of samples 1, 5, 6, 10 and 15.

3.2 Blood Color Image Analysis

For each sample image, we calculate the pixels' color information as a function of red, green, and blue (f(R, G, B)) using Matlab 7 software. To do so we created three planes that represent the three colors (red, green, blue) of the pixels' values of the images. Then, the average of all red values, green values, and blue values of the three planes were calculated to produce three single values for each image: the red value, R, the green value, G, and the blue value, B. These three values were stored in a matrix of size 3 × N, where N is the number of images as shown in **Figure 2**. In our case N is equal one thousand (1000) and hence the size of our matrix is 3 × 1000. In the **Figure 3** the values r_1, g_1, b_1 belongs to sample one 1 and r_l, g_l, and b_l belongs to the last sample, which is sample number 1000.

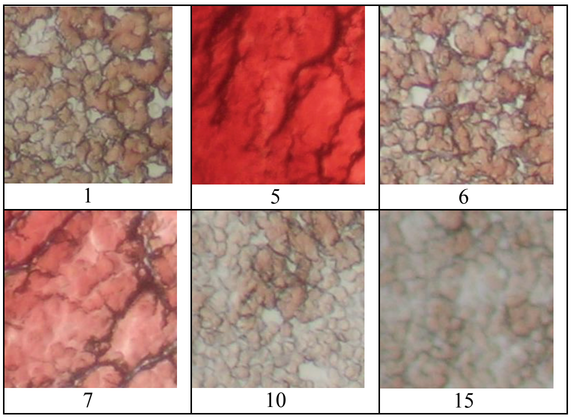


Figure 2. Six blood test samples

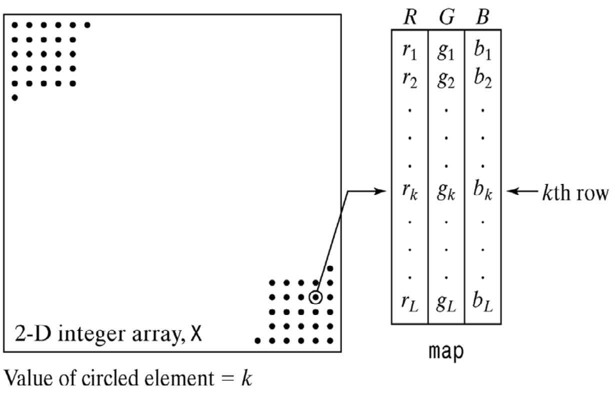


Figure 3. RGB analysis of each sample image

3.3 Building the Model

Prior to model the data, *i.e.* the RGB values, we have represented them visually to acquaint ourselves with some of their statistics and properties. Fore example, we plotted their histogram as shown in **Figure 4.**

As expected, in most cases, the red color values (plotted in dark blue) is the highest value, followed by green color values (plotted in pink), and finally the blue color values (plotted in yellow) is the smallest value of the three components. Examining the statistics of the hemoglobin values of the test samples supplied by the Prince George Hospital Lab. including the maximum and minimum values of the set, we found that the set maximum value is 185 g/L, and the minimum value is 57.30 g/L, and the average value is 132.51 g/L. This statistics helps determine the range of the set, which facilitates modeling the data. The hemoglobin values of 900 samples histogram is plotted and shown in **Figure 5** and the last 100 samples are plotted and shown in **Figure 6** for clarity and ease of tracking.

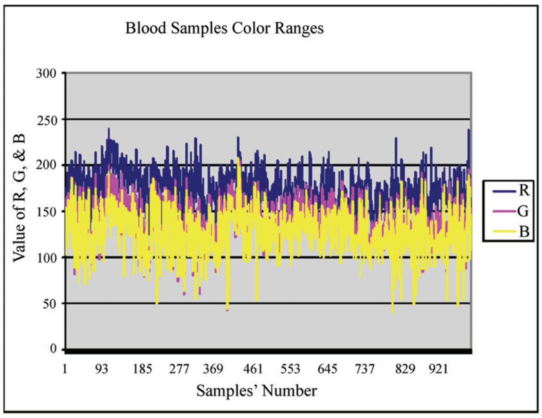


Figure 4. Histogram of the blood samples color ranges

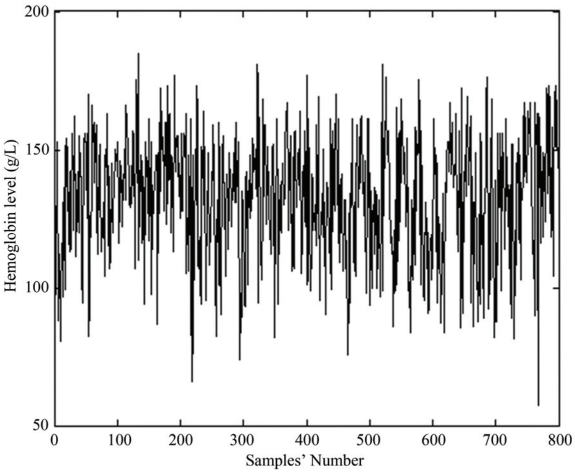


Figure 5. Histogram of the hemoglobin values of the 900 blood samples

Copyright © 2010 SciRes.

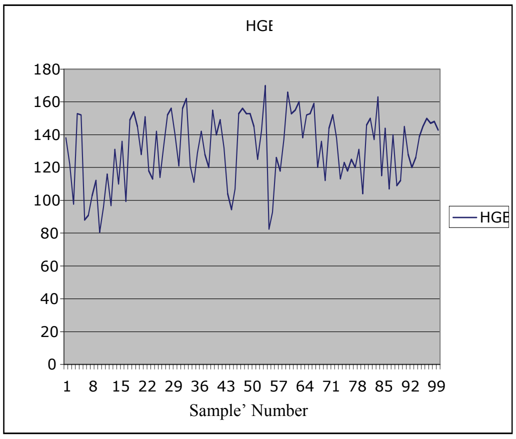


Figure 6. Histogram of the last 100 blood samples

From the data and the graph in **Figure 5**, we found that there are 10 samples outside the range (75-to-175 g/L) in the whole set which represent 1.0% of the total sample set. Based on this observation, we introduced an upper and a lower cut-off threshold values to eliminate those extreme values on both end of the range. We did that by equating any Hgb value greater than the upper threshold value to 175 g/L and any value that is less than the lower threshold to 75 g/L. Doing so does not alter any sample value group affiliation (*i.e.*, from high hemoglobin (healthy) to low hemoglobin (anemic). In addition such action will create a range of 100 (*i.e.*, 175-75g/L), which is easy to use.

At this point, several modeling methods were considered. In this paper, we used Eviews-5 software to produce the Logit regression model for this sample. Using Eviews software and the values of the R, G, and B, of the samples and if the sample indicates anemia or not, we produced the following Logit model:

$$L=(e^{-1.922 + 0.206R - 0.241G + 0.012B})/(1 + e^{-1.922 + 0.206R - 0.241G + 0.012B})$$

Where L is the hemoglobin level and R, G, and B are the color values of the blood sample.

3.4 Detecting Anemia

To determine if a person is anemic, we take a few drops of his or her blood, smear it on a glass and take its picture using a simple digital camera. Then we calculate the R, G, and B values of the image. We plug the values of the R, G, and B values in our LRM model and find the result. This method is simple and cost effective

4. Experimental Results

CBC has been a target for several automations attempts by researchers from different fields. None of the published paper attempted to model Hgb via regression models and images analysis. Zahir and Chowdhry [7] have presented a combined method that is based on artificial neural network (ANN) in conjunction with color image analysis. They presented results that are far better than those reported in [8]. In this research, Hgb value is sought to determine anemia. They reported that the network is trained with fixed training rate of 0.4 and for accuracies of 20% and 15%. They claim that accuracies below 15% demand more computing time. They added that the computing time is about 20 hours to realize an accuracy of 5%. The authors did not include explanation to support their claims nor that did they explain how can they train the NN to improve the results by 10% to 15% by increasing the computing time.

To test the performance and effectiveness of the LRM model, we have tested one thousand samples using this Logit model and compared them with the results of the hospital results and found to be almost identical. **Table 2** depicts the results we obtained and the number of faulty samples in each of the hemoglobin ranges shown. This Table shows the number of total errors is 65. This produces a 93.5% efficiency of the model. Out of this total 38 samples were in the hemoglobin range of 90-100. This range is a borderline level and in most cases doctors recommend that patients redo the test at a later time and recommend a specific diet. If these samples are not counted, then the efficiency of our model will be increased from 93.5% to 96.2 % accuracy. In addition, only 5 samples were at error when the person was anemic and the results shows he is not. Although this is a serious mistake but it rarely occurred and represent only 0.005% of the total sample.

5. Conclusions

The literature is scarce when it comes to simple, economical, and reliable automated methods for diagnosing anemia. There are, however, few methods like cyanmethemoglobin, which is reliable but very expensive and the WHO Hemoglobin Color Scale (HCS) method, which is inexpensive but not so reliable. In this paper, we

Table 2. Results of the logit regression model

HEMOGLOBIN MODEL RESULT			
No.	Hgb Range	# of Samples	# of Errors
1	0-80	8	1
2	80-90	29	4
3	90-100	66	38
4	100-110	81	15
5	110-120	104	4
6	120-130	134	2
7	130-140	186	2
8	140-175	392	0
Total		**1000**	**65**

Copyright © 2010 SciRes.

introduced a new logit regression based model that uses image analysis data to detect anemia. The simulation results of the proposed model are significantly higher than the published results. For a set of 1000 sample, our results show an accuracy of 96.2% if we do not consider 38 samples on the borderline between anemic and not anemic which is a grey area for all testing methods. Considering all samples, our model accuracy is 93.5%. In addition to its high accuracy, the proposed model is easy to implement and inexpensive. This model can be build in hardware at a cheep cost.

6. Acknowledgements

The authors would like to thank Northern Health Authority of British Columbia, Prince George Regional Hospital Laboratory for their continued support to this research.

REFERENCES

[1] J. B. Henry, "Clinical Diagnosis and Management by Laboratory Methods," W. B. Saunders, Philadelphia, 2001.

[2] http://www.nlm.nih.gov/medlineplus

[3] M. Rendell, M. Anderson, W. Schlueter, J. Mailliard, D. Honigs and R. Rosenthal, "Determination of Hemoglobin Levels in The Finger Using Near Infrared Spectroscopy", *Journal of Clinical & Laboratory Haematology*, Vol. 25, No. 2, April 2003, pp. 93-97.

[4] J. P. Greer, J. Foerster, J. N. Lukens, G. M. Rodgers, F. Paraskevas and B. Glader, "Wintrobe's Clinical hematology," 11th Edition, Lippincott Williams & Wilkins, Philadelphia, 2003.

[5] G. Ongun, U. Halici, K. Leblebicioglu and V. Atalay, "Feature Extraction and Classification of Blood Cells for an Automated Differential Blood Count System," *International Joint INNS-IEEE Conference on Neural Networks*, Washington DC, Vol. 4, 2001, pp. 2461-2466.

[6] CELL-DYN 3200 System Training Guide, 2004.

[7] S. Zahir, C. G. Rejaul and W. Payne, "Automated Assessment of Erythrocyte Disorders Using Artificial Neural Network," *IEEE International Symposium on Signal Processing and Information Technology*, Vancouver, 2006, pp. 776-780.

[8] H. Ranganath and N. Gunasekaran, "Artificial Neural Network Approach in Estimation of Hemoglobin in Human Blood," *International Computer Engineering Conference on New Technologies for the Information Society, ICENCO*, Cairo, 2004, pp. 341-344.

J. Software Engineering & Applications, 2010, 3, 761-766
doi:10.4236/jsea.2010.38088 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

The Design and Implement of TCP/IP Protocol Cluster on AVR Singlechip

Rong Pan, Hai Zhao, Jialiang Wang, Dan Liu, Penghua Cai

School of Information Science and Engineering, Northeastern University, Shenyang, China.
Email: panrong1012@126.com

Received June 3rd 2010; revised June 30th 2010; accepted July 6th 2010.

ABSTRACT

With the rapid development of the embedded technology, research and implement of the Internet of things will be a new technology revolution, yet the implement of the Internet of things is on the base of the communication between the things. For this reason, realizing the function of communication between singlechip is particularly important. Based on the characteristics of the embedded microcontroller, we analyzed the traditional PC TCP/IP protocol, and appropriately tailored TCP/IP protocol cluster on the basis of the characteristics of embedded singlechip. At last, we realized the reduced TCP/IP protocol cluster suitable for embedded singlechip, on AVR singlechip platform.

Keywords: *The Internet of Things, Webit, Embedded System, TCP/IP Protocol, Ethernet*

1. Introduction

The Internet of things means a kind of net that via information sense equipment such as FRID, infrared sensor, GPS, laser scanner and so on, in arranged protocol, join up between anything and the Internet to communicate information and realize intelligent identification, tracing, monitoring, and management. The concept of the Internet of things is suggested in the year of 1999. It is the "Internet communicated with things". It means two aspects: the first one is, the core and foundation of the Internet of things is still in the Internet, which based on, extending and expanding the Internet; the second one is, its client side extends and expands to anything, to make the information exchanging and communication [1-3].

With the rapid development of computer and network technology, Internet has become an important means of information transmission, more and more embedded equipments are necessary to achieve the Internet's network [4-5]. Relative to the PC, computing and storage resources of embedded systems are relatively limited; therefore to achieve all the TCP/IP protocol cluster in the embedded singlechip is quite unrealistic. So that, in order to save the system resources and ensure the reliability of the system, under the condition of improving the performance of embedded system, it's necessary to targeted modular simplify TCP/IP protocol.

2. Adoptive Equipment and Testing Platform

Webit is an overall solution that makes the equipments intelligent and networking. It is the new network equipment system structure with the elements of Internet and its basic idea is an independent, low-cost 3 W server embedded in equipment, to make the equipment has independent network intelligence.

Webit is an embedded Internet product decided by Liaoning Provincial Key Laboratory of Embedded Technology by themselves. Webit 1.0 is successfully pass technical appraisal and the trademark registration in the year of 2000, and Webit 2.0 (Internet non-standard electrical equipment access server) passed the appraising meeting of scientific and technological achievements held by science and technology commission of Liaoning province in May 2001. Considering that webit is AVR 8 bit singlechip, its storage unit is very limited, therefore it's very important to design a kind of TCP/IP protocol cluster suitable for the products.

The performance of Webit 2.0 as follows:

- Without depending on PC system structure;
- Using Atmel AVR RISC processors;
- User-defined Web pages;
- User-defined CGI programs used to control;
- 14 bit I/O interface (TTL level);
- TTL level UART supported 115200 bps;
- 10 M Ethernet interface (RJ-45);
- System programming (ISP);

Overview of Ethernet controller chip RTL8019AS in Webit:

RTL8019AS is a highly integrated Ethernet controller, it can simply answer plug and play NE2000 compatible

Copyright © 2010 SciRes.

adapter, which has two-fold and power decrease characteristics. Through the three-level control characteristic, RTL8019AS is the best ideal choice for network equipment GREEN PC in all already known things. The two-fold function can simulate send and receive the spread between twisted-pair and all two-fold Ethernet switches. This not only can make bandwidth stronger from 10 Mbps to 20 Mbps, but also avoid read muliaccess agreement because of Ethernet channel fight character. Microsoft's plug and play function can alleviate user lower income and focus on the adapter resources, such as the input and output, IRQ, memory address, etc. However, in special application without plug and play function of compatibility, RTL8019AS support s JUMPER and JUMPERLESS options.

In order to provide complete plug and play solution, RTL8019AS integrated 10BASET transceiver, and auto-examination function between AUI and BNC interface. In addition, 8 IRQ BUS and 16 basic addresses BUS provide comfortable environment for large resources situation.

RTL8019AS supports 16 k, 32 k and 64 k bytes memory BROM and the flash memory interface. It provides the page model function, which can only support 4 M bytes BROM under 16 k bytes of memory system space. BROM’s useless commands are used to release BROM memory space. RTL8019AS designed the singlechip with the 16 k bytes SRAM, so that not only provides more friendly function, but also saves SRAM storage resources.

3. The design of Webit Reduced TCP/IP Protocol Stack

In the AVR singlechip, due to the relatively limited resources, the TCP/IP protocol cluster of complete function cannot be achieved. So according to the characteristics of AVR singlechip, we cutting the original TCP/IP protocol cluster obtains the reduced TCP/IP protocol cluster.

Meanwhile, based on the architecture of TCP/IP protocol stack, we adapted the design method of network slice model. The architecture of TCP/IP protocol cluster after simplify contains the ARP, IP, ICMP, UDP, TCP protocol processing model, etc. [6-8]. Each layer of the architecture of TCP/IP protocol stack is designed as a module of independent function, handles their data. Different modules can be invoked by function to turn over datum to upper or lower processing module [9]. **Figure 1** shows the simplified TCP/IP protocol architecture.

Known by **Figure 1**, when the AVR singlechip receives data from network, data packet processing modules will base on certain condition to choose the ARP module of link layer or the IP protocol module of network layer to process.

Likewise, when the data packets are processed by UDP and TCP protocol modules of transport layer, it will turn over processed packets to IP protocol processing module, and make encapsulation for IP first address, the first (such as the fields like address, type of agreement, etc.) by corresponding function calls. Then transfer the datagram including IP first and TCP first to the lower layer by function call, until the data is sent smoothly. The TCP/IP protocol processing is shown in **Figure 2**.

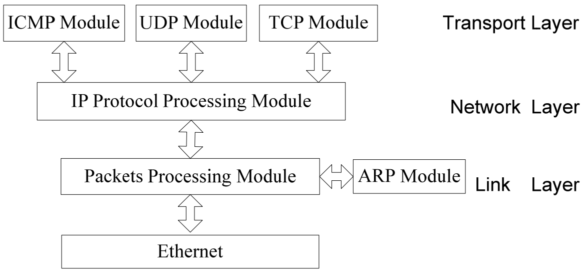


Figure 1. Simplified TCP/IP protocol architecture

4. The Design and Implement of Simplified Embedded TCP Protocol

First, in Webit, we format and size of the MAC and IP address, system address configuration, and size of the buffer were defined already. We make the address format of MAC, IP to become fixed value in system. The system configuration is used for setting specific value of the IP address, the port and MAC address. In this system, for the limited data needed the singlechip to process, so we don’t set the buffer larger than normal.

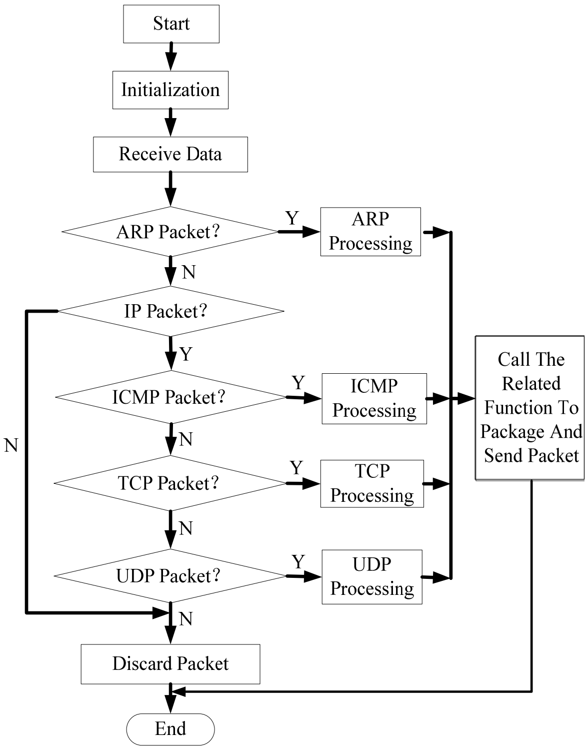


Figure 2. The TCP/IP protocol processing

Copyright © 2010 SciRes.

```
.DSEG
.ORG      0x60
LocalMAC:     .BYTE    6
LocalIP:      .BYTE    4
LocalPort:    .BYTE    2
RemoteMAC:    .BYTE    6
RemoteIP:     .BYTE    4
RemotePort:   .BYTE    2
Plugdelaytime: .BYTE    32
TCPCB:        .byte  30*2
RevBuffer:    .BYTE    260
```

4.1 The Implement of ARP Protocol

Because the embedded singlechip is normally in the service of the passive state. So while we design and implement the ARP protocol, we don't implement the function of address mapping table, neither realize the function of querying any client mapped IP into the MAC address, only need to achieve when other client to query the local Mac address. Packet and get feedback of the relationship between own IP and MAC address, and send.

When the embedded singlechip receives ARP packets from Ethernet, we according to the type of operation codes of the packets decide type of ARP packet, if the ARP request packet, compare destination IP address field of ARPP packet with the local settings of IP address. If it's equal, local MAC address packaging to responded ARP reply packet, if not, don't do processing, discard it. The processing flow of ARP packet is shown in **Figure 3**.

4.2 The Implement of IP Protocol

The IP protocol is the core of the TCP/IP protocol cluster. All the ICMP, UDP and TCP data transmit as IP datagram format. In the IP protocol processing modules, While implement the IP protocol module, we firstly received the IP date packet from the Ethernet and decide whether the destination IP address field values in the head of datagram equals local IP address, if not, discard; if consistent, check such field as the version number and checksum of the IP datagram, etc.

After examination, confirm the packet is right, and then decide to choose ICMP protocol, UDP protocol or TCP to submit to upper processing, according to the type of IP data. In addition, another function of IP protocol module we designed and implemented is to make the message encapsulation delivered from upper into IP data, then turn over IP data encapsulation to link layer to make data frame encapsulation and sending. The processing flow of IP protocol is shown in **Figure 4**.

4.3 The Implement of ICMP Protocol

ICMP protocol is a kind of information transfer control protocol. We think about the embedded singlechip as a server is responded the client commonly, as a passive device, it does not need to initiatively send back the message. So we only implement the receiving and handling the Echo Request between singlechip and other devices in the ICMP protocol module, and also send the Echo Reply. The implementation of ICMP protocol is as follows: read type code of the first byte of ICMP data packets, and check the ICMP packet types. If the type code is 8, the type of packets will be modified to 0, fill each field of packets to make the encapsulation of Echo Reply packets needed to be sent back, finally calls SendIP function, make ICMP data packets into IP datagram encapsulation to send. If the packets' type code isn't 8, discard the packet. The processing flow of ICMP protocol is shown in **Figure 5**.

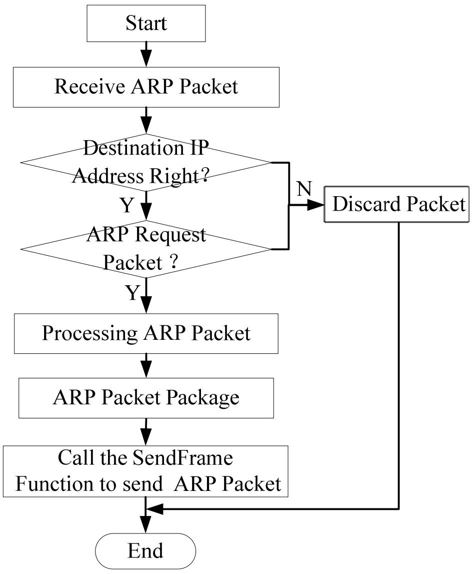


Figure 3. The ARP protocol processing

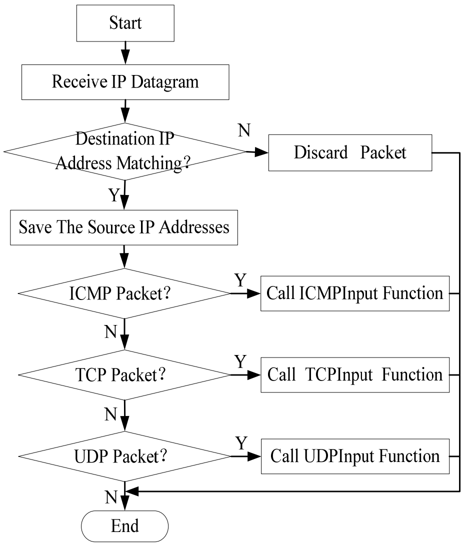


Figure 4. The IP protocol processing

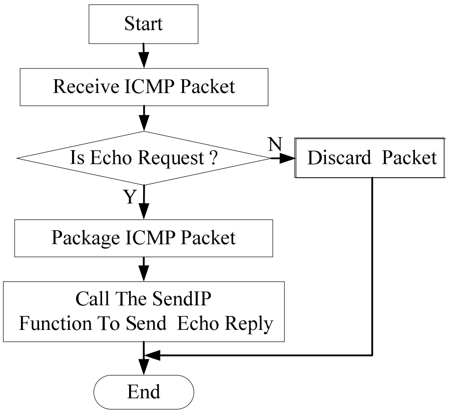


Figure 5. The ICMP protocol processing

4.4 The Implement of UDP Protocol

UDP protocol provides reliable, connectionless communication between applications, it transmits datum to the IP layer and sends out, but does not guarantee they can reach the destination.

When the UDP protocol modules receive packets, first locate the port fields of UDP packets, save the remote and purpose port of the UDP packets, then the compare the objective port of the packet with the port of local regulations, if not equal, discard it; if equal, call the corresponding function. Finally, set the source port, objective port, data length, checksum field in the header of UDP packet, add datum to be sent, make encapsulation and sending by IP layer. The processing flow of UDP protocol is shown in **Figure 6**.

4.5 The Implementation of TCP Protocol

Due to the limited resources of singlechip, and to handle TCP packet better, so while implementing the TCP protocol module, we reduced the common TCP/IP protocol, and did not implement the sliding window protocol, flow control and congestion control mechanisms. Meanwhile, we set two TCP connection control block in the TCP protocol modules, and adopt the response mode with single window. When receiving TCP packets, first locate mark field of TCP packets, if the TCP packet is required to build a new connection, check whether still exist spare TCP connection control block in the system. If present, this spare control block will be used as the control block for this connection, and establish connections. Conversely, if there is no spare TCP connection control block, and do nothing.

When the mark field of the TCP packets is another type, search whether exist TCP connection control block corresponding to the TCP packets. If present, judge according to the mark field of the value of SYN, FIN, ACK and so on, then choose corresponding function to process packets. If don't exist TCP connection control block corresponding to the TCP packets, don't do anything. Construction and closing the connection of the TCP protocol are through the “three handshakes” and “four times specific wave”. Setting the mark field in the TCP packets to different control bits is the specific approach. The processing flow of TCP protocol is shown in **Figure 7**.

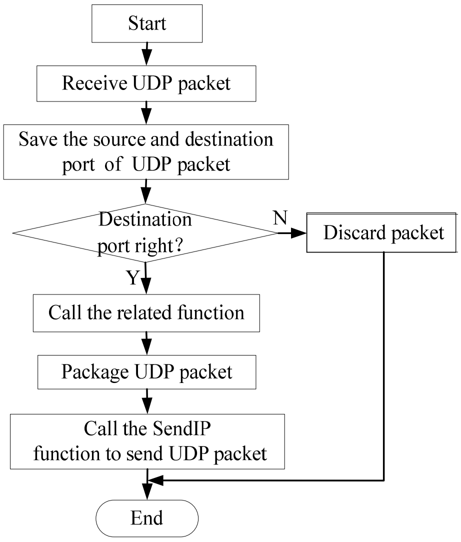

Start
Receive UDP packet
Save the source and destination port of UDP packet
Destination port right?
N
Discard packet
Y
Call the related function
Package UDP packet
Call the SendIP function to send UDP packet
End

Figure 6. The UDP protocol processing

5. Testing

In order to test whether the TCP/IP protocol realized can achieve the desired objective, we carried on a series of tests.

The Ping command is the most frequently used in net-

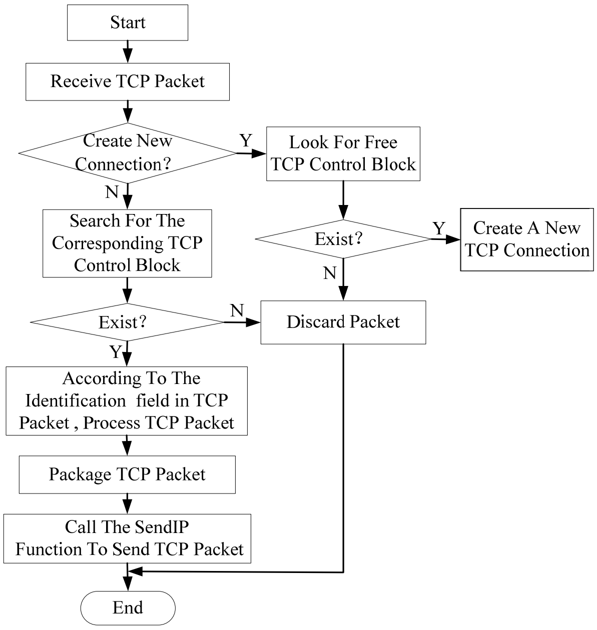

Start
Receive TCP Packet
Create New Connection?
Y
Look For Free TCP Control Block
N
Search For The Corresponding TCP Control Block
Exist?
Y
Create A New TCP Connection
N
Exist?
N
Discard Packet
Y
According To The Identification field in TCP Packet , Process TCP Packet
Package TCP Packet
Call The SendIP Function To Send TCP Packet
End

Figure 7. The TCP protocol processing

work. This command sends a network message and requests response via ICMP protocol. Therefore, through the Ping command we can determine whether the current network is connected correctly, and test whether the condition of network connections is available. So, for the testing of ARP, IP, and ICMP protocol, we can finish the testing through Ping command. The process of test is: first, connect Webit and PC, do the network configuration through WebitNetIfConfig function, make configuration of suitable IP address (in this test set IP of Webit as 192.168.180.94). Finally, input “ping 192.168.180.94” on the PC. The running test of **Figure 8** shows that the network is connected, network equipment is available. It states:

1) ARP module is normal, and can properly achieve the address mapping;

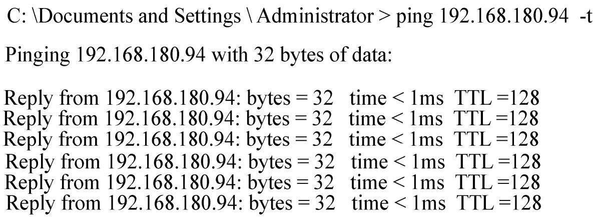
2) The IP protocol modules work normally, and can correctly analyze that this is an ICMP messages;

3) ICMP protocol modules work normally, and can correctly return responses message.

The response condition of ICMP packet is listed in the **Table 1**. The test is mainly by sending ping date packet to singlechip to verify the success rate of date sending. As the test date of the table shows, as the request packet sending to the singlechip is smaller, so the request can be effectively handled by server.

For the testing of TCP protocol, we can choose to write a simple Telnet-Server program based on TCP/IP protocol. Users can access on PC, according to system cue, users can input some simple commands to obtain relevant information. Testing method is: input “telnet 192.168.180.94” on PC. The result is: the system shows that the connection is established successfully, and the associated tip. Input something according to cue, and then obtain appropriate information. Finally, the test shows that TCP protocol modules in the TCP/IP protocol cluster are correct, and the network layer protocol is correct.

In addition, we also test the established time of TCP many times.



```
C: \Documents and Settings \ Administrator > ping 192.168.180.94  -t

Pinging 192.168.180.94 with 32 bytes of data:

Reply from 192.168.180.94: bytes = 32   time < 1ms  TTL =128
Reply from 192.168.180.94: bytes = 32   time < 1ms  TTL =128
Reply from 192.168.180.94: bytes = 32   time < 1ms  TTL =128
Reply from 192.168.180.94: bytes = 32   time < 1ms  TTL =128
Reply from 192.168.180.94: bytes = 32   time < 1ms  TTL =128
Reply from 192.168.180.94: bytes = 32   time < 1ms  TTL =128
```

Figure 8. The test results of ARP, ICMP and IP protocol

Table 1. Response of ICMP packet

Response of ICMP	Size	Request packets	Reply packets	Success rate
	128 Byte	50	50	100%

Table 2. The established time of TCP

The established time of TCP	Meanvalue	Standard deviation	Maximum
	2.314 /ms	0.53 /ms	2.437 /ms

As the test results listed in the **Table 2**, the average established time of TCP is about 2.314 ms, it means that server can quickly respond according to external request. In the process of test, while singlechip server accepting the connection request, it has well reliability. Mean while, it means after implementing the reduced TCP/IP protocol on the AVR singlechip, it can well meet the requirement of non-PC devices be connected to Internet.

For the testing of UDP protocol, we write a program according to the realized UDP protocol that creates two UDP Sockets, one is to achieve the function of sending UDP data, and the other is used to achieve the function of receiving UDP data, in order to test the correctness of the UDP protocol. The procedure of testing is:

1) Initialize the equipment;

2) Create the UDP Sockets (S1 and S2) through WebitUdpCreateSocket;

3) S1 sends UDP data to S2 through the function of WebitUdpSendTo, S2 receives data form S1 through the function of WebitUdpReceiveFrom, and output the received data to the serial port;

4) Close the two created Sockets (S1 and S2) through WebitUdpDestroySocket command.

Finally, the testing results show that S2 received data form S1 successfully; this means the UDP protocol of TCP/IP protocol cluster is correct.

6. Conclusions

This paper expounds the implement principle, method and technology of the TCP/IP protocol on WEBIT platform. In implement of the TCP/IP protocol we bring in the programming ideas of network layer and reduced TCP/IP system structure which adapt to the characteristics of singlechip [10]. At the same time, the paper also make certain explore and try in embedded singlechip and network applications finally, the feasibility of this protocol is confirmed in simulation experiment system, so it’s significant and full of reference value for embedded Internet system design and development.

REFERENCES

[1] G. Y. Xu, Y. C. Shi and W. K. Xie, “Pervasive Computing,” *Computer Journal*, Vol. 26, No. 9, 2003, pp. 1042-1052.

[2] D. A. Gregory and E. D. Mynatt, “Charting Past, Present and Future Research on Ubiquitous Computing,” *ACM Transaction on Computer-Human Interaction*, Vol. 7, No. l, 2002, pp. 29-58.

[3] H. Zhao and Y. Chen, “Pervasive Computing,” Northeastern University Press, Shenyang, 2005.

[4] H. Zhao, “Embedded Internet,” Tsinghua University Press, Beijing, 2002.

[5] T. KindBerg and A. Fox, “System Software for Ubiquitous Computing,” *IEEE Pervasive Computing*, Vol. 1, No. 1, 2002, pp. 70-81.

[6] V. Jonathan and P. Joseph, “Profiling and Reducing Processing Overheads in TCP/IP,” *IEEE/ACM Transactions on Networking* (*TON*), Vol. 4, No. 6, 1996, pp. 817-828.

[7] H. Jang, S.-H. Chung and D.-H. Yoo, “Design and Implementation of a Protocol Offload Engine for TCP/IP and Remote Direct Memory Access Based on Hardware/Software Coprocessing,” *Microprocessors & Microsystems*, Vol. 33, No. 5-6, 2009, pp. 333-342.

[8] J. F. D. Rezende, M. M. D. A. E. Lima, N. L. S. D. Fonseca, “Mobility over Transport Control Protocol/Internet Protocol (TCP/IP),” In: M. Llyas Ed., *The Handbook of Ad Hoc Wireless Networks*, CRC Press, Boca Raton, 2003, pp. 329-343.

[9] A. Dunkels, “Full TCP/IP for 8-bit Architectures,” *Proceedings of the* 1*st International Conference on Mobile Systems*, *Applications and Services*, San Francisco, May 2003, pp. 85-98.

[10] S. Ichiro, “Location-Aware Communication in Smart Spaces,” *Proceedings of* 2007 *International Conference on Multimedia and Ubiquitous Engineering* (*MUE* 2007), Seoul, 26-28 April 2007, pp. 1027-1034.

J. Software Engineering & Applications, **2010, 3, 767-775**
doi:10.4236/jsea.2010.38089 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

A Genetic Approach to Analyze Algorithm Performance Based on the Worst-Case Instances*

So-Yeong Jeon, Yong-Hyuk Kim

Department of Computer Science and Engineering, Kwangwoon University, Seoul, Korea.
Email: presentover@gmail.com, yhdfly@kw.ac.kr

Received June 29th 2010; revised July 15th 2010; accepted July 29th 2010.

ABSTRACT

Search-based software engineering has mainly dealt with automated test data generation by metaheuristic search techniques. Similarly, we try to generate the test data (i.e., problem instances) which show the worst case of algorithms by such a technique. In this paper, in terms of non-functional testing, we re-define the worst case of some algorithms, respectively. By using genetic algorithms (GAs), we illustrate the strategies corresponding to each type of instances. We here adopt three problems for examples; the sorting problem, the 0/1 knapsack problem (0/1KP), and the travelling salesperson problem (TSP). In some algorithms solving these problems, we could find the worst-case instances successfully; the successfulness of the result is based on a statistical approach and comparison to the results by using the random testing. Our tried examples introduce informative guidelines to the use of genetic algorithms in generating the worst-case instance, which is defined in the aspect of algorithm performance.

Keywords: *Search-Based Software Engineering, Automated Test Data Generation, Worst-Case Instance, Algorithm Performance, Genetic Algorithms*

1. Introduction

In search-based software engineering, researchers have been interested in the automated test data generation so that it would be helpful for testing the software. Since, in general, test data generation is an undecidable problem, metaheuristic search techniques have been used to find the test data. McMinn's survey [1] summarizes previous studies. In the part of non-functional testing, these studies had a bias to generate the test data that show the best/worst-case execution time. But if we analyze an algorithm, not the entire program, there can be many measures other than the execution time, in terms of non-functional testing.

Finding test data (or problem instances) for an algorithm is as important as finding those of the entire program. The reason is that algorithms in a program can affect the entire performance of the program and we can exploit problem instances for the algorithms in analyzing them. In fact, Johnson and Kosoresow [2] tried to find the worst-case instance for online algorithms, for the lower bound proof. Also, Cotta and Moscato [3] tried to find the worst-case instance for shell sort to estimate the lower bound of the worst-case complexity.

Nevertheless, to the best of our knowledge, such trials are currently quite fewer than those in the field of software testing. Of course, fore-mentioned trials are good as initiative studies. But the trials did not introduce various strategies for constructing metaheuristics to generate the worst-case instance.

Differently from the fore-mentioned studies, in this paper, we introduce various strategies to construct genetic algorithms (GAs) [4][1] in generating the worst-case (problem) instance which is defined in the aspect of algorithm performance. For this, we try to find the worst-case instance of three example problems for some algorithms; the (internal) sorting problem, the 0/1 knapsack problem (0/1KP), and the travelling salesperson problem (TSP). These are well-known problems and each can show different strategy to construct a GA. Since we tried not to use the problem-specific knowledge in the construction, our suggested GAs can be extended to generate the instances of other similar problems. For the sorting problem, we take as test algorithm not only shell sort but also many well-known

*The present Research has been conducted by the Research Grant of Kwangwoon University in 2010.

[1] For the sorting problem and the 0/1 knapsack problem, strictly speaking, we use memetic algorithms [5], which are GAs combined with local search. But we mainly focus on the GA rather than local search algorithm. Without classification, we just call our searching algorithm GA, in this paper.

Copyright © 2010 SciRes.

sorting algorithms; quick sort, heap sort, merge sort, insertion sort, and advanced quick sort. For the 0/1KP and the TSP, we test the algorithm based on a greedy approach, comparing to the known-optimal algorithms which are based on the dynamic programming.

The remaining part of this paper is organized in the following. In Section 2, we introduce GAs and the three adopted problems with their popular algorithms. Also we define the worst-case instance for each problem, in terms of non-functional testing. In Section 3, we present our GAs for finding the worst-case instance. In Section 4, we explain our experiment plan and the results. We make conclusions in Section 5.

2. Search Technique and Three Problems with Algorithms

2.1 Search Technique: Genetic Algorithms

The genetic algorithm (GA) [4] is a non-deterministic algorithm. A non-deterministic algorithm makes guesses, which may or may not be the answer of the problem the algorithm wants to solve. So, it consists of two phases; the guess phase and the evaluation phase. In the guess phase, guesses are made. In the evaluation phase, those guesses are evaluated by how close they are to the right answer. For evaluating guesses, the objective function is defined. This function takes a guess and returns the numerical value which indicates how close the guess is to the right answer. In other words, a deterministic algorithm searches for an object which maximizing the given objective function.

How does GA make a guess? By the principle of evolution. GA manages multiple guesses or *individuals*. We call the set of individual *population*. A new individual can be constructed by two operations; *crossover* and *mutation*. The crossover takes two individuals and returns one new individual. This new one is made by assembling each pattern of the two individuals. The mutation takes one individual and returns the individual which has slightly changed pattern from the taken individual. A pattern is found in the representation of the individual. Thus, the way an individual is represented is closely related to the way of making individuals. GA substitutes new individuals for some part of the population; this operation is called *replacement*.

How do we select individuals (in the population) for the crossover and the mutation as the input? Which individuals should we replace? By the *qualities* and the *fitnesses* [4] of the individuals. These are evaluated in the evaluation phase. The quality of an individual is the return value of the objective function. The fitness of an individual is evaluated using some part of the population or the entire population, as well as the individual to be evaluated. The fitnesses (not qualities) of the individuals are directly used to selection; fitness evaluation strategies are designed to increase the chances that some individuals with low qualities are selected. Note that using only qualities for selecting individuals can lose the diversity of the population and narrow the search range of GAs. Using the above operations, GA evolves the population until given stop condition is satisfied.

On the other hand, a *local search* algorithm can be combined with a GA. Given an individual, the local search tries to find out the individual with the best quality near the given individual. A GA combined with local search algorithms is called a memetic algorithm (MA) [5].

Since we want to generate the worst-case instance, each individual is a problem instance. Also, we should define the objective function so that this function returns the value indicating how close given individual is to the worst case. Thus the definition of the function depends on our definition of the worst case. Also, the way an instance is represented and strategies of crossover, mutation, replacement, fitness evaluation, and stop condition should be defined.

2.2 Sorting Problem

In the sorting problem, we are given an array of elements and their comparison operator. The correct solution of this problem is a sorted array in ascending (or descending) order. For the sorting problem, we assume that the comparison between elements takes so long time that it controls the total execution time. An instance of the problem is an array of elements to be sorted where the size of the array is fixed. Let the worst-case instance (array) be the instance that needs the most element-comparisons to sort. In this paper, we will test the following well-known sorting algorithms: quick sort, merge sort, heap sort, insertion sort, shell sort, and advanced quick sort.

Quick sort and merge sort [6] use the divide-and-conquer strategy. In the quick sort, a pivot element [6] is typically taken as the first element in the array to be divided into two partitions; we use the same method in the tested our quick sort. Heap sort [6] mainly uses a data structure called *heap* [6]. Insertion sort [6] inserts each element of the array into the already-sorted part of the array one by one. Shell sort [7] uses insertion sort repeatedly using *sequence of increments* [8]. In tested shell sort, we use the sequence as the reverse order of $\{h_n\}$, where $n \geq 0$ and $h_{n+1} = 3 \times h_n + 1$ with $h_0 = 1$, where the sequence is bounded above the size of sorted array. To improve the quick sort, the advanced quick sort [6] takes as a pivot element the median element of three elements which are randomly chosen from the array to be divided into partitions.

2.3 Zero/One Knapsack Problem

Let $item_i$ be an ordered pair (v_i, w_i) where v_i is its value and w_i is its weight. For a given set $S = \{item_1, item_2, \ldots,$

$item_n$}, we want to put items in S into the knapsack where the maximum capacity W is given as $c_w \times \sum w_i$ where i=1, 2, ..., n. Here $c_w \in (0, 1)$ is called the *weight coefficient*. The optimal solution of the problem is the subset A of S such that $\sum v_i$ ($item_i \in A$) is maximized and $\sum w_i \le W$ ($item_i \in A$), where $\sum v_i$ ($item_i \in A$) is called the *objective value*.

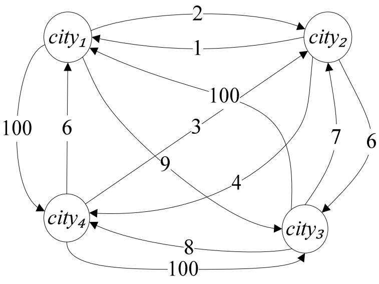
An algorithm based on the greedy approach [9] for this problem orders items in non-increasing order according to the 'value per unit weight' or *profit density*. Then, it tries to put the items in sequence satisfying that the total weight of the knapsack does not exceed W.

We will test the above algorithm in terms of the objective value. An instance of this problem is the set S where the size of S and c_w is fixed. Let the worst-case instance be the instance maximizing $(O-P)/O$, where P is the objective value obtained by the above algorithm and O is the objective value obtained by an optimal algorithm based on dynamic programming [9].

2.4 Travelling Salesperson Problem

A weighted, directed graph $G(V, E)$ is given, where $V = \{1, 2, ..., n\}$ is a set of cities. An edge $e(i, j) \in E$ weighted by c_{ij} represents that it costs c_{ij} to go from $city_i$ to $city_j$. An example is given in **Figure 1**. The optimal solution for the TSP is the tour, which is a Hamiltonian circuit of G, that takes the lowest (optimal) total cost if the tour exists. For our test, we assume that there always exists at least one tour although the tour is too expensive. The graph G can be represented as an $N \times N$ adjacency matrix as in **Figure 1**.

In a greedy approach we consider, we start from $city_1$. To make a tour, we visit the adjacent city to which we can go from the current city at the lowest cost under the restraint that the city to go has not been visited before. Once the tour is constructed, 2-opt [10] improves this tour. Since the given graph is directed, two tours are derived from a base tour at one move in 2-opt; one derived tour is the reverse order of the other derived tour.



$$C = \begin{pmatrix} 0 & 2 & 9 & 100 \\ 1 & 0 & 6 & 4 \\ 100 & 7 & 0 & 8 \\ 6 & 3 & 100 & 0 \end{pmatrix}$$

Figure 1. Two representations of an instance for TSP. NOTE. The cost to go from $city_1$ to $city_2$ is 2. This cost is the [1, 2]-th element of the matrix C. On the other hand, the cost to go from $city_2$ to $city_1$ is 1. This cost is the [2, 1]-th element of the matrix C

We will test an algorithm which uses the greedy approach with 2-opt in terms of the objective value (the tour cost). An instance of this problem is the graph $G(V, E)$ where the size of V is fixed. So, let the worst-case instance be the instance maximizing $(P-O)/O$, where P is the objective value obtained by the above algorithm and O is the objective value obtained by an optimal algorithm based on dynamic programming [9].

3. Genetic Algorithms

3.1 Framework and Things in Common

There exist some trials to design somewhat different GAs [11-13] rather than just adopting traditional design. We also adopt a non-traditional framework of GAs. In our framework, the initialization of a population is done first. Then the following procedure is repeated until the stop condition we set is satisfied: 1) Fitness evaluation is done first. 2) Two parents are selected from the population. 3) One new individual is created by the crossover of the two parents. The other two new individuals are created by the local search from the mutation of each parent; one individual from one parent and another from the other parent. Steps 2) and 3) are repeated until sufficiently many new individuals are created. 4) Some individuals are replaced from the population with the new individuals.

The strategy of the population initialization, fitness evaluation, selection, replacement, and stop condition are fixed in our GAs. The strategies of other operations are different for each problem; these strategies will be introduced in the next sections. The initialization of the population is based on random generation. For the fitness evaluation, we used one based on population sharing [4]. The formula is as follows:

$$F_i = f_i \,/ \sum_{j \in \{1,2,\ldots,n\}} s(d_{ij}), \qquad (1)$$

where

- F_i is the fitness of the i-th individual,
- f_i is given as $(C_i - C_{min}) + (C_{max} - C_{min}) / (k - 1)$,
- C_i is the quality of the i-th individual,
- C_{min} is the minimum quality among individuals,
- C_{max} is the maximum quality among individuals,
- k (k >1) is a *selective pressure*,
- n is the population size,
- $s(d_{ij})$ is given as $1-(d_{ij}/\sigma)$,
- d_{ij} is the distance between the i-th individual and the j-th one, and
- σ is the longest distance among all d_{ij}'s.

Note that our definition of the distance is slightly different for each problem but is based on the Manhattan distance. By examining the formula, we can see that the fitness evaluation strategy helps to select the individuals far from other individuals of the population and thus keep the diversity of the population.

For the selection, we use fitness-proportionate selection using roulette wheel [4]. In the replacement, individuals with low qualities are replaced with new individuals regardless of the qualities of new ones. The stop condition is satisfied if the population reaches the given maximum number of generations, or all the individuals of the population become 'the same'. Strictly speaking, 'the same' means that the distance between every pair of individuals is zero. We take the individual with the highest quality among individuals in the final population. Then we say that the GA found the individual. In this paper, individuals are (problem) instances for a test algorithm.

3.2 Sorting Problem

We restrict that an instance of the sorting problem is a permutation of N integers from 1 to N, where N is fixed for every individual in a GA. A permutation is represented as an array. The sorting algorithms we deal with are to sort the given permutation in ascending order (*i.e.*, 1, 2, 3, ..., N). For given sorting algorithm, the quality of an individual is the number of comparisons between elements needed to sort using the algorithm. Note that tested advanced quick sort uses pseudorandom-number generation and thus we repeat sorting the same permutation to take as the quality the average among the numbers of comparisons obtained by 50 repetitions. The distance between permutations A and B is defined as the sum of absolute differences between the numbers located in the same index of A and B.

As the crossover of the permutation encoding, we use PMX [14][2], which is popular. In our mutation of a permutation, say A, we decide a number at each index in A to be swapped with another number at a randomly chosen index. Whether or not to swap is according to given probability, say p_m. For the local search from a permutation, we consider each pair of numbers in the permutation. We try to swap them and test that the new permutation has better a quality than the original one. If it is true, then the new one is substituted for the original one. Otherwise, we try to swap other pairs until the local search reaches the given maximum count of improvements.

3.3 Zero/One Knapsack Problem

An instance of the 0/1KP is a list of N items. Here N is fixed for every individual in the population. We represent the list as an array. Note that an item is an ordered pair with value as its first coordinate and weight as its second coordinate; we represent an item as a two-dimensional point. We restrict the value and the weight are integers in [1, 100]. The quality of an individual is $(O-P)/O$, where P is the objective value obtained by the test algorithm and O is the objective value obtained by an optimal algorithm; the more details are described in Subsection 2.3. To get the distance between two individuals (or lists), we first arrange items of each list in order of their profit densities. This is for *normalization* [15]. Then we obtain the distance as the sum of the absolute differences between the profit densities of items at the same position in the two lists.

In the crossover of two individuals (or lists), we also rearrange items of each list as in the calculation of distance. We derive each item in the offspring from the items at the same position in both parents. The illustration of deriving the item is shown in **Figure 2(a)**. In the

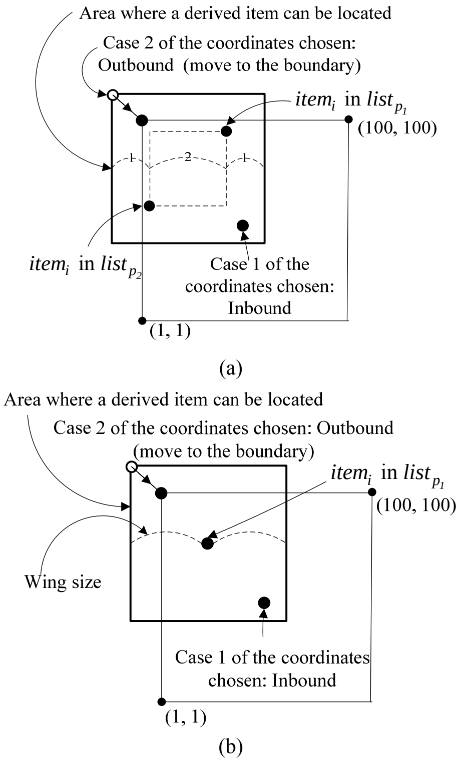

Area where a derived item can be located
Case 2 of the coordinates chosen: Outbound (move to the boundary)
$item_i$ in $list_{p_1}$
(100, 100)
1
2
1
$item_i$ in $list_{p_2}$
Case 1 of the coordinates chosen: Inbound
(1, 1)
(a)
Area where a derived item can be located
Case 2 of the coordinates chosen: Outbound (move to the boundary)
$item_i$ in $list_{p_1}$
(100, 100)
Wing size
Case 1 of the coordinates chosen: Inbound
(1, 1)
(b)

Figure 2. Crossover and Mutation in GAs for 0/1KP. (a) Crossover; (b) Mutation. NOTE. An item here is regarded as a two-dimensional coordinates. Part (a) is an illustration of deriving *item$_i$* in offspring in crossover. Two *item$_i$*'s in $list_{p_1}$ and $list_{p_1}$ make the area where *item$_i$* in offspring can be randomly chosen. Part (b) is an illustration of setting *item$_i$* in $list_{p_1}$ to be a new one in mutation. The square, whose center is *item$_i$* of $list_{p_1}$, indicates the area where *item$_i$* can be randomly chosen

[2] Strictly speaking, we use the PMX introduced in the following URL: http://www.rubicite.com/Genetic /tutorial/crossover5.php.

mutation from an individual, the probability of changing each item in the individual to be new one is p_m. Once an item is determined to be changed, we determine the area of the square where the point indicating a new item can be located. The center of the square is the point indicating the original item. The edge length of the square is determined using the following probability weight function of wing size or (*edge length*)/2: $y = a \times (x - x_1) + y_1$, where $a < 0$, y is the probability weight of choosing x as a wing size, y_1 is the probability weight of choosing x_1, and x_1 is the maximum wing size. We set y_1 to be 1 and set x_1 to be *floor*((*UB*−*LB*)/2), with $UB = 100$ and $LB = 1$. For a given wing size, the illustration of the changing an item to be a new one is shown in **Figure 2(b)**.

In the local search from an individual (or a list), we try to slightly move each item in the list and test that the new list has a better quality than old one. If it is true, then the new one is substituted for the original one. Otherwise, we try to move the next item in the list. The way of slightly moving the item (v, w) is as follows: moving to (v+1, w), (v−1, w), (v, w+1), (v, w−1), (v+1, w+1), (v−1, w−1), (v+1, w−1), or (v−1, w+1), excluding one out of the boundary of [1, 100]; we take the best way among all the possible ways.

3.4 Travelling Salesperson Problem

Every instance of the TSP is represented as an $N \times N$ adjacency matrix. The N, which is the number of cities, is fixed for every individual in the population. The (i, j) element of the matrix is the cost to go from $city_i$ to $city_j$. We restrict that every element of the matrix is an integer in [1, 100]. The TSP does not limit the values to be integers. But the bounds reduce the time for finding the worst case, which is still helpful. We will try to find the worst-case instance for two different versions of the problem. In one version, the input instance G is always represented as a symmetric matrix. In the other version, G may be represented as an asymmetric matrix. The distance between two individuals is the sum of absolute differences between (i, j) elements of two matrices. The quality of an individual is $(P-O)/O$, where P and O are described in Subsection 2.4.

For the crossover of two matrices, we use geographic crossover [16-18]. The geographic crossover can generate sufficiently diverse offspring with moderate difficulty in implementing. For the mutation from an individual (or a matrix), our approach is to move each (i, j) element of the matrix with given probability p_m. We regard an element as a point on the number line. If an element is decided to be moved, the element can be moved randomly within the interval whose center is the original element. The interval size is decided by the probability weight function of the size. The function is quite similar to one in the 0/1KP. The small size is taken more often than the big one. If we take large interval and the element moves out of [1, 100], we adjust the location to the closest boundary of the interval. We do not use the local search here.

4. Experimental Results

Using the framework and the strategies explained in Section 3, we tried to find the worst-case instances of test algorithms. We used the computer of which CPU model is Intel Core2 Duo T8100 @ 2.10 GHz. In **Table 1**, we show the parameters in common for every GA. The usage of the mutation probability (p_m) in **Table 1** is described in Subsections 3.2, 3.3, and 3.4 respectively for the three test problems. **Table 2** shows parameters in common for every random testing [1]. We ran the same GA fifty times to check whether our GAs are stochastically reliable. We say that two GAs are the same if and only if they are for the same problem and they belong to the same classification; we propose the classification in **Tables 3-5**, respectively for each problem. The weight coefficient in **Table 4** is described in Subsection 2.3. The probability weight function of wing size in the same table is described in Subsection 3.3. The probability weight function in **Table 5** is similar to one in **Table 4**. This function is referred to in Subsection 3.4.

Table 1. Parameters in common for every GA

Maximum number of generations	1,000
Selective pressure	3
# of individuals to be replaced for the next generation	30
Population size	100
Mutation probability p_m	0.15
# of independent runs of GA	50

Table 2. Parameters in common for every random testing

# of instances randomly generated	30,000
# of independent runs of the random testing	50

Table 3. Classification and parameters of GAs for algorithms solving sorting problem

Classification basis	Size of permutations: 10, 20, 30, or 40 (For advanced quick sort, we tested only on size 10 and 20.)
	Sorting algorithm: quick sort, merge sort, heap sort, insertion sort, shell sort, or advanced quick sort
Elements of permutation	{1,2, ..., N}
Limitation of improvement counts in local search	The smallest integer bigger than or equal to $N\times0.25+C_n^{k-1}$ (C_n^k denotes the combination.)

Table 4. Classification and parameters of GAs for algorithms solving 0/1KP

Classification basis	# of given items: 10, 20, or 30
	Weight coefficient cw: 0.25, 0.5, or 0.75
Range of value and weight of items	Integers in [1, 100]
Tangent slope of probability function of wing size.	−1 (wing size is an integer from 1 to 50)

Table 5. Classification and parameters of GAs for algorithms solving TSP

Classification basis	# of cities: 5 or 10
	Kind of instances: symmetric matrix or general matrix
Range of weighted cost	Integers in [1, 100]
Tangent slope of probability function of interval size in mutation.	−1 (interval size is an integer from 1 to 50)
# of cutting lines in a crossover	0.5× (# of cities)
NOTE. The crossover here is geographic crossover, which uses cutting lines.	

After the run of a GA, we find one instance that shows the maximum quality in the final population; this instance is the closest to the worst case which we defined. Among those maximum qualities which are found by repeating the same GA fifty times, we get the average and the standard deviation. These values are in **Figure 3**, **Figure 4**, and **Figure 5** respectively for each problem. Those figures are represented by the histogram with error bars; the length of error bar is 2 times the corresponding standard deviation. Some error bars in the figures are not identified because the corresponding standard deviation is close to 0. Note that in the figures, ‘Avg’ means the average, ‘Std’ means the standard deviation, ‘Random’ means the random testing method. In **Figure 3**, ‘Ascend’ means the permutation in ascending order (*i.e.*, 1, 2, 3, 4, ..., *N*) and ‘Descend’ means the permutation in descending order (*i.e.*, *N*, *N*−1, ..., 4, 3, 2, 1). In **Figure 5**, ‘symmetric TSP’ means the TSP which does not allow any problem instance with an asymmetric matrix representation whereas ‘general TSP’ means the TSP which allows such problem instances. The CPU seconds taken by a run of the GA for each classification are given in **Table 6**, **Table 7**, and **Table 8**, respectively for each problem.

For test sorting algorithms, the quality of an individual (a permutation) was defined as the number of comparisons between elements when the permutation is sorted using the algorithm. In **Figure 3**, the worst cases of our quick sort, insertion sort, and advance quick sort takes more element comparisons than those of other test sorting algorithms. But seeing the result when the sorting

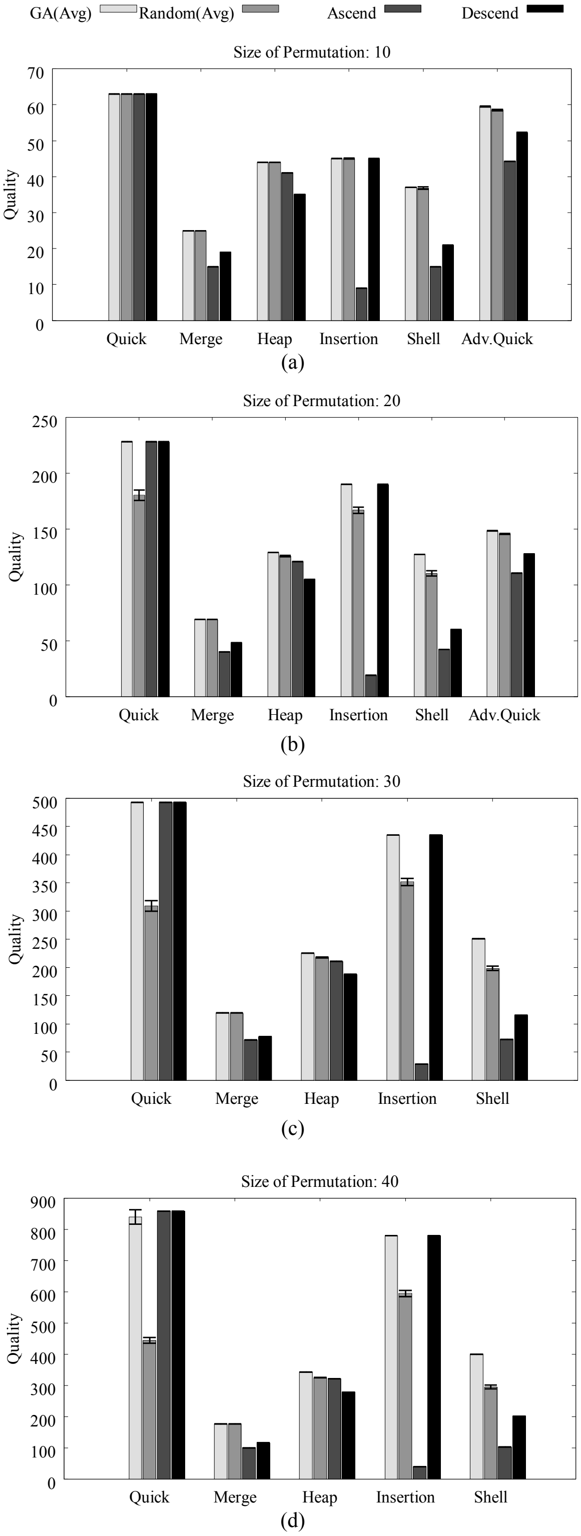
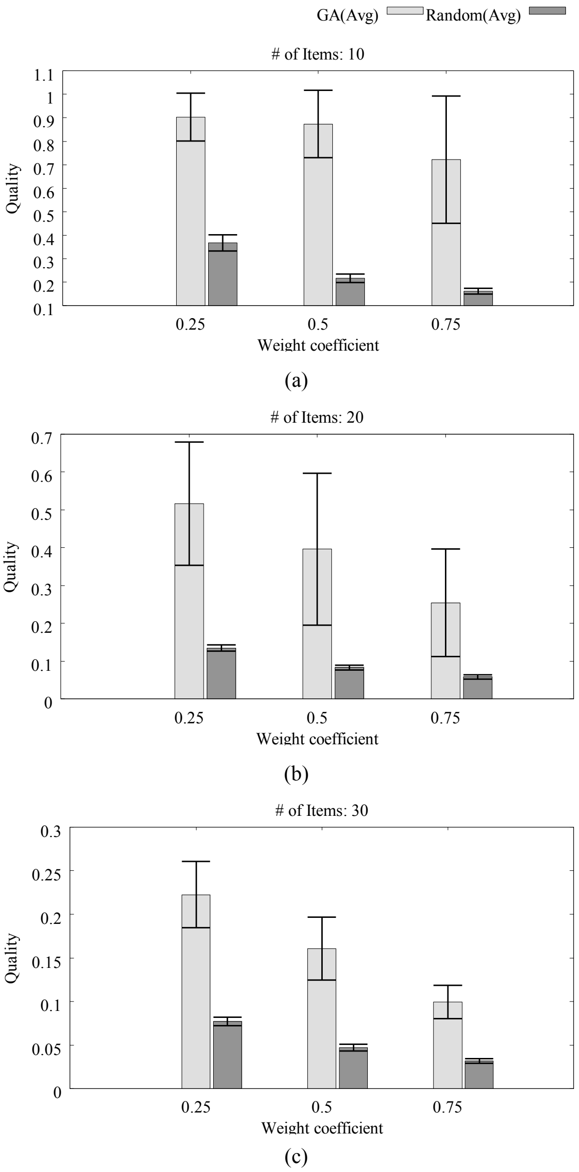


Figure 3. Quality of the worst-case instances (sorting problem). (a) Size of Permutation: 10; (b) Size of Permutation: 20; (c) Size of Permutation: 30; (d) Size of Permutation: 40. NOTE. The definition of the quality of an instance is given in Subsection 3.2



(a)

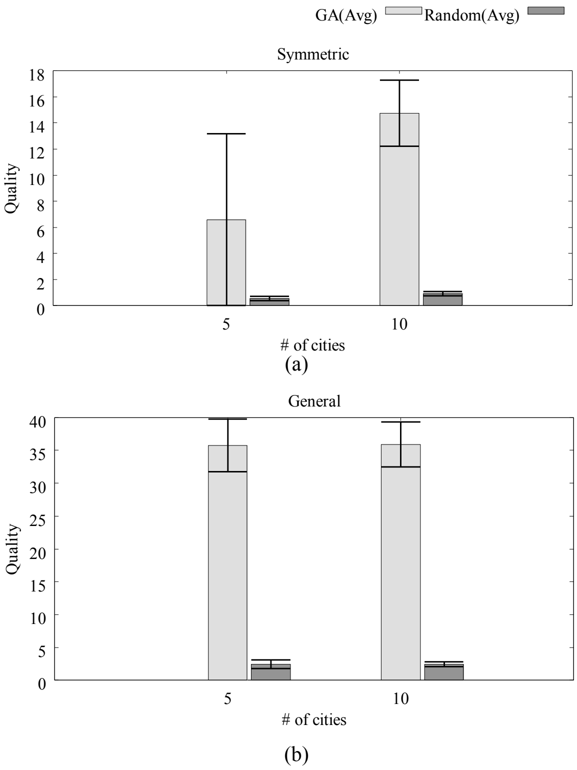
(b)

(c)

Figure 4. Quality of the worst-case instances (0/1KP). (a) The number of given items: 10; (b) The number of given items: 20; (c) The number of given items: 30. NOTE. The definition of the quality of an instance is given in Subsection 3.3

Table 6. CPU seconds taken by a GA for each classification (sorting problem)

Sorting algorithm / Size of permutations	10	20	30	40
Quick	16	75	214	456
Merge	20	122	415	1,059
Heap	19	123	414	1,093
Insertion	19	134	496	1,359
Shell	17	85	282	729
Advanced Quick	1,352	12,502	N/A	N/A



(a)

(b)

Figure 5. Quality of the worst-case instances (TSP). (a) The symmetric TSP; (b) The general TSP. NOTE. The definition of the quality of an instance is given in Subsection 3.4

Table 7. CPU seconds taken by a GA for each classification (0/1KP)

Weight coefficient / # of given items	10	20	30
0.25	70	607	2,415
0.5	75	904	5,520
0.75	105	1,927	6,000

Table 8. CPU seconds taken by a GA for each classification (TSP)

# of cities / Kind of instances	Symmetric matrix	General matrix
5	18	18
10	472	470

permutation's size is 10 to 20, we can predict that as the permutation size grows, the worst-case of our advanced quick sort takes fewer comparisons than that of our quick sort and insertion sort. On the other hand, the worst cases of merge sort, heap sort, and shell sort takes fewer comparisons than those of other algorithms.

For our test algorithm solving the 0/1KP (*i.e.*, the algorithm based on a greedy approach which we explained in Subsection 2.3), the quality of an individual is $(O-P)/O$, where P is the objective value obtained by the test algorithm and O is the objective value obtained by an optimal algorithm. Regardless of the number of items (*i.e.*, the magnitude of the searching space), the upper bound of the qualities is 1. We can see the result in **Figure 4**. For the same number of items, the higher the weight coefficient is, the lower the average quality of the worst cases found by our GA was. For the same coefficient, the more the given number of items is, the lower the average quality of the worst cases found by our GA was. When 10 items are given, the average quality of the worst cases was generally more than 0.7; our test algorithm is possibly not good enough in terms of optimality.

For our test algorithm solving the TSP (*i.e.*, the algorithm based on a greedy approach with 2-opt which we explained in Subsection 2.4), the quality of an individual is $(P-O)/O$, the meaning of O and P is similar to those in the 0/1KP. By examining the formula of the quality, we can assume that the upper bound of the qualities does not exist. In **Figure 5**, on the same number of cities, the average quality of the worst case in the symmetric TSP was lower than that of the worst case in the asymmetric TSP. The average quality of the worst case found by our GA in the symmetric TSP was about 14 when 10 cities are given, whereas one found in the asymmetric TSP was about 35 on the same condition; our test algorithm is generally not good enough in terms of optimality.

Overall, the results found by GAs were superior to those obtained by the random testing; we can conclude that our GAs have some effectiveness.

5. Conclusions

There are few trials to find the worst case for testing algorithms by GAs and the existing trials do not introduce various strategies for constructing GAs. Therefore, by taking as test examples the internal sorting problem, the 0/1KP, and the TSP with some algorithms for each, we gave guidelines to the use of the GA in generating the worst-case instance for an algorithm. First, we defined the objective function of our GAs for the purpose of the analysis. In this paper, the objective function returns the quality of a problem instance; this quality indicates how-close the instance is to the worst case. Next, we introduced the framework of GAs and the specific strategies for each test problem.

For the sorting problem, we adopted as test algorithm quick sort, merge sort, heap sort, insertion sort, shell sort, and advanced quick sort. We defined the worst-case instance for a sorting algorithm as the instance that takes the most number of the element comparisons in using the algorithm. A problem instance can be represented as a permutation. We here used the PMX for the crossover. The mutation here swaps arbitrary elements in the permutation.

For the 0/1KP and the TSP, we tested the algorithm based on a greedy approach, comparing to an optimal algorithm. We defined the worst-case instance as the instance at which the test algorithm shows the most different objective value from that obtained by the optimal algorithm. In the case of the 0/1KP, a problem instance can be represented as the list of two-dimensional point whose coordinates are integers. Our suggested crossover is based on the idea of uniform crossover, but extended to two-dimensional version. The mutation runs point by point; each point in the list is moved to another close point with a high probability or to another far point with a low probability. In the case of the TSP, we represent a problem instance as a square matrix. We used the geographic crossover for the representation. The mutation runs element by element in the given matrix. Although the element here is just a scalar (not the two-dimensional point), the idea of mutation is similar to that of mutation for the 0/1KP.

For the test algorithms, our GAs has some effectiveness as the experimental results are superior to those obtained by the random testing. The results of finding the worst cases show the following: 1) At the worst case, merge sort, heap sort, and shell sort takes fewer comparisons than other sorting algorithm. 2) Our greedy approach is not good enough in terms of optimality.

Our guidelines can help to analyze algorithms and can be used to test software. Since our guidelines are just for using GAs, we suggest giving guidelines for using other metaheuristics to generate the worst cases. Also note that many essential parts are still remained to analyze the algorithms. For future work, we suggest finding some weak points of a test algorithm and improve the algorithm by analyzing the worst-case instance.

REFERENCES

[1] P. McMinn, “Search-Based Software Test Data Generation: A Survey,” *Software Testing Verification And Reliability*, Vol. 14, No. 2, 2004, pp. 105-156.

[2] M. Johnson and A. Kosoresow, “Finding Worst-Case Instances of, and Lower Bounds for, Online Algorithms Using Genetic Algorithms,” *Lecture Notes in Computer Science*, Vol. 2557, 2002, pp. 344-355.

[3] C. Cotta and P. Moscato, “A Mixed-Evolutionary Statistical Analysis of an Algorithm’s Complexity,” *Applied Mathematics Letters*, Vol. 16, No. 1, 2003, pp. 41-47.

[4] D. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning,” Kluwer Academic Publishers, Boston, 1989.

[5] P. Moscato, “Memetic Algorithms: A Short Introduction,” In: D. Corne, M. Dorigo and F. Glover, Eds., *New ideas in optimization*, Mcgraw-Hill, London, 1999, pp. 219-234.

[6] M. Main and W. Savitch, “Data Structures and Other Objects Using C++,” 3rd Edtion, Pearson/Addison-Wesley, 2004.

[7] R. Sedgewick, “Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching,” 3rd Edtion, Addison-Wesley, 1998.

Copyright © 2010 SciRes.

[8] D. Knuth, "The Art of Computer Programming, Volume 3: Sorting and Searching," 2nd Edtion, Addison-Wesley, New York, 1998.

[9] R. Neapolitan and K. Naimipour, "Foundations of Algorithms Using C++ Pseudocode," 3rd Edtion, Jones and Bartlett Publishers, Inc., 2008.

[10] C. Papadimitriou and K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity," Prentice-Hall, New Haven, 1981.

[11] L. Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination," In: G. Rawlins, Ed., *Foundations of Genetic Algorithms*, Morgan Kauffman, San Mateo, 1991, pp. 265-283.

[12] G. Harik, F. Lobo and D. Goldberg, "Compact Genetic Algorithm," *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, 1999, pp. 287-297.

[13] J. Grefenstette, "Genetic Algorithms for Changing Environments," *Proceedings Parallel Problem Solving from Nature*, Amsterdam, Vol. 2, 1992, pp. 137-144.

[14] D. Goldberg and R. Lingle, "Alleles, Loci, and the Traveling Salesperson Problem," *Proceedings of the International Conference on Genetic Algorithms*, Hillsdale, 1985, pp. 154-159.

[15] S. Choi and B. Moon, "Normalization in Genetic Algorithms," *Genetic and Evolutionary Computation—GECCO-2003*, *Lecture Notes in Computer Science*, Vol. 2723, Springer-Verlag, Berlin, 2003, pp. 862-873.

[16] T. Bui, B. Moon, "On Multi-Dimensional Encoding/Crossover," *Proceedings of the 6th International Conference on Genetic Algorithms*, San Francisco, 1995, pp. 49-56.

[17] B. Kahng and B. Moon, "Toward More Powerful Recombinations," *Proceedings of the 6th International Conference on Genetic Algorithms*, San Francisco, 1995, pp. 96-103.

[18] C. Im, H. Jung and Y. Kim, "Hybrid Genetic Algorithm for Electromagnetic Topology Optimization," *IEEE Transactions on Magnetics*, Vol. 39, No. 1, 2003, pp. 2163-2169.

Copyright © 2010 SciRes.

J. Software Engineering & Applications, **2010, 3, 776-783**
doi:10.4236/jsea.2010.38090 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

Applications of Norm and Situation Calculus in the Semantic Web Service Composition*

Jun Zhao[1], Xiaozhu Gu[2]

[1]School of Economics and Management, Ningxia University, Yinchuan, China; [2]School of Mathematical and Computer Sciences, Ningxia University, Yinchuan, China.
Email: wwwzhaojun@163.com, guxiaozhugood@163.com

Received June 23rd 2010; revised July 8th 2010; accepted July 24th 2010.

ABSTRACT

Semantic Web service studies are carried out around the OWL-S, however, OWL-S model does not contain the description of preferences and constraints when we choose and use them in different organizations, different cultures, different sectors and actors because of OWL capacity limitations in rules. It means that Semantic Web service composition can not compose users' subjective services from the user requirements. The paper makes up for the semantic shortcoming of OWL-S through using norm semantic scalability, and achieves the formal description of the norm by using the situation calculus. Finally, the paper also takes a tourism composite service as an example how to extend the ability of the semantic description of OWL-S, which shows us the combination of the norm and the situation calculus. It is positive significance for eventually realizing semantic extension of OWL-S.

Keywords: *Norm, Situation Calculus, Web Service, Semantic Web, Web Service Composition*

1. Introduction

Web services are described using WSDL, but WSDL only from the syntactic layer of the Web service description. It does not support the rich semantic descriptions. Semantic Web can be regarded as an extension and expansion of the current Web technology. Web services are an increasingly important resource of web, it require semantic to provide wide range semantic support. OWL-S is bridges to connect Web services and semantic Web. Many of the current research on semantic Web services are carried around the OWL-S. However, as the capacity limitations of the rules of OWL, OWL-S model does not include the description of preferences and constraints in users' selection and use of services, which always manifests in the form of rules. The user's preferences and scene constraints can't be understood and processed well by the computer because of the expression limitations. It also means that intelligent system does not start from user requirements during service composition, and does not assemble more subjective service in accordance with users' will. Therefore, the article will extend the semantic of OWL-S through semantic capabilities of the norm. However, norm is non-formal and ambiguous. This paper attempts to use situation calculus to formally describe the norm in order to ensure the correctness of norm and the user correct expression of various scene constraints, and provide more accurate and subjective service in expression of the preferences.

2. Related Researches

Sheila A. McIlraith, Srini Narayanan and others of Stanford University apply the theory of situation calculus and Petri net to study the operational semantics of OWL-S. Firstly, they study the conversion relationship between the atomic service description and situation calculus. Then, they use situation calculus to study the formal semantics of atomic service descriptions in OWL-S. The situation calculus as an intermediate language is converted into the Petri net, and then the Petri net is used to study the implementation reasoning of the operational semantics of semantic Web services [1].

Andrea and Ferrara, who come from DIS-University di Roma use an approach of the process algebra to study the Web service composition, establish mutual mapping between process algebra and Business Process Execution Language for Web Services based on the research of Srini Narayanan and others [2].

SWSI (Semantic Web Services Initiative), mainly by DARPA and the EU (European Union) funded a project to build a combination of Semantic Web and Web services technology integration framework that enables ser-

*This research was supported by the Natural Science Foundation of China (No. 70961007).

Copyright © 2010 SciRes.

vice providers and service requesters in the service discovery retrieval, matching, call, monitor and so on to achieve the greatest possible integration of automation and dynamic.

Yongshang Cheng and Zhijian Wangf, who come from College of Computer and Information Engineering, Hehai University have invented formal model of semantic Web services on the basis of colored Petri nets [3] .They formally describes several basic web service composition models in OWL-S, which is constructed to meet the demand of service, but it lacks of the expression of services semantic information.

Jun Liao, Hao Tan and Jinde Liu, who come from the University of Electronic Science and Technology, describe the Web service composition by using Pi calculus. The paper introduces an example using Pi calculus to describe the Web service composition [4].

From the above research and projects we can see that researchers have made some valuable results in the semantic Web and Web service composition context, and many research problems wait to be solved to provide more powerful semantics services especially in the semantic extensions, but the theoretical system of service composition engineering and implementation technology are still immature because the study of history is not long.

This paper introduces norm theory of Organizational Semiotics to solve the problem of semantic deficiency in Web service composition. In order to achieve a deeper level of semantic extension, it achieves the formal description of norm in semantic expansion by the theory of situation calculus in order to ensure the accuracy of semantic extension.

3. Applications of Norm and Situation Calculus in the Semantic Web Service Composition

Norm is also known as social norms which is the common rules of conduct and standards of every member in a social group or smaller groups, which is in the Oxford Encyclopedia of the explanation.

As rules in expression limitations of the OWL, OWL-S model does not include the users' the description of preferences and constraints in the selection and use of services. It try to introduce norm to provides a new theoretical platform in the semantic Web service composition in order to solve Web services Semantic shortage described by the OWL-S in the semantic Web service composition. This approach considers many the users' initiative and willingness. It joins the human factor in the service composition to provide a more personalized composition services.

Norm determines whether and when an event occurs, and also decided meeting the particular needs should call the areas of service layer of what services. It can describe the activities relationship of the system control. The norm detailed description includes the following sections [5].

Whenever< Conditions set >If< State Sets >Then <Agent>Is <Deontic Operator>To< Action sets >Consequence< Result Sets>

Conditions Sets specified the conditions when an Agent executed an act. Further instructions can be specified by If < State set>, which show that it happened in what state. <Agent> Refers to the responsibility Agent, Agent may be people here, also may be software. < Deontic logic operators> including allowing (Permitted), must (Obliged), prohibition (Prohibited)etc. <Action Sets> determined action sets applied in the case of conditions to met for Agent. <Result sets> is the results after the successful implementation of the specification, which is also the goals of users implement the act.

We can see that norm is different from the relations of causality [6]. Causality is usually only a brief description, if the conditions are met, then certain behavior could take place. Therefore causality is rigid and limited, and there is not available for human decision. On the contrary, norms are a better reflection of how people activities in the business environment. Therefore it is more suited to the description of user needs in the real business. Changes of user needs in the growing environment can be described through the norm. This required model can fully focus on changes factors. It will help to provide dynamically alternative solutions among the service layer and process layer on user needs.

Situation Calculus is a multi-type and first-order logic language, and has some second-order characteristics. It is a formal planning approach and describes a logical basis for dynamic system. For problem solving of the dynamic field and logic programming, it is first proposed in 1963 by McCarthy who is the master of artificial intelligence. Goal-oriented autonomous behavior reasoning can be carried out in a dynamic environment by situation calculus. In the situation calculus, a scenario is a snapshot of the world, and dynamic changes in the world are all the result of actions. Scenarios class is applied to express Situation Calculus, and the scene change is the results of the actions. In ontology, the action is the basic means to change the state of things. Therefore, a possible world active can be viewed as a string of action series. Situation calculus constitutes by the following three elements: *action, situation and flow.*

Action is the basic means to change station of things. All Changes in the state of things are the result of the implementation of the action sequences. A specific predicate *Poss* is used to indicate that an action is executable.

Situation is the dynamic world. All changes in the world are the result of action, which is a string of limited action sequences. Action function is expressed as: action situation → situation. If the action a is implemented under the scenario s, then another situation will be got, and the situation s can be expressed as do (**a**, **s**).

Flow is used to represent the properties of things in the world and the link between them [7]. When an action occurs, the flow will change. Flow ***F***(***x***, ***s***) takes scene s as its last parameter. The initial situation is expressed as s0. The value of the flow is constantly changing from the initial state of the world to the current world state.

In order to characterize the dynamic changing world, situation calculus theory uses the following actions: *Action Precondition Axiom, Successor State Axiom, Effect Axiom* etc. [8].

Action Precondition Axiom: Every action of the field has a corresponding action precondition axioms, describing a prerequisite for action to perform.

$Poss(a,s) \Leftrightarrow f_i$ f_i is a prerequisite sets for the implementation of a.

Successor *State Axiom*: Each flow in field has a corresponding *Successor State Axiom,* which is described the implementation of atomic actions how to affect the flow and state changes.

$$Poss(a,s) \wedge \gamma_F^+(x,a,s) \rightarrow F(x, do(a,s))$$

$$Poss(a,s) \wedge \gamma_F^-(x,a,s) \rightarrow \neg F(x, do(a,s))$$

$Poss(a,s)$ is a special flow and action a is enforceable under state s.

Formula $\gamma_F^+(x,a,s)$ is a positive effect axiom that describes the related collection of actions and conditions to makes the value of flow F true after the execution. Formula $\gamma_F^-(x,a,s)$ is a negative effect axiom that describes the related collection of actions and conditions to makes the value of flow F false after the execution.

In addition to the above several axioms, there are other flows that can be used to describe atomic service or a combination of services as formal semantics of services in situation calculus. Other flows are the following types:

$Kref(a,s)$ expresses that the value of a is known under the scenario s.

$Kwhether(a,s)$ expresses that the true value of a is known under the scenario s.

$Knows(a,s)$ expresses that the function value of a is known under the scenario s.

This essay will introduce users Desirable Axiom and constraint Axiom in order to express the will degree of user requirements in the situation calculus [9-10]. We also attempt to define $\neg Constra\,\mathrm{int}(a,s)$ which action a under the situation s can be implemented as long as the users' conditions are not limited [11].

A set of items compared norm expression with situation calculus to achieve the service semantic expansion and formal description. It is shown as **Table 1**.

Table 1. Mapping norm to situation calculus

Items of norm	Items of the situation calculus
Input	Conditions Input Function $Kref(a,s)$
Conditions set State set	Action Precondition Axiom: $Poss(a,s) \Leftrightarrow f_i$
Obliged	Desirable Axiom $Desirable(a,s)$
Permitted	$\neg Constra\,\mathrm{int}(a,s)$
Prohibited	Constraint Axiom $Constra\,\mathrm{int}(a,s)$
Action set	Action predicate $Poss(a,s)$
Result Set	Effect Axioms and Condition Output Function: $Kwhether(a,s)$ $Kwhether(a,s)$

4. Case Studies

4.1 Example Background

Tourism is an integrated industry which includes food, housing, transportation, travel, shopping and entertainment. A trip involves a number of services and tourism resources, and tourist information is rich in space and time content. Travel is the complex process restricted by artificial and natural factors. How to use Web services by the network to provide "personal travel" service and how to design personalized service process according to the needs of our customers become a problem. The problem has plagued further development of the tourism industry and prevented from increasing the overall quality of tourism services to increase tourists' satisfaction. Therefore the examples will make a combination of the weather check, ticket booking, tourist attractions, reservations and other travel services with personal characters. The service composition flow is shown in **Figure 1**.

The example mainly involves the following web services:

1) $InquireWeather(City, Date)$: Weather inquiry service.

2) $InquireDis\mathrm{tance}(Start, Destination)$: Check the distance between the two places. *Start* is the place of departure. *Destination* is the sightseeing place.

3) $BuyAirPlaneTicket(City, Date)$: Booking airline tickets.

4) $BuyTrainTicket(City, Date)$: Booking train tickets.

5) $BookScenicSpots(City)$: Attractions reservation.

6) $BookInsurance(TypesOfInsurance)$: Buy insurance.

7) $\mathrm{Re}ntCar(CarModel)$: Service of cars on hire.

8) $BookHotel(City)$: Hotel reservation.

4.2 Norm Analysis

A detailed analysis of the norm is as below **Tables 2-9**.

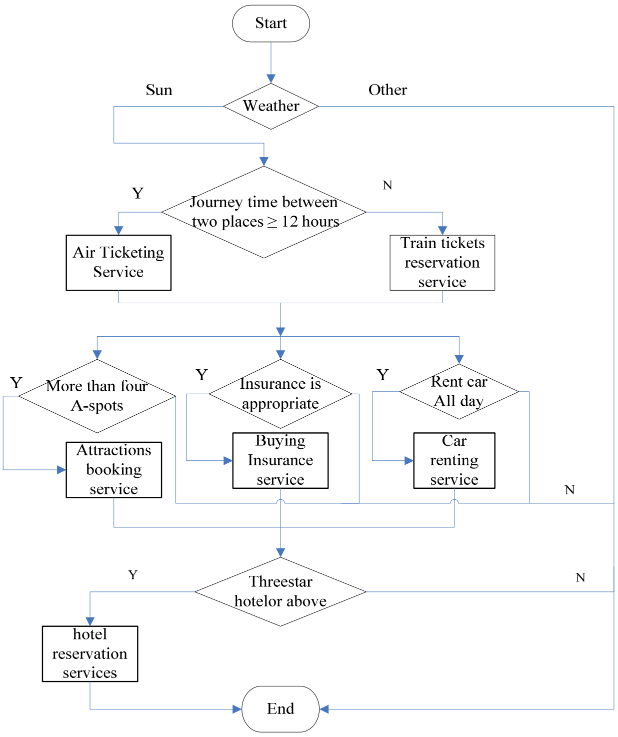


Figure 1. The chart of reservation service composition flow

Table 2. Weather norm

Norm 1	Searching start
In Charge of	User
Information Recognition	City, Date
Trigger Condition	User query the weather of sightseeing site Precondition: Enter valid city and date Then: Weather query service returns query results
Detailed Norm	**if** the input date and destination of users are valid **then** Weather enquiry service **is obliged to** return the query results

Table 3. Distance query norm

Norm 2	Searching start
In Charge of	User
Information Recognition	Start, Destination
Trigger Condition	User queries the distance between the start and destination Precondition: The query results of norm 1 is sun Then: Return the distance between start and destination
Detailed Norm	**Whenever** the weather of the indicated Date is sunny in norm1 **if** the input start and destination of users are valid **then** distance query service **is obliged to** return the distance between two places

Table 4. Airline reservation norm

Norm 3	Booking start
In Charge of	User
Information Recognition	Bank Card ID, User ID
Trigger Condition	Users make flight reservations Precondition: Output is sunny in norm1and the output is greater than or equal 1000 kilometers in norm 2 Then: Users input bank card password ,book ticket
Detailed Norm	**Whenever** Output is sunny in norm1and the output is greater than or equal 1000 kilometers in norm 2 **if** bank card ID and user ID are valid **then user is obliged to** input bank card password to transfer

Table 5. Train ticket reservation norm

Norm 4	Booking start
In Charge of	User
Information Recognition	Bank Card ID ,User ID
Trigger Condition	Users book train tickets Precondition: Output is sunny in norm1and the output is greater than or equal 1000 kilometers in norm 2 Then: Users input bank card password ,book ticket
Detailed Norm	**Whenever** Output is sunny in norm 1and the output is greater than or equal 1000 kilometers in norm 2 **if** bank card ID and user ID are valid **then user permitted** input bank card password to transfer

Table 6. Attractions reservation norm

Norm 5	Booking start
In Charge of Information Recognition	User City
Trigger Condition	Users reserve scenic spot Precondition: City is valid, the grade of attractions are four A-level or above Then: Return the attractions which are qualified
Detailed Norm	**Whenever** norn 3 and norm4 are successful **if** the grade of attractions are four A-level or above **then users permitted** input bank card password and transfer reserve attractions if the grade of attractions are below four A-level then norm 3 or norm 4 **is prohibited** to book

Table 7. Insurance booking norm

Norm 6	Booking start
In Charge of Information Recognition	User City
Trigger Condition	Users reserve insurance Precondition: City is valid, users choose appropriate insurance Then: Users input the bank card password and transfer; Users reserve insurance
Detailed Norm	**Whenever** norm 3 or norm 4 are successful **if** thegrade of attractions are four A-level or above **then** user **is obliged to** input bank card password and transfer in order to reserve attraction

Table 8. Car rental reservation norm

Norm 7	Booking start
In Charge of Information Recognition	User City
Trigger Condition	Users reserve the car which is rental Precondition: City is valid, the rental car is full-time Then: Users input bank card password in order to reserve rental cars
Detailed Norm	Whenever norm 3 or norm 4,norm 5,norm 6 are successful **if** the rental car is full-time **then** users **is obliged to** input bank card password in order to reserve the rental car

Table 9. Hotel reservation norm

Norm 8	Booking start
In Charge of Information Recognition	User Bank Card ID ,User ID
Trigger Condition	Users reserve hotel Precondition: Norm 3 or norm 4 and norm 5 norm 6 norm 7 are successful; The hotel is three-star or above Then: Booking hotel
Detailed Norm	**Whenever** Norm 3 or norm 4 and norm 5 norm 6 norm 7 are successful **if** hotel is above 3A **then** user **is obliged to** input bank card password in order to reserve hotels

4.3 Mapping from Norm to Situation Calculus

1) The Situation Calculus Formally Describes the Atomic Services in Example

a) Weather atomic service: Before the atomic service of Weather is implemented, agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$Poss(InquireWeather(City, Date), s) \rightarrow$$

$$Kref\,(City, s) \wedge Kref\,(Date, s)$$

$InquireWeather(City, Date)$ is a weather query action, the action on the premise that you need to know the $City$ and $Date$, which can be represented by flow $Kref\,(City, s)$ and $Kref\,(Date, s)$.

Users want to travel as long as the weather was not dust storms and heavy rain. It is formalized as the following formula with the situation calculus.

$$\neg Constra\ \text{int}(Tranvel, s) \equiv$$

$$\neg(Weather(Equal, Storm, s) \wedge$$

$$Weather(Equal, Sadstorm, s));$$

The output of the Weather Service is formalized as the following formula with the situation calculus.

$$Poss(InquireWeather, s) \wedge \gamma(x, InquireWeather, s)$$

$$\rightarrow Knows(Sun / Other, do(InquireWeather, s))$$

b) The atomic service of distance query: Before the atomic service of Distance Query is implemented, agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$Poss(Distance(Start, Destination), s) \rightarrow$$

$$Kref\,(Start, s) \wedge Kref\,(Destination, s)$$

The output of the Distance Query Service is formalized as the following formula with the situation calculus.

$$Poss(Dis\tan ce, s) \wedge \gamma(x, Dis\tan ce, s) \rightarrow$$

$$Knows(\ Figure, do(Dis\tan ce, s))$$

c) Ticket booking atomic services: Before the atomic service of Airline Reservation is implemented, agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$Poss(BuyAirplaneTicket(City, Date), s) \rightarrow$$

$$Kref\,(City, s) \wedge Kref\,(Date, s)$$

If user wants to buy air tickets must be that the distance between two places is greater than or equal to 1,000 kilometers. It is formalized as the following formula with the situation calculus.

$$Desirable(BuyAirplaneTicket(City, Date), s) \equiv$$

$$Dis\tan ce((Start, Destinatin), Longer1000);$$

The effectiveness and output parameters of atomic service buy ticket are formalized as the following Positive Effect Axiom and Negative Effect Axiom.

$$\gamma_F^+ : a = BuyAirplaneTicket(City, Date) \wedge$$
$$InstockAirplaneTicket(City, Date)$$
$$Poss(a, s) \wedge a = BuyAirplaneTicket(City, Date) \wedge$$
$$InstockAirplaneTicket(City, Date) \rightarrow$$
$$own(AirplaneTicket, do(a, s))$$

2) The Situation Calculus Formally describes the Composition Services in Example

The Example involve several combination services: *BookTicket* is composed of the Distance Query Service, ticket booking service and ticket reservation service, which implement are controlled mainly by *if* control operator; *Journey* is composed of the Attractions Reservation Service; Insurance Booking Service and Car Rental Reservation Service, which implement are controlled mainly by *Anyordered* control operator; The composition service *Tranvel* is composed of the composition service *BookTicket* ; composition service *Journey* and Hotel Reservation Service, which implement are controlled mainly by *Condition* control operator.

a) Composition service *BookTicket* : Distance Query Service, ticket booking service and ticket reservation service is composed *BookTicket* , which implement are controlled mainly by *if* control operator.

The prerequisite was formalized as the following formula in the situation calculus:

$$(Knows(Over1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyAirplaneTicket(City, Date), s) \rightarrow$$
$$Knows(Sun, do(InquireWeather(City, Date), s))) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyTrainTicket(City, Date), s) \rightarrow$$
$$Knows(Sun, do(InquireWeather(City, Date), s)));$$

Before the composition service of *BookTicket* is implemented, agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$(Knows(Over1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyAirplaneTicket(City, Date), s) \rightarrow$$
$$Kref(City, s) \wedge Kref(Date, s)) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyTrainTicket(City, Date), s) \rightarrow$$
$$Kref(City, s) \wedge Kref(Date, s))$$

Based on the above two formulas, we can get the Action Precondition Axiom.

$$(Knows(Over1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyAirplaneTicket(City, Date), s) \rightarrow$$
$$Knows(Sun, do(InquireWeather(City, Date), s)) \wedge$$
$$Kref(City, s) \wedge Kref(Date, s)) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyTrainTicket(City, Date), s) \rightarrow$$
$$Knows(Sun, do(InquireWeather(City, Date), s)) \wedge$$
$$Kref(City, s) \wedge Kref(Date, s))$$

The effectiveness and output parameters of composition service are formalized as the following Positive Effect Axiom and Negative Effect Axiom.

$$\gamma_F^+ : a_1 = BuyAirplaneTicket(City, Date) \wedge$$
$$InstockAirplaneTicket(City, Date)$$
$$a_2 = BuyTrainTicket(City, Date) \wedge$$
$$InstockTrainTicket(City, Date)$$
$$(Knows(Over1000km,$$
$$do(InquireDistance(Start, Destination), s)) \wedge$$
$$Poss(a_1, s) \wedge \gamma_F^+(x, a_1, s) \rightarrow$$
$$Own(AirplaneTicket, do(a_1, s))) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDistance(Start, Destination), s)) \wedge$$
$$Poss(a_2, s) \wedge \gamma_F^+(x, a_2, s) \rightarrow$$
$$Own(TrainTicket, do(a_2, s)))$$
$$\gamma_F^- : a_1 = \neg BuyAirplaneTicket(City, Date) \vee$$
$$NoInstockAirplaneTicket(City, Date)$$
$$a_2 = \neg BuyTrainTicket(City, Date) \vee$$
$$NoInstockTrainTicket(City, Date)$$
$$(Knows(Over1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(a_1, s) \wedge \gamma_F^-(x, a_1, s) \rightarrow$$
$$\neg Own(AirplaneTicket, do(a_1, s))) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(a_2, s) \wedge \gamma_F^-(x, a_2, s) \rightarrow$$
$$\neg Own(TrainTicket, do(a_2, s)))$$

The output of the Composition Service *BookTicket* was formalized as the following formula with the situation calculus.

$$(Knows(Over1000km,$$

$$\begin{array}{l}
do(InquireDis\tan ce(Start, Destination), s)) \wedge \\
Poss(a_1, s) \wedge \gamma(x, a_1, s) \rightarrow \\
Knows(AirplaneTicketMessage, do(a_1, s))) \vee \\
(Knows(Below1000km, \\
do(InquireDis\tan ce(Start, Destination), s)) \wedge \\
Poss(a_2, s) \wedge \gamma(x, a_2, s) \rightarrow \\
Knows(TrainTicketMessage, do(a_2, s)))
\end{array}$$

b) Composition service *Journey* : It is composed with Attractions Reservation Service, Insurance Booking Service and Car Rental Reservation Service. Implement of the service is controlled by *Anyordered* control operator.

The implementation order of atomic services which are involved in the composition service is not required, so the composition service can be controlled implemented *Anyordered* control operator.

The prerequisite was formalized as the following formula with the situation calculus.

$$\begin{array}{l}
Poss(BookScenicSpots(City), s) \wedge \\
Poss(BookInsurance(TypesOfInsurance), \\
do(BookScenicSpots, s)) \wedge \\
Poss(\mathrm{Re}\, ntCar(CarModel), \\
do(BookInsurance, do(BookScenicSpots, s))) \rightarrow \\
Knows(GreaterFourA,\ do(InquireScenicSpots, s)) \wedge \\
Knows(TypesOfInsurance, \\
do(InquireInsurance(TypesOfInsurance, s))) \wedge \\
Knows(CarModel, do(Inquire\,\mathrm{Re}\, ntCar, s)) \vee \\
Poss(\mathrm{Re}\, ntCar(CarModel), s) \wedge \\
Poss(BookInsurance(TypesOfInsurance), \\
do(\mathrm{Re}\, ntCar, s)) \wedge\ Poss(BookScenicSpots(City), \\
do(BookInsurance, do(\mathrm{Re}\, ntCar, s))) \rightarrow \\
Knows(CarModel, do(Inquire\,\mathrm{Re}\, ntCar, s)) \wedge \\
Knows(TypesOfInsurance, \\
do(InquireInsurance(TypesOfInsurance, s))) \wedge \\
Knows(GreaterFourA, do(InquireScenicSpots, s))
\end{array}$$

Before the composition service of *Journey* is implemented, the agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$\begin{array}{l}
Poss(BookScenicSpots(City), s) \wedge \\
Poss(BookInsurance(TypesOfInsurance), \\
do(BookScenicSpots, s)) \wedge \\
Poss(\mathrm{Re}\, ntCar(CarModel), do(BookInsurance, \\
do(BookScenicSpots, s))) \rightarrow \\
Kref\,(City, s) \wedge Kref\,(TypesOfInsurance, s) \wedge \\
Kref\,(CarModel, s) \vee\ Poss(\mathrm{Re}\, ntCar(CarModel), s) \wedge \\
Poss(BookInsurance(TypesOfInsurance), \\
do(\mathrm{Re}\, ntCar, s)) \wedge\ Poss(BookScenicSpots(City), \\
do(BookInsurance, do(\mathrm{Re}\, ntCar, s))) \rightarrow \\
Kref\,(CarModel, s) \wedge Kref\,(TypesOfInsurance, s) \wedge \\
Kref\,(City, s)
\end{array}$$

The above two formulas, we can get the Action Precondition Axiom.

$$\begin{array}{l}
Poss(BookScenicSpots(City), s) \wedge \\
Poss(BookInsurance(TypesOfInsurance), \\
do(BookScenicSpots, s)) \wedge \\
Poss(\mathrm{Re}\, ntCar(CarModel), do(BookInsurance, \\
do(BookScenicSpots, s))) \rightarrow \\
Knows(GreaterFourA, do(InquireScenicSpots, s)) \wedge \\
Knows(TypesOfInsurance, \\
do(InquireInsurance(TypesOfInsurance, s))) \wedge \\
Knows(CarModel, do(Inquire\,\mathrm{Re}\, ntCar, s)) \wedge \\
Kref\,(City, s) \wedge Kref\,(TypesOfInsurance, s) \wedge \\
Kref\,(CarModel, s) \vee\ Poss(\mathrm{Re}\, ntCar(CarModel), s) \wedge \\
Poss(BookInsurance(TypesOfInsurance), \\
do(\mathrm{Re}\, ntCar, s)) \wedge\ Poss(BookScenicSpots(City), \\
do(BookInsurance, do(\mathrm{Re}\, ntCar, s))) \rightarrow \\
Knows(CarModel, do(Inquire\,\mathrm{Re}\, ntCar, s)) \wedge \\
Knows(TypesOfInsurance, \\
do(InquireInsurance(TypesOfInsurance, s))) \wedge \\
Knows(GreaterFourA, do(InquireScenicSpots, s)) \wedge \\
Kref\,(CarModel, s) \wedge \\
Kref\,(TypesOfInsurance, s) \wedge Kref\,(City, s)
\end{array}$$

In this combination of services, users require the attractions level to be 4A or above and the rental car is full-time on behalf of driving, the above can be mapped to the Desirable Axiom with the situation calculus.

$$\begin{array}{l}
Desirable(BookScenicSpots(City),\ s) \equiv \\
\mathrm{Grade}(GreaterFourA, s) \\
Desirable(\mathrm{Re}\, ntCar(CarModel),\ s) \equiv \\
\mathrm{Re}\, ntCarModel(WholeDay, s)
\end{array}$$

The composition service *Tranvel* is composed of the composition service *BookTicket* . Composition service *Journey* and Hotel Reservation Service are controlled by *Condition* control operator.

Here the *BookTicket* , *Journey* and Hotel Reservation service are seen as the three actions: $a_1 = BookTicket$, $a_2 = Journey$, $a_3 = BookHotel$.

a_1 and a_2 are the implementation precondition of

a_3, the implementation of composition service *Travel* can be controlled by the *Condition* control operator.

The prerequisite of composition service *Travel* was formalized as the following formula in the situation calculus.

$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge Poss(a_3, s) \rightarrow$$
$$Knows(Greater3Star, do(InquireHotel, s))$$

Before the composition service of *Travel* is implemented, the agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge Poss(a_3, s) \rightarrow$$
$$Kref(City, s)$$

The above two formulas, we can get the Action Precondition Axiom.

$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge Poss(a_3, s) \rightarrow$$
$$Kref(City, s) \wedge Knows(Greater3Star, do(InquireHotel, s))$$

The effectiveness and output parameters of composition service are formalized as the following Positive Effect Axiom and Negative Effect Axiom.

$$\gamma_F^+ : a_3 = BookHotel(City) \wedge Instock(Room)$$
$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge$$
$$Poss(a_3, s) \wedge \gamma_F^+(x, a_3, s) \rightarrow$$
$$Own(TanvelMessage, do(a_3, s))$$

The output of the Composition Service was formalized as the following formula with the situation calculus.

$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge$$
$$Poss(a_3, s) \wedge \gamma(x, a_3, s) \rightarrow$$
$$Knows(TanvelMessage, do(a_3, s))$$

5. Summary

This paper introduces the norm to strengthen the semantics from user needs perspective based on the defects of lack of semantics described in OWL-S semantic service of the semantic web service composition. A new way is provided to increase semantic. However, the norm is non-formal and ambiguous, the situation calculus is applied to realize formal norm in order to ensure the accuracy of semantic Web service composition.

REFERENCES

[1] S. N. Sheila and S. Mcilraith, “Analysis and Simulation of Web Services,” *Computer Networks*, Vol. 42, May 2003, pp. 675-693.

[2] A. Ferrara, “Web Services:A Process Algebra Approach,” *Proceedings of the International Conference on Service Oriented Computing*, New York, 2004, pp.1-18

[3] Y. S. Cheng and Z. J. Wang, “Research on Formal Description and Verification of Semantic Service Composition,” *Computer Science*, Vol. 35, No. 4, April 2008, pp. 40-43.

[4] J. Liao, H. Tan and J. D. Liu, “Describing and Verifying Web Service Using Pi-Calculus,” *Chinese Journal of Computers*, Vol. 28, No. 4, April 2005, pp. 635-643.

[5] R. Stamper and K. Liu, “Organisational Dynamics, Social Norms and Information Norms,” *Proceedings of Hawaii International Conference on System Sciences*, Hawaii, 1994, pp. (IV)645-654.

[6] H. Q. Yang, S. Z. Li, J. P. Zhao and W. Pu, “Research on Architecture Design and Requirement Modeling Based on Semiotics of Adaptive Healthcare Information System,” *Journal of Computer Engineering and Applications*, Vol. 45, No. 9, September 2009, pp. 244-248.

[7] G. Wang, “Research and Implement on the Behaviors of Multiple Intelligent Virtual Human Based on the Situation Calculus,” JiangSu University, Zhenjiang, 2006

[8] D. M. Xu and H. Q. Wang, “Muti-Agent Collabo-Ration for B2B Workflow Monitoring,” *Knowledge-Based System*, Vol. 15, No. 8, November 2002, pp. 485-491.

[9] S. Liang, “Research on Technology of Automatic Services Composition Based on Semantic Web,” Chinese Academy of Sciences, Beijing, 2003.

[10] S. McIlraith and T. C. Son, “Adapting Golog for Programming the Semantic Web,” *International Conference on Knowledge Representation and Reasoning*, Toulouse, 2002, pp. 482-493.

[11] Z. S. Gao, “A Multi-Agent Based Approach for Automated Semantic Web Services Composition,” XiHua University, Chengdu, 2006.

Copyright © 2010 SciRes.

J. Software Engineering & Applications, **2010, 3, 784-795**
doi:10.4236/jsea.2010.38091 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

Efficient Fast Multiplication Free Integer Transformation for the 1-D DCT of the H.265 Standard

Mohamed Nasr Haggag[1], **Mohamed El-Sharkawy**[2], **Gamal Fahmy**[1], **Maher Rizkalla**[1]

[1]Electrical Engineering Department, Assiut University, Assiut, Egypt; [2]Egypt Japan University of Science and Technology Borg Al Arab, Alexandria, Egypt.
Email: mrizkall@iupui.edu

Received January 21st 2010; revised June 30th 2010; accepted July 5th 2010.

ABSTRACT

In this paper, efficient one-dimensional (1-*D*) *fast integer transform algorithms of the DCT matrix for the H.*265 *standard is proposed. Based on the symmetric property of the integer transform matrix and the matrix operations, which denote the row/column permutations and the matrix decompositions, along with using the dyadic symmetry modification on the standard matrix, the efficient fast* 1-*D integer transform algorithms are developed. Therefore, the computational complexities of the proposed fast integer transform are smaller than those of the direct method. In addition to computational complexity reduction one of the proposed algorithms provides transformation quality improvement, while the other provides more computational complexity reduction while maintaining almost the same transformation quality. With lower complexity and better transformation quality, the first proposed fast algorithm is suitable to accelerate the quality-demanding video coding computations. On the other hand, with the significant lower complexity, the second proposed fast algorithm is suitable to accelerate the video coding computations.*

Keywords: *Fast Algorithm, HDTV, H.*265, *ICT, Order*-16 *Transform, Video Coding*

1. Introduction

NOWDAYS the demand for higher quality digital video products and faster digital video applications in our daily life activities is increasing. These demands start from our daily necessary needs, like video conferencing, television and surveillance, up to our entertainment, iPods, internet video streaming, digital cameras, and all high definition (HD) products [1,2]. There have been two primary standards organizations driving the definition of video coding. The International Telecommunications Union (ITU), which is an organization focused on telecommunication applications and has created the series of H.26x standards for low bit rate video telephony. These include H.261, H.262, H.263 and H.264. The other organization is the International Standards Organization (ISO), which is more focused on consumer applications and has defined the MPEG series standards for compressing moving pictures. The MPEG standard series include MPEG-1, MPEG-2 and MPEG-4 [1,2].

In [3], the 16 × 16 2-D matrix for the H.265 standard DCT is revealed, using the decomposition and the modification techniques used in [4-7], this paper will introduce two proposed algorithm that will aim to reduce the complexity of the algorithm implementation in addition to making it multiplication-free.

The rest of this paper is organized as follows. In Section 2, review of the integer transformation for the H.265 standard is described. In Section 3, the two proposed efficient fast integer transform algorithms for the 2-D H.265 standard are introduced with the proposed matrix factorizations. Then the computational complexities of these proposed algorithms are discussed. In Section 4, analysis and comparison of transformation quality between the proposed fast algorithms and the original method is shown. In Section 5, comparison of computational complexity done in Section 3 and quality evaluation done in Section 4 is discussed. Finally, we give a conclusion.

2. Review of the Integer Transformation for the H.265 Standard

The H.265 is the newest yet to be released standard for

Copyright © 2010 SciRes.

high definition video processing. From [3], the matrix of the 2-D 16 × 16 integer cosine transformation for the H.265 standard is shown in Equation (1).

The matrix elements in Equation (1) shows that there is symmetric properties between the left side and right side of the matrix, this property will be exploited using matrix decomposition in order to this matrix into the product of sparse matrices in the next section.

$$T=\begin{bmatrix}32&32&32&32&32&32&32&32&32&32&32&32&32&32&32&32\\45&43&40&35&29&21&13&4&-4&-13&-21&-29&-35&-40&-43&-45\\44&38&25&9&-9&-25&-38&-44&-44&-38&-25&-9&9&25&38&44\\43&29&4&-21&-40&-45&-35&-13&13&35&45&40&21&-4&-29&-43\\42&17&-17&-42&-42&-17&17&42&42&17&-17&-42&-42&-17&17&42\\40&4&-35&-43&-13&29&45&21&-21&-45&-29&13&43&35&-4&-40\\38&-9&-44&-25&25&44&9&-38&-38&9&44&25&-25&-44&-9&38\\35&-21&-43&4&45&13&-40&-29&29&40&-13&-45&-4&43&21&-35\\32&-32&-32&32&32&-32&-32&32&32&-32&-32&32&32&-32&-32&32\\29&-40&-13&45&-4&-43&21&35&-35&-21&43&4&-45&13&40&-29\\25&-44&9&38&-38&-9&44&-25&-25&44&-9&-38&38&9&-44&25\\21&-45&29&13&-43&35&4&-40&40&-4&-35&43&-13&-29&45&-21\\17&-42&42&-17&-17&42&-42&17&17&-42&42&-17&-17&42&-42&17\\13&-35&45&-40&21&4&-29&43&-43&29&-4&-21&40&-45&35&-13\\9&-25&38&-44&44&-38&25&-9&-9&25&-38&44&-44&38&-25&9\\4&-13&21&-29&35&-40&43&-45&45&-43&40&-35&29&-21&13&-4\end{bmatrix} \tag{1}$$

3. Proposed Algorithms

In this section, two proposed algorithm for efficient fast multiplication-free for the H.265 standard are presented. The proposed algorithms are done using a combination of Modified Integer Cosine Transformation, matrix decomposition and dyadic symmetry. The common part in the complexity reduction is discussed first, then each algorithm is presented individually and its complexity is calculated with it. The aim of the proposed algorithms is to reduce the computational complexities, which are referred to as the numbers of additions and shift operations as much as possiblewhile maintaining reasonable error margin. The DCT matrix for the H.265 is given as ***T*** in Equation (1).

The symmetric property of the transformation matrix is exploited to decompose it into the product of two sparse matrices. so the transformation matrix in Equation 1 can be rewritten in Equation (2) [4,7].

$$T = T_T \cdot P_1 \tag{2}$$

$$T_T=\begin{bmatrix}32&32&32&32&32&32&32&32&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&-4&-13&-21&-29&-35&-40&-43&-45\\44&38&25&9&-9&-25&-38&-44&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&13&35&45&40&21&-4&-29&-43\\42&17&-17&-42&-42&-17&17&42&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&-21&-45&-29&13&43&35&-4&-40\\38&-9&-44&-25&25&44&9&-38&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&29&40&-13&-45&-4&43&21&-35\\32&-32&-32&32&32&-32&-32&32&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&-35&-21&43&4&-45&13&40&-29\\25&-44&9&38&-38&-9&44&-25&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&40&-4&-35&43&-13&-29&45&-21\\17&-42&42&-17&-17&42&-42&17&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&-43&29&-4&-21&40&-45&35&-13\\9&-25&38&-44&44&-38&25&-9&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&45&-43&40&-35&29&-21&13&-4\end{bmatrix}$$

$$P_1 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

Where

$$T_T = P_r \cdot T_r \tag{3}$$

$$Pr = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

and

$$T_r = \begin{bmatrix}
32 & 32 & 32 & 32 & 32 & 32 & 32 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
44 & 38 & 25 & 9 & -9 & -25 & -38 & -44 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
42 & 17 & -17 & -42 & -42 & -17 & 17 & 42 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
38 & -9 & -44 & -25 & 25 & 44 & 9 & -38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
32 & -32 & -32 & 32 & 32 & -32 & -32 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
25 & -44 & 9 & 38 & -38 & -9 & 44 & -25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
17 & -42 & 42 & -17 & -17 & 42 & -42 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & -25 & 38 & -44 & 44 & -38 & 25 & -9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & -13 & -21 & -29 & -35 & -40 & -43 & -45 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13 & 35 & 45 & 40 & 21 & -4 & -29 & -43 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -21 & -45 & -29 & 13 & 43 & 35 & -4 & -40 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 29 & 40 & -13 & -45 & -4 & 43 & 21 & -35 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -35 & -21 & 43 & 4 & -45 & 13 & 40 & -29 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 40 & -4 & -35 & 43 & -13 & -29 & 45 & -21 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -43 & 29 & -4 & -21 & 40 & -45 & 35 & -13 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 45 & -43 & 40 & -35 & 29 & -21 & 13 & -4
\end{bmatrix}$$

The function of $\boldsymbol{P_1}$ is the post-process matrix for the input data to the matrix multiplication, and the post-process only uses the additions and subtracts. The computational complexities of $\boldsymbol{P_1}$ are 16 additions. In Equation (2), the elements of $\boldsymbol{T_T}$ are scattered; the rearrangement of the elements in $\boldsymbol{T_T}$ is required to group the elements of $\boldsymbol{T_T}$ into two 8 × 8 independent matrices. $\boldsymbol{P_r}$ is a pre-process matrix that permutes the rows of $\boldsymbol{T}$, so that it is rewritten as in Equation (3) [4,7].

As shown in Equation (3), the matrix $\boldsymbol{P_r}$ can be used as a pre-process matrix. $\boldsymbol{P_r}$ doesn't have any complexity at all while serving the purpose of rearranging $\boldsymbol{T_T}$ into $\boldsymbol{T_r}$ where the matrix can be easily represented as the direct sum of its two non-zero areas. The result of the direct sum is shown in Equation (4).

$$T_r = T_{00} \oplus T_{11} \tag{4}$$

Where

$$T_{00} = \begin{bmatrix} 32 & 32 & 32 & 32 & 32 & 32 & 32 & 32 \\ 44 & 38 & 25 & 9 & -9 & -25 & -38 & -44 \\ 42 & 17 & -17 & -42 & -42 & -17 & 17 & 42 \\ 38 & -9 & -44 & -25 & 25 & 44 & 9 & -38 \\ 32 & -32 & -32 & 32 & 32 & -32 & -32 & 32 \\ 25 & -44 & 9 & 38 & -38 & -9 & 44 & -25 \\ 17 & -42 & 42 & -17 & -17 & 42 & -42 & 17 \\ 9 & -25 & 38 & -44 & 44 & -38 & 25 & -9 \end{bmatrix}$$

And

$$T_{11} = \begin{bmatrix} -4 & -13 & -21 & -29 & -35 & -40 & -43 & -45 \\ 13 & 35 & 45 & 40 & 21 & -4 & -29 & -43 \\ -21 & -45 & -29 & 13 & 43 & 35 & -4 & -40 \\ 29 & 40 & -13 & 45 & -4 & 43 & 21 & -35 \\ -35 & -21 & 43 & 4 & -45 & 13 & 40 & -29 \\ 40 & -4 & -35 & 43 & -13 & -29 & 45 & -21 \\ -43 & 29 & -4 & -21 & 40 & -45 & 35 & -13 \\ 45 & -43 & 40 & -35 & 29 & -21 & 13 & -4 \end{bmatrix}$$

The computation of $\boldsymbol{T_{00}}$ can be called the computation of the even frequency part in the transform matrix, and the computation of $\boldsymbol{T_{11}}$ can be called the computation of the odd frequency part in the transform matrix [4].

In Equation (4), the integers of matrix $\boldsymbol{T_{00}}$ have the symmetry property, using matrix decomposition $\boldsymbol{T_{00}}$ can be expressed as the product of the three sparse matrices $\boldsymbol{R_1}$, $\boldsymbol{T_0}$ and $\boldsymbol{R_2}$ as was done with the original matrix $\boldsymbol{T}$ [4,7]. The result of the decomposition for $\boldsymbol{T_{00}}$ is shown in Equation (5).

$$T_{00} = R_r \cdot T_0 \cdot R_1 \tag{5}$$

Where

$$R_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0 = \begin{bmatrix} 32 & 32 & 32 & 32 & 0 & 0 & 0 & 0 \\ 42 & 17 & -17 & -42 & 0 & 0 & 0 & 0 \\ 32 & -32 & -32 & 32 & 0 & 0 & 0 & 0 \\ 17 & -42 & 42 & -17 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -9 & -25 & -38 & -44 \\ 0 & 0 & 0 & 0 & 25 & 44 & 9 & -38 \\ 0 & 0 & 0 & 0 & -38 & -9 & 44 & -25 \\ 0 & 0 & 0 & 0 & 44 & -38 & 25 & -9 \end{bmatrix}$$

And

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

As shown in the above equation, $\boldsymbol{R_1}$ can be implemented using additions and subtractions only and have a complexity of 8 additions, while $\boldsymbol{R_r}$ doesn't have any complexity at all [4], as for $\boldsymbol{T_0}$ it can be expressed by the direct sum of matrices $\boldsymbol{T_{0E}}$ and $\boldsymbol{T_{0O}}$, as shown in Equation (6).

$$T_0 = T_{0E} \oplus T_{0O} \tag{6}$$

Where

$$T_{0E} = \begin{bmatrix} 32 & 32 & 32 & 32 \\ 42 & 17 & -17 & -42 \\ 32 & -32 & -32 & 32 \\ 17 & -42 & 42 & -17 \end{bmatrix}$$

$$T_{0O} = \begin{bmatrix} -9 & -25 & -38 & -44 \\ 25 & 44 & 9 & -38 \\ -38 & -9 & 44 & -25 \\ 44 & -38 & 25 & -9 \end{bmatrix}$$

The symmetry of the matrix $\boldsymbol{T_{0E}}$ can be further exploited using matrix decomposition to decompose the matrix into the product of sparse matrices ***U*₁** and ***U*₂** [4], as shown in Equation (7).

$$T_{0E} = U_1 \cdot U_2 \tag{7}$$

Where

$$U_1 = \begin{bmatrix} 32 & 32 & 0 & 0 \\ 0 & 0 & -17 & -42 \\ 32 & -32 & 0 & 0 \\ 0 & 0 & 42 & -17 \end{bmatrix}$$

And

$$U_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

In Equation (7), $\boldsymbol{U_2}$ can be implemented using additions and subtractions only and have a complexity of 4 additions, while $\boldsymbol{U_1}$ can be implemented using 10 additions and 10 shifts. This would sum up to a complexity of 14 addition and 10 shift operations for Equation (7) [4].

Although there is no symmetry present in matrix $\boldsymbol{T_{0O}}$, the matrix addition can be used to segment the matrix into two matrices with more value coherence in their elements, as shown in Equation (8).

$$T_{0O} = K_0 + K_1 \tag{8}$$

Where $$K_0 = \begin{bmatrix} -9 & -24 & -40 & -36 \\ 24 & 36 & 9 & -40 \\ -40 & -9 & 36 & -24 \\ 36 & -40 & 24 & -9 \end{bmatrix}$$

And $$K_1 = \begin{bmatrix} 0 & -1 & 2 & -8 \\ 1 & 8 & 0 & 2 \\ 2 & 0 & 8 & -1 \\ 8 & 2 & 1 & 0 \end{bmatrix}$$

After using the matrix addition, $\boldsymbol{K_0}$ can be simplified using matrix decomposition into the product of matrices $\boldsymbol{K_2}$ and $\boldsymbol{K_3}$. The result of the decomposition for $\boldsymbol{K_0}$ is shown in Equation (9).

$$K_0 = K_2 \cdot K_3 \tag{9}$$

Where

$$K_2 = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 4 & 0 \\ 0 & 4 & -1 & 0 \\ -4 & 0 & 0 & 1 \end{bmatrix}$$

And

$$K_3 = \begin{bmatrix} -9 & 8 & -8 & 0 \\ -8 & 0 & 9 & -8 \\ 8 & 9 & 0 & -8 \\ 0 & -8 & -8 & -9 \end{bmatrix}$$

As shown in the above equation; $\boldsymbol{K_1}$ can be implemented using additions and shifts only and have a complexity of 8 additions and 8 shifts, the addition between $\boldsymbol{K_1}$ and the product of $\boldsymbol{K_2}$ and $\boldsymbol{K_3}$ has a complexity of 4 additions, As for the matrix $\boldsymbol{K_2}$ it have a complexity of 4 additions and 4 shifts, while on the other hand $\boldsymbol{K_3}$ have a complexity of 12 additions and 4 shifts. This in the end sums the complexity of $\boldsymbol{T_{0O}}$ to 28 additions and 16 shifts [4,5]. All of the above decomposition and summation sum the complexity of $\boldsymbol{T_{00}}$ to 66 additions and 32 shifts [4,5].

Turning to matrix T_{11} which represents the computation of the odd frequency part, the matrix as shown in Equation (10), doesn't have the symmetric property within its elements, in order to be able to decompose this matrix the modification techniques will be used. For the decomposition of this matrix two proposed algorithms will be presented in this paper.

3.1 Proposed Algorithm 1

For Proposed Algorithm 1, the odd frequency modified integer cosine transformation matrix based on the dyadic symmetry concept used by Wai-Kuen Cham in [7] is obtained by modifying the positions of the elements in the matrix to provide the matrix basic vectors with orthogonality regardless of the matrix elements values [7]. The matrix T_{11} and the modified matrix are shown in Equation (10) below.

$$T_{11} \colon= T_{1\text{mod}} \tag{10}$$

Where

$$T_{11} = \begin{bmatrix} -4 & -13 & -21 & -29 & -35 & -40 & -43 & -45 \\ 13 & 35 & 45 & 40 & 21 & -4 & -29 & -43 \\ -21 & -45 & -29 & 13 & 43 & 35 & -4 & -40 \\ 29 & 40 & -13 & 45 & -4 & 43 & 21 & -35 \\ -35 & -21 & 43 & 4 & -45 & 13 & 40 & -29 \\ 40 & -4 & -35 & 43 & -13 & -29 & 45 & -21 \\ -43 & 29 & -4 & -21 & 40 & -45 & 35 & -13 \\ 45 & -43 & 40 & -35 & 29 & -21 & 13 & -4 \end{bmatrix}$$

and

$$T_{1\text{mod}} = \begin{bmatrix} -4 & -13 & -21 & -29 & -35 & -40 & -43 & -45 \\ 35 & 40 & 43 & 45 & -4 & -13 & -21 & -29 \\ -21 & -29 & 4 & 13 & 43 & 45 & -35 & -40 \\ 45 & 43 & -40 & -35 & 29 & 21 & -13 & -4 \\ -43 & -45 & 35 & -40 & -21 & 29 & 4 & -13 \\ 13 & -4 & -29 & 21 & -40 & 35 & 45 & -43 \\ -29 & 21 & -13 & 4 & 45 & -43 & 40 & -35 \\ 40 & -35 & 45 & -43 & 13 & -4 & 29 & -21 \end{bmatrix}$$

Replacing $\boldsymbol{T_{11}}$ with $\boldsymbol{T_{1\text{mod}}}$ as the new odd frequency transformation matrix [7], this matrix can be segmented into two matrices with more value coherence in their elements as was done in the matrix $\boldsymbol{T_{00O}}$, as shown in Equation (11).

$$T_{1\text{mod}} = T_{m1} + T_{m2} \tag{11}$$

Where

$$T_{m1} = \begin{bmatrix} -4 & -16 & -24 & -32 & -36 & -44 & -44 & -44 \\ 36 & 44 & 44 & 44 & -4 & -16 & -24 & -32 \\ -24 & -32 & 4 & 16 & 44 & 44 & -36 & -44 \\ 44 & 44 & -44 & -36 & 32 & 24 & -16 & -4 \\ -44 & -44 & 36 & -44 & -24 & 32 & 4 & -16 \\ 16 & -4 & -32 & 24 & -44 & 36 & 44 & -44 \\ -32 & 24 & -16 & 4 & 44 & -44 & 44 & -36 \\ 44 & -36 & 44 & -44 & 16 & -4 & 32 & -24 \end{bmatrix}$$

and $T_{m2} = \begin{bmatrix} 0 & 3 & 3 & 3 & 1 & 4 & 1 & -1 \\ -1 & -4 & -1 & 1 & 0 & 3 & 3 & 3 \\ 3 & 3 & 0 & -3 & -1 & 1 & 1 & 4 \\ 1 & -1 & 4 & 1 & -3 & -3 & 3 & 0 \\ 1 & 1 & -1 & 4 & 3 & -3 & 0 & 3 \\ -3 & 0 & 3 & -3 & 4 & -1 & 1 & 1 \\ 3 & -3 & 3 & 0 & 1 & 1 & -4 & 1 \\ -4 & 1 & 1 & 1 & -3 & 0 & -3 & 3 \end{bmatrix}$

In Equation (11), the segment $\boldsymbol{T_{m1}}$ from the matrix addition segmentation done on the matrix $\boldsymbol{T_{1\text{mod}}}$ has symmetric properties; hence it can be further simplified using matrix decomposition algorithm into the product of three sparse matrices [7]. The result of this matrix decomposition is shown in Equation (12).

$$T_{m1} = M_1 \cdot M_2 \cdot M_3 \cdot 4 \tag{12}$$

Where $M_1 = \begin{bmatrix} -2 & 0 & 1 & -1 & -1 & 3 & -1 & 0 \\ 3 & -1 & 1 & 1 & 0 & 2 & 0 & 1 \\ -1 & -3 & 1 & 0 & 1 & 0 & 2 & -1 \\ 0 & 1 & 0 & 1 & 3 & 1 & -1 & -2 \\ 1 & -1 & -3 & -2 & 0 & 1 & 0 & -1 \\ 1 & 1 & 1 & 0 & -2 & 0 & 1 & -3 \\ 0 & -2 & 0 & 1 & -1 & -1 & -3 & -1 \\ -1 & 0 & -2 & 3 & -1 & 1 & 1 & 0 \end{bmatrix}$

$$M_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

and

$$M_3 = \begin{bmatrix} -1 & 1 & -2 & 0 & 1 & 0 & 2 & 0 \\ -2 & 0 & 1 & -1 & -2 & 0 & 1 & 0 \\ 2 & -1 & 0 & 0 & -1 & -1 & 2 & 0 \\ 1 & 1 & 0 & -2 & 0 & 1 & 0 & 2 \\ 0 & 2 & 1 & 1 & 0 & -2 & 0 & 1 \\ -1 & -2 & 0 & 0 & 1 & -1 & 0 & 2 \\ 0 & 0 & -2 & 1 & -2 & 0 & -1 & 1 \\ 0 & 0 & 1 & 2 & 0 & 2 & 1 & 1 \end{bmatrix}$$

As shown from the equation above; $\boldsymbol{M_1}$ can be implemented using additions and shifts only and have a complexity of 48 additions and 8 shifts, while $\boldsymbol{M_2}$ can be implemented using only additions and have a complexity of 16 additions, As for the matrix $\boldsymbol{M_3}$ it have a complexity of 32 additions and 8 shifts. This in the end sums the complexity of $\boldsymbol{T_{m1}}$ to 96 additions and 40 shifts [7]. On the other hand, $\boldsymbol{T_{m2}}$ can be implemented using additions and shifts only, and has a complexity of 72 additions and 8 shifts [4,5,7].

By using equations from Equation (2) to Equation (12), the efficient fast multiplication-free integer transformation for the 2-D DCT matrix for the H.265 standard for proposed algorithm 1 is given as shown in Equation (13).

$$T = P_r.\left\{\left[R_r \cdot \left[\left(U_1 \cdot U_2\right) \oplus \left(K_1 + \left(K_2 \cdot K_3\right)\right)\right] \cdot R_1\right] \oplus \left[T_{m1} + \left(M_1 \cdot M_2 \cdot M_3 \cdot 4\right)\right]\right\} \cdot P_1 \tag{13}$$

3.2 Proposed Algorithm 2

In the proposed algorithm presented in this section the same complexity reduction and decomposition techniques will be done, the difference will be that in the odd frequencies matrix in the step done in Equation (10) instead of only using dyadic symmetry to rearrange the elements of the matrix, a modification in values of the elements will be done so that it matches the matrix Tm1 in Equation (11). The resultant odd frequency matrix is given in Equation (14).

$$T_{11}\colon= T_{1\text{mod}2} \tag{14}$$

Where

$$T_{1\text{mod}2} = \begin{bmatrix} -4 & -16 & -24 & -32 & -36 & -44 & -44 & -44 \\ 36 & 44 & 44 & 44 & -4 & -16 & -24 & -32 \\ -24 & -32 & 4 & 16 & 44 & 44 & -36 & -44 \\ 44 & 44 & -44 & -36 & 32 & 24 & -16 & -4 \\ -44 & -44 & 36 & -44 & -24 & 32 & 4 & -16 \\ 16 & -4 & -32 & 24 & -44 & 36 & 44 & -44 \\ -32 & 24 & -16 & 4 & 44 & -44 & 44 & -36 \\ 44 & -36 & 44 & -44 & 16 & -4 & 32 & -24 \end{bmatrix}$$

The odd frequency matrix for proposed algorithm 2 $\boldsymbol{T_{1\text{mod}2}}$ can be decomposed for complexity reduction in

Copyright © 2010 SciRes.

the same manner as done in Equation (12) in Subsection 3.1. This means that the two proposed algorithm have exactly the same decomposition and segmentation, the only exception is that the odd frequency part for proposed algorithm 2 consists of $\boldsymbol{T_{m1}}$ only while for proposed algorithm 1 it consists of the addition of $\boldsymbol{T_{m1}}$ and $\boldsymbol{T_{m2}}$.

By using equations from Equation (2) to Equation (14), the efficient fast multiplication-free integer transformation for the 2-D DCT matrix for the H.265 standard for proposed algorithm 2 is given as shown in Equation (15).

$$T = P_r \cdot \left\{ \left[R_r \cdot \left[\left(U_1 \cdot U_2 \right) \oplus \left(K_1 + \left(K_2 \cdot K_3 \right) \right) \right] \cdot R_1 \right] \oplus \left[\left(M_1 \cdot M_2 \cdot M_3 \cdot 4 \right) \right] \right\} \cdot P_1 \qquad (15)$$

For the proposed algorithms and the original algorithm, the complexity evaluation is done by calculating the number of additions, shifts and multiplications needed to implement it. The complexity evaluation summary for the proposed and original algorithms is shown in **Table 1**.

4. Analysis and Comparison of Transformation Quality

In order to test the efficiency of the proposed algorithms, evaluation of the quality of the reconstructed video compared to the original video is done using the quality assessment metrics; the three quality metrics used in this paper are the MSE, PSNR and the SSIM. The tests done in this section are applied to standard high definition video quality assessment sequences as developed by Dr. Karl Mauthe at Taurus Media Technik. The full description of the test sequences used is shown in **Table 2**.

Using the Matlab computational tool the quality metrics for original and the proposed algorithms were calculated for 100 frames of the four different standard test sequences. The aim is to evaluate the algorithms quality and reliability, and determine the efficiency of each of the proposed algorithms compared to the original.

Table 1. Complexity Evaluation Comparison table for Original and Proposed Algorithms

	Complexity		
Operation	Original Algorithm	Proposed Algorithm 1	Proposed Algorithm 2
Additions	240	242	162
Multiplications	256	0	0
Shifts	0	58	50

Table 2. Test Sequences Information

Sequence	#Frames	Short Description
Blue sky	250	Top of two trees against blue sky. High contrast, small color differences in the sky, many details. Camera rotation.
Pedestrian Area	375	Shot of a pedestrian area. Low camera position, people pass by very close to the camera. High depth of field. Static camera.
Riverbed	250	Riverbed seen through the water. Very hard to code.
Station	313	View from a bridge to Munich station. Evening shot. Long zoom out. Many details, regular structures (tracks).

The quality metrics results of the quality assessment for the Blue Sky sequence are shown in the **Figures 1**, **2** and **3**.

The quality metrics results of the quality assessment for the Pedestrian Area sequence are shown in the **Figures 4**, **5** and **6**.

The quality metrics results of the quality assessment for the Riverbed sequence are shown in the **Figures 7**, **8** and **9**.

The quality metrics results of the quality assessment for the Station_2 sequence are shown in the **Figures 10**, **11** and **12**.

The figures from 1 to 12 show all the test results done to evaluate the quality of the proposed algorithms compared to the original algorithm, all these results are

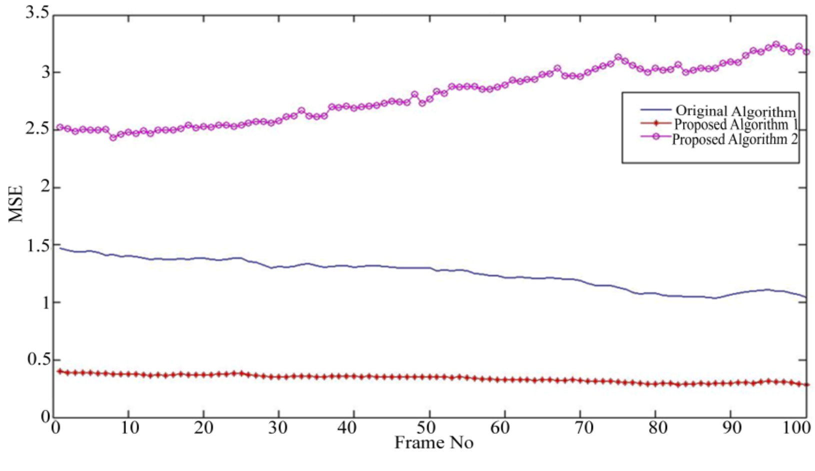


Figure 1. Mean MSE for Original and Proposed Algorithms

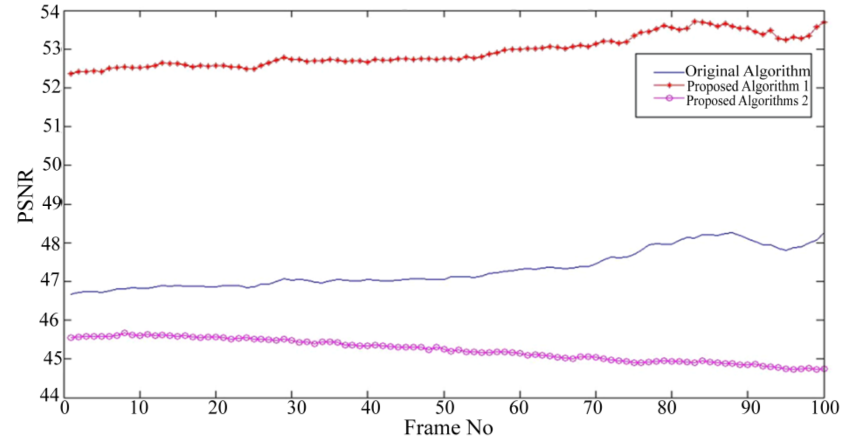


Figure 2. Mean PSNR for Original and Proposed Algorithms

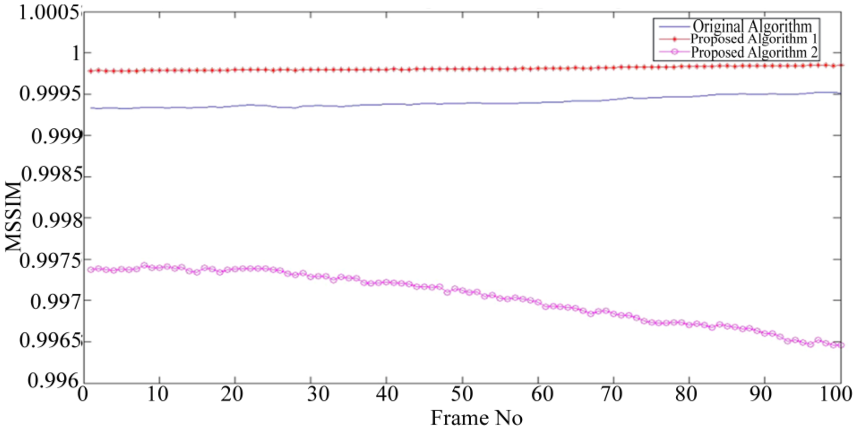


Figure 3. Mean SSIM for Original and Proposed Algorithms

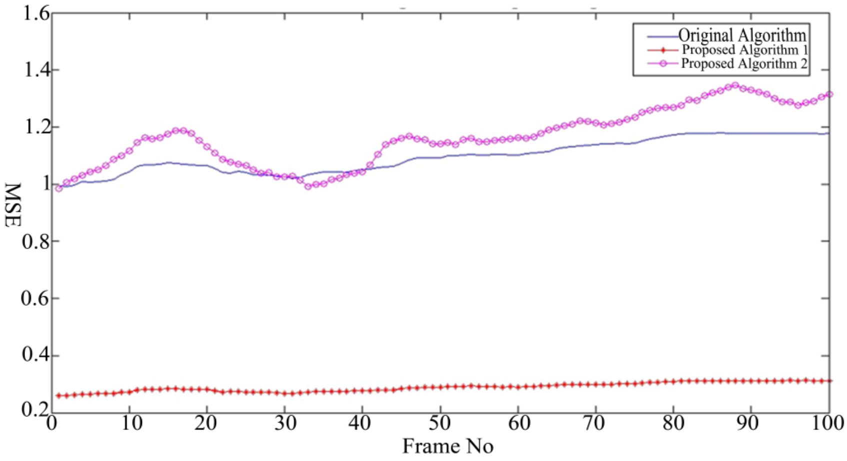


Figure 4. Mean MSE for Original and Proposed Algorithms

summarized and combined with the complexity of all the algorithms to determine the efficiency of the proposed algorithms. **Tables 3, 4** and **5** show the summarized results for the MSE, PSNR and the MSSIM respectively.

5. Comparison of Computational Complexity and Quality Evaluation

In this section, evaluation of the proposed algorithms versus the original algorithm is done through a comparison

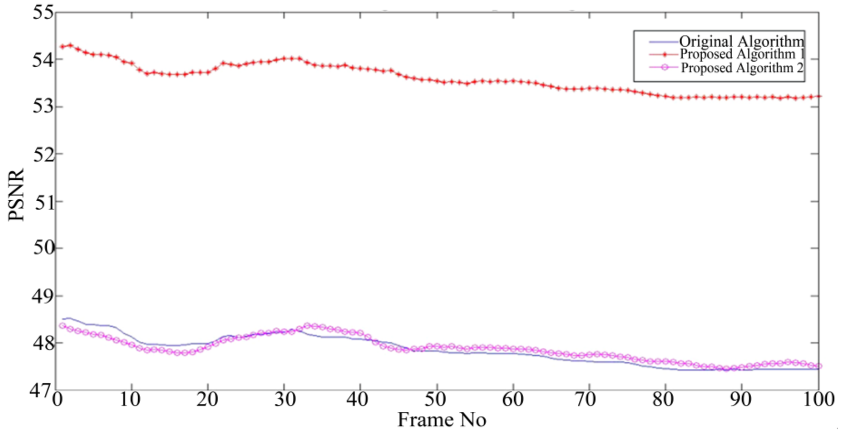


Figure 5. Mean PSNR for Original and Proposed Algorithms

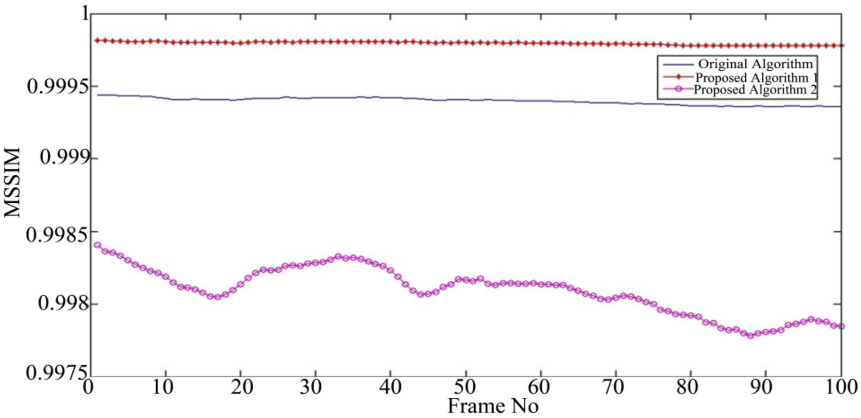


Figure 6. Mean SSIM for Original and Proposed Algorithms

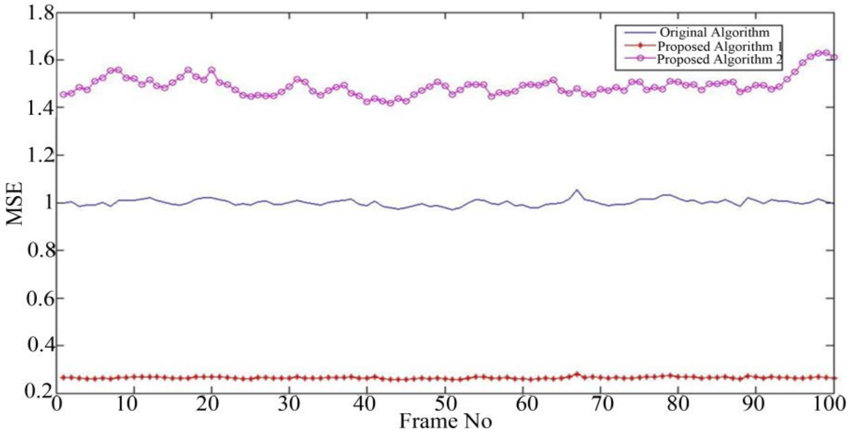


Figure 7. Mean MSE for Original and Proposed Algorithms

Table 3. Average MSE Comparison table for Original and Proposed Algorithms

	Average MSE		
Sequence	Original Algorithm	Proposed Algorithm 1	Proposed Algorithm 2
Blue_Sky	1.2513	0.3399	2.8050
Pedestrian_Area	1.0966	0.2893	1.1636
Riverbed	1.0009	0.2646	1.4918
Station	0.9314	0.2449	1.0008

Table 4. Average PSNR Comparison table for Original and Proposed Algorithms

	Average PSNR		
Sequence	Original Algorithm	Proposed Algorithm 1	Proposed Algorithm 2
Blue_Sky	47.3093	52.9384	45.2233
Pedestrian_Area	47.8787	53.5946	47.8456
Riverbed	48.2731	54.0194	47.3802
Station	49.0446	54.7329	48.3484

Table 5. Average SSIM Comparison table for Original and Proposed Algorithms

	Average MSSIM		
Sequence	Original Algorithm	Proposed Algorithm 1	Proposed Algorithm 2
Blue_Sky	0.9994	0.9998	0.9970
Pedestrian_Area	0.9994	0.9998	0.9981
Riverbed	0.9995	0.9998	0.9976
Station	0.9995	0.9998	0.9982

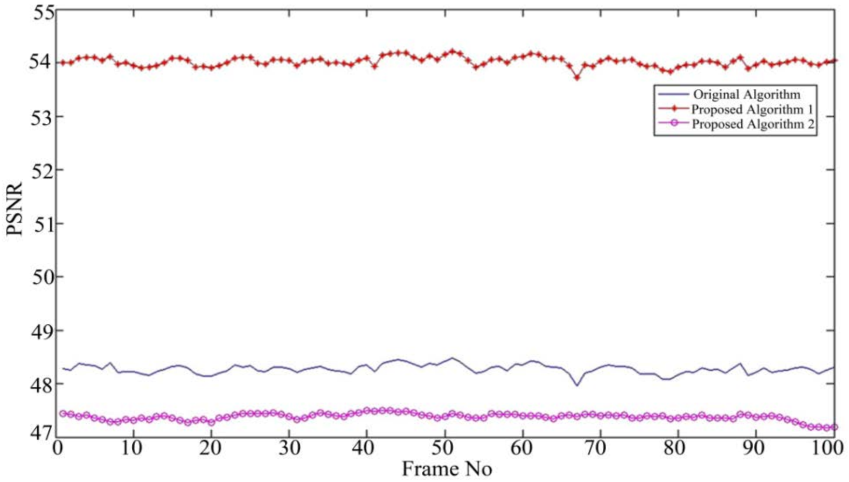


Figure 8. Mean PSNR for Original and Proposed Algorithms

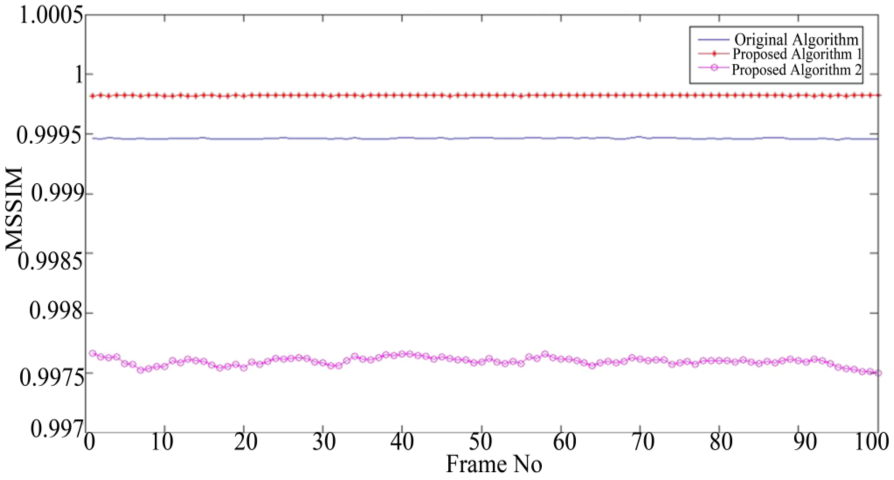


Figure 9. Mean SSIM for Original and Proposed Algorithms

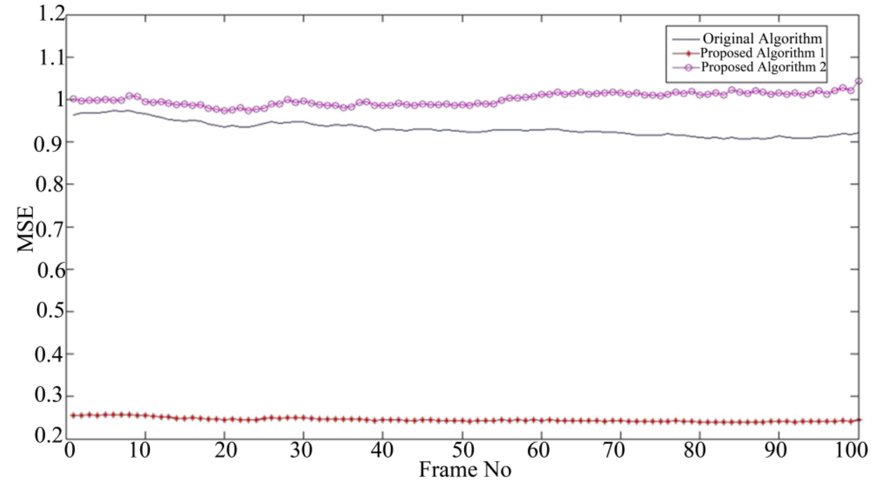


Figure 10. Mean MSE for Original and Proposed Algorithms

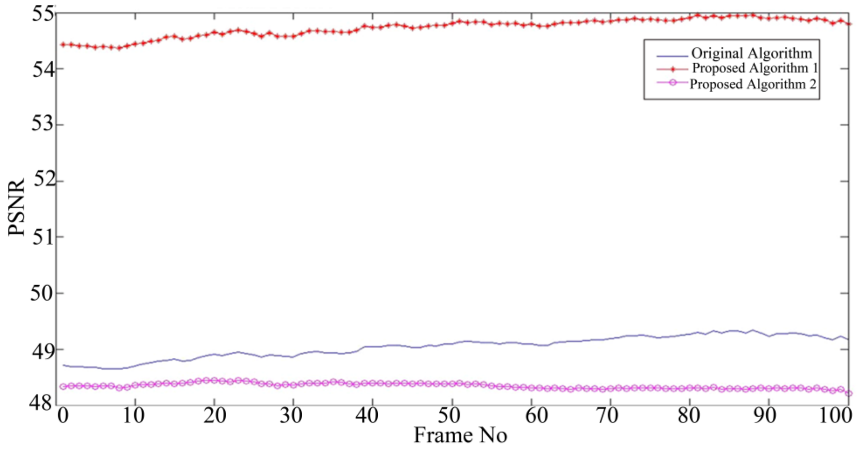


Figure 11. Mean PSNR for Original and Proposed Algorithms

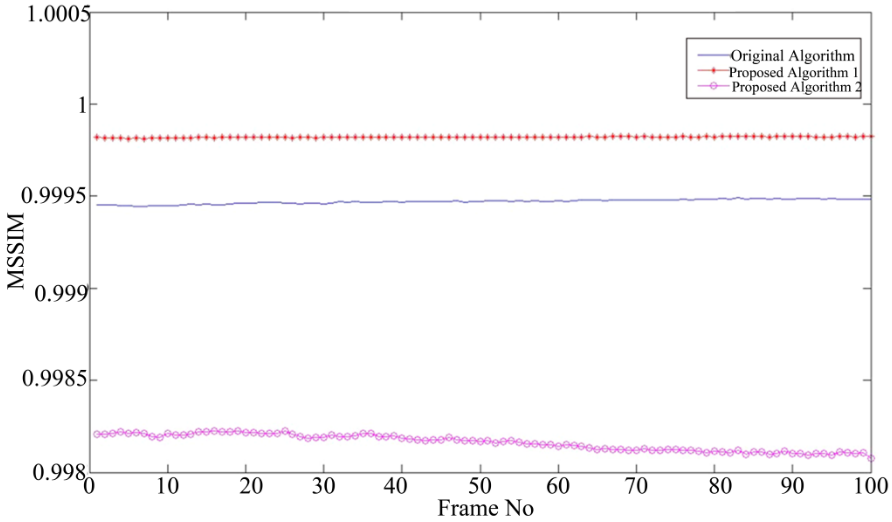


Figure 12. Mean SSIM for Original and Proposed Algorithms

between the results of Section 3 and Section 4.

In terms of quality assessment **Table 3** clearly shows the advantage of Proposed Algorithm 1 over Proposed Algorithm 2 and even the Original algorithm as it was able to achieve the smallest values for the MSE index over 4 different test sequences. **Table 4** backs the results shown in **Table 3**, and also shows that the different in quality measured by the PSNR index between the originnal algorithm and Proposed Algorithm 2 is not high, which means that both achieve comparable qualities. **Table 5** shows that according to a more (HVS) based index, Proposed Algorithm 1 still achieve the best result with the original algorithm coming in second place, and proposed algorithm achieving lower structural similarity than both. However the values in this table indicate that the results for the three algorithms are all almost in the same range, and achieve what is considered high structural similarity. These improved results achieved by the proposed algorithms compared to the original one are due to applying the dyadic symmetry on the transformation matrix odd-frequency part which makes the transformation matrix as a whole symmetric and hence eliminates the transformation error as the product of the transformation matrix multiplied by its transpose will yield an identity matrix.

As for the complexity point of view, taking into consideration that multiplications are the process in terms of complexity and hardware followed by additions and finally shifts. **Table 1** clearly shows the superiority of proposed Algorithm 2 over the other two algorithms in terms of less complexity, and emphasis the fact that both of the proposed algorithms are multiplication-free, also the table shows that proposed Algorithm 1 is far less complex than the original algorithm, although it requires two more additions than the original algorithm, it has no multiplications and small number of shifts.

Finally form all the results obtained in this section, it can be concluded that the proposed Algorithm 1 achieves the highest quality in encoding and decoding while maintaining less complexity than the original algorithm, which means that it would be suitable for quality-oriented applications, however on the other hand proposed Algorithm 2 achieves the dramatically less complexity than the original algorithm without having any noticeable or detectable quality degradation, which makes this algorithm suitable for speed or hardware oriented applications.

6. Conclusions

A new 16 × 16 DCT matrix was recently introduced for the highly anticipated H.265 standard, this DCT matrix is developed for high definition videos encoding and decoding, the aim is to make them less complex and faster for video communication and transmission, like in high definition broadcasting and storage. Two new algorithms were proposed in this paper. The first technique is a quality oriented algorithm while offering multiplication-free complexity. The second algorithm is a complexity and speed oriented algorithm while maintaining almost the same quality offered by the original algorithm.

The aim of proposing an efficient fast 1-D algorithm for this DCT matrix is to reduce the complexity and hence the hardware and increase the speed of computation to meet the constantly improving demands in the fields of communication and transmission. Quality Assessment Tests were carried out and the quality metrics MSE, PSNR and SSIM were calculated to evaluate the performance of the proposed algorithms compared to the original one. The test results showed that the first proposed algorithm offers better quality, objective and subjective, while offering less complexity and multiplication-free computation. While the second proposed algorithm offers almost the same quality, objective and subjective, while offering much less complexity and multiplication-free computation than the original algorithm and the first proposed one.

REFERENCES

[1] J.-B. Lee and H. Kalva, “The VC-1 and H.264 Video Compression Standards for Broadband Video Services,” Springer, New York, 2008.

[2] H. 264/MPEG-4 Part 10: Overview. http://www.vcodex.com/files/h264_overview_orig.pdf

[3] http://www.h265.net/index.php?s=dct

[4] C.-P. Fan and G.-A. Su, “Efficient Fast 1-D 8×8 Inverse Integer Transform for VC-1 Application,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 4, 2009, pp. 584-590.

[5] C.-P. Fanm and G.-A. Su, “Efficient Low-Cost Sharing Design of Fast 1-D Inverse Integer Transform Algorithms for H.264/AVC and VC-1,” *IEEE Signal Processing Letters*, Vol. 15, 2008, pp. 926-929.

[6] W.-K. Cham, “Development of Integer Cosine Transformation by the Principle of Dyadic Symmetry,” *IEEE Proceedings*, Vol. 136, No. 4, August 1989, pp. 276-282.

[7] J. Dong, K. N. Ngan, C.-K. Fong and W.-K. Cham, “2D Order-16 Integer Transforms for HD Video Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 10, October 2009, pp. 1463-1474.

J. Software Engineering & Applications, **2010, 3, 796-802**
doi:10.4236/jsea.2010.38092 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

Contour-Based Image Segmentation Using Selective Visual Attention

Engin Mendi[1], Mariofanna Milanova[2]

[1]Department of Applied Science, University of Arkansas at Little Rock, Little Rock, United States; [2]Department of Computer Science, University of Arkansas at Little Rock, Little Rock, United States.
Email: esmendi@ualr.edu

Received July 12th 2010; revised July 27th 2010; accepted August 12th 2010.

ABSTRACT

In many medical image segmentation applications identifying and extracting the region of interest (ROI) accurately is an important step. The usual approach to extract ROI is to apply image segmentation methods. In this paper, we focus on extracting ROI by segmentation based on visual attended locations. Chan-Vese active contour model is used for image segmentation and attended locations are determined by SaliencyToolbox. The implementation of the toolbox is extension of the saliency map-based model of bottom-up attention, by a process of inferring the extent of a proto-object at the attended location from the maps that are used to compute the saliency map. When the set of regions of interest is selected, these regions need to be represented with the highest quality while the remaining parts of the processed image could be represented with a lower quality. The method has been successfully tested on medical images and ROIs are extracted.

Keywords: *Active Contours, Selective Visual Attention, Image Segmentation, Telemedicine*

1. Introduction

Identifying and extracting the region of interest (ROI) accurately is an important step before coding and compressing the image data for efficient transmission or storage. The main requirement for multimedia encoding techniques is achieving high level ratio of compression for effective use of bandwidth and energy consumption. There is an increased demand for faster transmitting diagnostic medical images in telemedicine applications. ROI must be compressed by lossless or near lossless algorithm while on the other hand, the background region must be compressed with some loss of information that is still recognizable using JP2K standard or Inverse Difference Pyramidal (IDP) decomposition (**Figure 1**).

There are a wide variety of approaches for the segmentation problem. One of the popular approaches is active contour models, also called snakes. The basic idea is to start with a curve around the object to be detected, the curve moves towards an "optimal" position and shape by minimizing its own energy. Based on the Mumford-Shah functional [1-3] for segmentation, Chan and Vese [4] proposed a new level set model for active contours to detect objects whose boundaries are not necessarily defined by a gradient.

Visual attention is the process of selecting and getting visual information based on saliency in the image itself (bottom-up), and on prior knowledge about scenes, objects and their interrelations (top-down) [5,6]. Visual attention addresses both problems by selectively enhancing perception at the attended location, and by successively shifting the focus of attention to multiple locations. It is also important for selecting the object of interest from the input information and [7] provides the brain with a mechanism of focusing computational resources on one object at a time, either driven by low-level image properties (bottom-up attention) or based on a specific task (top-down attention). Moving the focus of attention to locations one by one enables sequential recognition of objects at these locations. The more one knows about an image, the higher the top-down influence part will be. On the other side, for an unknown image, the bottom-up attention mechanism is very important. This is the case when no medical doctor is sending remotely the image.

Hu *et al.* [8] used visual attention algorithm to define a method leading to the automatic choice of the best features for a given medical application. Mancas presents application of computational attention in medical images

Copyright © 2010 SciRes.

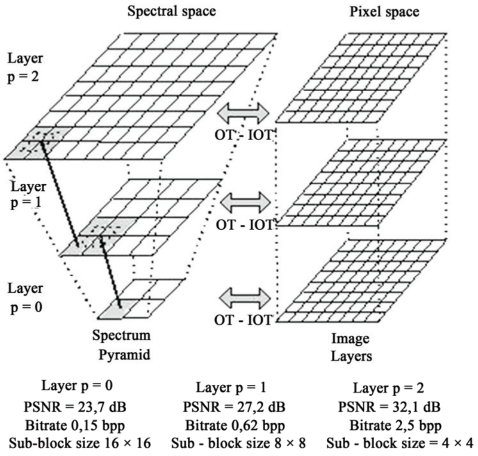


Figure 1. Illustration of the image decomposition called Inverse Difference Pyramid (IDP) [9]

[10]. Attention may be due to: 1) local properties (a feature saliency depends on its neighborhood); 2) global properties (a feature saliency depends on the whole visual field). Attention model can be applied directly on the medical images in order to find rare grey level: for instance liver images, where only the grey level variations should be enough to detect pathologies.

Here ROI was extracted with active contours based on selective visual attention. Chan-Vese active contour model is used for image segmentation and attended locations are determined by SaliencyToolbox [11] which is extension of the saliency map-based model of bottom-up attention [12], by a process of inferring the extent of a proto-object at the attended location from the maps that are used to compute the saliency map. In this paper we extend our previous study of markless segmentation of medical images [13]. Here we compare results using different local and global features for a coarse localization of possibly pathological areas. We also show the results extracting multiple ROIs in a single image. The paper is organized as follows: Section 2 provides an overview of the Chan-Vese model. Section 3 presents the bottom-up salient region selection model. Section 4 describes the application of our approach. Section 5 presents the conclusions of this paper in a summary.

2. Chan-Vese Model

The Mumford-Shah model [1-3] is a variational problem for approximating a given image by a piecewise smooth image of minimal complexity. Let u be differentiable on R and allowed to be discontinuous across C, Mumford-Shah energy functional is as follows:

$$F(\phi,C)=\frac{1}{\sigma^2}\int_R(\phi-f)^2dx+\int_{R\setminus C}\|\nabla\phi\|^2dx+\lambda|C| \tag{1}$$

where R is the image domain, f is the feature intensity, C is the curve, ϕ is the smoothed image, $|C|$ is the arc length of C and σ, λ are positive parameters. Segmentation problem is restated as finding optimal approximations of g by piece-wise smooth functions u, whose restrictions to the regions are differentiable.

The Chan-Vese model [4] is a special case of the Mumford Shah model by restricting (1) to piece-wise constant functions ϕ and looking for the best approximation ϕ of f taking only two values. Then the energy functional in (1) is expressed in terms of the level set function by replacing the C by Lipschitz function ϕ:

$$F(\phi,c_1,c_2)=\int_R\left\{|\nabla H_\varepsilon(\phi)|+\lambda\left[H_\varepsilon(\phi)(c_1-f)^2\right.\right.$$
$$\left.\left.+(1-H_\varepsilon(\phi))(c_2-f)^2\right]\right\}dx \tag{2}$$

where H is the Heaviside function, defined by:

$$H(z)=\begin{cases}1, & if\ \ z\geq 0\\ 0, & if\ \ z<0\end{cases}$$

and H_ε is the regularization of H.

Constant functions c_1 and c_2 of level sets can be expressed by minimizing the energy functional with respect to the constants and keeping the level sets fixed:

$$c_1(\phi)=\frac{\int_D fH_\varepsilon(\phi)dx}{\int_D H_\varepsilon(\phi)dx} \tag{3}$$

$$c_2(\phi)=\frac{\int_D f(1-H_\varepsilon(\phi))dx}{\int_D (1-H_\varepsilon(\phi))dx} \tag{4}$$

Combining the energy terms and replacing the singular term $H_\varepsilon{}'(\phi)$ by $|\nabla\phi|$, the corresponding Euler-Lagrange equation for ϕ, using gradient descent in artificial time leads to:

$$\frac{\partial\phi}{\partial t}=|\nabla\phi|\left\{\kappa(\phi)-\lambda[(c_1-f)^2-(c_2-f)^2]\right\} \tag{5}$$

where $\kappa(\phi)$ is the curvature of the level sets and $\kappa(\phi)=-div(\frac{\nabla\phi}{|\nabla\phi|})$. A multigrid scheme on the discretized Euler-Lagrange Equation (5) is used for the minimization of Chan-Vese energy functional.

$$\inf_{\phi,c_1,c_2}F(\phi,c_1,c_2) \tag{6}$$

which is

$$\min_\phi F(\phi,c_1,c_2)=\int_D\left\{|\nabla\phi|+\lambda[(c_1-f)^2\right.$$
$$\left.+(c_2-f)^2]\phi\right\}dx \tag{7}$$

The explicit formula provided by (5) is solved by using gradient descent procedure as described in [14].

3. Bottom-Up Salient Region Selection Model

The model of bottom-up salient region selection presented by [7,11] based on the model of saliency-based bottom-up attention by Itti-Koch [15,16] is implemented as part of the SaliencyToolbox [11]. This model introduces a process of inferring the extent of a proto-object at the attended location from the maps that are used to compute the saliency map.

Itti-Koch model [15,16] is a bottom-up selective visual attention based on serially scanning a saliency map that is computed from local feature contrasts, for salient locations in the order of decreasing saliency (**Figure 2**). Presented with a manually preprocessed input image, their model replicates human viewing behavior for artificial and natural scenes.

Visual input [14] is first decomposed into a set of topographic feature maps. Different spatial locations then compete for saliency within each map, such that only locations which locally stand out from their surround can persist. All feature maps feed, in a purely bottom-up manner, into a master saliency map. The purpose of the saliency map is to represent the saliency at every location in the visual field by a scalar quantity and to guide the selection of attended locations, based on the spatial distribution of saliency. However this model's usefulness [17] as a front-end for object recognition is limited by the fact that its output is merely a pair of coordinates in the image corresponding to the most salient location.

This model is extended [7,11] by a process of inferring the extent of a proto-object, contiguous region of high activity in feature map, at the attended location from the maps that are used to compute the saliency map. This is achieved by introducing feedback connections in the saliency computation hierarchy in order to estimate the proto-object region based on the maps and salient locations computed in Itti-Koch model [15,16]. Different visual features that contribute to attentive selection are combined into one single topographically oriented saliency map which integrates the normalized information from the individual feature maps into one global measure of conspicuity.

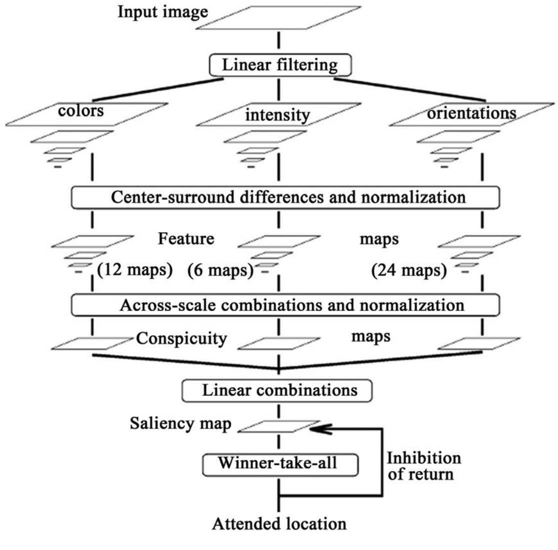


Figure 2. General architecture of Itti-Koch model [14]

The locations [7] in the saliency map compete for the highest saliency value by means of a winner take-all (WTA) networks of integrate-and-fire neurons. The winning of this process is attended to, and the saliency map is inhibited. Continuing WTA competition produces the second most salient location, which is attended to subsequently and then inhibited, thus allowing the model to simulate a scan path over the image in the order of decreasing saliency of the attended locations.

4. Experimental Results

Image segmentations of attended locations of four medical images were used in the application of the new approach to image segmentation. All conspicuity maps, saliency maps, WTAs and attended locations are operated by SaliencyToolbox [11]. The saliency map is summed by conspicuity maps that provide information of color, intensity and orientation. The attended locations are set as initial contours to be segmented by using Chan-Vese Model [4].

For example, in **Figure 3**, seborrheic keratosis is segmented from a skin image. **Figure 4** shows multiple basal cell carcinoma segmentation. **Figure 5** and **Figure 6** show segmentation of cherry angiomas of the trunk and basal cell carcioma of the cheek, respectively. **Table 1** shows the stimulated time (ms) that attended locations (AL) took. Global low level attention is applied directly on the medical images. Low level features bring some top down information about grey levels. A final attention map, for example **Figure 3(g)**, can help the contour segmentation algorithm by focusing only at separated regions with the greatest chance of being pathological. This approach works on the images where pathological pixel grey level is different from normal tissues grey-level.

For telemedicine applications, we have integrated image segmentation with adaptive compression technique. The proposed compression technique is based on the hypothesis that image resolution exponentially decreases from the fovea to the retina periphery. This hypothesis

Table 1. Simulated time (ms) of attended locations

	1st AL	2nd AL	3rd AL	4th AL
Figure 3	239,98			
Figure 4	39,171	200,151		
Figure 5	174,193	72,18		
Figure 6	98,189	157,132	167,242	185,115

Copyright © 2010 SciRes.

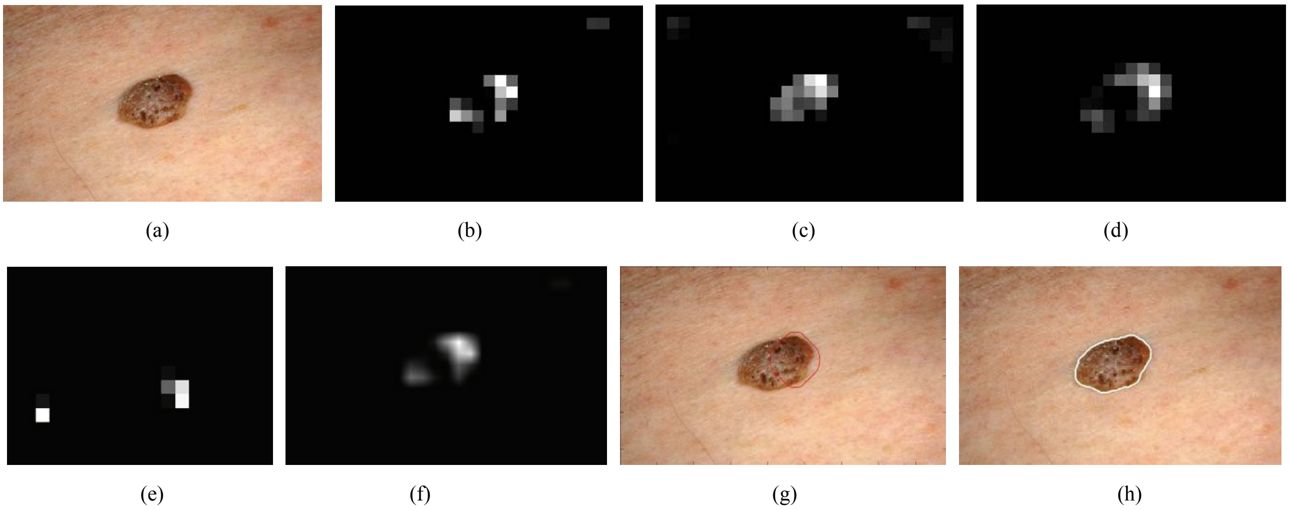


Figure 3. (a) Input image; (b) conspicuity map for color contrast; (c) conspicuity map for intensity contrast; (d) conspicuity map for orientation contrast; (e) saliency map combined by conspicuity maps; (f) WTA map for the attended location; (g) attended location; (h) active contours based on the attended location

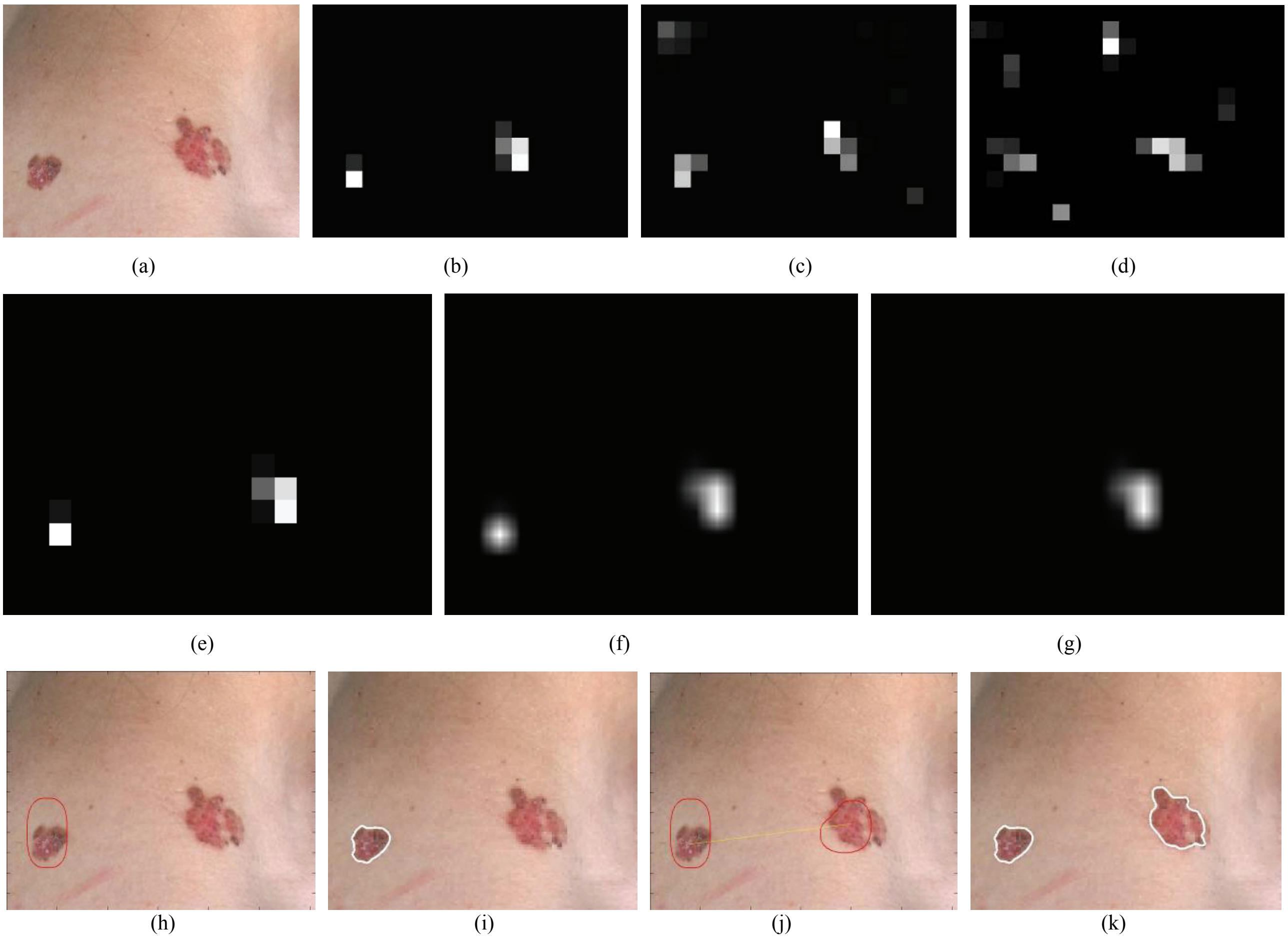


Figure 4. (a) Input image; (b) conspicuity map for color contrast; (c) conspicuity map for intensity contrast; (d) conspicuity map for skin contrast ; (e) saliency map combined by conspicuity maps; (f) WTA map for the first attended location; (g) WTA map for the second attended location; (h) first attended location; (i) active contours based on first attended location; (j) second attended location; (k) active contours based first two attended locations

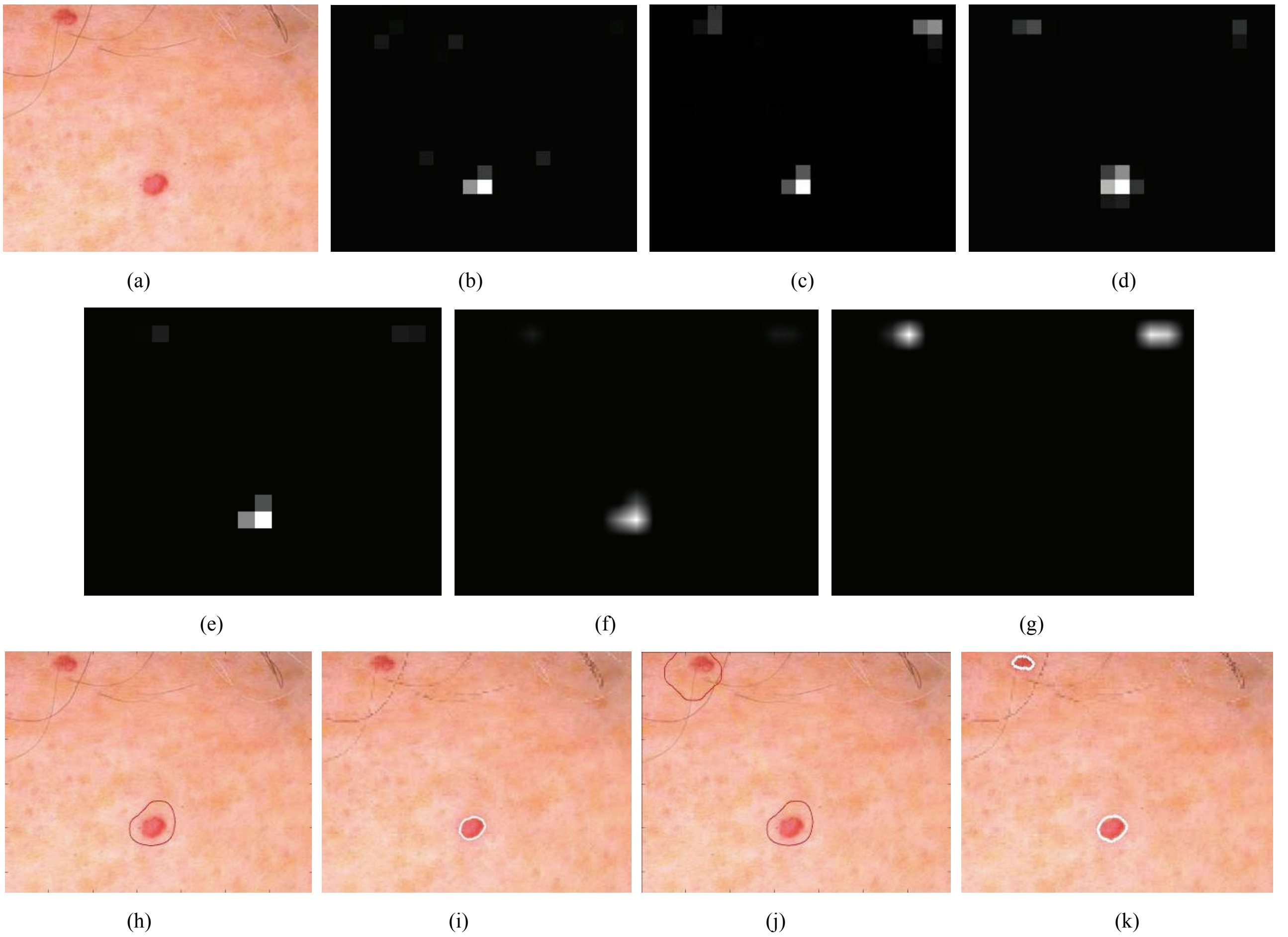
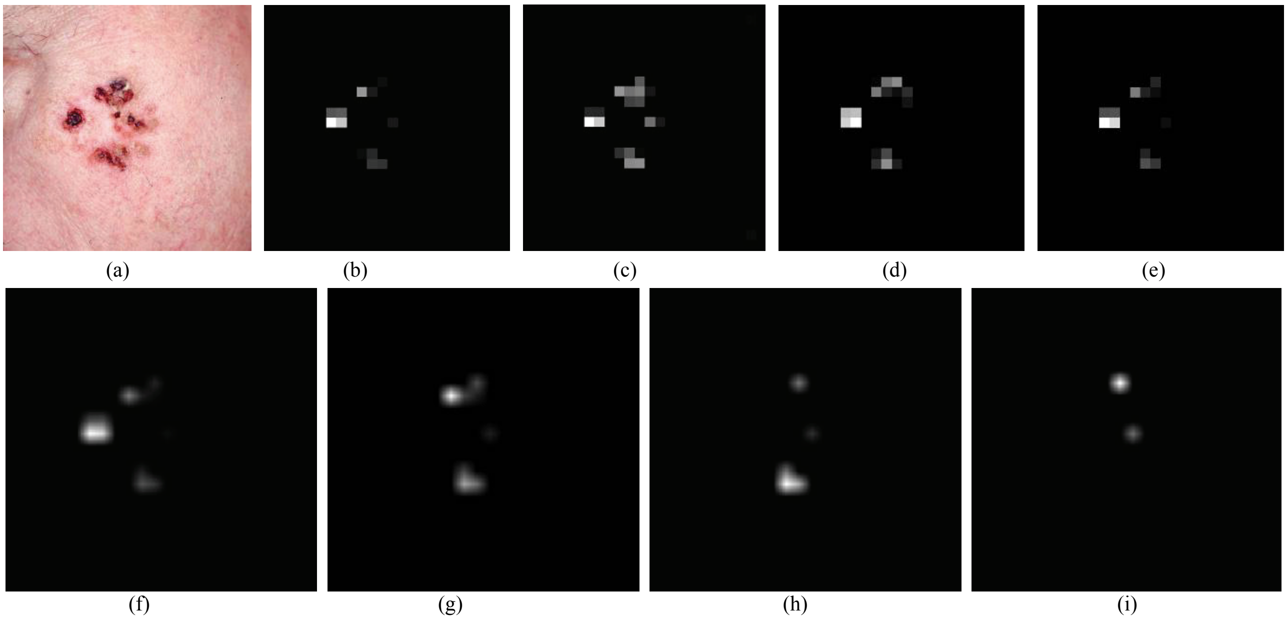


Figure 5. (a) Input image; (b) conspicuity map for color contrast; (c) conspicuity map for intensity contrast; (d) conspicuity map for orientation contrast; (e) saliency map combined by conspicuity maps; (f) WTA map for the first attended location; (g) WTA map for the second attended location; (h) first attended location; (i) active contours based on first attended location; (j) second attended location; (k) active contours based first two attended locations



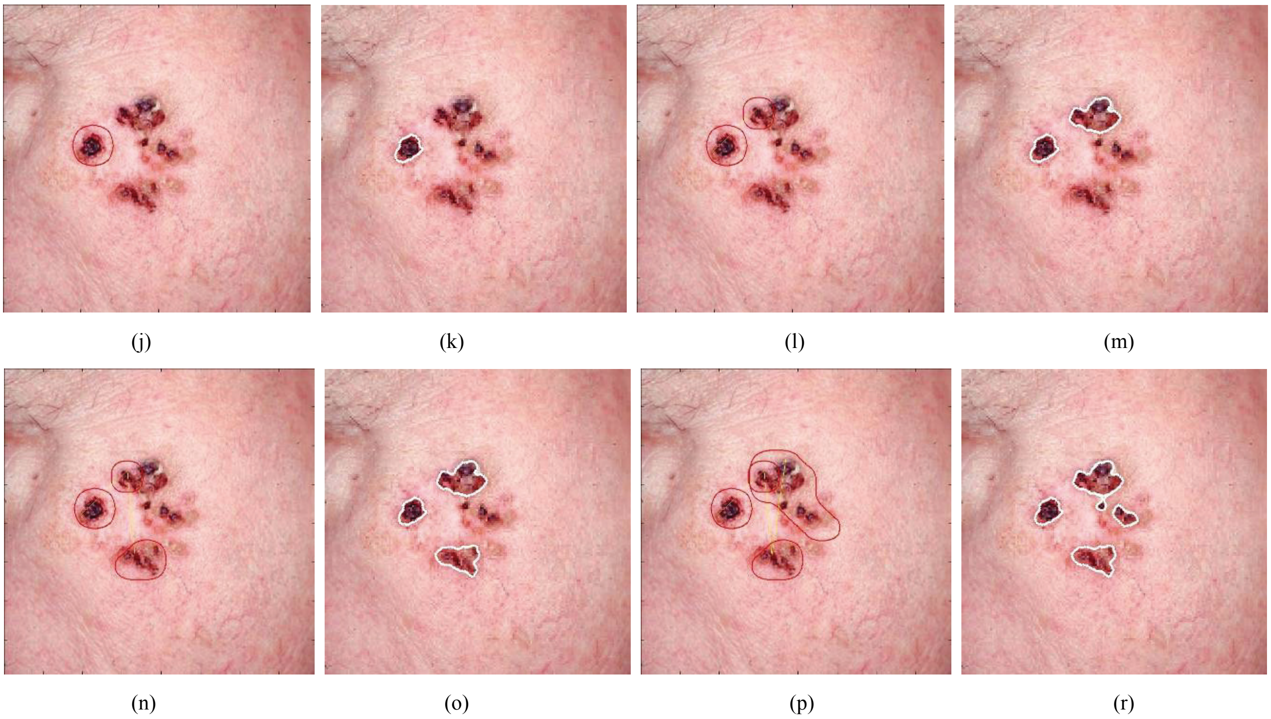


Figure 6. (a) Input image; (b) conspicuity map for color contrast; (c) conspicuity map for intensity contrast; (d) conspicuity map for orientation contrast; (e) saliency map combined by conspicuity maps; (f-i) WTA maps for the first, second, third and fourth attended locations, respectively; (j, l, n, p) WTA maps for the first, second, third and fourth attended locations, respectively; (k, m, o, r) active contours based first fourth attended locations, respectively

can be represented computationally with different resolutions. The visual attention points may be considered as the most highlighted areas of the visual attention model. These points are the most salient regions in the image. When going further from these points of attention, the resolution of the other areas dramatically decrease. Different authors work with different filters and different kernel size to mimic this perceptual behavior [18]. These models ignore contextual information representation. When the set of regions of interest is selected, these regions need to be represented with the highest quality while the remaining parts of the processed image could be represented with a lower quality. In result, higher compression is obtained. The adaptive compression technique proposed is based on new image decomposition called Inverse Difference Pyramid (IDP) [9]. This approach is developed by analogy with the hypothesis for the way humans do image recognition using consecutive approximations with increasing similarity. A hierarchical decomposition is used for the image representation. The approximations in the consecutive decomposition layers are represented by the neurons in the hidden layers of the neural networks (NN) [19]. The most specific features of IDP method are that the images are processed in consecutive layers with higher quality. This approach offers the ability to transfer the image via Internet layer by layer, without sending the same information twice.

5. Conclusions

The paper presents a new markerless approach for medical image segmentation by combining saliency attention maps with active contours. The Chan-Vese active contour model [4] has been implemented by setting attended locations as initial contours. Attended locations are extracted with SaliencyToolbox [11]. It is anticipated that this process will be useful for identifying and extracting the ROI accurately. The combination of the two techniques minimizes user interaction and speeds up the entire segmentation process. The method has been successfully tested on medical images and the ROI is extracted. The proposed approach works for allocating tumors in medical images.

REFERENCES

[1] D. Mumford and J. Shah, “Optimal Approximation by Piecewise Smooth Functions and Associated Variationol Problems,” *Communications on Pure and Applied Mathematics*, Vol. 42, 1989, pp. 577-685.

[2] D. Mumford and J. Shah, “Boundary Detection by Minimizing Functionals, I,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, 1985, pp. 22-26.

[3] J. Shah, “Piecewise Smooth Approximations of Functions,” *Calculus of Variations and Partial Differential Equations*,

Vol. 2, 1994, pp. 315-328.

[4] T. F. Chan and L. A. Vese, “Active Contours without Edges,” *IEEE Transactions on Image Processing*, Vol. 10, No. 2, 2001, pp. 266-277.

[5] D. Walters, U. Rutishauser, C. Koch and P. Perona, “Selective Visual Attention Enables Learning and Recognition of Multiple Objects in Cluttered Scenes,” *Computer Vision and Image Understanding*, Vol. 100, No. 1-2, 2005, pp. 41-63.

[6] M. Milanova, S. Rubin, R. Kountchev, V. Todorov and R. Kountcheva, “Combined Visual Attention Model for Video Sequences,” *IEEE ICPR’*2008 *International Conference on Pattern Recognition*, Tampa, 2008, pp. 1-4.

[7] D. Walter, “Interactions of Visual Attention and Object Recognition: Computational Modeling, Algorithms, and Psychophysics,” PhD Thesis, California Institute of Technology, Pasadena, 2006.

[8] X.-P. Hu, “Hot Spot Detection based on Feature Space Representation of Visual Search,” *IEEE Transactions on Medical Images*, Vol. 22, No. 9, 2003, pp. 1152-1162.

[9] A. Kountchev and A. Mironov, “Algorithms for Pyramid Image Decomposition,” In: N. Mastorakis Ed., *Intelligent Systems and Computer Science*, WSEAS Press, Danvers, 1999, pp. 196-200.

[10] M. Mancas, “Image Perception: Relative Influence of Bottom-Up and Top-Down Attention,” *Attention in Cognitive Systems*, *Lecture Notes in Computer Science*, Vol. 5395, Greece, 2008, pp. 212-226.

[11] D. Walther and C. Koch, “Modeling Attention to Salient Proto-Objects,” *Neural Networks*, Vol. 19, No. 9, 2006, pp. 1395-1407.

[12] C. Koch and S. Ullman, “Shifts in Selective Visual-Attention—Towards the Underlying Neural Circuitry,” *Human Neurobiology*, Vol. 4, No. 4, 1985, pp.219-227.

[13] E. Mendi and M. Milanova, “Image Segmentation with Active Contours Based on Selective Visual Attention,” 8*th WSEAS International Conference on Signal Processing* (*SIP*’09), Istanbul, 2009, pp. 79-84.

[14] K. Fundana, N. C. Overgaard and A. Heyden, “Deformable Shape Priors in Chan-Vese Segmentation of Image Sequences,” *Proceedings International Conference on Image Processing*, San Antonio, Vol. 1, 2007, pp. 285-288.

[15] L. Itti, C. Koch and E. Niebur, “A Model of Saliency-Based Visual Attention for Rapid Scene Analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 11, 1998, pp. 1254-1259.

[16] C. Koch and L. Itti, “Computational Modeling of Visual Attention,” *Nature Reviews Neuroscience*, Vol. 2, 2001, pp. 194-203.

[17] D. Walther, U. Rutishauser, C. Koch and P. Perona, “On the Usefulness of Attention for Object Recognition,” *Workshop on Attention and Performance in Computational Vision at ECCV*, Prague, 2004, pp. 96- 103.

[18] M. Mancas, B. Gosselin and B Macq, “Perceptual Image Representation,” *EURASIP Journal of Image and Video Processing*, Vol. 2007, 2007, pp. 1-9.

[19] R. Kountchev, S. Rubin, M. Milanova, V. Todorov and R. Kountcheva, “Non-Linear Image Representation Based on IDP with NN,” *WSEAS Transactions on Signal Processing*, Istanbul, Vol. 5, No. 9, 2009, pp. 315-325.

Copyright © 2010 SciRes.

J. Software Engineering & Applications, **2010, 3, 803-812**
doi:10.4236/jsea.2010.38093 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

Verifying Monoid and Group Morphisms over Strongly Connected Algebraic Automata

Nazir Ahmad Zafar, Ajmal Hussain, Amir Ali

Faculty of Information Technology, University of Central Punjab, Lahore, Pakistan.
Email: {dr.zafar, ajmal, amiralishaihid}@ucp.edu.pk

Received June 12th 2010; revised July 10th 2010; accepted July 15th 2010.

ABSTRACT

Automata theory has played an important role in theoretical computer science since last couple of decades. The algebraic automaton has emerged with several modern applications, for example, optimization of programs, design of model checkers, development of theorem provers because of having certain interesting properties and structures from algebraic theory of mathematics. Design of a complex system requires functionality and also needs to model its control behavior. Z notation has proved to be an effective tool for describing state space of a system and then defining operations over it. Consequently, an integration of algebraic automata and Z will be a useful computer tool which can be used for modeling of complex systems. In this paper, we have linked algebraic automata and Z defining a relationship between fundamentals of these approaches which is refinement of our previous work. At first, we have described strongly connected algebraic automata. Then homomorphism and its variants over strongly connected automata are specified. Next, monoid endomorphisms and group automorphisms are formalized. Finally, equivalence of endomorphisms and automorphisms under certain assumptions are described. The specification is analyzed and validated using Z/Eves toolset.

Keywords: *Formal Methods, Z notation, Algebraic Automata, Validation and Verification*

1. Introduction

Now a day, the complex and critical systems have shifted from electro-mechanical systems to computerized systems. As a result, it requires controlling these systems by computer software. When software is used in such complex systems its failure may cause a huge loss in terms of deaths, injuries or financial losses. Therefore, constructing correct software is as important as its other counterparts including hardware and electro-mechanical systems [1]. Formal methods are mathematical approaches used for specification of properties of software and hardware systems [2]. Using formal methods, we can describe a mathematical model of a system and then it can be analyzed and validated by computer tools such as model checkers and theorem provers [3]. Current formal approaches cannot be applied to develop a system using a single formal technique and as a result its integration is required with other traditional approaches. That is why integration of approaches has become a well-researched area [4-10]. There are a large variety of specification techniques which are suitable for specific aspects in the software development process. For example, Z, algebraic techniques, VDM, RAISE and B are usually used for defining data types while petri-nets, process algebra and automata are best suited for modeling dynamic aspects of systems [11]. Because of having well-defined mathematical syntax and semantics of the formal techniques, it is required to identify and link these approaches for modeling of consistent, correct and complete computerized systems.

There exists a lot of work on integration of approaches but does not exist much work on formalization of graphical based notations. The work [12,13] of Dong *et al.* is close to ours in which they have integrated Object Z and Timed Automata. Another piece of good work is listed in [14,15] in which R. L. Constable has given a constructive formalization of some important concepts of automata using Nuprl. A combination of Z with statecharts is given in [9]. A relationship is investigated between Petri-nets and Z in [16,17]. An integration of B and UML is given in [18,19]. Wechler W. has introduced algebraic structures in fuzzy automata [20]. In [21], a treatment of fuzzy automata and fuzzy language theory is given when the set of possible values is a closed interval [0, 1]. Ito M. has described formal languages and automata from the algebraic point of view in which the algebraic structures of automata are investigated and then a kind of global theory is treated [22]. Kaynar D. K *et al.* has presented a

Copyright © 2010 SciRes.

modeling framework of timed computing systems [23]. In [24], Godsil C. *et al*. has presented some ideas of algebraic graphs with an emphasis on current topics rather than on classical structures of graphs.

Automata theory has proved to be a useful tool in theoretical computer science since last couple of decades. The algebraic automaton is an advanced form of automata which has very interesting properties and structures from algebraic theory of mathematics. The applications of algebraic theory of automata are not limited to computers but are being seen in many other disciplines of science and engineering, for example, representing characteristics of natural phenomena in biology [25] and modeling of chemical systems using cellular automata [26]. Another interesting application, in which a system is described for synthesizing physics-based animation programs based on hybrid automata, is presented in [27]. Because of having special algebraic characteristic, an automaton can easily be extended to develop algebraic automata.

In our previous work, some preliminary results on integration of algebraic automata and Z notation were presented in terms of a formal proof of equivalence in endomorphisms and automorphisms over strongly connected automata [28,29]. Few inconsistencies and errors are observed there which are presented after correction. For example, the set of states in the strongly connected automata must be finite which was not described correctly. The homomorphism was defined as a change in state using schema structure which is now corrected to define as a relationship between two different structures of strongly connected automata. The similar corrections have been made in defining endomorphism, isomorphism and automorphism. Because all of these morphisms are reused in defining monoids and groups therefore rest of the specifications is refined accordingly. First, we have given formal specification of strongly connected algebraic automaton and then homomorphism with its variants over the automata are formalized and generalized. Next, monoid endomorphisms and group automorphisms are described. Finally, a formal proof of their equivalence is given under certain assumptions. The formal models are analyzed and validated using Z/Eves toolset.

In Section 2, an overview of Z notation is given. In Section 3, Formal models of morphisms over algebraic automata are provided. In Section 4, an approach is presented to analyze the resultant formal models. Conclusion and future work are discussed in Section 5.

2. An Introduction to Z Notation

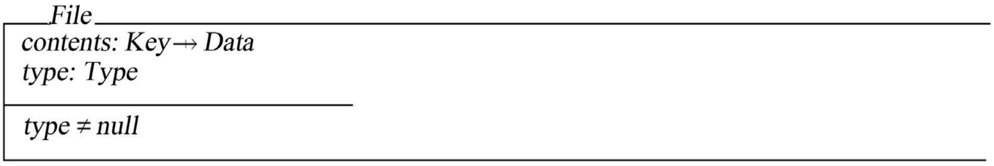
Formal methods are approaches used for describing and analyzing properties of software and hardware systems [30]. There are several ways in which formal methods may be classified. One frequently-made distinction is between property oriented and model oriented methods [31]. Property oriented approaches are used to describe the operations defining their relationships. Property oriented methods usually consist of two parts. The first one is the signature part which is used for defining syntax of an operation and the other one is an equations part used for defining semantics of the operation by a set of axioms called the rules. The OBJ language [32] and algebraic specification of abstract data types [33] are examples of property oriented methods. Model oriented methods are used to construct a model of a system and then allows to defining operations over it [34]. The Z is a model oriented approach based on set theory and first order predicate logic [35] used for specifying behavior of abstract data types and sequential programs.

A brief overview of Z is given by taking a case from the book "Using Z: specification, refinement and proof" by Woodcock and Davies [36]. A programming interface is taken as case study for file system. A list of operations which is defined after defining the entire system is: 1) read: used to read a piece of data from a file, 2) write: used to write a piece of data to a file, 3) access: may change the availability of a file for reading and writing over the file of the system.

A file is represented as a schema using a relation between storage keys and data elements. Basic set types are used for simple specification. The variables name, type, keys and data elements of a file are represented as Name, Type, Key and Data respectively in Z notation. An axiomatic definition is used to define a variable null which is used to prove that the type of a file cannot be null even there are no contents on a file.

[*Name*, *Type*, *Key*, *Data*]; *null*: *Type*

A file consists of its contents and type which are specified by contents and type respectively. The schema structure is usually used because of keeping specification flexible and extensible. In the predicate part, an invariant is described proving that the file type is non null even there are no contents in it. As a file can associate a key with at most one piece of a data and hence the relation contents is assumed to be a partial function.



File

$contents: Key \rightarrow\!\!\!\!\!+ Data$

$type: Type$

$type \neq null$

The read operation is defined over the file to interrogate its state. A read operation requires an existing key as input and provides the corresponding data as output. The symbol Ξ is used when there is no change in the state. Now the structure Ξ*File* means that the bindings of *File* and *File*' are of same type and equal in value. The decorated file, *File*', represents the next state of the file. In the read operation, it is unchanged because the variable *k?* is given as input and the output is returned to the variable *d!*. The symbols *?* and *!* are used with input and output variables respectively. In the predicate part of the schema, first it is ensured that the input key *k?* is in the domain of contents. Then the value of data against it is returned to the output variable *d!*.

Read

$\Xi File$
$k?: Key$
$d!: Data$

$k? \in dom\ contents$
$d! = contents\ k?$

Another operation is defined to write contents over the given file. The symbol Δ is used when there is a change in the state. In the schema defined below, the structure Δ*File* gives a relationship between *File* and *File'* representing that the bindings of *File* are changed. In this case, the write operation defined below replaces the data stored under an existing key and provides no output. The old value of contents is updated with maplet $k? \mapsto d?$. The symbol $\oplus$ is an override operator used to replace the previous value of a key with the new one in a given function.

Write

$\Delta File$
$k?: Key$
$d?: Data$

$k? \in dom\ contents$
$contents' = contents \oplus \{(k? \mapsto d?)\}$
$type = type'$

As a system may contain a number of files indexed using a set of names and some of which might be open hence it consists of two components namely collection of files known to the system and set of files currently open. The variable *file* is used as a partial function to associate the file name and its contents. The variable *open* is of type of power set of *Name*. The set of files which are open must be a subset of set of total files as described below in the schema.

System

$file: Name \rightarrow\!\!\!\!\!+ File$
$open: \mathbb{P}\ Name$

$open \subseteq dom\ file$

As the open and close operations neither change name of any file nor add and remove any file hence both are the access operations. It may change the availability of a file for reading or writing. In the schema *FileAccess* given below, the variable *n?* is used to check if the file to be accessed exist in the system. Further, it is also described that the file is left unchanged.

FileAccess

$\Delta System$
$n?: Name$

$n? \in dom\ file$
$file' = file$

If we require creating another system with same pattern but with different components, then renaming can be used rather than creating the new system from the scratch. Renaming is sometimes useful because in this way we are able to introduce a different collection of variables with the same pattern. For example, we might wish to introduce variables *newfile* and *newopen* under the constraint of existing system *System*. In this case, the new system *NewSystem* can be created in horizontal form by defining: *NewSystem* ≙ *System*[*newfile*/*file*, *newopen*/*open*] which is equivalent to the schema given below in the vertical form.

```
NewSystem
newfile: Name ⇸ File
newopen: ℙ Name
─────────────────
newopen ⊆ dom newfile
```

3. Algebraic Automata and Morphisms

As we know that automata theory has become a basis in the theoretical computer science because of its applications in modeling scientific and engineering problems [37-40]. Algebraic automaton has some properties and structures from algebraic theory of mathematics. The algebraic automata have emerged with several modern applications in computer science. Further, the applications of algebraic theory are not limited to computers but are being seen in many other disciplines of science, e.g., representing chemical and physical phenomena in chemistry and biology. It is a well known fact that a given automata may have different implementations and consequently its time and space complexity must be different which is one of the major issues in modeling using automata. Therefore it is required to describe the formal specification of automata for its optimal implementation. Formal models of algebraic automata are given in this section. At first, formal description of some important concepts of algebraic automata using Z is given. Homomorphism and its variants are described. Then, formal description of monoid (group) on a set of endomorphisms (automorphisms) is presented. Finally an equivalence of monoid endomorphisms and group automorphisms over strongly connected automata is described. It is to be noted that the definitions used in this section are based on the book "Algebraic Theory of Automata and Languages" [22].

3.1 Strongly Connected Algebraic Automaton

A strongly connected automaton is a one for which if for any two given states there exists a string s such that the delta function connects these states through the string s. A strongly connected algebraic automaton (SCAA) is a 5-tuple (Q, ∑, δ, T, Epsilon), where 1) Q is a finite nonempty set of states, 2) ∑ is a finite set of alphabets, 3) δ is a transition function which takes a state and a string and produces a state, 4) T is a set of strings where each string is based on set of alphabets and 5) Epsilon is a null string. To formalize SCAA, Q and ∑ are denoted by *S* and *X* respectively at an abstract level of specification [*S*, *X*].

To describe a set of states for SCAA, a variable *states* is introduced. Since, a given state q is of type Q therefore *states* must be of type of power set of Q. For a set of alphabets, the variable *alphabets* is used which is of type of power set of *X*. As we know that δ is a function because for each input (*q, u*), where *q* is a state and *u* is a string there must be a unique state, which is image of (*q*, *u*) under the transition function δ. Hence we can declare δ as, $S \times \text{seq}\, X \rightarrow S$.

The *delta* function takes a state and a string as input and produces the same state or new state as output. Because we need to compute the set of all the strings based on the set of alphabets and hence the fourth variable is used and denoted by *strings* which is of type of power set of set of all the sequences, *i.e.*, ℙ(seq *X*). As we know that a sequence can be empty and hence a fifth variable is used representing it and is denoted by *epsilon* of type *seq X*. The schema structure is used for composition because it is very powerful at abstract level of specification and helps in describing a good specification approach. All of the above components are encapsulated and put in the schema named as *SCAA*. The formal description of the connected automaton is described.

Invariants: 1) The null string is an element of *strings.* 2) If the transition function takes a state and null string as input, then it produces the same state. 3) For each (*s*, (⟨*a*⟩⁀*u*)), where s is state, an alphabet and *u* is a string, the *delta* function is defined as: *delta* (*s*, (⟨*a*⟩ ⁀ *u*)) = *delta* ((*delta* (*s*, ⟨*a*⟩)), *u*). 4) For any two states *q1* and *q2*, there exists a string s such that: *delta* (*q*1, *s*) = *q2*.

3.2 Homomorphism and its Variants

The word homomorphism means "same shape" and is an interesting concept because a similarity of structures can be verified by it. It is a structure in abstract algebra which preserves a mapping between two algebraic structures, for example, monoid, groups, rings, vector spaces.

SCAA

$states\colon \mathbb{F}\ S$

$alphabets\colon \mathbb{F}\ X$

$strings\colon \mathbb{F}\ (seq\ X)$

$delta\colon S\ \times seq\ X \rightarrow S$

$epsilon\colon seq\ X$

$epsilon \in strings$

$\forall q\colon S \mid q \in states \bullet delta\ (q, epsilon\ ) = q$

$\forall\ q\colon S;\ a\colon X;\ u\colon \mathrm{seq}\ X \mid q \in \mathrm{states}\ \wedge a \in alphabets \wedge u \in strings$

$\quad \bullet\ delta\ (q,(\langle a\rangle \frown u))= delta\ ((delta\ (q, \langle a\rangle)), u)$

$\forall q1, q2\colon S \mid q1 \in states \wedge q2 \in states$

$\quad \bullet\ \exists s\colon \mathrm{seq}\ X \mid \mathrm{s} \in \mathrm{strings} \bullet\ delta\ (q1\ s) = q2$

Now we give formal specification of it and its variants over strongly connected automata. In [22], Ito M. has given a concept of homomorphism and its variants over algebraic automata.

Let A = (Q1, ∑1, δ1) and B = (Q2, ∑2, δ2) be two strongly connected automata represented by *SCAAa* and *SCAAb* respectively in Z. Let ρ be a mapping from Q1 into Q2. If ρ (δ1(q, x)) = δ2(ρ(q), x) holds for any q ∈ Q1 and x ∈ ∑1, then ρ is called a homomorphism from Q1 to Q2. The formal definitions of both the automata are described by using the concept of renaming.

SCAAa ≙ *SCAA*[*statesa*/*states*, *alphabetsa*/*alphabets*, *stringsa*/*strings*, *deltaa*/*delta*, *epsilona*/*epsilon*]

SCAAb ≙ *SCAA*[*statesb*/*states*, *alphabetsb*/*alphabets*, *stringsb*/*strings*, *deltab*/*delta*, *epsilonb*/*epsilon*]

A formal definition of homomorphism from A into B in terms of a schema is given below. It consists of three components, *i.e.*, *SCAAa*, *SCAAa* and *row*. The variable *row* is a mapping from *Q1* into *Q2*. The sets *Q1* and *Q2* are used for states of *SCAAa* and *SCAAb* respectively.

Homomorphism

SCAAa

SCAAb

$row\colon S \longrightarrow S$

$\forall q\colon S;\ s\colon\ \mathrm{seq}\ X \mid\ q \in\ statesa \wedge s \in\ stringsa \wedge ran\ row\ \subseteq stringsb$

$\quad \bullet\ row\ (deltaa\ \ (q, s)) = deltab\ ((row\ q), s\ )$

Invariants: 1) For every q in set of states and s in set of strings of the first automata, if the mapping *row* satisfies the condition: *row* (*deltaa* (*q*, *s*)) = *deltab* ((*row q*), *s*) then it conforms a homomorphism from automata SCAAa into SCAAb.

If SCAAa = SCAAb in the homomorphism then it is called an endomorphism. The mapping *row* is defined from set of states *S* into itself. We have induced the formal definition of endomorphism from the definition of homomorphism because it is a special case of it.

Endomorphism

Homomorphism

$statesa = statesb \wedge\ alphabetsa = alphabetsb$

$stringsa = stringsb \wedge\ deltaa = deltab \wedge\ epsilona = epsilonb$

Let us define the bijection over two given sets. Let X and Y are two nonempty sets. A mapping π from X to Y is called one to one if different elements of X have different images in Y. That is, ∀ x1, x2 ∈ X; y ∈ Y • π(x1)

= y and π(x2) = y ⇒ x1 = x2. The mapping π is called onto if each element of Y is an image of some element of X. If a mapping is one to one and onto then it is called bijective. If the mapping defined in case of homomorphism is bijective from algebraic automata SCAAa to SCAAb then it is called an isomorphism and the automata are said to be isomorphic. Formal description of isomorphism from SCAAa to SCAAb is given below.

Isomorphism

Homomorphism

$\forall\, q1, q2{:}\, S;\ q{:}\, S \mid q1 \in statesa \wedge q2 \in statesa \wedge q \in statesb$

$\quad \bullet\ (q1, q) \in row \wedge (q2, q) \in row \Rightarrow q1 = q2$

$ran\ row = statesb$

Invariants: 1) For all *q1*, *q2* in set of states of SCAAa and *q* in set of states of SCAAb, if images of *q1* and *q2* are same under the mapping *row* then the elements *q1* and *q2* must be same. 2) Each element of the set of states of automata SCAAb is an image of some element of automata SCAAa under the mapping *row*.

If SCAAa = SCAAb then an isomorphism becomes an automorphism. Its formal description is given below along with its invariants which are not explained here because it is a repetition as given above in the schema *Isomorphism*.

Automorphism

Isomorphism

$statesa = statesb \wedge alphabetsa = alphabetsb \wedge stringsa = stringsb$

$deltaa = deltab \wedge epsilona = epsilonb$

3.3 Formal Models of Monoid and Group Morphisms

Let G be a nonempty set. The structure (G,*) under binary operation * is monoid if (i) ∀ x, y ∈ G, x*y ∈ G, (ii) ∀ x, y, z ∈ G, (x*y)*z = x*(y*z), that is associative property is satisfied, (iii) ∀ x ∈ G, there exists an e ∈ G such that x*e = e * x = x, e is an identity of G.

Let us suppose that E(A) = set of all the endomorphisms over the strongly connected automata A. The formal specification of E(A) in terms of the schema *Endomorphisms* is described below. Two variables are taken, the first one, is a set of all endomorphisms which is of type of power set of *Endomorphism* and is denoted by *endomorphisms* and, the second one, is a binary operation denoted by *boperation*. It takes two endomorphisms as input and produces a new endomorphism. The components of *Endomorphisms* are defined in first part and invariants in the second part of it. Now we verify that E(A) is a monoid under binary operation defined above.

Endomorphisms

$endomorphisms{:}\ \mathbb{F}\ Endomorphism$

$boperation{:}\ Endomorphism \times Endomorphism \rightarrow Endomorphism$

$\forall e1, e2{:}\ Endomorphism \mid e1 \in endomorphisms \wedge e2 \in endomorphisms$

$\quad \bullet\ \exists e3{:}\ Endomorphism \mid e3 \in endomorphisms \bullet boperation\ (e1, e2) = e3$

$\forall e1, e2, e3{:}\ Endomorphism$

$\quad \mid e1 \in endomorphisms \wedge e2 \in endomorphisms \wedge e3 \in endomorphisms$

$\quad \bullet\ boperation\ ((boperation\ (e1, e2)), e3)$

$\qquad = boperation\ (e1, (boperation\ (e2, e3)))$

$\forall e{:}\ Endomorphism \mid e \in endomorphisms$

$\quad \bullet\ \exists\, ee{:}\ Endomorphism \mid ee \in endomorphisms$

$\qquad \bullet\ boperation\ (e, ee) = e \wedge boperation\ (ee, e) = e$

The algebraic structure (G, *) is called a group if 1) it is a monoid and 2) for any element x there exists y in G such that x*y = y*x = e. That is the inverse of each element of G exists. Let us suppose that G (A) = set of all automorphisms. For its formal specification, three variables are assumed. The first one is a set of all automorphism of type of power set of *Automorphism*. The second one is an identity element and the last one is binary operation denoted by *boperation*. It takes two automorphisms as input and produces a new automorphism as an output.

```
Automorphisms
 automorphisms: 𝔽 Automorphism
 ae: Automorphism
 boperationa: Automorphism × Automorphism → Automorphism
 ─────────────
 ∀ a1, a2: Automorphism | a1 ∈ automorphisms ∧ a2 ∈ automorphisms
   • ∃ a3: Automorphism | a3 ∈ automorphisms • boperationa (a1, a2) = a3
 ∀a1, a2, a3: Automorphism
   | a1 ∈ automorphisms ∧ a2 ∈ automorphisms ∧ a3 ∈ automorphisms
   • boperationa((boperationa (a1, a2)), a3)
      = boperationa (a1, (boperationa (a2 ,a3 )))
 ∀ a: Automorphism | a ∈ automorphisms
   • boperationa ( a, ae) = a ∧ boperationa (ae, a ) = a
 ∀a: Automorphism | a ∈ automorphisms
   • ∃ ai: Automorphism | ai ∈ automorphisms
      • boperationa (a, ai ) = ae ∧ boperationa (ai, a) = ae
```

Invariant: The last one property verifies existence of inverse of each element in set G (A). The first three properties are same as in case of monoid endomorphisms defined above.

Now we verify that the sets of endomorphisms and automorphisms over strongly connected algebraic automata are same. This verification is done in terms of a schema named as *Equivalence*. There are three inputs to this schema which are *Endomorphism*, *Endomorphisms* and *Automorphisms*. At first, it is described that set of all endomorphisms are bijective and then it is checked that the number of elements must be same in both of the morphisms.

```
Equivalence
 Endomorphism
 Endomorphisms
 Automorphisms
 ─────────────
 ∀ e: Endomorphism | e ∈ endomorphisms
   • ∀ q1, q2, q: S | q1 ∈ statesa ∧ q2 ∈ statesa ∧ q ∈ statesb
      • (q1, q) ∈ e . row ∧ (q2, q) ∈ e . row ⇒ q1 = q2
 endomorphisms ∈ dom # ∧ automorphisms ∈ dom #
 # endomorphisms = # automorphisms
```

Invariant: 1) There is a one to one correspondence between the set of *Endomorphisms* and *Automorphisms*. 2) Both the sets *Endomorphisms* and *Automorphisms* are of same type. 3) The total number of elements in both the morphisms are equal.

4. Model Analysis

As we know there does not exist any computer tool which may guarantee about the correctness of a computer model. That is why we believe that even the specification is written, in any of the formal languages, it may contain potential errors. These errors may range from syntax errors to hazardous inconsistencies. The Z/Eves is one of the most powerful tools which can be used for analyzing the specification written in Z. It is integrated with various facilities which provide rigorous analysis increasing confidence over the system to be developed. Further, it is to be mentioned that it has automated deduction capability and it supports the entire Z notation.

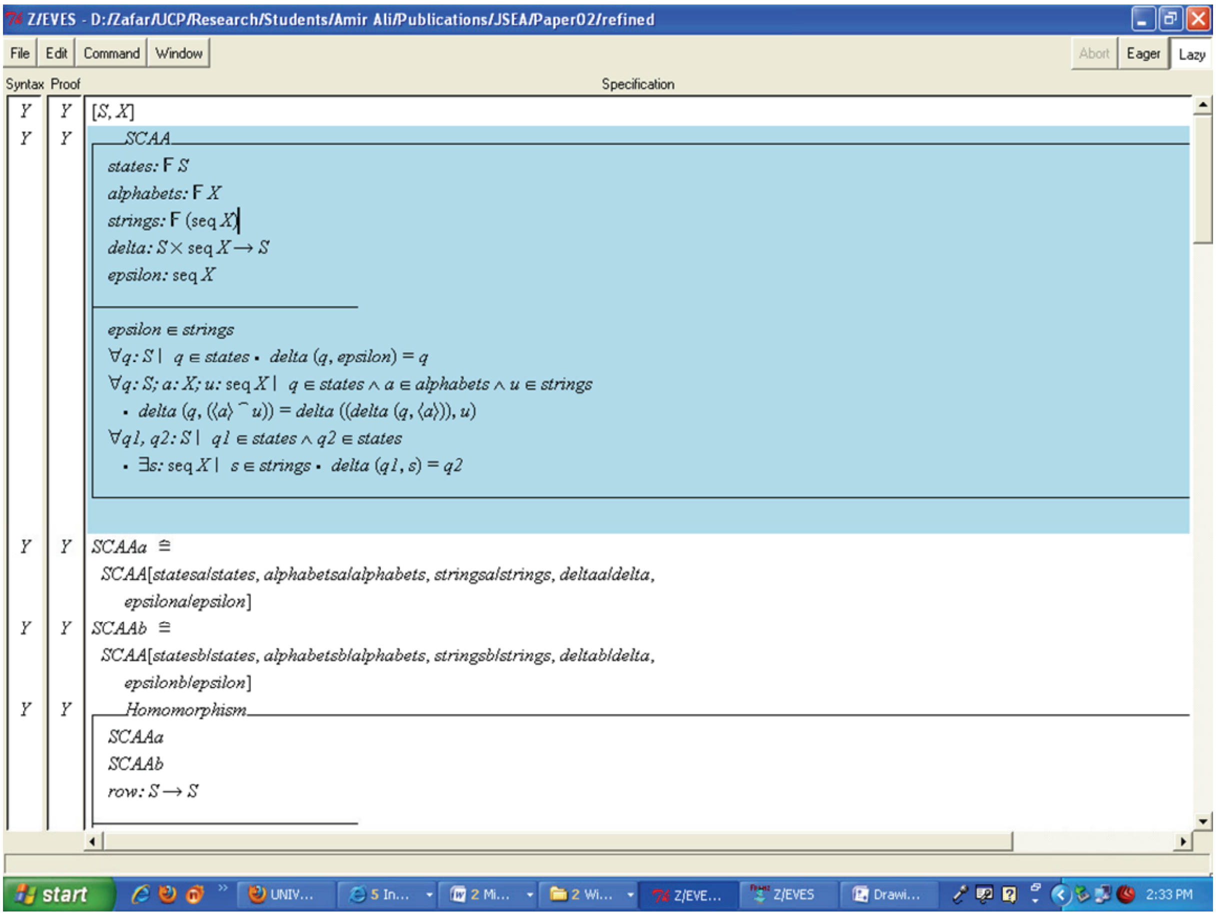
A snapshot of the analyzed specification is presented below in **Figure 1**. The first column on left most of the figure shows a status of the syntax checking and the second one presents the status of proof correctness. The symbol 'Y' shows that specification is correct syntactically and proof is correct while the symbol 'N' represents that errors are identified. In the schema, it is checked that the specification is correct in syntax and has a correct proof.

Formal specification of a system can be manipulated by using a set of rules based on mathematical formulae provided by such computer tools. These tools allow us to explore properties of the systems to prove that the specification of a system has certain meanings. The Z/Eves is one of the powerful tools used in our research to explore and analyze the specification which is a resultant of the integration of approaches proposed in this paper.

5. Conclusions

The main objective of this research was linking algebraic automata and formal methods. To achieve this objective, an integration of some fundamental concepts of strongly connected algebraic automata and Z notation is proposed. At first we described formal specification of strongly connected algebraic automaton. Then we described two important concepts of homomorphism and isomorphism over the same automata. Extended forms of homomorphism and isomorphism were formalized by making the reuse of the components. In next, monoid endomorphisms and group automorphisms were described. Finally, a formal proof of equivalence between monoid endomorphisms and group automorphisms over strongly connected algebraic automata was verified. It is to be mentioned that preliminary results of this research were presented in [28] and [29] where some inconsistencies and error were identified which are refined and corrected in this work.

Why and what kind of integration is required, were two basic questions in our mind before initiating this research. Automata are best suited for modeling behavior while formal methods are very useful describing properties and state space of a system. An exhaustive survey of existing work was made before initiating this research.



```
Syntax  Proof  Specification
Y       Y      [S, X]
Y       Y      SCAA
               states: F S
               alphabets: F X
               strings: F (seq X)
               delta: S × seq X → S
               epsilon: seq X
               ---
               epsilon ∈ strings
               ∀q: S | q ∈ states • delta (q, epsilon) = q
               ∀q: S; a: X; u: seq X | q ∈ states ∧ a ∈ alphabets ∧ u ∈ strings
                 • delta (q, (⟨a⟩ ⁀ u)) = delta ((delta (q, ⟨a⟩)), u)
               ∀q1, q2: S | q1 ∈ states ∧ q2 ∈ states
                 • ∃s: seq X | s ∈ strings • delta (q1, s) = q2
Y       Y      SCAAa ≙
                SCAA[statesa/states, alphabetsa/alphabets, stringsa/strings, deltaa/delta,
                   epsilona/epsilon]
Y       Y      SCAAb ≙
                SCAA[statesb/states, alphabetsb/alphabets, stringsb/strings, deltab/delta,
                   epsilonb/epsilon]
Y       Y      Homomorphism
               SCAAa
               SCAAb
               row: S → S
```

Figure 1. Snapshot of the tool used for model analysis

Some interesting work [41-43] was found but our work and approach are different because of abstract and conceptual level integration of Z and algebraic automata. We believe that this work will be useful in development of integrated tools increasing their modeling power. It is to be mentioned that most of the researchers have either taken some examples in defining integration of approaches or addressed only some aspects. There is also a lack of formalism which can be supported by computer tools. Our work is different from others because we have given a generic approach to link Z and algebraic automata which is verified by the Z/Eves toolset.

After integration, we have observed that a natural relationship exists between Z and algebraic automata. This work is also important because formalizing graph based notation is not easy as there has been little tradition of formalization in it due to concreteness of graphs [44]. Our work is useful for researchers interested in integration of approaches. We believe that this research is also useful because it is focused on general principles and concepts and this integration can be used for modeling systems after required reduction. The Z/Eves toolset made it possible to capture the conceptual errors which otherwise could not have been identified.

REFERENCES

[1] A. Hall, "Correctness by Construction: Integrating Formality into a Commercial Development Process," *Lecture Notes in Computer Science*, Springer, Vol. 2391, 2002, pp. 139-157.

[2] C. J. Burgess, "The Role of Formal Methods in Software Engineering Education and Industry," Technical Report: CS-EXT-1995-045, University of Bristol, Bristol, 1995.

[3] B. A. L. Gwandu and D. J. Creasey, "The Importance of Formal Specification in the Design of Hardware Systems," School of Electron & Electrical Engineering, Birmingham University, Birmingham, 1994.

[4] H. A. Gabbar, "Fundamentals of Formal Methods," Modern Formal Methods and Applications, Springer Netherlands, 2006.

[5] E. A. Boiten, *et al.*, "Integrated Formal Methods (IFM 04)," Springer, Canterbury, 2004.

[6] J. Davies and J. Gibbons, "Integrated Formal Methods (IFM07)," UK, Springer, 2007.

[7] J. Romijn, G. Smith and J. V. D. Pol, "Integrated Formal Methods (IFM 05)," Springer, Eindhoven, 2005.

[8] K. Araki, A. Galloway and K. Taguchi, "Integrated Formal Methods (IFM99)," Springer, York, 1999.

[9] R. Bussow and W. Grieskamp, "A Modular Framework for Integration of Heterogeneous Notations and Tools," *Integrated Formal Methods* (*IFM* 99), Springer-Verlag, York, 1999, pp. 211-230.

[10] W. Grieskamp, T. Santen and B. Stoddart, "Integrated Formal Methods," Springer, Dagstuhl Castle, 2000.

[11] T. B. Raymond, "Integrating Formal Methods by Unifying Abstractions," *Lecture Notes in Computer Science*, Springer, Vol. 2999, 2004, pp. 441-460.

[12] J. S. Dong, R. Duke and P. Hao, "Integrating Object-Z with Timed Automata," 12*th IEEE International Conference on Engineering Complex Computer Systems*, Auckland, 2005, pp. 488-497.

[13] S. Dong, *et al.*, "Timed Patterns: TCOZ to Timed Automata," 6*th International Conference on Formal Engineering Methods*, Seattle, 2004, pp. 483-498.

[14] R. L. Constable, *et al.*, "Formalizing Automata II: Decidable Properties," Cornell University, New York, 1997.

[15] R. L. Constable, *et al.*, "Constructively Formalizing Automata Theory," Foundations of Computing Series, MIT Press, Cambridge, 2000.

[16] X. He, "Pz Nets a Formal Method Integrating Petri Nets with Z," *Information & Software Technology*, Vol. 43, No. 1, 2001, pp. 1-18.

[17] M. Heiner and M. Heisel, "Modeling Safety Critical Systems with Z and Petri Nets," *International Conference on Computer Safety*, *Reliability and Security*, *Springer*, *Lecture Notes In Computer Science*, Toulouse, Vol. 1698, 1999, pp. 361-374.

[18] H. Leading and J. Souquieres, "Integration of UML and B Specification Techniques: Systematic Transformation from OCL Expressions into B," *Asia-Pacific Software Engineering Conference*, Gold Coast, 2002, pp. 495-504.

[19] H. Leading and J. Souquieres, "Integration of UML Views using B Notation," *Proceeding of Workshop on Integration and Transformation of UML Models*, Málaga, 2002.

[20] W. Wechler, "The Concept of Fuzziness in Automata and Language Theory," Akademic-Verlag, Berlin, 1978.

[21] N. M. John and S. M. Davender, "Fuzzy Automata and Languages: Theory and Applications," Chapman & Hall, CRC, USA, 2002.

[22] M. Ito, "Algebraic Theory of Automata and Languages," World Scientific Publishing, Singapore, 2004.

[23] D. K. Kaynar and N. Lynchn, "The Theory of Timed I/O Automata," Morgan & Claypool, 2006.

[24] C. Godsil and G. Royle, "Algebraic Graph Theory," Springer, New York, 2001.

[25] Z. Aleksic, "From Biology to Computation," IOS Press Publishers, IOS Press, Amsterdam, 1993.

[26] L. B. Kier, P. G. Seybold and C. K. Cheng, "Modeling Chemical Systems Using Cellular Automata," Springer, New York, 2005.

[27] T. Ellman, "Specification and Synthesis of Hybrid Automata for Physics-based Animation," *Automated Software Engineering*, Vol. 13, No. 3, 2006, pp. 395-418.

[28] N. A. Zafar, A. Hussain and A. Ali, "Refinement: Formal Proof of Equivalence in Endomorphisms and Automorphisms over Strongly Connected Automata," *Journal of Software Engineering and Applications*, Vol. 2, No. 2, 2009, pp. 77-85.

[29] N. A. Zafar, A. Hussain and A. Ali, "Formal Proof of

Equivalence in Endomorphisms and Automorphisms over Strongly Connected Automata," *Proceedings of the* 2008 *International Conference on Computer Science and Software Engineering*, Wuhan, Vol. 2, 2008, pp. 792-795.

[30] D. P. Tuan, "Computing with Words in Formal Methods," University of Canberra, Canberra, Australia, 2000.

[31] M. Conrad and D. Hötzer, "Selective Integration of Formal Methods in Development of Electronic Control Units," *Proceedings of the* 2*nd IEEE International Conference on Formal Engineering Methods*, Brisbane, 1998, pp.144-155.

[32] M. Brendan and J. S. Dong, "Blending Object-Z and Timed CSP: An Introduction to TCOZ," *Proceedings of the* 20*th International Conference on Software Engineering*, Kyoto, 1998, pp. 95-104.

[33] A. T. Nakagawa, *et al.*, "Cafe an Industrial-Strength Algebraic Formal Method," Elsevier Science & Technology, Amsterdam, 2000.

[34] J. V. Guttag and J. J. Horning, "The Algebraic Specification of Abstract Data Types," *Acta Informatica*, Springer Berlin, Vol. 10, No. 1, 2004, pp. 27-52.

[35] J. M. Spivey, "The Z Notation: A Reference Manual," International Series in Computer Science, Prentice Hall, 1989.

[36] J. M. Wing, "A Specifier: Introduction to Formal Methods," *IEEE Computer*, Vol. 23, No. 9, 1990, pp. 8-24.

[37] J. Woodcock and J. Davies, "Using Z: Specification, Refinement and Proof," Prentice Hall, International Series in Computer Science, 1996.

[38] J. A. Anderson, "Automata Theory with Modern Applications," Cambridge University Press, Cambridge, 2006.

[39] L. L. Claudio, *et al*., "Applications of Finite Automata Representing Large Vocabularies," *Software Practice & Experience*, Vol. 23, No. 1, 1993, pp. 15-30.

[40] Y. V. Moshe, "Nontraditional Applications of Automata Theory," *Proceedings of the International Conference on Theoretical Aspects of Computer Software*, *Lecture Notes in Computer Science*, Vol. 789, Sendai, 1994, pp. 575-597.

[41] D. I. A. Cohen, "Introduction to Computer Theory," 2nd Edition, John Wiley & Sons Inc., New York, 1996.

[42] J. P. Bowen, "Formal Specification and Documentation Using Z: A Case Study Approach," International Thomson Computer Press, London, 1996.

[43] S. A. Vilkomir and J. P. Bowen, "Formalization of Software Testing Criterion," South Bank University, London, 2001.

[44] C. T. Chou, "A Formal Theory of Undirected Graphs in Higher Order Logic," *Proceeding of* 7*th International Workshop on Higher Order Logic*, *Lecture Notes in Computer Science*, Valletta, Vol. 859, 1994, pp. 144-157.

J. Software Engineering & Applications, **2010, 3, 813-819**
doi:10.4236/jsea.2010.38094 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

Study of Integral Variable Structure Control Method for Stability of SI Engine Idling Speed

Yang Zhang[1]**, Nobuo Kurihara**[2]

[1]Science Program in Mechanical Systems, Hachinohe Institute of Technology, Hachinohe, Japan; [2]Graduate School of Engineering, Hachinohe Institute of Technology, Hachinohe, Japan.
Email: kurihara@hi-tech.ac.jp

Received June 29th 2010; revised July 15th 2010; accepted July 30th 2010.

ABSTRACT

The intake air control system of a gasoline engine is a typical nonlinear system, and included among the adverse factors that always induce poor idle-speed control stability are dead time and disturbances in the intake air control process. In this paper, to improve the responsiveness when idling with regard to disturbances, a mean-value engine model (MVEM) with dead time was constructed as the control object, and the two servo structures of sliding mode control (SMC) were studied for better idle control performance, especially in transient process of speed change. The simulation results confirmed that under the constraint condition of control input, the robustness of idle speed control that is being subjected to torque disturbances and noise disturbances can be greatly improved by use of the servo structure II.

Keywords: *SI Engine, Idle Speed Control, Variable Structure, Integral Sliding Mode Control, Simulation*

1. Introduction

Idle speed is the minimum operating speed of a combustion engine. Most of the driving time of urban traffic consists of periods when the engine is idling. Furthermore, if the idle speed can be reduced to 100 rpm (revolutions per minute) by improving the control method, fuel consumption will be reduced by 2 to 5%. Therefore, significant fuel economy and emissions improvements can be achieved by lowering the idle speed of an engine. In order to achieve a relatively lower idle speed while at the same time preventing the engine from stalling, it is necessary to maintain a stable idle speed in the presence of disturbances, both known disturbances (e.g. stationary steering and evaporation-gas purges) and unknown.

The automotive engine is a typical nonlinear, time-delay, time-varying parameter system. Recently there are many studies that apply control theories such as LQG, PID, adaptive control to idle-speed control [1,2]. In particular, because variable structure control is suitable for systems that are linear or nonlinear, continual or discrete, certain or uncertain, the application of sliding mode control is regarded as a solution to the problem of improving idle-speed control. Nevertheless, there are some studies is applying this theory [3-5]. However, idle-speed control should be investigated practically; in other words, the issues that typically constrain the application of SMC are how to improve the transient process of an idle control system and how to alleviate chattering, especially quasi-sliding mode is used against chattering recently [6,7]. Consequently, in this paper, idle speed control was studied based on a non-linear gasoline engine model, and an servo system of SMC was modified. As PID (proportional–integral–derivative) controllers are usually used for idle speed control in practical application, a PID control and sliding mode control (SMC) were employed to improve idle-speed control. Several disturbances such as torque and fuel disturbances were added into the engine model to test the responsiveness and stability of three control methods. Sudden start operation was also studied during fuel disturbances. Moreover, an integral type of control input instead of quasi-sliding mode is employed for chattering alleviation. The system was simulated by MATLAB/Simulink. According to the results of simulations, the robustness of idle-speed control was improved by using SMC.

2. Sliding Mode Control Design

SMC is a type of variable structure control in which the dynamics of a nonlinear system are altered by the application of a high-frequency switching control [8]. In other words, SMC uses practically infinite gain to force the trajectories of a dynamic system to slide along a restricted sliding mode subspace. This is an important, robust control approach that provides an adaptive approach

Copyright © 2010 SciRes.

to dealing with parametric, uncertain parametric, and uncertain disturbance systems. If a switching surface is appropriately designed with desirable characteristics, the system will exhibit desirable behavior when confined to this switching surface. In order to achieve the target value, idle-speed control can be studied as a servo system. Here, two servo systems using SMC were designed as follows.

2.1 Servo System I Using SMC

Consider the system

$$\dot{x} = Ax + Bu$$

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ b_n \end{bmatrix} u \qquad (1)$$

$$y = x_1$$

Where *x* is the state vector, *y* is the output, and *u* is the scalar. To achieve a servo system, a new state *z* is inserted and a new input *r* as a target input is also necessary. Accordingly, Equation (1) is extended as in Equation (2).

$$\dot{z} = r - x_1$$

$$\dot{x} = Ax + Bu + Er$$

$$\begin{bmatrix} \dot{z} \\ \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} z \\ x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ b_n \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} r \qquad (2)$$

The switching surface is defined as

$$\sigma(x) = Sx$$

$$\dot{\sigma}(x) = S\dot{x} \qquad (3)$$

When the system is in sliding mode state, a dynamic characteristic is exhibited. Firstly, taking into consideration non-linear factors and uncertain disturbances, the equivalent control for the SMC is usually employed.

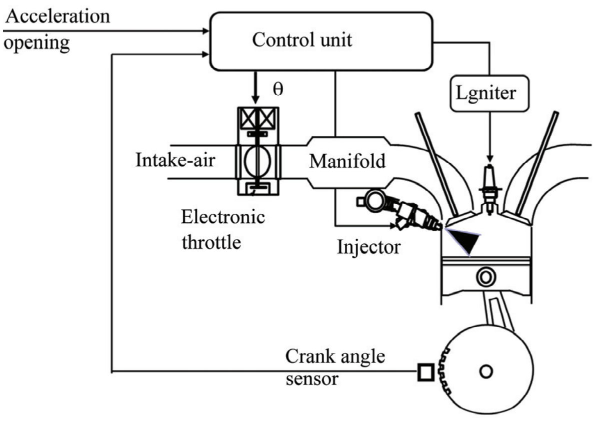


Figure 1. Engine control system

$$u = u_{eq} + u_{nl} \qquad (4)$$

As is well known, when the system is on the switching surface, it maintains $\sigma = \dot{\sigma} = 0$. Hence, by using $\dot{\sigma} = S\dot{x}$, substituting Equation (2) into Equation (3) gives the following:

$$\dot{\sigma} = S\dot{x} = SAx + SBu_{eq} + SEr = 0 \qquad (5)$$

$$u_{eq} = -(SB)^{-1}(SAx + SEr)$$

Here, $(SB)^{-1}$ can be used, but only when SB is the nonsingular matrix, *i.e.* $|SB| \neq 0$. So one part of the control input can be obtained, and another part u_{nl} is used only for uncertain factors such as disturbances and non-linear issues. In general, u_{nl} is designed as follows:

$$u_{nl} = K\,\mathrm{sgn}(\sigma) \qquad (6)$$

K is the switching gain, and. $\mathrm{sgn}(\sigma)$ is the signal function that can makes the system robust, but in the same way it leads to chattering, which is undesirable.

Next, the parametric vector *S* in the switching surface equation is required for deciding the existence of sliding mode. There are a few methods for the *S* design. Pole deployment (see Equation (7)) and the Riccati equation (see Equation (8)) are usually employed.

$$\dot{x} = \left(I - B\left(SB\right)^{-1} S\right)\left(Ax + Er\right) \qquad (7)$$

$$\begin{aligned} &PA_\tau + A_\tau^T P - PBB^T P + Q = 0 \\ &A_\tau = A + \varepsilon I \end{aligned} \qquad (8)$$

$\varepsilon \geq 0$ is assumed for the stability margin coefficient ε.

$$S = B^T P$$

Finally, according to Lyapunov's second theorem on stability, the reachability of sliding mode is assumed to be proved.

$$\dot{V} = \sigma\dot{\sigma} = -\kappa\,\mathrm{sgn}(\sigma)\sigma = -\kappa|\sigma| < 0 \qquad (9)$$

where $\kappa > 0$ But in the theory of SMC, the two main issues are the improvement of robustness with regard to uncertain disturbances and the alleviation of chattering, and these issues restrict the scope of application of SMC to an extent.

2.2 Servo System II Using Modified SMC

In this section, to solve the two aforementioned problems, the design method of SMC was modified. The structure is shown in **Figure 2.**

The reader will remember the expansion servo system described in Equation (2). Here another state variable was used, and the derivative $\ddot{z}$ of the target value and the model output was inserted, which was expected to improve the robustness of whichever engine control system

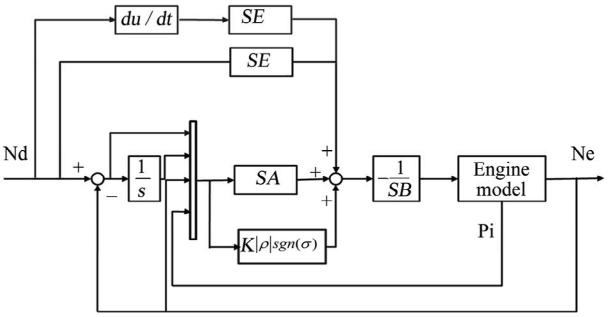


Figure 2. Block diagram

is being used in the operating mode of holding a lower idle speed or of a sudden start.

Equation (2) can be rewritten as follows:

$$\dot{x} = Ax + Bu + Er + E'\dot{r}$$

$$\begin{bmatrix} \ddot{z} \\ \dot{z} \\ \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & H_1 & H_2 & \cdots & H_n \\ 0 & 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & 0 & a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} \dot{z} \\ z \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ b_n \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} r + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \dot{r} \tag{10}$$

$$H_i = -a_{1i} \quad \ddot{z} = \dot{\theta}_r - \dot{x}_1 \quad \dot{z} = \theta_r - x_1 \quad z = \int(\theta_r - x_1)dt$$

The *S* in the hyper-plane was designed by using the Riccati equation if and only if it has a solution. The equivalent control input was solved by using Equation (8). The alleviation of chattering was expected, but while the smoothing function seems to alleviate or remove chattering, it will in all probability weaken the robustness of the system. As a result, the original signum function was kept, and the authors hoped that gain *K* would gradually get smaller so that chattering would be alleviated with a lessening of switch gain *K*. For this purpose, gain *K* was attained with the integration of the switching function σ [9]. Thus, gain *K* was higher when the system was in reaching mode, and fell when the system was in sliding mode. This not only alleviated chattering, but also maintained the robustness of the system.

$$u_{eq} = -(SB)^{-1}\left(SAx + SE\theta_r + SE\dot{\theta}_r\right) \tag{11}$$

$$u_{nl} = -K|\rho|\operatorname{sgn}(\sigma)(SB)^{-1} \qquad K > 0 \tag{12}$$

$$\rho = \left|\int_0^t (\sigma + \kappa\rho)\,dt\right| \qquad \kappa < 0$$

When the integer $\rho > 0$, $\kappa\rho > 0$ and when $\rho < 0$, $\kappa\rho > 0$, it was assumed that the larger switching gain at reaching mode will be alleviated.

Substituting (11), (12) into (9) gives

$$\dot{V} = \sigma\dot{\sigma} = -K|\rho|\operatorname{sgn}(\sigma)\sigma = -K|\rho||\sigma| < 0 \tag{13}$$

The existence of the sliding mode and reachability is proved based on Lyapunov's second theorem on stability. The modified servo control system of SMC was verified by its application to idle speed control. The effect of chattering alleviation is shown in **Figure 3**. The cSMC denotes conventional SMC and the pSMC denotes modified SMC.

3. Application to Idling Speed Control System

The control system of a spark-ignition engine (SI engine), which was chosen as the target of this study on idle-speed control is shown in **Figure 1**. The intake air flow is adjusted to reach a certain engine speed by adjusting the angle and position of the electronic throttle. The crank angle sensor detects the engine speed; and the angle and position of the electronic throttle depend on the controller in the engine control unit. Then the throttle adjusts the amount of air supply to the cylinder. After that fuel proportionate to the air flow is injected into the cylinder. Thus, the torque produced by the combustion of the fuel maintains a constant engine speed.

The idle-speed control system for the regulation of intake air flow is shown in **Figure 4**. The basic input is the equivalent electronic throttle opening angle θ, which is restricted within a finite opening scale and the controlled output is engine speed Ne [rpm]. Because of the inversely-proportional relationship between the dead time L and the engine speed Ne [rpm], the dead time is derived from the feedback of the output of the system. In view of the disturbances in an actual engine, three disturbances are loaded into the engine model. Two disturbances, D1 and D2, are added as transitional disturbances before and after the dead time, taking evaporative purge and stationary steering into consideration. The measured engine speed D3 is also added as a steady disturbance to simulate the combustion fluctuation of an actual engine.

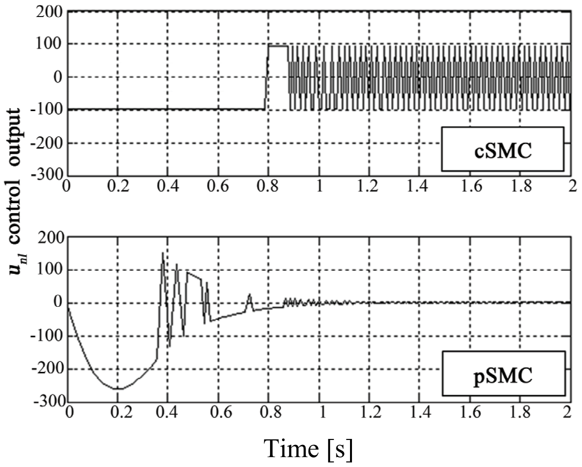


Figure 3. Non-linear control output by step response

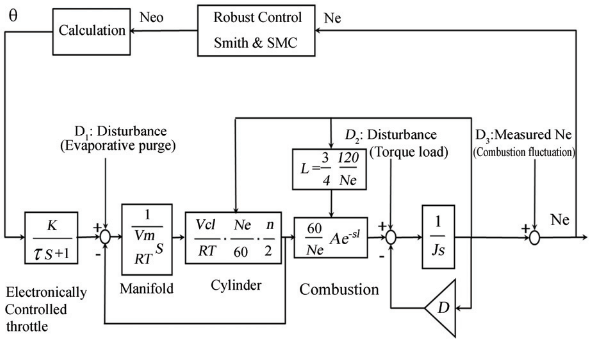


Figure 4. Model for idle-speed control

a) Intake air flow is in direct proportion to throttle angle. Since the throttle is driven by the motor, it causes a delay, the time constant τ . K is the air flow conversion factor; the input is θ and output is intake air flow Q_{in} $\left[kg/s\right]$, so that the state equation is

$$\dot{Q}_{in} = -\frac{1}{\tau}Q_{in} + \frac{K}{\tau}\theta + \frac{1}{\tau}D_1 \tag{14}$$

D_1 is the air leak and fuel purge disturbance.

b) In the manifold and cylinder, the input is Q_{in} and the output is from manifold to cylinder intake air flow Q_o ,

$$\dot{P}_m = \frac{RT}{V_m}\left(Q_{in} - Q_o\right) \tag{15}$$

$$Q_o = \frac{V_{cl}nN_e}{120RT}P_m \tag{16}$$

Equation (16) is substituted into Equation (15), therefore:

$$\dot{P}_m = \frac{RT}{V_m}(Q_{in} - \frac{V_{cl}nN_e}{120RT}P_m) \tag{17}$$

V_m is the manifold volume; R is the air constant; T is the temperature; and the cylinder pressure is assumed to be equal to the manifold pressure.

c) In combustion, the engine torque τ_e is in direct proportion to the cylinder intake air flow Q_o , hence

$$\tau_e = \frac{60}{N_e}Q_oZe^{-sL} \tag{18}$$

Z is the comparison coefficient of unit conversion, and the dead time is obtained by 1st order Padé approximant[10]. The state variable D_e is:

$$\dot{D}_e = \frac{2}{L}\left(\frac{60Z}{N_{eo}}Q_o - D_e\right) \tag{19}$$

The dead time L is a function of engine speed, which gives:

$$L = \frac{3}{4}\cdot\frac{2\times 60}{N_e} \tag{20}$$

Obtaining torque requires intake, compression and combustion; hence the dead time is 3 strokes long. Equation (21) is obtained from Equations (16) and (19).

$$\tau_e = 2D_e - \frac{ZnV_{cl}}{2JRT}P_m \tag{21}$$

d) The engine inertia moment is J, the friction coefficient is D_f, and the disturbance is D_2. Then:

$$\dot{N}_e = \frac{1}{J}\left(\tau_e - N_eD_f + D_2\right) \tag{22}$$

Equation (21) is substituted into Equation (22), therefore:

$$\dot{N}_e = \frac{2}{J}D_e - \frac{ZnV_{cl}}{2JRT}P_m - \frac{D_f}{J}N_e + \frac{1}{J}D_2 \tag{23}$$

The block diagram shown in **Figure 2** is arrived at based on the above equations. In addition, the equivalent state space is:

$$\begin{aligned} \dot{x} &= Ax + Bu + Fd \\ y &= Cx \end{aligned} \tag{24}$$

$$x = \left[N_e \quad P_m \quad D_e \quad Q_{in}\right]T$$

$$A = \begin{bmatrix} 0 & -\frac{ZnV_{cl}}{RTL} & \frac{2}{J} & -\frac{D_f}{J} \\ \frac{RT}{V_m} & -\frac{N_{eo}nV_{cl}}{120V_m} & 0 & 0 \\ 0 & \frac{ZnV_{cl}}{RTL} & 0 & 0 \\ -\frac{1}{\tau} & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ K/\tau \end{bmatrix}, F = \begin{bmatrix} 0 & 1/J \\ 0 & 0 \\ 0 & 0 \\ 1/\tau & 0 \end{bmatrix}, C = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T$$

4. Simulation

In order to confirm the robustness of the idle-speed control when disturbances are present, the system was simulated using MATLAB/Simulink. The parameters of the engine model in **Figure 2** are set according to **Table 1**.

Firstly, the initial idle speed of the engine was 700 rpm, but, taking into consideration the existence of disturbances such as fuel purges and the use of power windows, a unit step input was added at 4 seconds for disturbance (D1) that results from an evaporation-gas purge in the fuel module and another unit step input was added at 12 seconds for disturbance (D2) that results from stationary steering or other types of torque variations in the crankshaft module. The simulation results are shown in **Figure 5**.

Here cSMC represents a conventional SMC controller,

Table 1. Engine coefficient for simulation

τ	Time Constant	0.1[s]
K	Air flow conversion factor	0.01
R	Gas constant	287.2[J/Kg・K]
T	Intake air tempurature	296[K]
V_m	Manifold volume	0.00317[m^3]
V_{cl}	Cylinder volume	0.00045[m^3]
n	Number of cylinders	4
Z	Unit conversion coefficent	60000
J	Engine inertia moment	0.15[Kg・m^2]
Df	Friction coefficient	0.412[Nm・s]

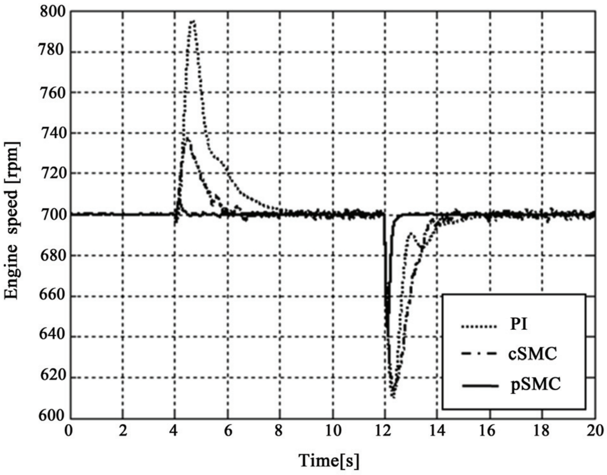


Figure 5. Responses for two disturbances

and pSMC represents the modified SMC controller. When using a PI controller, the system takes 4 seconds and 2.2 seconds, respectively, to deal with two disturbances. When using a cSMC, it takes 2.3 seconds and 2.2 seconds. When using the pSMC, it takes 0.2 seconds and 0.3 seconds. Also, it appears that pSMC suppressed the two disturbances greatly, as the engine speed merely deviates to 706 rpm and 645 rpm. On the other hand, PI and cSMC have a long response time and provide a weak compensation effect.

Secondly, considering the stability of the feedback loop in conditions of noise (D3), engine speed fluctuations measured from an actual engine as background noise were used, and were added to the engine speed output of the engine model; meanwhile two step disturbances also were applied as in the simulation above. The three control methods were simulated under the above-mentioned conditions. The simulation result is shown in **Figure 6**. The adjustment time when using PI was longer, the deviation was bigger than in the case where the other two methods were used, although the noise disturbance is loaded in the feedback loop. When pSMC was used the system still maintained its robustness.

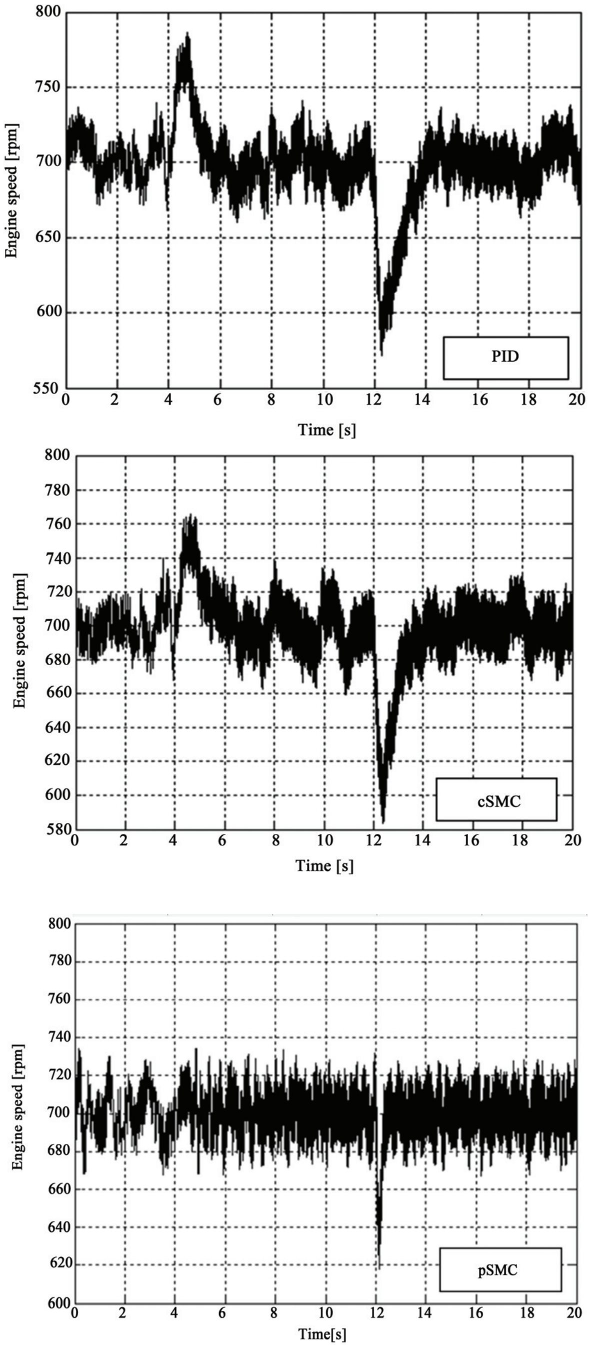


Figure 6. Simulation results with loading engine-speed signal measured in engine experiment

Third, the responsiveness and tracing ability were considered when the engine makes a sudden start from a lower idle speed to a higher speed such as 2000 rpm. As far as it's known, there is usually little fuel loss during the conditions of a sudden start, due to some fuel drops adhering to the manifold, a phenomenon which sometimes leads to an undesirable loss in speed. Therefore, a sudden start occurring at 10 seconds was assumed. The simulation results are shown in **Figure 7**. When using PI

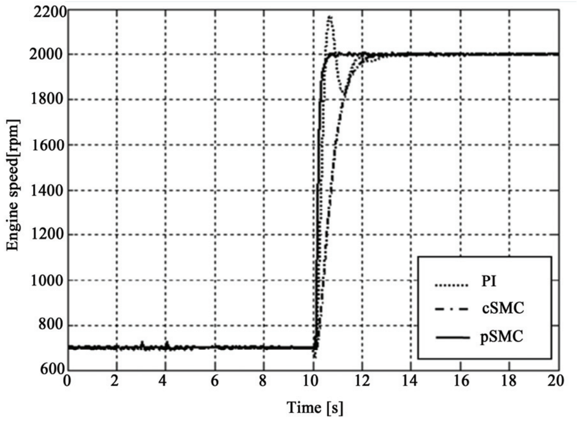


Figure 7. Simulation result of sudden start from 700 rpm

an overshoot occurred, when using cSMC was no overshoot but the adjustment time was not short. It appears that, the transition of the sudden start when using pSMC is faster than when using PI or cSMC, the time taken by pSMC for adjustment being approximately 0.3 seconds. So a steady and fast engine start can be achieved by using pSMC.

On the other hand, all real world control systems must deal with constraints. The constraints acting on a process can originate from amplitude limits on control signal, slew rate limits of the actuators and limits on output signals. As a result of constraints, the actual plant input will be different from the output of the controller. When this happens, the controller output does not drive the plant asexpected. Input constraints always appear in the in the form of rate constraints: valves and other actuators with limited slew rates. These constraints, especially of the saturation type, are also often active when a process is running at its most profitable condition. They may limit production rate.

In this case, Although the chattering are supposed to be alleviated by the way in Section 2, we also need to set the relative appropriate switching gain to make sure of control input θ operation within normal range. Under the target input of a step response which was set from 700 rpm to 2000 rpm, the system was simulated. **Figure 8** shows the results by using modified SMC. When $K = 50$ and $\kappa = -50$,There is no sign that the sum control input of u_{eq} and u_{nl} exceeded the normal opening angle(here the ratio of opening angle to full opening angle is regard as control input), and the chattering was also alleviated greatly along with the gain getting smaller.

Based on the aforementioned simulation results, it can be seen that a control system with a modified sliding mode controller is more effective with respect to either of the two disturbances, as well as regarding noises generated by the actual engine, so proving the robustness of pSMC. And the tracing ability of idle speed also appears to be improved in the work condition under the sudden start. In addition, for industrial application, chattering is assumed to be alleviated by the use of a class switching function with an integral item.

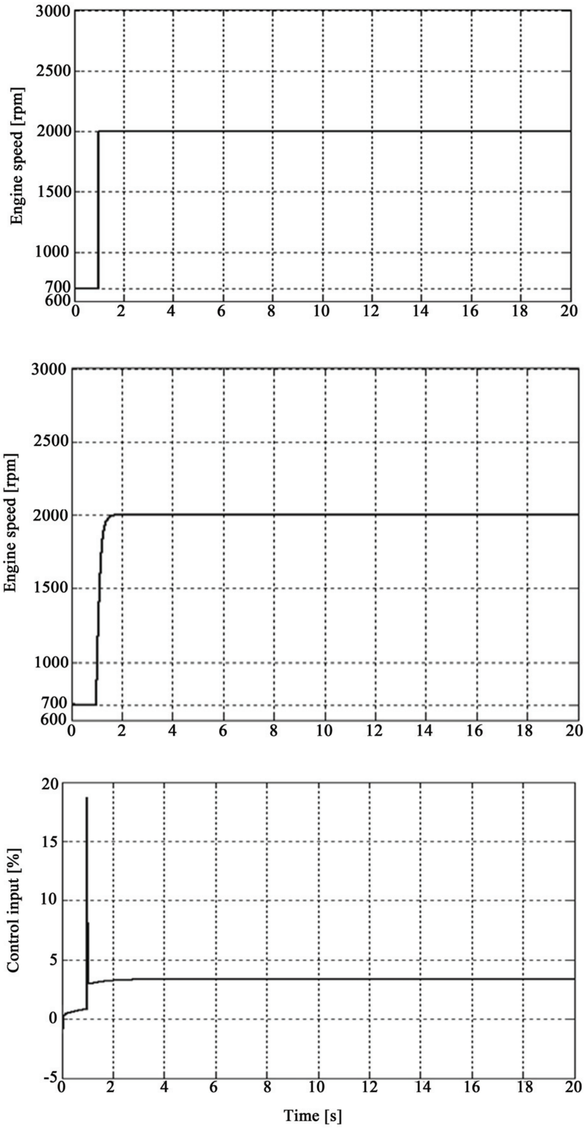


Figure 8. Simulation result under control input constraint

5. Conclusions

In this paper, two servo systems of SMC were studied, and were applied to idle speed control in order to improve the stability of idling engine speed and raise fuel economy. A mean value engine model that includes dead time was employed as the control object. The throttle opening and the engine speed were regarded as the input and output of idle system respectively. The electronic throttle angle and engine speed was taken as the input and the output of system. Using the nominal state-space model, three controllers of PI controller, conventional

SMC and modified SMC were constructed. Taking into consideration actual engines, three disturbances D1, D2 and D3 were added in the author's simulation, where, D1 and D2 were transition disturbances caused by sudden operations, and D3 was a measured value of engine speed fluctuation. The simulation results show excellent responsiveness and stability when modified SMC was used because it can shorten the transient process. Furthermore, under the permission range of constraint condition of control input, chattering which is regarded as an obstacle to the application of SMC, was alleviated by servo structure II.

REFERENCES

[1] F.-C. Hsieh and B.-C. Chen, "Adaptive Idle-speed Control for Spark-Ignition Engines," *SAE Paper*, 2007-01-1197.

[2] J. S. FU and N. Kurihara, "Intake Air Control of SI Engine Using Dead-Time Compensation," *SAE Paper*, 2003-01-3267.

[3] B. Kwak and Y. J. Park, "Robust Vehicle Stability Controller based on Multiple Sliding Mode Control," *SAE Paper*, 2001-01-1060.

[4] M. Kajitani and K. Nonami, "High Performance Idle-speed Control Applying the Sliding Mode Control with H Robust Hyperplane," *SAE Paper*, 2001-01-0263.

[5] Y. Zhang, T. Koorikawa and N. Kurihara, "Evaluation of Sliding Mode Idle-speed Control for SI Engines," *Japanese Society of Automotive Engineers*, Vol. 40, No. 4, 2009, pp. 997-1002.

[6] X. D. Li, T. W. S. Chow and J. K. L. Ho, "Quasi-Sliding Mode Based Repetitive Control for Nonlinear Continuous-Time Systems with Rejection of Periodic Disturbances," *Automatic Journal of IFAC*, Vol. 45, No. 1, 2009, pp. 103-108.

[7] R. Ramos, D. Biel, E. Fossas and F. Guinjoan, "A Fixed-Frequency Quasi-Sliding Control Algorithm: Application to Power Inverters Design by means of FPGA Implementation," *IEEE Transactions on Power Electronics*, Vol. 18, No. 1, 2003, pp. 344-355.

[8] V. I. Utkin, "Variable Structure Systems with Sliding Modes," *IEEE Transactions on Automatic Control*, Vol. 22, No. 1, 1977, pp. 212-222.

[9] J. I. Tahara, K. Tsuboi, T. Sawano and Y. Nagata, "An Adaptive VSS Control Method with Integral Type Switching Gain," *Proceedings of the IASTED International Conference Robotics and Applicatons*, Tampa, 2001, pp. 106-111.

[10] O. J. M. Smith, "A Controller to Overcome Dead Time," *Indian Scientist Association in Japan*, Vol. 6, No. 2, 1959, pp. 28-33.

Copyright © 2010 SciRes.

J. Software Engineering & Applications, **2010, 3, 820-826**
doi:10.4236/jsea.2010.38095 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

Identifying and Modeling Non-Functional Concerns Relationships

Hakim Bendjenna[1], Pierre-Jean Charrel[1], Nacer Eddine Zarour[2]

[1]University of Toulouse and Institut de Recherche en Informatique de Toulouse, Toulouse, France; [2]Lire Laboratory, University Mentouri of Constantine, Constantine, Algeria.
Email: bendjenna@irit.fr, charrel@univ-tlse2.fr, nasro-zarour@umc.edu.dz

Received June 8th 2010; revised July 6th 2010; accepted July 10th 2010.

ABSTRACT

Requirements elicitation step is of paramount importance in the requirements engineering process. In the distributed environment of so-called inter-company cooperative information system, this step is a thorny issue. To elicit requirements for an inter-company cooperative information system, we early proposed a methodology called MAMIE (from MAcro to MIcro level requirements Elicitation) with an accompanied tool. In MAMIE methodology, requirements are the result of composing functional and non-functional concerns. Before non-functional concerns composition, it's primary to identify relationships between them. According to the most existing approaches, a non-functional concern may have a negative, positive or null contribution on the other non-functional concerns. In this paper, we argue that using only these three contributions types is not sufficient to express relationships which may exist between non-functional concerns. Thus, we propose a process which aims to identify non-functional concerns' relationships and model them using a fuzzy cognitive map. The resulting model is composed of non-functional concerns, relationships between them and the weight of these relationships expressed with linguistics fuzzy values. Using fuzzy cognitive maps to model non-functional concerns relationships allows moving from the conventional modelling toward developing a computer based model. An example from the textile industry is used to illustrate the applicability of our process.

Keywords: *Non-Functional Concerns, Non-Functional Requirements, Soft-Goals, Non-Functional Concerns Relationships, Fuzzy Cognitive Maps, Delphi Method*

1. Introduction

Eliciting requirements for an Inter-company Cooperative Information System (ICIS) is challenged by different factors; most of which are related to communication between stakeholders [1]. The geographic and temporal distance between stakeholders increases the difficulty to elicit requirements [2]. In our previous research [3-5] we proposed a methodology named MAMIE (from MAcro to MIcro level requirements Elicitation) to elicit requirements for an ICIS. One of the main contributions of MAMIE methodology is coupling together goals, scenarios and viewpoints. Thus, requirements which do emerge from MAMIE methodology exhibit a high degree of completeness and consistency and, crucially, will embody the principal companies' goals. In MAMIE, goals, scenarios and viewpoints interact in a logical complementary way, where each concept plays a specific role: goals describe the macro-level of requirements; scenarios are used to describe the medium-level of requirements; whereas viewpoints describe the micro-level of requirements. A goal may be: a Functional Concern (FC) or a Non-Functional Concern (NFC). We use here the term of concern instead of requirement in order to refer to a high level of abstraction. A FC represents a primary business goal (e.g. build a car) while a NFC (e.g. security, performance, compatibility) is a global property and usually refers to a quality of FCs. Requirements are issued from the composition of FCs and NFCs. Effectively, a system's utility is determined by both its functionality and its non-functionality characteristics [6], known as NFCs, non-functional requirements, or also soft-goals. Nonetheless, there has been a lop-sided emphasis in the functionality of the system, even though the functionality is not useful or usable without the necessary non-functional characteristics [6].

Before composing NFCs, the analyst must identify and specify relationships between them. The most existing approaches differentiate three types of relationships: negative (-), positive (+) or null (no contribution) [7-9]. The opportunity to compose NFCs depends on the type

Copyright © 2010 SciRes.

of these relationships. For example, when two NFCs which contribute negatively to each other are composed, then one NFC will influence negatively the correct working of the other. Based on relationships types among NFCs, the analyst may decide if the composition is or not.

The interactions between NFCs need to prioritize NFCs and then to define the degree of contribution of a NFC to another NFC: two or more NFCs may not contribute with the same degree to a specific NFC. For example, Functionality and Aesthetics are two NFCs that contribute positively to the NFC Service quality. However, Functionality ensures Service quality more accurately than Aesthetics. Thus, defining explicit knowledge about NFCs interactions may help to improve the composition process and then the requirements elicitation process. This paper uses fuzzy cognitive maps [10,11] to model interactions between NFCs. This formalism is known to be used in ill-structured problems solving. It allows observing the significance of each factor and its influence on other factors and the final decision [12]. In our context, a fuzzy cognitive map is composed of (1) NFCs, (2) the type of relationships between them, and (3) the weights of these relationships which indicate their importance.

The rest of this paper is structured as follows. In Section 2 we discuss related works. In Section 3 we give the main features of fuzzy cognitive maps. In Section 4 we present the three steps of a process to build a FCM which identify and model NFCs and their relationships using a fuzzy cognitive map. Section 5 shows the application of this process to a case study. Finally, concluding remarks are provided as well as future research issues.

2. Related Works

Most of the early work on NFCs focused on measuring how much a software system is in accordance with the set of NFCs that it should satisfy, using some form of quantitative analysis [13-16], offering predefined metrics to assess the degree to which a given software object meets a particular NFC.

Recently, a number of works proposed to use approaches which explicitly deal with NFCs before metrics are applicable [11,17-19]. These works propose the use of techniques to justify design decisions on the inclusion or exclusion of requirements which will impact on the software design. Unlike the metrics approaches, these latter approaches are concerned about making NFCs a relevant and important part of the software development process.

Boehm and In (1996) propose a knowledge base where NFCs are prioritized through stakeholders' perspectives, dealing with NFCs at a high level of abstraction. Kirner (1996) describe properties for six NFCs from the real-time system domain: performance, reliability, safety, security, maintainability and usability. This work provides heuristics on how to apply the identified properties to meet the NFCs and later measure these NFCs.

A significant advance was introduced when NFCs where treated as competing goals that are extensively refined and traded off among each other in an attempt to arrive at acceptable solutions. The non-functional requirements Framework is one of the few works to deal with non-functional requirements starting from the early stages of software development through a broader perspective. The non-functional requirements Framework [6] views non-functional requirements as goals that might conflict among each other and must be represented as soft-goals to be satisficed. The soft-goal concept was introduced to cope with the abstract and informal nature of non-functional requirements. 4 types of contribution have been proposed to describe relationships among non-functional requirements: make (++), help (+), hurt (-), break (--). However, as important as getting a well-formed and as-complete-as-possible set of contribution types, we need to understand how to identify and model these relationships. None of the above work tackles this problem.

3. Fuzzy Cognitive Maps: A Brief Description

In this section, we describe some features of fuzzy cognitive maps. Cognitive Maps (CMs) were proposed by R. Axelrod (1976) in order to solve ill-structured problems. A CM is a signed digraph designed to capture the causal assertions of a person with respect to a certain domain and then uses them in order to analyze the effects of alternative, e.g. policies, business decisions, etc. upon certain goals. A cognitive map is based on two notions:

- The concept which is represented by a variable.
- The causal belief which defines a relationship among variables. These relationships link variables to each other and they can be either positive or negative.

Variables that cause a change are called cause variables while those that undergo the effect of the change in the cause variable are called effect variables. If the relationship is positive, an increase or decrease in a cause variable causes the effect variable(s) to change in the same direction. If the relationship is negative, then the change which the effect variable undergoes is in the opposite direction. **Figure 1** is a graphical representation of a cognitive map, where variables (X, Y, Z, F, W) are represented as nodes, and causal relationships as directed arrows between variables, thus constructing a signed digraph.

Another way of representing a cognitive map is possible through an adjacency matrix where one can clearly observe the sign of the relationship, while keeping in mind that in case of there being an absence of relationship between these two factors, the corresponding entry

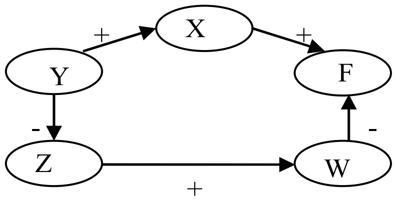


Figure 1. An example of Cognitive Map [10]

will be empty [10]. **Figure 2** shows this matrix (E) that represents an example of a CM.

CMs were developed in simulation, organizational strategies modelling, support for strategic problem formulation and decision analysis, knowledge bases construction, managerial problems diagnosis, failure modes effects analysis, modelling of social and psychological processes, modelling virtual worlds and analysis of their behaviour, requirements analysis and systems requirements specification [12,20,21].

B. Kosko (1986) introduces Fuzzy Cognitive Maps (FCMs) *i.e.* weighted cognitive maps with fuzzy weights. It is argued that FCM eliminate the indeterminacy problem of the total effect. Since its development, fuzzy set theory has been advanced and applied in many areas such as experts systems and decision making, control engineering, pattern recognition, etc. [21]. It is argued that people use fuzzy data, vague rules, etc. and fuzzy sets as a mathematical way to represent vagueness [22].

$$E = \begin{array}{c|c|c|c|c|} & W & X & Y & Z \\ \hline W & & + & & \\ \hline X & & & & + \\ \hline Y & & - & & \\ \hline Z & & & & \\ \hline \end{array}$$

Figure 2. Adjacency matrix associated with a cognitive map [10]

Different approaches were proposed for the specification of the fuzzy weights in a FCM [23]. One suggestion is to ask the experts to assign a real number from the interval (0, 1) for each relationship and then calculate the average of these numbers. However, it is difficult for the experts to assign a real number in order to express their beliefs with regard to the strength of relationships. This is the reason why partially ordered linguistic variables such as weak < moderate < strong, etc. are preferred instead of real number.

4. A Process to Model NFCs Interaction Using Fuzzy Cognitive Maps

To model interactions between NFCs using a FCM, we present in this section a process composed of three steps: 1) Identify NFCs, 2) Identify relationships between NFCs, 3) Specify the fuzzy weights and so provide the FCM. A simple view of this process is depicted in **Figure 3.**

4.1 Step 1: Identify NFCs

The first step consists on identifying all the NFCs of a system. NFCs are global properties (assumptions, constraints, etc.) that can influence part or the whole system [6]. NFCs can be identified using several approaches such as those used to identify goals. Here we have taken the main ideas from Chung (2000) who proposed a catalogue of NFCs.

The analyst with the other stakeholders identifies which subset of these NFCs is applicable to the system. For each entry in the catalogue, we must decide whether it would be useful in our system or not.

For example, if the owner of a vehicle has to indicate, during registration, his/her bank account details, so that automatic transfers can be performed automatically, then

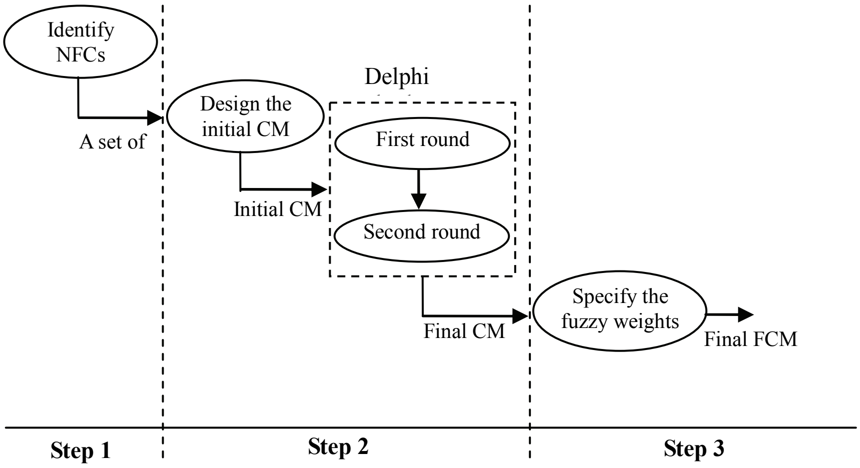


Figure 3. Steps needed to model NFCs interactions using a FCM

Security is an issue that the system needs to address. Other NFCs may be identified in this example from catalogue are Legal Issues and Correctness. Certain NFCs appear time and again during system development and most of them are likely to be domain-specific [6].

4.2 Step 2: Identify Relationships between NFCs

The objective of this step is to understand and define the relations between the identified NFCs. When the NFCs and their relations are clearly recognized, it is possible to establish the final CM.

In order to determine relations between NFCs, the analyst asks advices to a panel of experts. The analyst chooses experts according to their experience and background in the field. The number of experts may depend on the characteristics of the field. One of the most recent studies suggests a range of 10 to 18 to be an ideal number for each panel of experts [24]

The analyst starts by designing an initial/draft CM, either alone or with help of an expert. In order to reach consensus between experts, several techniques have been proposed [25]. Here, due to its simplicity we adopt Delphi methodology. Delphi is used to structure the communication process within a group of experts in order to reach a consensus regarding a complex problem [26]. Delphi method is organised in two consulting rounds. After the first round the experts receive feedback reports. They have the opportunity of improving their own opinion based on this feedback.

At the beginning of the second round experts are provided with information about deviations from the first round. Often a larger consensus than the first round can be observed. The goal is to obtain consensus and get all experts to go toward the average [26].

To apply Delphi in our context, we define the following steps:

1) *The first round.*

- The analyst gives the initial CM to all chosen experts.
- The analyst asks them to check and comments the initial CM.
- Experts may change the type of relationships between NFCs in the initial CM (e.g. from negative contribution to positive contribution), they may also delete existing relationships; add new relationships or new NFCs.
- The analyst revises the draft CM on the basis of the answers of experts and builds a new CM. For each relationship in the new CM, the analyst determines the number of experts who issues the same evaluation (positive, negative or null). These results are collected in a table.

2) *The second round.*

- The new CM with the table resulting from the first round are sent to experts for revision. Instructions for giving advices are the same as in the first round.
- The analyst collects the results of the second round. A consensus is reached either because the experts are influenced by the others in the second round, or because they have realized that their previous opinion was erroneous.
- The analyst builds the final version of the CM, based on the answers of the experts.

The purpose of the next step is to extend this CM to a FCM.

4.3 Step 3: Specify the Fuzzy Weights and so Provide the FCM

Up to the previous step, the cognitive map has been produced. In this cognitive map, no certain strengths for causal relations between NFCs are considered. The objective of this step is to provide such strength for the relations using the fuzzy set theory. To do so, each mutual relationship includes one linguistic fuzzy weight which determines the accuracy of the expert choice. Following W. R. Zhang (1989) (1992) [27,28] we use the linguistic fuzzy weights instead of real values for weights, since they make it easier for the planners to express their beliefs.

These linguistic fuzzy weights bring about a more thorough and understandable vision for the decision makers by mapping the ideas of the experts into a logic which could be processed [29].

In order to identify the linguistic fuzzy weights the analyst identifies the response to the following question for each relationship in the CM:

How strong, the causal relationship between NFCs in the final CM is?

The response to this question is an element from the following set: {Weak, Moderate, Strong, and Undefined}, where Weak < Moderate < Strong.

In order to express their beliefs in the strength of a certain causal relationship as being strong, moderate, or weak, the experts assign fuzzy weights to all of the relationships in the cognitive map. The corresponding fuzzy weights range is between 0 and 1. The goal of this step is not to reach a consensus between experts, thus the analyst is not limited by any number of experts. To give weights to the CM, the analyst chooses a number of experts from the panel of experts who agree with the CM obtained in the previous step.

The evaluations of the weight of a specific relationship may be different over experts. In order to aggregate all these evaluations, we propose to compute the average of these weights. The result may classify the strength of relationships as weak, moderate or strong.

5. A Case Study

A group of specialized companies in a regional environment wish to cooperate together in order to produce a range of textile products. These groups of 5 participating companies with their locations are the following:

- The fiber producer company (Morocco).
- The knitting mills and weaver company (France).
- The dyer and finisher company (France).
- The designer company (Spain).
- And the manufacturer company (Italy).

The analyst who is in France applies MAMIE methodology to elicit requirements for the future system. In this paper, we present the result of applying the process presented to identify and model relationships between NFCs.

5.1 Step 1: Identify NFCs

The analyst with the other stakeholders has identified the following set of NFCs: Product quality (NFC_1), Functionality (NFC_2), Product cost (NFC_3) and Competitiveness (NFC_4).

5.2 Step 2: Identify Relationships between NFCs

The result of this step is a final CM. To obtain it, the analyst chooses 10 experts. S/he starts by building an initial CM with one expert. **Figure 4** depicts the result of this sub-step.

In order to obtain the final CM, the analyst applies Delphi method. **Table 1** summarises the advices of the experts on the initial CM: for each relationship between NFCs, the table collects the number of experts who agree with its type (positive, negative or null).

We may remark that:

- The experts reach a consensus for almost all the relationships, *i.e.* in most cases they have responded in the same way.
- Product aesthetics (NFC_5) is a new NFC added by 6 experts. They estimate that it has a positive relation with Product quality NFC (NFC_1)

According to the consequences of the first round of Delphi methodology, the initial cognitive map needs some improvements and corrections.

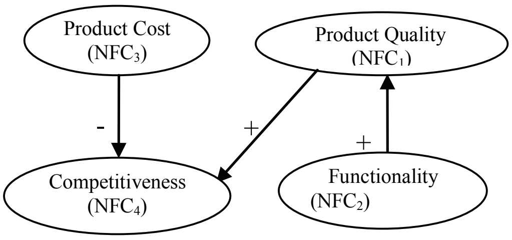


Figure 4. The initial CM

Table 1. First round: distribution of the experts' responses

Relationships	Positive relationship	Negative Relationship	No relationship
NFC_2-NFC_1	10	0	0
NFC_1-NFC_4	9	1	0
NFC_3-NFC_4	3	6	1
NFC_5-NFC_1	6	0	4

After applying these revisions, the second round starts up. The revised cognitive map with the frequency of responses obtained at the end of the first round is sent to the experts. The analyst asks them to explore the relations in the new cognitive map and insert their opinions. The results of the second round are collected in **Table 2**. We observe that the experts have made some compromises. **Figure 5** depicts the final CM.

Table 2. Second round: distribution of the experts' responses

Relationships	Positive relationship	Negative relationship	No relationship
NFC_2-NFC_1	10	0	0
NFC_1-NFC_4	9	1	0
NFC_3-NFC_4	1	8	1
NFC_5-NFC_1	10	0	0

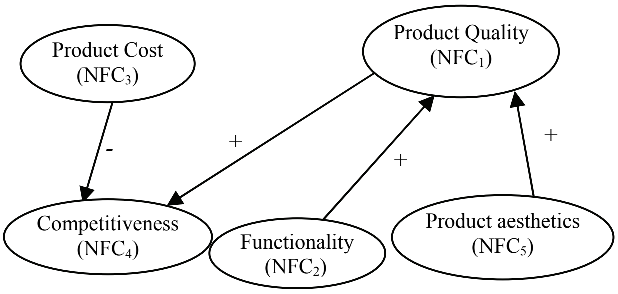


Figure 5. The final CM

5.3 Specify the Fuzzy Weights and so Provide the FCM

In order to determine the weights of the relations identified in the previous step, the analyst asks the following question to the experts:

How strong, do you believe, the causal relationship between NFCs in the final CM is?

The experts express their beliefs in the strength of a certain causal relationship, by assigning a fuzzy weight ranging between 0 and 1 where, we consider that:

- 0 < Low < 0.25;
- 0.25 <= Moderate < 0.75;
- 0.75 <= Strong <= 1.

Table 3 summarizes the results where E_i denotes the

Table 3. Experts' evaluations for relationships fuzzy weights

Relationships	E1	E2	E3	E4	E5	E6	E7
NFC_2-NFC_1	0.80	0.85	0.90	1,00	0.80	0.95	0.90
NFC_1-NFC_4	0.85	0.90	0.80	0.90	1,00	0.92	0.87
NFC_3-NFC_4	0.30	0.28	0.40	0.45	0.50	0.40	0.35
NFC_5-NFC_1	0.40	0.35	0.45	0.50	0.48	0.55	0.60

i-th expert and i=1,..,7, *i.e.* 7 experts participate to this step.

The weight of each relationship is the average of the experts' evaluation for this relationship. The value returned by this function allows attributing a linguistic fuzzy weight. **Table 4** shows the results.

The final FCM is the labelled CM with its fuzzy linguistic weights depicted by **Figure 6.**

Figure 7 depicts a screen-dump of MAMIE-Tool when NFCs are identified and specified with their relationships.

The managers and planners can use this FCM to increase the competitiveness and to improve the product quality by analyzing different choices; such as decreasing the product cost. Thus, using a FCM to model relationships between NFCs provides a way to identify how to reach goals

Table 4. Linguistics weights of relationships between NFCs

Relationships	Average value	Weights of relationships
NFC_2-NFC_1	0.88	Strong
NFC_1-NFC_4	0.89	Strong
NFC_3-NFC_4	0.38	Moderate
NFC_5-NFC_1	0.47	Moderate

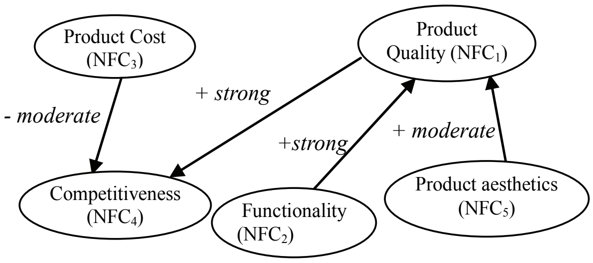
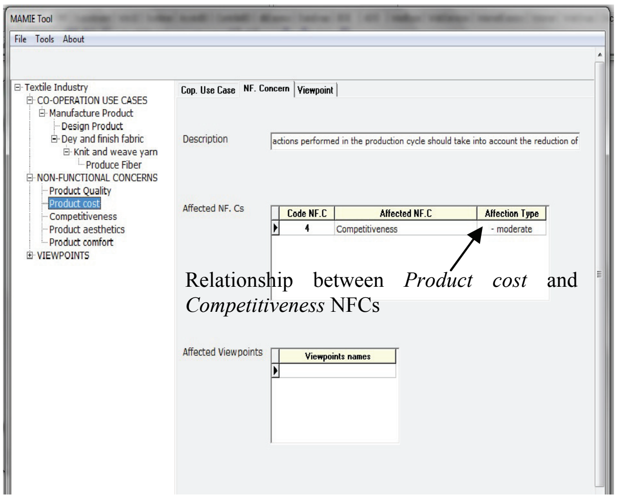


Figure 6. The final FCM



Description: actions performed in the production cycle should take into account the reduction of

Affected NF. Cs

Code NF.C	Affected NF.C	Affection Type
4	Competitiveness	- moderate

Relationship between *Product cost* and *Competitiveness* NFCs

Affected Viewpoints

Viewpoints names

Figure 7. MAMIE-Tool interface-NFCs' relationships

6. Conclusion and Future Works

This paper presents a process to identify and model relationships between non-functional concerns. To do so, we chose fuzzy cognitive maps which have been introduced to model ill-structured complex systems. Building a fuzzy cognitive map follows an approach similar to human reasoning and the human decision-making process. They can successfully represent knowledge and human experience; introduce concepts to represent the essential elements and the cause and effect relationships among the concepts to model the behaviour of any system. It is a very convenient, simple, and powerful tool, which is used in numerous fields.

The process presented in this paper starts by identifying non-functional concerns. Then, the analyst builds an initial cognitive map for the identified non-functional concerns. In order to obtain consensus for a final cognitive map between experts, Delphi method is used. A fuzzy linguistic value for a specific relationship is obtained by calculates the average of all the evaluations values given by experts. Then s/he classifies it as weak, moderate or strong. Using fuzzy linguistics labels makes the fuzzy cognitive map more sensible. In addition, the presented process enables the experts to simulate idea from various viewpoints.

We believe that using a fuzzy cognitive map to model non-functional concerns relationships proves useful and looks promising for a move from the conventional modelling toward developing computer based model.

Thus, further works will concentrate on two objectives:

- Design an expert system based on fuzzy cognitive maps.
- Study the use of the presented process in the context of Aspect Oriented Requirements Engineering (AORE) area to handle interactions between aspects. AORE aims at addressing crosscutting concerns by means of aspects to provide their identification, separation, representation and composition.

7. Acknowledgements

This research is partially supported by the PHC TASSILI project under the number 10MDU817. We thank the anonymous reviewers for providing valuable comments.

REFERENCES

[1] F. P. Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering," *IEEE Computer Journal*, Vol. 20, No. 1, 1987, pp. 10-19.

[2] B. H. C. Cheng and J. M. Atlee, "Research Directions in Requirements Engineering," *Proceedings of the Future of Software Engineering Conference*, Minneapolis, IEEE Computer Society, 2007, pp. 285-303.

Copyright © 2010 SciRes.

[3] H. Bendjenna, N. Zarour and P. J. Charrel, "MAMIE, A Methodology to Elicit Requirements for an Inter-Company Cooperative Information System," *Proceeding of IEEE Conference on Innovation on Software Engineering* (*ISE*), Vienna, IEEE Computer Society, 2008, pp. 290-295.

[4] H. Bendjenna, N. Zarour and P. J. Charrel, "Enhancing the Elicitation Technique Selection Process in a Cooperative Distributed Environment," *Proceeding of the* 14*th Requirements Engineering Foundation for Software Quality Conference* (*REFSQ*), Montpellier, *Lecture Notes in Computer Science*, Vol. 5025, Springer, Heidelberg, 2008, pp. 23-36.

[5] H. Bendjenna, M. Amroune, N. Zarour and P. J. Charrel, "Prioritizing Non-functional Concerns in MAMIE Methodology," *Proceeding of the Third International Conference on Knowledge Science, Engineering and Management* (*KSEM*), Vienna, *Lecture Notes in Artificial Intelligence*, Vol. 5914, Springer, Heidelberg, 2009, pp. 253-262.

[6] L. Chung, B. Nixon, E. Yu and J. Mylopoulos, "Non-Functional Requirements in Software Engineering," Kluwer Academic Publishers, New York, 2000.

[7] P. J. A. Diaz, F. Trilnik and M. R. Campo, "How to Handle Interaction Concerns?" *Proceedings of the ACM Conference on Object-Oriented Programming*, *Systems*, *Languages*, *and Applications OOPSLA*'00, Minneapolis, 2000, pp. 32-52.

[8] J. Liu, D. Batory and S. Neduniry, "Modelling Interactions in Feature Oriented Software Design," *International Conference on Feature Interactions in Telecommunications and Software Systems* (*ICFI*), Leicester, 2005, pp. 178-197.

[9] F. Sanen, E. Truyen, W. Joosen, N. Loughran, A. Rashid, A. Jackson, A. Nedos and S. Clarke, "Study on Interaction Issues," *AOSD-Europe Deliverable*, Vol. 44, Leuven, 2006, pp. 1-31.

[10] R. Axelrod, "Structure of Decision: The Cognitive Maps of Political Elites," Princeton University Press, Princeton, 1976.

[11] H. J. Zimmermann, "Fuzzy Set Theory and its Applications," Kluwer Academic Publishers, Dordrecht, 1991.

[12] S. Bueno and J. L. Salmeron, "Fuzzy Modeling Enterprise Resource Planning Tool Selection," *Computer Standards & Interfaces*, Vol. 30, No. 3, 2008, pp. 137-147.

[13] B. Boehm, "Characteristics of Software Quality," North Holland Press, New York, 1978.

[14] N. E. Fenton and S. L. Pfleeger, "Software Metrics: A Rigorous and Practical Approach," 2nd Edition, International Thomson Computer Press, London, 1997.

[15] S. E. Keller, L. G. Khan and R. B. Panara, "Specifying Software Quality Requirements with Metrics," In: R. H. Thayer and M. Dorfman, Eds., *Tutorial*: *System and Software Requirements Engineering*, IEEE Computer Society Press, Piscataway, 1990, pp. 145-163.

[16] M. R. Lyu, "Handbook of Software Reliability Engineering," McGraw-Hill, New York, 1996.

[17] B. Boehm and H. In, "Identifying Quality-Requirement Conflicts," *IEEE Software Journal*, Vol. 13, No. 2, 1996, pp. 25-35.

[18] L. M. Cysneiros, J. C. S. P Leite and J. S. M. Neto, "A Framework for Integrating Non-Functional Requirements into Conceptual Models," *Requirements Engineering Journal*, Vol. 6, No. 2, 2001, pp. 97-115.

[19] T. G. Kirner and A. M. Davis, "Non-functional Requirements of Real-Time Systems," *Advances in Computers Journal*, Vol. 42, No. 1, 1996, pp. 1-37.

[20] D. Kardaras and B. Karakostas, "The Use of Fuzzy Cognitive Maps to Simulate the Information Systems Strategic Planning Process," *Information and Software Technology Journal*, Vol. 41, No. 1, 1999, pp. 97-210.

[21] B. Kosko, "Fuzzy Cognitive Maps," *International Journal on Man-Machine Studies*, Vol. 24, No. 1, 1986, pp. 65-75.

[22] J. Bezdek, "Fuzzy Models-What Are They, and Why," *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, 1993, pp. 1-6.

[23] R. Taber, "Knowledge Processing with Fuzzy Cognitive Maps," *Expert Systems with Applications Journal*, Vol. 2, No. 1, 1991, pp. 83-87.

[24] C. Okoli and S. Pawlowski, "The Delphi Method as a Research Tool: An Example, Design Considerations and Applications," *Information & Management*, Vol. 42, No. 1, 2004, pp. 15-29.

[25] N. Bryson, A. Mobolurin and A. Joseph, "Generating Consensus Fuzzy Cognitive Maps," *Intelligent Information Systems*, Vol. 8, No. 10, 1997, pp. 231-235.

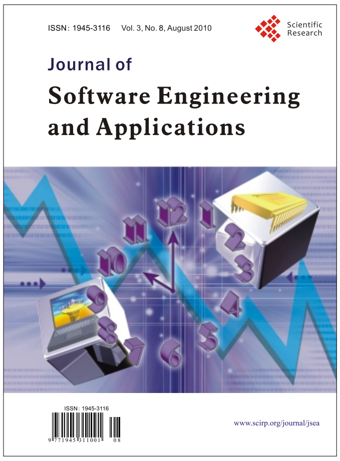
[26] N. C. Dalkey and O. Helmer, "An Experimental Application of the Delphi Method to the User of Experts," *Management Science Journal*, Vol. 9, No. 3, 1963, pp. 458-467.

[27] W. R. Zhang, S. S. Chen and J. C. Bezdek, "A Generic System for Cognitive Map Development and Decision Analysis," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, Vol. 19, No. 1, 1989, pp. 31-39.

[28] W. R. Zhang, S. S. Chen, W. Wang and R. King, "A Cognitive-Map-Based Approach to The Coordination of Distributed Cooperative Agents," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, Vol. 22, No. 1, 1992, pp. 103-114.

[29] G. J. Klir and B. Youan, "Fuzzy Sets and Fuzzy Logic, Theory and Applications," Prentice Hall, India, 2005.

Call for Papers


ISSN: 1945-3116 Vol. 3, No. 8, August 2010
Scientific Research
Journal of
Software Engineering and Applications
ISSN: 1945-3116
www.scirp.org/journal/jsea

Journal of Software Engineering and Applications (JSEA)

ISSN 1945-3116 (print) ISSN 1945-3124 (online)
www.scirp.org/journal/jsea

JSEA publishes four categories of original technical articles: papers, communications, reviews, and discussions. Papers are well-documented final reports of research projects. Communications are shorter and contain noteworthy items of technical interest or ideas required rapid publication. Reviews are synoptic papers on a subject of general interest, with ample literature references, and written for readers with widely varying background. Discussions on published reports, with author rebuttals, form the fourth category of JSEA publications.

Editor-in-Chief

Dr. Ruben Prieto-Diaz, Universidad Carlos III de Madrid, Spain

Subject Coverage

- Applications and Case Studies
- Artificial Intelligence Approaches to Software Engineering
- Automated Software Design and Synthesis
- Automated Software Specification
- Component-Based Software Engineering
- Computer-Supported Cooperative Work
- Software Design Methods
- Human-Computer Interaction
- Internet and Information Systems Development
- Knowledge Acquisition
- Multimedia and Hypermedia in Software Engineering
- Object-Oriented Technology
- Patterns and Frameworks
- Process and Workflow Management
- Programming Languages and Software Engineering
- Program Understanding Issues
- Reflection and Metadata Approaches
- Reliability and Fault Tolerance
- Requirements Engineering
- Reverse Engineering
- Security and Privacy
- Software Architecture
- Software Domain Modeling and Meta-Modeling
- Software Engineering Decision Support
- Software Maintenance and Evolution
- Software Process Modeling
- Software Reuse
- Software Testing
- System Applications and Experience
- Tutoring, Help and Documentation Systems

Notes for Prospective Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. All papers are refereed through a peer review process. For more details about the submissions, please access the website.

Website and E-Mail

Website: http://www.scirp.org/journal/jsea E-Mail: jsea@scirp.org

J. Software Engineering and Applications, **2010, 3, 737-826.**
Published Online August 2010 in SciRes (http://www.SciRP.org/journal/jsea).

TABLE OF CONTENTS

Volume 3 Number 8 August 2010

Copyright©2010 SciRes.