Advances in Intelligent Systems and Computing 285

Radek Silhavy
Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy
Zdenka Prokopova *Editors*

Modern Trends and Techniques in Computer Science

3rd Computer Science On-line Conference 2014 (CSOC 2014)

Springer

Advances in Intelligent Systems and Computing

Volume 285

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/11156

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Radek Silhavy · Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy · Zdenka Prokopova
Editors

Modern Trends and Techniques in Computer Science

3rd Computer Science On-line Conference 2014 (CSOC 2014)

Springer

Editors
Radek Silhavy
Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy
Zdenka Prokopova
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

ISSN 2194-5357 ISSN 2194-5365 (electronic)
ISBN 978-3-319-06739-1 ISBN 978-3-319-06740-7 (eBook)
DOI 10.1007/978-3-319-06740-7
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014937958

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book constitutes the refereed proceedings of the *3rd Computer Science On-line Conference 2014 (CSOC 2014)*, held in April 2014.

We are promoting new scientific conference concepts by organizing the CSOC conference. Modern online communication technology improves the traditional concept of scientific conferences. It brings equal opportunity to participate to all researchers around the world. Therefore, CSOC 2014 uses innovative methodology to allow scientists, postdocs, and doctoral students to share their knowledge and ideas online.

The conference intends to provide an international forum for the discussion of the latest high-quality research results in all areas related to Computer Science. The topics addressed are the theoretical aspects and applications of Artificial Intelligences, Computer Science, and Software Engineering. The authors present new approaches and methods to real-world problems, and particularly, exploratory research that describes novel approaches in their field.

The 53 papers presented in the proceedings were carefully reviewed and selected from 95 paper submissions. At least two respected reviewers reviewed each submission. 58 % of all submissions were received from Europe, 27 % from Asia, 7 % from America, and 5 % from Africa.

The editors believe that readers will find the proceedings interesting and useful for their own research work.

March 2014

Radek Silhavy
Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy
Zdenka Prokopova

Program Committee

Program Committee Chairs

Zdenka Prokopova, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: prokopova@fai.utb.cz
Zuzana Kominkova Oplatkova, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: kominkovaoplatkova@fai.utb.cz
Roman Senkerik, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: senkerik@fai.utb.cz
Petr Silhavy, Ph.D., Senior Lecturer, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: psilhavy@fai.utb.cz
Radek Silhavy, Ph.D., Senior Lecturer, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: rsilhavy@fai.utb.cz

Program Committee Members

Dr. Luis Alberto Morales Rosales, Head of the Master Program in Computer Science, Superior Technological Institute of Misantla, Mexico.
Mariana Lobato Baes, M.Sc., Research Professor, Superior Technological of Libres, Mexico.
Abdessattar Chaâri, Professor, Laboratory of Sciences and Techniques of Automatic control and Computer engineering, University of Sfax, Tunisian Republic.
Gopal Sakarkar, Shri Ramdeobaba College of Engineering and Management, Republic of India.
V. V. Krishna Maddinala, Assistant Professor, GD Rungta College of Engineering and Technology, Republic of India.
Anand N. Khobragade, Scientist, Maharashtra Remote Sensing Applications Centre, Republic of India.
Abdallah Handoura, Assistant Professor, Computer and Communication Laboratory, Telecom Bretagne—France.

Technical Program Committee Members

Eric Afful Dazie
Michal Bliznak
Donald Davendra
Radim Farana
Zuzana Kominkova Oplatkova
Martin Kotyrba
Erik Kral
David Malanik
Michal Pluhacek
Zdenka Prokopova
Martin Sysel
Roman Senkerik
Petr Silhavy
Radek Silhavy
Jiri Vojtesek
Eva Volna

Organizing Committee Chair

Radek Silhavy, Ph.D., Senior Lecturer, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: rsilhavy@fai.utb.cz

Conference Organizer (Production)

OpenPublish.eu s.r.o.
Svornosti 1908
755 01 Vsetin
Czech Republic
Web: www.openpublish.eu
E-mail: csoc@openpublish.eu

Contents

Part I
Artificial Inteligence

Intelligence Digital Image Watermark Algorithm Based on Artificial Neural Networks Classifier

Cong Jin and Shu-Wei Jin

Abstract An intelligence robust digital image watermarking algorithm using artificial neural network (ANN) is proposed. In new algorithm, for embedding watermark, the original image first is divided into some $N_1 \times N_2$ small blocks, different embedding strengths are determined by RBFNN classifier according to different textural features of every block after DCT. The experimental results show that the proposed algorithm are robust against common image processing attacks, such as JPEG compression, Gaussian noise, cropping, mean filtering, wiener filtering, and histogram equalization etc. The proposed algorithm achieves a good compromise between the robustness and invisibility, too.

Keywords Digital watermarking · ANN · Classification · Textural feature · Invisibility · Robustness

1 Introduction

Multimedia data is easily copied and modified, so necessity for copyright protection is increasing. Digital watermarking [1, 2] has been proposed as the technique for copyright protection of multimedia data. A watermarking algorithm requires both invisibility and robustness, which exist in a trade-off relation. Many watermarking systems based on artificial neural network (ANN) have been already proposed [3–5]. In [3], watermarking is viewed as a form of communications. A blind watermarking algorithm is presented using Hopfield neural network to calculate the capacity of watermarking. Zhang et al. [4] used a back-propagation

C. Jin (✉) · S.-W. Jin
School of Computer Science, Central China Normal University,
Wuhan 430079, People's Republic of China
e-mail: jincong@mail.ccnu.edu.cn

S.-W. Jin
e-mail: dede91@mail.ccnu.edu.cn

R. Silhavy et al. (eds.), *Modern Trends and Techniques in Computer Science*,
Advances in Intelligent Systems and Computing 285, DOI: 10.1007/978-3-319-06740-7_1,

(BP) ANN to learn the characteristics of relationship between the watermark and the watermarked image. The false decoded rate of the watermark can be greatly reduced by the trained ANN. In [5], ANN is used to model human visual system (HVS) [6] and an image-adaptive scheme of watermarking strength decision method is presented for the watermarking on DCT coefficients. Robust digital image watermarking schemes based on ANN are proposed in [5], too. We know that the robustness of watermarks depends on the watermark embedding strength. For transform domain watermark, if selecting higher watermark embedding strength, it has good robustness and bad invisibility; and if selecting lower watermark embedding strength, it has bad robustness and good invisibility. Therefore, when we choose a watermark embedding strength, we should consider a trade-off between invisibility and robustness.

Different from existing methods, in this paper, we don't discuss the capacity of watermarking (topic of [3]). In [4], the properties of HVS don't be considered. Although the good watermark performance was obtained by using ANN in [5], the watermark extraction processes are not blind with referring to the original image. In this paper, we propose a new watermarking algorithm based on ANN. After dividing the original image into some non-overlapping small blocks of size $N_1 \times N_2$, ANN is used to determine different watermarking embed strengths according to different textural features of every block.

2 ANN Classifier

Multi-layer perceptron has many advantages such as simple structure [7], rapid training process and good extend ability etc. So it can be applied to many fields, especially, in the aspects of pattern classification and function approach.

ANN-based classifiers can incorporate both statistical and structural information and achieve better performance than the simple minimum distance classifiers [8]. An ANN possesses the following characteristics [9]: (1) It is capable of inferring complex non-linear input–output transformations. (2) It learns from experience, so, has no need for any a priory modeling assumptions. Therefore, based on the advantages of ANN, multi-layered ANN, usually employing the back propagation (BP) algorithm, is also widely used in digital watermarking [10]. In this paper, an effective classification approach using ANN according to the image textural features is proposed.

In this paper, we let ANN have four layers including an input layer with five neurons, first hidden layer with six neurons, second hidden layer with eight neurons and one output neuron. Where, the number of the input neuron is determined by dimension number of textural feature vector. The output of ANN is the maximum watermarking strength. All the input features and output have to be normalized so that they always fall within a specified range before training. The inputs and the output are normalized to fall in the interval $[-1, 1]$ and $[0, 1]$, respectively.

3 Select Textural Features

Let I be a gray-level original image of size $M_1 \times M_2$, and we divide I into non-overlapping $N_1 \times N_2$ blocks, J,

$$J = \bigcup_{i=1}^{M_1/N_1} \bigcup_{j=1}^{M_2/N_2} J_{(i,j)} \tag{1}$$

where, $M_1 > N_1$ and $M_2 > N_2$.

For one non-overlapping block $J_{(i,j)}$ in I, we will extract five-dimensional vectors of features. We use five features: one from the image histogram (mean gray level) and four from the gray level co-occurrence matrix (contrast, entropy, correlation, and angular second moment). Specific definitions of these features are given below.

1. First-order gray level parameter

   In this category, the feature is derived from the gray level histogram. It describes the first-order gray level distribution without considering spatial interdependence. As a result, it can only describe the echogenicity of texture as well as the diffuse variation characteristics within the every $N_1 \times N_2$ block. The feature selected from this category is:

(a) The mean gray level (*MGL*)

$$MGL = \frac{1}{N_1 \times N_2} \sum_{m=1}^{N_1} \sum_{n=1}^{N_2} J_{(i,j)}(m,n),\ 1 \le i \le \frac{M_1}{N_1},\ \ 1 \le j \le \frac{M_2}{N_2} \tag{2}$$

where $J_{(i,j)}(m, n)$ is the gray level of $N_1 \times N_2$ block $J_{(i,j)}$ at pixel (m, n).

2. Second-order gray level features

   This category of features describes the gray level spatial inter-relationships and hence, they represent efficient measures of the gray level texture homogeneity. These features can be derived using several approaches such as first-order gradient distribution, gray level co-occurrence matrix, edge co-occurrence matrix, or run-length matrix. Because the gray level co-occurrence matrix seems to be a well-known statistical technique for feature extraction, we will generate these features from the gray level co-occurrence matrix. We generated a gray-level co-occurrence matrix from every $N_1 \times N_2$ block. The formal definition of this matrix is as follows:

$$Co(s,t) = \frac{1}{N_1 \times N_2} cardinality\{((k_1,k_2),\ (l_1,l_2)) \in J_{(i,j)} : \quad |k_1 - l_1| = dx, \quad |k_2 - l_2| = dy \tag{3}$$

$$sign((k_1 - l_1) \cdot (k_2 - l_2)) = sign(dx \cdot dy),\ J_{(i,j)}(k_1, k_2) = s,\ J_{(i,j)}(l_1, l_2) = t\Big\} \tag{4}$$

where $Co(s, t)$ is the gray level co-occurrence matrix entry at gray levels s and t, and (dx, dy) is a prescribed neighborhood definition taken in case to be (0, 12) representing an axial neighborhood definition. In other words, the entry (s, t) of this matrix describes how often the two gray levels s and t are neighbors under the given neighborhood definition. Note that this definition does not discriminate between negative and positive shifts and hence, the co-occurrence matrix is expected to be symmetric using this definition. The four features are defined as follows

(b) The contrast (*CON*)

$$CON = \sum_{s,t} (s - t)^2 \cdot Co(s, t) \tag{5}$$

The contrast feature is a difference moment of the $Co(s, t)$ matrix and is a standard measurement of the amount of local variations presented in an image. The higher the values of contrast are, the sharper the structural variations in the image are.

(c) The entropy (*ENT*)

$$ENT = -\sum_{s,t} Co(s,t) \cdot \log(Co(s,t)) \tag{6}$$

(d) The correlation (*COR*)

$$COR = \frac{\sum_{s,t} stCo(s, t) - m_x \cdot m_y}{S_x \cdot S_y} \tag{7}$$

where,

$$m_x = \sum_s s \sum_t Co(s, t),\ m_y = \sum_t t \sum_s Co(s, t), \quad S_x^2 = \sum_s s^2 \sum_t Co(s, t) - m_x^2,$$
$$S_y^2 = \sum_t t^2 \sum_s Co(s, t) - m_y^2$$

(e) The angular second moment (*ASM*)

$$ASM = \sum_{s,t} (Co(s, t))^2 \tag{8}$$

Angular second moment gives a strong measurement of uniformity. Higher non-uniformity values provide evidence of higher structural variations.

4 Embedding and Extraction of Watermark

4.1 Select DCT Coefficients for Inserting Watermark

Our goal is to embed the roust watermark into to the DCT frequency bands of I. Before the embedding procedure, we need to transform the spatial domain pixels into DCT domain frequency bands. After we perform the $N_1 \times N_2$ block DCT on I, we get the coefficients in the frequency bands, F,

$$F = DCT(I), \quad \text{and} \quad F = \bigcup_{i=1}^{M_1/N_1} \bigcup_{j=1}^{M_2/N_2} F_{(i,j)} \tag{9}$$

For one non-overlapping block (i, j) in I, the resulting $N_1 \times N_2$ DCT bands $J_{(i,j)}$ can be represented by

$$F_{(i,j)} = \bigcup_{k=1}^{N_1 \times N_2} \{F_{(i,j)}(k)\}, \quad 1 \le i \le \frac{M_1}{N_1}, \quad 1 \le j \le \frac{M_2}{N_2} \tag{10}$$

$F_{(i,j)}(k)$ are zigzag ordered DCT coefficients. Afterwards, we are able to embed the watermark in the DCT domain. Assuming that the binary-valued watermark to be embedded is W, having size $S_1 \times S_2$. A pseudo-random number traversing method is applied to permute the watermark to disperse its spatial relationship. With a pre-determined key, key_0, in the pseudo-random number generating system, we have the permuted watermark W^P,

$$W^P = permute(W, key_0). \tag{11}$$

And we use W^P for embedding the watermark bits into the selected DCT frequency bands. The human eye is more sensitive to noise in lower frequency components than in higher frequency ones. However, the energy of most natural images is concentrated in the lower frequency range, and watermark data in the higher frequency components might be discarded after quantization operation of lossy compression. In order to invisibly embed the watermark that can survive lossy data compressions, a reasonable trade-off is to embed the watermark into the middle-frequency range of the image. In this paper, the middle band of the DCT domain is chosen. But, with regard to JPEG, casting watermarks in the middle band of the $N_1 \times N_2$ block-based DCT domain is more robust.

In the case of embedding a watermark based on $N_1 \times N_2$ block DCT, only $k = \left\lfloor N_1 \times N_2 \times \frac{S_1 \times S_2}{M_1 \times M_2} \right\rfloor$ coefficients for each $N_1 \times N_2$ block $F_{(i,j)}$ will be used for the watermark embedding. First, the DCT coefficients of each block are reordered in zig-zag scan. Then, the coefficients from the $(L + 1)$th to the $(L + k)$th, *i.e.* a sequence of values $F_{(i,j)} = \{F_{(i,j)}(L+1), F_{(i,j)}(L+2), \ldots, F_{(i,j)}(L+k)\}$ are taken according to the zig-zag ordering of the DCT spectrum, where the first L coefficients are skipped for embedding the middle band.

4.2 Embedding the Watermark

The amount of modification each coefficient undergoes is proportional to the magnitude of the coefficients itself as expressed by

$$\hat{F}_{(i,j)}(L+u) = \begin{cases} F_{(i,j)}(L+u) + \alpha_{(i,j)}(u) \cdot \left|F_{(i,j)}(L+u)\right| \cdot w^P_{(i,j)}(u), & \text{if } w^P_{(i,j)}(u) = 1 \\ F_{(i,j)}(L+u), & \text{otherwise} \end{cases} \tag{12}$$

$$u = 1, 2, \ldots, k; \quad 1 \le i \le \frac{M_1}{N_1}, \quad 1 \le j \le \frac{M_2}{N_2}$$

where, $\alpha_{(i,j)}(u)$ indicates uth parameter controlling the watermarking strength of block (i, j) given by the ANN. $\hat{F}_{(i,j)}$ is then inserted back into the image in place of $F_{(i,j)}$, and we obtain the watermarked DCT coefficients, F',

$$F' = \bigcup_{i=1}^{M_1/N_1} \bigcup^{M_2/N_2} \hat{F}_{(i,j)} \tag{13}$$

After performing inverse DCT on F', we get the watermarked image, $\hat{I}$,

$$\hat{I} = inverse\_DCT(F') \tag{14}$$

4.3 Extracting the Watermark

To extract a watermark in a possibly watermarked image I_w, firstly, I_w is decomposed into non-overlapping $N_1 \times N_2$ blocks and the DCT is computed for each blocks. k DCT coefficients are selected by using the same method with

embedding watermark for each block. Another necessary procedure is to calculate the suggested strength $\alpha_{(i,j)}(u)$ of the corresponding block (i,j). Moreover, we let

$$Th_{(i,j)} = \frac{1}{k}\sum_{u=1}^{k} F_{(i,j)}(L+u) \tag{15}$$

we are able to extract the permuted watermark according to the following equation

$$W'^{P}_{(i,j)}(u) = \begin{cases} 1, & if \ \left|\hat{F}_{(i,j)}(L+u) - F_{(i,j)}(L+u)\right| \geq \alpha_{(i,j)}(u) \cdot Th_{(i,j)} \\ 0, & otherwise \end{cases} \tag{16}$$

$$W'^{P} = \bigcup_{i=1}^{M_1/N_1} \bigcup_{j=1}^{M_2/N_2} W'^{P}_{(i,j)}(u), \quad u = 1, 2, \ldots, k \tag{17}$$

We use key_0 in Eq. (11) to acquire the extracted watermark W' from W'^{P},

$$W' = inverse\_permute\,(W'^{P}, key_0) \tag{18}$$

5 Experimental Results

5.1 Experiments

To evaluate the performance of the proposed watermarking algorithm, a set of experiments is performed under the following conditions.

The "*Flower*" image with size $M_1 \times M_2 = 512 \times 512$, is used the original image, which is shown in Fig. 1. We have the embedded watermark with size $S_1 \times S_2 = 128 \times 128$, shown in Fig. 2. Size $N_1 \times N_2$ of small block is 8×8. Hence, the number of bits to be embedded in one 8×8 non-overlapping block is $k = 128^2/512^2 \cdot 64 = 4$, $L = 6$, and $F_{(i,j)} = \{F_{(i,j)}(7),\ F_{(i,j)}(8),\ F_{(i,j)}(9),\ F_{(i,j)}(10)\}$. Which is shown in Fig. 3. After watermark embedding in the DCT domain, we take the inverse DCT, and obtain the watermarked image. We measure the invisibility of the watermarked images and the robustness of the extracted watermarks against various attacks. The Peak Signal to Noise Ratio (PSNR) and the Normal Correlation (NC) shown in Eqs. (19) and (20) are used to measure the invisibility and the robustness, respectively.

$$PSNR = 10\log_{10}\frac{M_1 \times M_2 \times \sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\left(\hat{I}(i,j)\right)^2}{\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\left(I(i,j) - \hat{I}(i,j)\right)^2} \tag{19}$$

Fig. 1 Original image with size 512 × 512

Fig. 2 Watermark image with size 128 × 128

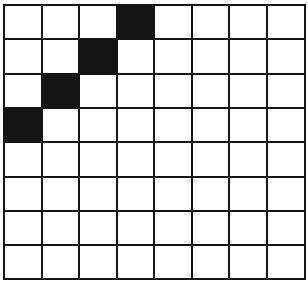


Fig. 3 4 bits are embedded in every 8 × 8 block

$$NC = \frac{\sum_{i=1}^{S_1} \sum_{j=1}^{S_2} W(i,j)W'(i,j)}{S_1 \times S_2} \times 100\ \% \tag{20}$$

where W and W' are original watermark and extracted watermark, respectively.

The ANN training procedure is terminated either when the training error is less than 10^{-4} or 2,000 iterations. The training error is the mean square error. The learning rate and momentum term were chosen as 0.1–0.15 and 0.8–0.9, respectively. The initial weight values, momentum term, and learning rate are the parameters of BP algorithm. The most commonly used winner-takes-all method was used for selecting the ANN output. The hidden and output neuron functions were defined by the logistic sigmoid function $f(x) = 1/(1 + \exp(-x))$.

The watermarked image using proposed algorithm is depicted in Fig. 4.

Fig. 4 The watermarked image, PSNR = 44.63

Table 1 PSNR of watermarked images

Image	Flower	Baboon	Lena	Goldhill	Peppers
PSNR	44.63	45.31	46.82	43.94	45.58

5.2 Results

5.2.1 Invisibility

The goal is to measure the invisibility of the watermarked images. PSNR is used to measure the invisibility of the watermarked image, where the higher PSNR, the more transparency of the watermark. Basics image "*Flower*", another four images are chosen as the tested images. The four images of size 512 × 512 are *Baboon*, *Lena*, *Goldhill*, and *Peppers*. By using the proposed algorithm, we see that the watermark is almost invisibility to the human eyes. Table 1 shows PSNR of tested images after embedding watermark.

5.2.2 Robustness

1. Wiener filtering Wiener filtering is a kind of the common signal attack for digital watermark. Table 2 collects the results in terms of the average PSNR and NC about above mentioning five images, and Wiener filtering with different windows for the watermarked image.

To illustrate the effect of the Wiener filtering to an individual tested image, Fig. 5a shows the watermarked images of "*Flower*" under 7 × 7 window and Fig. 5b shows the corresponding extracted watermark. The experiment results demonstrated that the proposed algorithm has great robustness against Wiener filtering.

Table 2 PSNR and NC for Wiener filtering attack

Windows	PSNR	NC
3 × 3	40.07	0.9261
5 × 5	35.98	0.8492
7 × 7	30.62	0.8019

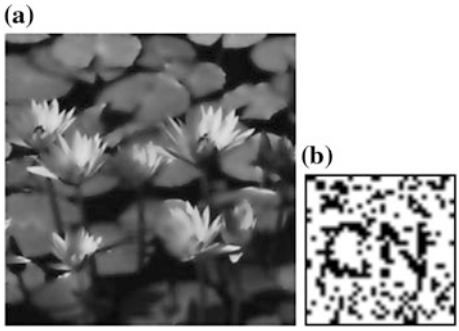


Fig. 5 **a** Watermarked image after Wiener filtering with a 7 × 7 window. **b** The watermark extracted from (**a**), NC = 0.8219

2. Mean filtering Mean filtering is another kind of common watermark attacks method. Table 3 collects the results in terms of the average PSNR, NC about above mentioning five images, and mean filtering with different windows for the watermarked image, respectively.

To illustrate the effect of the mean filtering to an individual tested image, Fig. 6a shows the watermarked images of "*Flower*" under 5 × 5 window and Fig. 6b shows the corresponding extracted watermark. Experiment results show that the proposed algorithm has very robustness against mean filtering.

3. Histogram equalization Histogram equalization is a kind of the common signal processing operation. Figure 7a shows the watermarked images of "*Flower*" after attacked by histogram equalization and Fig. 7b shows the corresponding extracted watermark.

Above experiment results confirm that proposed algorithm is robustness to histogram equalization attack, but also has more invisibility.

4. Gaussian noise The addition Gaussian noise is also a very common signal attack. We introduce Gaussian noise to watermarked image, Gaussian noise is zeros mean, and variance are 0.0005, 0.001, 0.002, 0.003 respectively. Figure 8a shows the watermarked images of "*Flower*" after attacked by Gaussian noise, and Fig. 8b–e shows the corresponding extracted watermark. The experiment results confirmed that the proposed algorithm has great robustness against Gaussian noise.
5. JPEG Compression We demonstrate the robustness of new algorithm. Since most digital images on networks are compressed, resistance against lossy

Table 3 PSNR and NC for mean filtering attack

Windows	PSNR	NC
3 × 3	38.25	0.8564
5 × 5	33.16	0.8091
7 × 7	29.67	0.7023

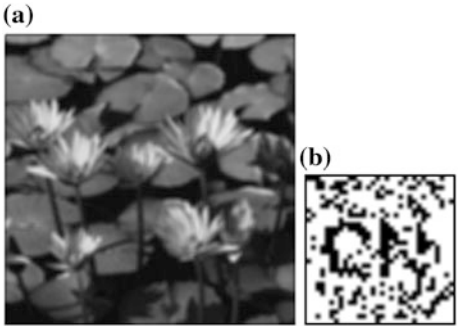


Fig. 6 **a** Watermarked image after mean filtering with a 5 × 5 window. **b** The watermark extracted from (**a**), NC = 0.7952

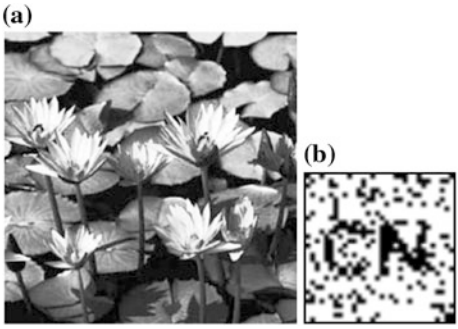


Fig. 7 **a** Watermarked image attacked by histogram equalization. **b** The watermark extracted from (**a**), NC = 0.8584

compression is essential. We obtain the watermark extraction results for the JPEG compression of watermarked '*Flower*' image with different quality factors (QF) 45, 40, 20, 15 to Fig. 4 respectively. The extraction results are shown in the Fig. 9a–d.

The experiment results demonstrated sufficiently that the proposed algorithm has great robustness against JPEG compression.

6. Cropping We cropped a part of the watermarked image, 1/4 cropped and filled with pixels valued 0 and 1/2 cropped and filled with pixels valued 255 respectively, and the extraction results are shown in Fig. 10. These experiments show that proposed algorithm is robustness to cropping attack.

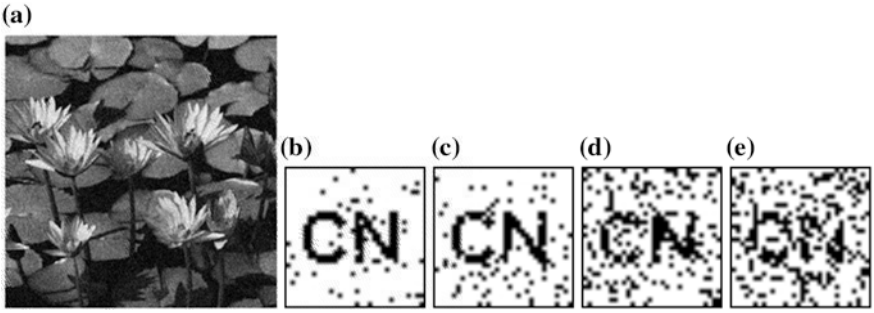


Fig. 8 **a** Watermarked image attacked by Gaussian noise with zero-mean and variance = 0.003, **b** extracted watermark when variance = 0.0005, **c** extracted watermark when variance = 0.001, **d** extracted watermark when variance = 0.002, and **e** extracted watermark when variance = 0.003

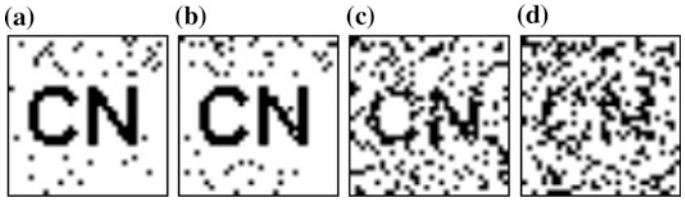


Fig. 9 Extracted images with a quality factors of 45, 40, 20, 15 respectively

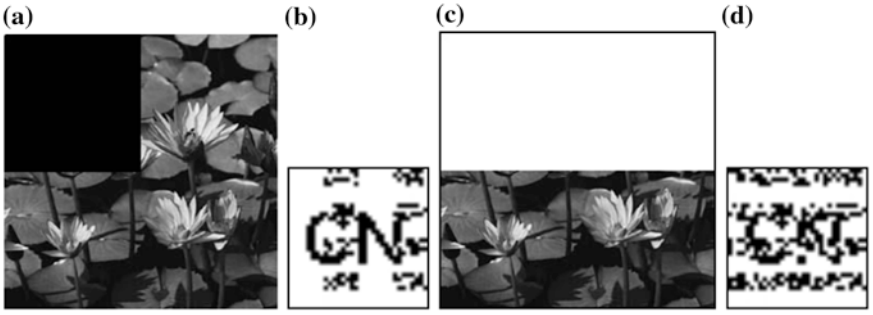


Fig. 10 Cropped the watermarked image 1/4 shown in (**a**) or 1/2 in (**c**) by different methods and the extracted watermarks are shown in (**b**) and (**d**) respectively

5.2.3 Comparison with Other Methods

We compare proposed algorithm to [11] method to *Lena*, *Goldhill* and *Peppers* images. The experiment results after various attacked are shown in Table 4. In Table 4, "Proposed" is average NC value to three images *Lena*, *Peppers*, and *Goldhill*.

Table 4 NC after attacked by JPEG compression with different QF of [11]'s methods

QF (%)	90	80	70	60	50	40	30	20	10
Lena[a]	0.99	0.99	0.97	0.95	0.96	0.90	0.82	0.67	0.34
Goldhill[a]	0.98	0.97	0.96	0.95	0.90	0.88	0.81	0.69	0.30
Peppers[a]	0.99	0.99	0.97	0.96	0.96	0.93	0.86	0.72	0.35
Proposed	1.000	1.000	0.991	0.989	0.985	0.941	0.847	0.768	0.626

[a] is proposed by [11]

Table 5 PSNR and NC after various attacks of [12]'s methods

Various attacks	[12][b]	[13][b]	[14][b]	[15][b]	Proposed
PSNR	38.92	40.89	40.08	26.77	40.86
LPF 3 × 3	0.86	0.92	0.91	1.00	0.9406
Median 3 × 3	1.00	0.87	0.84	1.00	1.0000
JPEG (80)	0.86	0.84	0.89	1.00	1.0000
JPEG (75)	0.82	0.81	0.85	1.00	1.0000
JPEG (50)	0.55	0.69	0.79	1.00	0.9259
Cropping 1/4	1.00	0.89	0.93	1.00	0.9772
Scaling 256 × 256	1.00	0.67	1.00	0.99	0.9898

[b] are proposed by [12–15], respectively

Numerical values in Table 4 show that new algorithm has shown the best performance for robustness against JPEG compression attacks. The performance of new algorithm is also compared with the results reported in [12–15] respectively. "Proposed" is average PSNR or NC value to five tested images *Baboon*, *Flower*, *Lena*, *Peppers*, and *Goldhill*. Numerical values in Table 5 show that the proposed algorithm has shown the very good performance for invisibility and robustness against various attacks. It is also seen that NC values for the method in [12] is little better compared to the proposed algorithm in case of lowpass filtering (LPF), JPEG compression (50 %), Cropping 1/4, and Scaling 256 × 256, but PSNR is much lower than the latter, did not achieve a good trade-off between invisibility and robustness.

6 Conclusion

Proposed new watermarking algorithm has the following advantages: (1) It can resist to common image processing attacks. Furthermore, watermark has good invisibility, and which makes that the proposed algorithm solves the conflict between invisibility and robustness better. (2) It is very difficult only to use a threshold determined by the experiments. By the proposed algorithm, this problem is a good solution. (3) The proposed algorithm shows the complex texture information of the image can be classified by ANN, and which makes that the proposed algorithm is very suitable to design digital image watermark algorithm.

Acknowledgments This work was supported by the science and technology research program of Wuhan of China (Grant No. 201210121023).

References

1. Jin, C.: Adaptive robust image watermark scheme based on fuzzy comprehensive evaluation and analytic hierarchy process. SIViP **6**, 317–324 (2012)
2. Li, L.D., Yuan, X.P., Lu, Z.L., Pan, J.S.: Rotation invariant watermark embedding based on scale-adapted characteristic regions. Inf. Sci. **180**, 2875–2888 (2010)
3. Zhang, F., Zhang, H.B.: Applications of a neural network to watermarking capacity of digital image. Neurocomputing **67**, 345–349 (2005)
4. Zhang, J., Wang, N.C., Xiong, F.: Hiding a logo watermark into the multi-wavelet domain using neural networks. In: 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02), pp. 477–482 (2002)
5. Frank, H.L.L., Chen, Z.Y., Tang, L.: Novel perceptual modeling watermarking with MLF neural networks. Int. J. Inf. Technol. **10**, 82–85 (2004)
6. Tsai, H.H., Liu, C.C.: Wavelet-based image watermarking with visibility range estimation based on HVS and neural networks. Pattern Recogn. **44**, 751–763 (2011)
7. Haykin, S.: Neural Networks, A Comprehensive Foundation. Macmillan, New York (1994)
8. Ohlsson, N., Helander, M., Wohlin, C.: Quality improvement by identification of fault-prone modules using software design metrics. In: International Conference on Software Quality, pp. 1–13 (1996)
9. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, New York (1995)
10. Haykin, S.: Neural Networks. Prentice Hall, New Jersey (1999)
11. Lin, W.H., Wang, Y.R., et al.: A blind watermarking method using maximum wavelet coefficient quantization. Expert Syst. Appl. **36**, 11509–11516 (2009)
12. Nasir, I., Weng, Y., Jiang, J.: A new robust watermarking scheme for color image in spatial domain. In: IEEE Conference on Signal Image Technologies and Internet-Based System, pp. 942–947. China (2007)
13. Zhang, D., Wu, B., Sun, J., Huang, H.: A new robust watermarking algorithm based on DWT. In: The Second International Congress on Image and Signal Processing, pp. 1–6. Tianjin, China (2009)
14. Luo, K., Tian, X.: A new robust watermarking scheme based on wavelet transform. In: Congress Image Signal Processing, pp. 312–316. Hainan, China (2008)
15. Verma, B., Jain, S.: A spatial domain robust non-oblivious watermarking scheme for image database. In: The Second Indian International Conference on Artificial Intelligence, pp. 3071–3085. Puna, India (2005)

PPSA: A Tool for Suboptimal Control of Time Delay Systems: Revision and Open Tasks

Libor Pekař and Pavel Navrátil

Abstract During the development of algebraic controller design in a special ring for time delay systems (TDSs) a problem of a suitable free controller parameters setting appeared. The first author of this contribution recently suggested a natural idea of placing the dominant characteristic numbers (poles) and zeros of the infinite-dimensional feedback control system on the basis of the desired overshoot for a simple finite-dimensional matching model and shifting of the rest of the spectrum. However, the original procedure called the Pole-Placement Shifting based controller tuning Algorithm (PPSA) was not developed and described entirely well. The aim of this paper is to revise the idea of the PPSA and suggest a possible ways how to improve or extend the algorithm. A concise illustrative example is attached to clarify the procedure for the reader as well.

Keywords Time delay systems · Pole placement controller tuning · Optimization · Direct-search algorithms · Evolutionary algorithms · SOMA · Nelder-Mead algorithm · Gradient sampling algorithm · Model matching

1 Introduction

Time delay systems (TDSs) constitute a huge class of processes and systems that are affected by any form of delay or latency, either in the input–output relation (as it is known in classical engineering problems) or inside the system dynamics (in this case notions of internal or state delays are introduced). The latter models and

L. Pekař (✉) · P. Navrátil
Faculty of Applied Informatics, Tomas Bata University in Zlín, Zlín, Czech Republic
e-mail: pekar@fai.utb.cz

P. Navrátil
e-mail: pnavratil@fai.utb.cz

R. Silhavy et al. (eds.), *Modern Trends and Techniques in Computer Science*,
Advances in Intelligent Systems and Computing 285, DOI: 10.1007/978-3-319-06740-7_2,

processes those are much more involved for analysis and control can be found in many theoretical and practical applications covering various fields of human activity, such as technology, informatics, biology, economy, etc., see e.g. [1–4].

A typical feature of TDSs is their infinite spectrum, due to transcendental nature of the characteristic equation, i.e. they have an infinite number of solution modes and corresponding system poles. This unpleasant attribute makes them difficult to analyze and design a control law as well. Linear time-invariant TDSs can be modeled and described by transfer functions by means of the Laplace transform. In most cases, roots of the transfer function denominator coincide with system poles.

The ring of quasipolynomial meromorphic functions (R_{MS}), originally developed and introduced in [5] and revised and extended in [6], represents a possible tool for description and control design of TDSs. However, in many cases, namely, for unstable TDSs, the control algorithm must deal with also infinitely many feedback characteristic poles the positions of which depend on the selectable controller parameters. The use of pole-placement (pole-assignment, root-locus) tuning algorithms can be a possible way how to solve the setting problem, see e.g. [7–9]. However, these algorithms deal with poles only ignoring closed-loop zeros and/or they have been derived for state-space controllers.

The idea of the Pole-Placement Shifting based controller tuning Algorithm (PPSA) provides slightly different approach [10]. It is based on the analysis of a simple finite-dimensional model where the relative maximum overshoot, relative dumping and relative time-to-overshoot of the reference-to-output step response are calculated and serve as a control performance indicators. Then, according to the selected values, the desired positions of dominant (i.e. the rightmost) poles and zeros are calculated, and poles and zeros of the infinite-dimensional feedback system are shifted to the prescribed positions while the rest of the spectrum is pushed to the left (i.e. to the "stable" region). In some sense, it represents a matching problem. The initial solution (i.e. controller parameter setting) is obtained using the Quasi-Continuous Shifting Algorithm (QCSA) [7, 8] which is followed by the use of an advanced numerical optimization algorithm. The method was independently developed in [9]; however, there are some essential differences—the reader is referred e.g. to [10] for details.

However, the original algorithm was described neither precisely nor in details and it contains some shortcomings and errors. Thus, the aim of this contribution is to revise and consolidate the PPSA and raise some open tasks how to improve and accelerate the algorithm. In this connection, the reader is kindly asked to participate on the solution of these problems in the future if he or she is interested in them.

To make the procedure clearer (to the reader) a short illustrative example on the control of an unstable time delay system by means of Matlab-Simulink environment is provided.

2 Time Delay Systems: Introductory Description

Since the reader is supposed to be a non-expert in system and control theory and the description and control design of TDSs is not the primary topic of this contribution, only a very concise overview of TDS models is provided such that all necessary information are given him or her.

A possible formulation of a TDS model (either a plant or a delayed control feedback loop) can be done using the transfer function in a complex variable s as the direct consequence of the use of the Laplace transform as follows

$$G(s) = \frac{b(s)}{a(s)} \tag{1}$$

where $a(s)$, $b(s)$ are quasipolynomials of a general form

$$x(s) = s^n + \sum_{i=0}^{n} \sum_{j=1}^{h_i} x_{ij} s^i \exp\left(-s\eta_{ij}\right); \quad \eta_{ij} \geq 0, x_{ij} \in \mathbb{R} \tag{2}$$

where η_{ij} express delays and $\mathbb{R}$ means the set of real numbers. If delays are included only in the numerator $b(s)$, they influence the input-output relation; in the contrary, the system contain internal delays and equation $a(s) = 0$ has infinitely many solution. These solution values constitute (in overwhelming majority of cases) system poles, more precisely, poles $s_i, i = 1, 2, \ldots$ are singularities of $G(s)$ satisfying

$$\lim_{s \to s_i} G(s) = \pm\infty; \quad \exists n_0, \forall n \geq n_0 : \lim_{s \to s_i} (s - s_0)^n G(s) < \infty \tag{3}$$

Zeros have the same meaning as in (3) yet for $1/G(s)$ instead of $G(s)$, i.e. they coincide with the roots of $b(s)$ (in most cases).

3 Problem Formulation

Now consider that $G(s)$ means the control feedback transfer function. Some control design approaches yield this function with the denominator containing delays along with free real controller parameters from the set $\mathbf{K} = \{k_1, k_2, \ldots, k_r\} \neq \varnothing \in \mathbb{R}^n$. This results in the infinite-dimensional (delayed) control feedback. Naturally, the numerator can own delays (and controller parameters) as well.

The idea of the PPSA is to match some number of the rightmost (i.e. the dominant) poles and zeros of $G(s)$ with all poles and zeros of a finite-dimensional model $G_m(s)$. Thus the selected poles and zeros of $G(s)$ are quasi-continuously

shifted to the desired positions by small steps and the rest of both spectra (of poles and zeros) try to push to the left (i.e. to the stable complex semiplane) as far as possible. The shifting can be done e.g. using the QCSA or via an advanced algorithm, [11–13], minimizing a suitable cost function reflecting the distance of dominant poles from prescribed positions and the spectral abscissa (i.e. the value of the real part of the rightmost pole/zero). By doing this, the values of $\mathbf{K}$ are being adjusted and hence the controller parameters are being tuned.

A crucial problem is to choose a suitable number of prescribed poles and zeros, i.e. degrees of the numerator, $N(s)$, and denominator, $D(s)$, of $G_m(s)$. Let us denote the numerator and the denominator as $N(s, \mathbf{K}_N)$ and $D(s, \mathbf{K}_D)$, respectively, where $\mathbf{K}_N$ and $\mathbf{K}_D$ mean free real parameters of the numerator and denominator, respectively, with $r_N = |\mathbf{K}_N| \geq 0$, $r_D = |\mathbf{K}_D| > 0$. It is initially assumed that equations $N(s_i, \mathbf{K}_N) = 0$, $D(s_j, \mathbf{K}_D) = 0$ are independent for arbitrary yet fixed s_i, s_j with $i = 1, 2, \ldots n_N \leq r_N$, $j = 1, 2, \ldots n_D \leq r_D$, that is

$$\begin{aligned} \operatorname{rank}\left[\frac{\partial}{\partial k_{N,l}} N(s_i, \mathbf{K}_N)\right]_{\substack{i=1,2,\ldots n_N \\ l=1,2,\ldots r_N}} &= n_N \\ \operatorname{rank}\left[\frac{\partial}{\partial k_{D,l}} D(s_j, \mathbf{K}_D)\right]_{\substack{j=1,2,\ldots n_D \\ l=1,2,\ldots r_D}} &= n_D \end{aligned} \tag{4}$$

Then the following conditions must hold: As indicated above, the number of prescribed poles, n_D, and zeros, n_N, must be less or equal to the number of corresponding free parameters to obtain a solvable matching problem. Moreover, if one needs to enable shifting the rest of the spectrum to the left, some parameters might not be bounded with desired position of roots, hence

$$0 \leq n_D < r_D, \quad 0 < n_N < r_N \tag{5}$$

where $\Delta n_D = r_D - n_D, \Delta n_N = r_N - n_N$ serve for adjusting the rightmost real parts of the rest of spectra. Naturally, the number of all desired solutions can not exceed the number of all free parameters, which gives rise to

$$n_D + n_N < r \tag{6}$$

In addition, the model has to be strictly proper, i.e.

$$n_D < n_N \tag{7}$$

Conditions (5)–(7) ought to be taken into account when designing the finite-dimensional model.

4 PPSA Strategies

Three possible revised modifications of the PPSA follows. A thorough algorithm description is consequently supported by its vague explanation and discussion in all three cases. Let is use these notations in the algorithms: $\mathbf{K} = \mathbf{K}_N \cup \mathbf{K}_D$ where numerator coefficients of $G(s)$ read $\mathbf{K}_N = \mathbf{K}_{N\setminus D} \cup \mathbf{K}_{\overline{ND}}$ with $r_{\overline{ND}} = \left|\mathbf{K}_{\overline{ND}}\right| = |\mathbf{K}_N \cap \mathbf{K}_D|$, $r_{N\setminus D} = \left|\mathbf{K}_{N\setminus D}\right| = \left|\mathbf{K}_N \setminus \mathbf{K}_{\overline{ND}}\right|$, whereas denominator ones analogously are $\mathbf{K}_D = \mathbf{K}_{D\setminus N} \cup \mathbf{K}_{\overline{ND}}$ with $r_{D\setminus N} = \left|\mathbf{K}_{D\setminus N}\right| = \left|\mathbf{K}_D \setminus \mathbf{K}_{\overline{ND}}\right|$. Simply, $r_N = r_{N\setminus D} + r_{\overline{ND}}$, $r_D = r_{D\setminus N} + r_{\overline{ND}}$.

Algorithm 1 (*PPSA strategy 1: "Poles First Independently"*)
Input. Closed-loop reference-to-output transfer function $G(s)$ with $r_{N\setminus D} > 0$.

Step 1. Set $n_D = r_D - 1$, thus $\Delta n_D = r_D - n_D = 1$. (Or just select $n_D < r_D$ as high as desirable).

Step 2. Verify that there can exist a non-negative number n_N satisfying

$$0 \le n_N < \min\left\{n_D, r_{N\setminus D}\right\} \tag{8}$$

If (8) holds, fix n_N and go to Step 3; otherwise, set $n_D = n_D + 1$. If $n_D < \min\{r_D, r_{N\setminus D} + 1\}$, i.e. $n_D < r_D$ and $n_D \le r_{N\setminus D}$, go to Step 2, else terminate the procedure (a solution does not exist).

Step 3. Choose a simple matching model of a stable finite-dimensional system with the numerator of degree n_N, the denominator of degree n_D and the unit static gain governed by the transfer function $G_m(s)$. The model can be prescribed e.g. according to the desired dynamic behavior of the feedback loop. Its poles and zeros are referred as "prescribed" below.

Step 4. Set a part of the spectrum of poles via the number n_D of coefficients from the set $\mathbf{K}_D$ into the prescribed positions while the rest of denominator parameters are chosen arbitrarily. If these poles are dominant, initialize the counter of currently shifted poles as $n_{sp} = n_{sp,m} + n_{sp,opt} = n_D + 1$ where $n_{sp,m} = n_D$ and $n_{sp,opt} = 1$. If not, then $n_{sp} = n_{sp,m} + n_{sp,opt} = n_D$, $n_{sp,m} = n_D$, $n_{sp,opt} = 0$.

Step 5. Check that (4) holds for the number n_{sp} of the rightmost poles and $\mathbf{K}_D$. If not, go to Step 4 and reset the initial assignment; otherwise, shift the number $n_{sp,m}$ of the rightmost feedback system poles towards the prescribed locations (i.e., keep in the close proximity of them), e.g. using the QCSA, whereas the number $n_{sp,opt}$ of poles is pushed to the left. If necessary, increase $n_{sp,opt} \Rightarrow n_{sp}$. If $n_{sp} = r_D$ and/or the shifting is no more successful, go to Step 6.

Step 6. If all $n_{sp,m}$ poles are dominant, go to Step 7. Otherwise, select a suitable cost function $\Phi_P(\mathbf{K}_D)$ reflecting the distance of dominant poles of $G(s)$ from prescribed positions and the spectral abscissa. Minimize $\Phi_P(\mathbf{K}_D)$ starting with results from Step 5 (using e.g. an advanced iterative algorithm, [11–13]). Fix $\mathbf{K}_D$.

Step 7. Place a part of the spectrum of zeros of $G(s)$ using the number of n_N coefficients from the set $\mathbf{K}_{N\setminus D}$ into the prescribed positions and the remaining

parameters in $\mathbf{K}_{N\setminus D}$ are chosen arbitrarily. If these zeros are dominant, initialize the counter of currently shifted zeros as $n_{sz} = n_{sz,m} + n_{sz,opt} = n_N + 1$ where $n_{sz,m} = n_N$ and $n_{sz,opt} = 1$; otherwise, set $n_{sz} = n_{sz,m} + n_{sz,opt} = n_N$, $n_{sz,m} = n_N$, $n_{sz,opt} = 0$.

Step 8. Check that (4) holds for the number n_{sz} of the rightmost zeros of $G(s)$ and for current values of $\mathbf{K}_{N\setminus D}$. If it is approved, $n_{sz,m}$ zeros are to be incessantly moved to the prescribed positions whereas $n_{sz,opt}$ zeros are pushed to the left. If necessary, increase $n_{sz,opt} \Rightarrow n_{sz}$. If $n_{sz} = r_{N\setminus D}$ and/or the shifting is no more successful, go to Step 9.

Step 9. If all $n_{sz,m}$ zeros are dominant, the algorithm is finished. Otherwise, select a suitable cost function $\Phi_Z(\mathbf{K}_{N\setminus D})$ reflecting the distance of dominant zeros of $G(s)$ from prescribed positions and the spectral abscissa. Minimize $\Phi_Z(\mathbf{K}_{N\setminus D})$ with initial setting of $\mathbf{K}_{N\setminus D}$ obtained from Step 8.

Output. The vector of controller parameters $\mathbf{K} = \mathbf{K}_{N\setminus D} \cup \mathbf{K}_D$, positions of the rightmost poles and zeros and the spectral abscissae.

The above presented strategy of the PPSA places the feedback poles to the desired positions first, and consequently, transfer function numerator parameters not included in the numerator serve as tuning tool for inserting zeros to the desired loci. Thus, zeros are placed independently from poles by means of $\mathbf{K}_{N\setminus D}$. In both the cases, the rest of the spectrum is pushed to the left as far as possible to minimize the spectral abscissa. If this quasi-continuous shifting is not successful, a trade-off between the zeros/poles matching task and the spectral abscissa is optimized. Note that condition (8) stem from (5) and (7) while (6) always holds for this strategy.

In fact, the QCSA or a shifting technique presented in [14] enables to shift a conjugate pair of roots along the real axis using a single controller parameter, i.e. it is possible to write $n_{sp,m} + n_{sp,opt,R} + n_{sp,opt,C} \leq r_D$ and $n_{sz,m} + n_{sz,opt,R} + n_{sz,opt,C} \leq r_{N\setminus D}$ where a subscript R denotes real roots whereas C means complex conjugate pairs.

If $r_{D\setminus N} > 0$, it is possible to apply the strategy reversely, i.e. to set zeros first and, afterwards, to place poles. However, the presented variant prefers poles since they affect the system dynamics more significantly.

Let us present now another (a simpler) strategy combining both, the poles and zeros matching, under one procedure.

Algorithm 2 (*PPSA strategy 2: "Poles and Zeros Together"*)
Input. Closed-loop reference-to-output transfer function $G(s)$.

Step 1. Set $n_D = r_D - 1$, or just select $n_D < r_D$ as high as desirable.
Step 2. Verify that there exists a non-negative number n_N satisfying

$$0 \leq n_N < \min(n_D, r - n_D, r_N) \tag{9}$$

If (9) holds, fix n_N and go to Step 3; otherwise, set $n_D = n_D - 1$. If $r_D > n_D \geq \max\{r - n_D, r_N\}$, go to Step 2; contrariwise, a solution does not exist.

Step 3. Choose a simple model $G_m(s)$ of a stable finite-dimensional system with the numerator of degree n_N, the denominator of degree n_D, the unit static gain and prescribed (desired) zeros and poles.

Step 4. Set finite subsets of both the spectra, poles and zeros, via the number n_D of coefficients from the set $\mathbf{K}_D$ and by means the number n_N of coefficients from the set $\mathbf{K}_N$, respectively, into the prescribed positions of $G_m(s)$ while the rest of parameters from $\mathbf{K}$ are chosen arbitrarily. If all these poles are dominant, initialize the counter of currently shifted poles as $n_{sp} = n_{sp,m} + n_{sp,opt} = n_D + 1$ where $n_{sp,m} = n_D$ and $n_{sp,opt} = 1$; otherwise, $n_{sp} = n_{sp,m} + n_{sp,opt} = n_D$, $n_{sp,m} = n_D$, $n_{sp,opt} = 0$. Similarly for zeros, if they are the rightmost ones, set $n_{sz} = n_{sz,m} + n_{sz,opt} = n_N + 1$, $n_{sz,m} = n_N$, $n_{sz,opt} = 1$; in the contrary, $n_{sz} = n_{sz,m} + n_{sz,opt} = n_N$, $n_{sz,m} = n_N$, $n_{sz,opt} = 0$.

Step 5. Check that (5) holds for the number n_{sp} of the rightmost poles and $\mathbf{K}_D$, and for n_{sz} dominant zeros along with $\mathbf{K}_N$. If not, go to Step 4 and reset the initial assignment; otherwise, shift mutually the number $n_{sp,m}$ and $n_{sz,m}$ rightmost feedback system poles and zeros, respectively, towards the prescribed locations the number $n_{sp,opt}$ and $n_{sz,opt}$ of poles and zeros, respectively, is pushed to the left along the real axis. If necessary, increase $n_{sp,opt} \Rightarrow n_{sp}$ and/or $n_{sz,opt} \Rightarrow n_{sz}$. If $n_{sp,m} + n_{sp,opt,R} + n_{sp,opt,C} \leq r_D$ and $n_{sz,m} + n_{sz,opt,R} + n_{sz,opt,C} \leq r_N$ and $n_{sz,m} + n_{sp,m} + n_{sp,opt,C} + n_{sz,opt,R} + n_{sz,opt,C} \leq r$, or the shifting is no more successful, go to Step 6.

Step 6. If all $n_{sp,m}$ poles and $n_{sp,z}$ zeros are dominant, the procedure is finished. Otherwise, select a suitable cost function $\Phi(\mathbf{K})$ reflecting the distance of dominant poles and zeros of $G(s)$ from prescribed positions and spectral abscissae of both the spectra. Minimize $\Phi(\mathbf{K})$ starting with results from Step 5.

Output. The vector of controller parameters $\mathbf{K}$, positions of the rightmost poles and zeros and the spectral abscissae.

The methodology is useful in case $r_{N\setminus D} = 0$ (and/or $r_{D\setminus N} = 0$). Roughly speaking to summarize it, poles and zeros are moved simultaneously over a common set $\mathbf{K}$ of adjustable parameters, therefore their positions are not independent to each other.

A trade-off between Algorithm 1 and Algorithm 2 can be done by a procedure when only a subset $\mathbf{K}_{\overline{ND},D} \subset \mathbf{K}_{\overline{ND}}$ is dedicated to poles while a subset $\mathbf{K}_{\overline{ND},N} \subset \mathbf{K}_{\overline{ND}}$ is given to zeros to be modified, where $\mathbf{K}_{\overline{ND},D} \cap \mathbf{K}_{\overline{ND},N} = \emptyset$. Hence, these disjunctive sets provide a certain kind of independency.

The last conceivable strategy consists in the accurate setting of a part of the spectrum of zeros, which results in that some parameters from $\mathbf{K}_N$ are dependent to others, and consequently, find the optimal setting of independent parameters by strategies from Algorithm 1 or Algorithm 2. This idea, however, does not guarantee the dominancy of the placed zeros.

Due to the limited space, these two strategies mentioned above will be a topic of any of our future papers.

5 Illustrative Example

A very concise demonstrative example follows to provide the reader with the idea of control of TDS and the PPSA.

In [15] a mathematical model of a skater on the swaying bow, which represents an unstable TDS system, was introduced, and a corresponding controller designed in the R_{MS} ring was derived in [16]. The eventual reference-to-output transfer function reads

$$\begin{aligned} G(s) &= \frac{bb_Q(s)}{(s+m_0)^4 m_Q(s)} \exp(-(\tau+\vartheta)s) \\ b_Q(s) &= b\left(q_3 s^3 + q_2 s^2 + q_1 s + q_0\right)(s+m_0)^4 + p_0 m_0^4 s^2\left(s^2 - a\,\exp(-\vartheta s)\right) \\ m_Q(s) &= s^2\left(s^2 - a\,\exp(-\vartheta s)\right)\left(s^3 + p_2 s^2 + p_1 s + p_0\right) \\ &\quad + b\,\exp(-(\tau+\vartheta)s)\left(q_3 s^3 + q_2 s^2 + q_1 s + q_0\right) \end{aligned} \tag{10}$$

where delays τ, $\vartheta \geq 0$ stand for the skater's and servo latencies, respectively, b, a are real plant parameters. Note that the spectral assignment for the polynomial factor $(s+m_0)^4, m_0 > 0$ is trivial, then the goal is to find unknown parameters of $m_Q(s)$. To cancel the impact of the quadruple real pole $s_1 = -m_0$ to the feedback dynamics, it must hold that $m_0 \gg -\alpha(\mathbf{K})$ where $\alpha(\mathbf{K})$ expresses the spectral abscissa of the quasipolynomial factor. Hence, we have $\mathbf{K} = \{p_2, p_1, p_0, q_3, q_2, q_1, q_0\}$ with $\mathbf{K}_{\overline{ND}} = \{q_3, q_2, q_1, q_0, p_0\}$, $\mathbf{K}_{N\setminus D} = \emptyset$, $\mathbf{K}_{D\setminus N} = \{p_2, p_1\}$, $\mathbf{K}_N = \mathbf{K}_{\overline{ND}}$, $\mathbf{K}_D = \mathbf{K}$, that is $r = r_D = 7, r_{\overline{ND}} = r_N = 5$, $r_{N\setminus D} = 0, r_{D\setminus N} = 2$. Let us follow Algorithm 2 which is suitable in this case since $r_{N\setminus D} = 0$ and hence Algorithm 1 can not be used.

We attempt to set $n_D = 2$, then the conditions (9) reads $0 \leq n_N < 2$; therefore, let $n_D = 1$ and consider the model

$$G_m(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} = k\frac{s - z_1}{(s - s_1)(s - \bar{s}_1)} \tag{11}$$

According to the desired dynamic properties, we prescribe a zero $z_1 = -0.18$ and a complex conjugate pair of poles $s_1 = -0.1 + 0.2\mathrm{j}$. Since the initially place roots are not dominant with abscissas for poles and zeros as $\alpha_P(\mathbf{K}) = 0.8959$ and $\alpha_Z(\mathbf{K}) = -0.1373$, respectively, set $n_{sp} = 2, n_{sz} = 1$ and perform Steps 5-6 of the PPSA by means of the QCSA.

In Figs. 1 and 2 distances of the rightmost poles pair σ and the zero ζ from the prescribed ones are displayed, and the evolution of $\mathbf{K}$ during the quasi-continuous shifting is provided in Fig. 3.

Further, the SOMA is used to minimize the cost function $\Phi(\mathbf{K}) = |\sigma_1 - s_1| + |\zeta_1 - z_1| + 0.01\alpha_{r,P}(\mathbf{K}) + 0.01\alpha_{r,Z}(\mathbf{K})$ where $\alpha_{r,P}(\mathbf{K})$, $\alpha_{r,Z}(\mathbf{K})$ mean the spectral abscissa of the rest of poles and zeros, respectively. It is worth noting that the

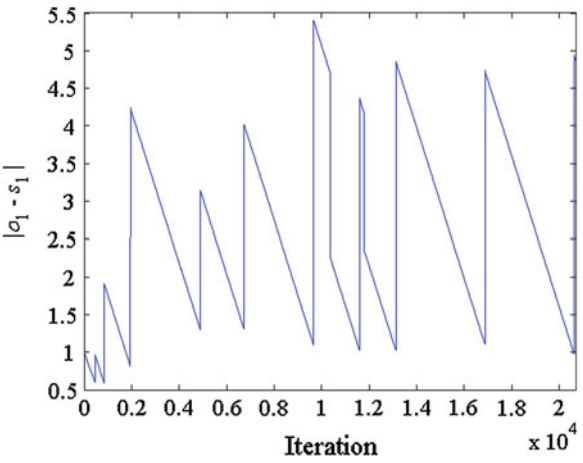


Fig. 1 Evolution of $|\sigma_1 - s_1|$ using the PPSA with QCSA

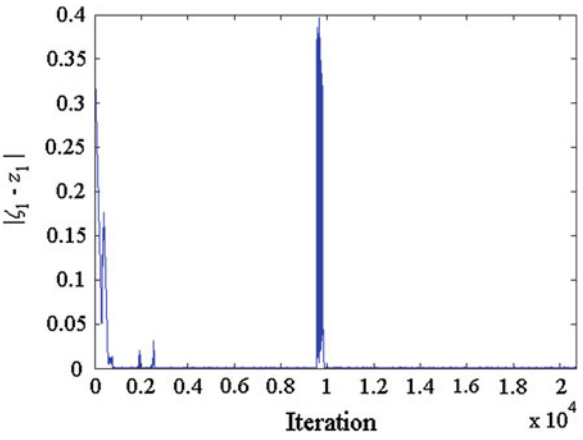


Fig. 2 Evolution of $|\zeta_1 - z_1|$ using the PPSA with QCSA

optimization yields only a slightly improvement giving the eventual spectra and the parameters set as in (12). However, final poles and zeros positions are quite far from the desired ones, which proves the fact about TDS that the desired spectrum can not be chosen arbitrarily in general.

$$\begin{aligned}\Omega_{P,opt} &= \{-0.1158 \pm 0.0674\text{j}, -0.1161 \pm 5.1163\text{j}, -0.1211 \pm 1.2103\text{j},\ldots\}\\ \Omega_{Z,opt} &= \{-0.1801, -0.2247 \pm 0.1032\text{j}, -0.7607, -2.817 \pm 8.1939\text{j},\ldots\}\\ \mathbf{K}_{opt} &= [5235.169,\ 9829.219,\ 1060.87,\ 78.2405,\ 30.9684,\ 1.763,\ 947.517]^T\end{aligned} \tag{12}$$

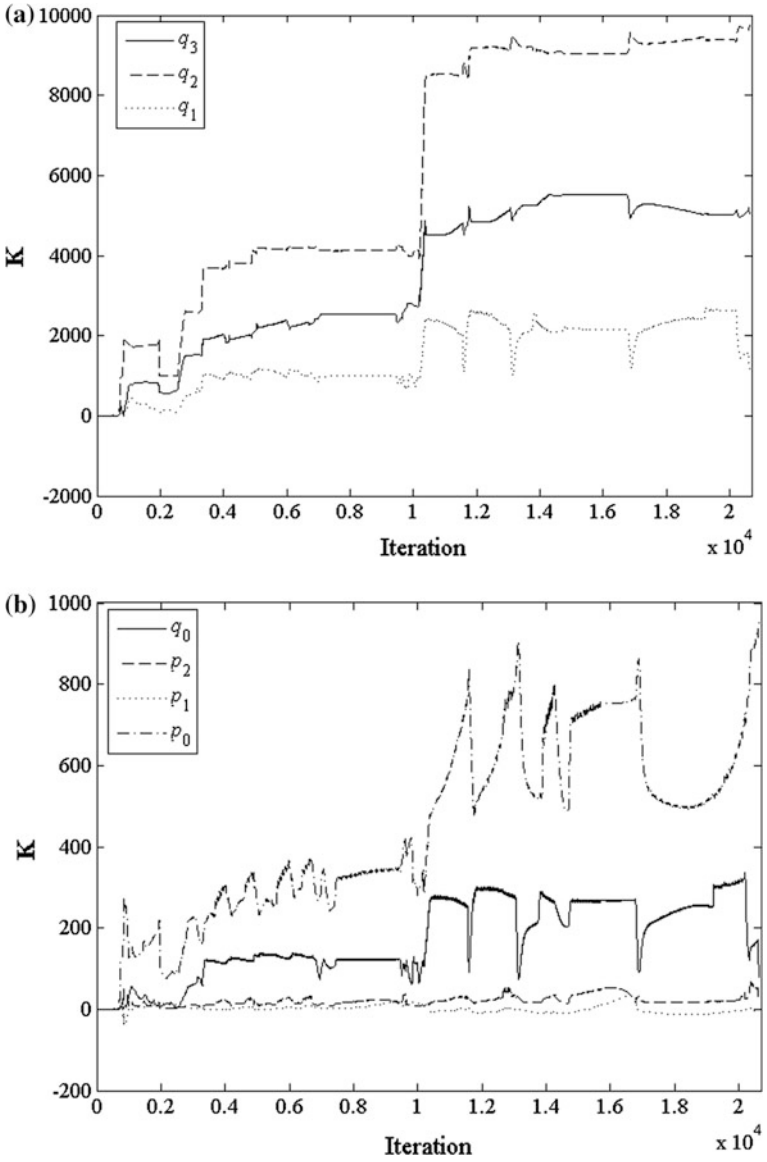


Fig. 3 Evolution of **K** using the PPSA with QCSA

6 Discussion

Let us now present some ideas how to modify, extend or improve the PPSA, regarding computation acceleration, shifting strategies, model selection etc.

Considering these aspects in the chronological order according to the running of Algorithm 1 or Algorithm 2, we can start with the selection of a finite-dimensional matching model. In the example above, it is supposed that the feedback dynamics is primarily given by positions of the rightmost poles and zeros where the model is found from the desired maximum overshoot, time-to-overshoot and the relative dumping. Naturally, other strategies how to prescribe the model (with corresponding roots) can be adopted. Moreover, the dominancy of the roots can be evaluated in a different way, e.g. in [14], the method based on the "weights" of modes of the impulse response was presented.

The initial shifting, convergence and the speed of the PPSA may be improved by the use of other "approaching" strategies, e.g. only roots of the same type (real, complex) are approaching to each other, or by thorough consideration that a complex conjugate pair means two separate roots instead of one (as it used here).

Last but not least another optimization procedures can be utilized in, e.g. the well-known and efficient NM algorithm [13] or some of many modern evolutionary or genetic algorithms. In fact, computationally the most time-consuming operation is the finding of the spectrum; hence the aim is to minimize the number of these spectral evaluations. For instance, it would be desirable to parallelize an existing spectrum-searching procedure and to utilize distributed computations on graphical cards, e.g. Compute Unified Device Architecture (CUDA) or Open Computing Language (OpenCL).

7 Conclusion

It is always difficult to tackle optimal or suboptimal control design or controller tuning for TDS. The presented paper has summarized and revised the basic principles of the PPSA which is based on quasi-continuous feedback poles and zeros shifting to the described dominant ones according to a selected finite-dimensional feedback model. The semi-finite result from the shifting has been then improved by an optimization procedure. Two possible PPSA strategies have been introduced and discussed, and the explanation has been supported by an illustrative example. In the future research, the other possible strategies will be analyzed and, moreover, the two presented ideas will be tested, compared and enhanced by tools discussed in this paper. Hence, the reader is kindly asked to participate on the future research, with the accent to provide us with the computational and programming support, to benchmark and verify the discussed ideas.

Acknowledgements The authors kindly appreciate the financial support which was provided by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

1. Hale, J.K., Verduyn Lunel S.M.: Introduction to functional differential equations. Appl. Math. Sci. **99**, (1993)
2. Kolmanovskii, V.B., Myshkis, A.: Introduction to the theory and applications of functional differential equations. Cluwer Academy, Dordrecht (1999)
3. Niculescu S.I.: Delay effects on stability. Lecture Notes in Control and Information Sciences, vol. 269. Springer, Berlin (2001)
4. Richard, J.P.: Time-delay systems: An overview of some recent advances and open problems. Automatica **39**, 1667–1694 (2003)
5. Zítek, P., Kučera, V.: Algebraic design of anisochronic controllers for time delay systems. Int J Control **76**, 1654–1665 (2003)
6. Pekař, L.: A ring for description and control of time-delay systems. In: WSEAS Transaction on Systems 11. Special Issue on Modelling, Identification, Stability, Control and Applications, pp. 571–585 (2012)
7. Michiels, W., Engelborghs, K., Vansevevant, P., Roose, D.: Continuous pole placement for delay equations. Automatica **38**, 747–761 (2002)
8. Michiels, W., Vyhlídal, T.: An eigenvalue based approach for the stabilization of linear time-delay systems of neutral type. Automatica **41**, 991–998 (2005)
9. Michiels, W., Vyhlídal, T., Zítek, P.: Control design for time-delay systems based on quasi-direct pole placement. J Process Control **20**, 337–343 (2010)
10. Pekař, L.: On a controller parameterization for infinite-dimensional feedback systems based on the desired overshoot. WSEAS Trans. Syst. **12**, 325–335 (2013)
11. Zelinka, I.: SOMA-self organizing migrating algorithm. In: Onwobolu, G.C., Babu, B.V. (eds.) New optimization techniques in engineering, pp. 167–217. Springer, Berlin (2004)
12. Vanbiervliet, T., Verheyden, K., Michiels, W., Vandewalle, S.: A nonsmooth optimization approach for the stabilization of time-delay systems. ESIAM: Control Optim. Calc. Var. **14**, 478–493 (2008)
13. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–313 (1965)
14. Vyhlídal, T.: analysis and synthesis of time delay system spectrum. Ph.D. thesis. Faculty of Mechanical Engineering, Czech Technical University in Prague, Prague (2003)
15. Zítek, P., Kučera, V., Vyhlídal, T.: Meromorphic observer-based pole assignment in time delay systems. Kybernetika **44**, 633–648 (2008)
16. Pekař, L., Prokop, R.: Algebraic optimal control in RMS ring: a case study. Int. J. Math. Comput. Simul. **7**, 59–68 (2013)

Logistic Warehouse Process Optimization Through Genetic Programming Algorithm

Jan Karasek, Radim Burget and Lukas Povoda

Abstract This paper introduces process planning, scheduling and optimization in warehouse environment. The leading companies of the logistics warehouse industry still do not use planning and scheduling by automatic computer methods. Processes are planned and scheduled by an operational manager with detailed knowledge of the problem, processed tasks and commodities, warehouse layout, performance of employees, parameters of equipment etc. This is a quantum of information to be handled by a human and it can be very time-consuming to plan every process and schedule the timetable. The manager is usually also influenced by stress conditions, especially by the time of holidays when everyone is making supplies and the performance of the whole warehouse management goes down. The main contribution of this work is (a) to introduce the novel automatic method for optimization based on the evolutionary method called genetic programming, (b) to give a description of a tested warehouse, and (c) to show the metrics for performance measurement and to give a results which states the baseline for further research.

Keywords Genetic programming · Logistics · Optimization · Scheduling

J. Karasek (✉) · R. Burget · L. Povoda
Faculty of Electrical Engineering and Communication, Department of Telecommunications, Brno University of Technology, Technicka 12, 616 00 Brno, Czech Republic
e-mail: karasekj@feec.vutbr.cz
URL: http://vutbr.cz/en/; http://splab.cz/en/

R. Burget
e-mail: burgetrm@feec.vutbr.cz

L. Povoda
e-mail: xpovod00@stud.feec.vutbr.cz

R. Silhavy et al. (eds.), *Modern Trends and Techniques in Computer Science*, Advances in Intelligent Systems and Computing 285, DOI: 10.1007/978-3-319-06740-7_3,

1 Introduction

The processes which come under operational level management competences such as planning and scheduling of daily routine tasks are important parts of everyday decision making. When these problems are handled in a big company with hundreds of employees, they become more and more complex. Furthermore, the complexity of the problem may arise from various sources, such as attributes related to performance of employees and attributes which describe the equipment, logistic warehouse and commodities layout, processed tasks, sub-tasks, and others. In a nutshell, the problem often becomes so complex that it is very difficult or nearly impossible to solve it only by skilled operational manager or any kind of mathematical programming method. By the time of writing this paper, the leading companies of the logistics and warehousing industry still have not used automated methods for process planning and scheduling.

The main aim of this paper lies (a) in introduction of the novel automatic method for process planning and scheduling based on genetic programming algorithms, (b) in description of a tested logistic warehouse, and (c) in introduction of the metrics for performance measurement and the initial results as reference points for further research with a more detailed view on one single example where the performance of the automated system has been proved to surpass the human operator. The main contribution of this paper is to help community dealing with logistics and warehouse optimization to set the baseline results for further development and joint research in the considered problem domain and to provide the metrics for performance measurement.

The rest of the paper is organized as follows. Section 2 deals with the work related to the problem considered in this paper. Similar problems and methods dealing with process planning and scheduling are discussed there. Section 3 describes in a nutshell a novel automatic method based on a genetic programming algorithm. Section 4 describes the standard layout of the logistic warehouse center which is also used as the reference point for further research. Section 5 describes a simple metrics for progress measurement, results of an operational manager, and results reached by a proposed system. Furthermore, this section also briefly describes an example where the proposed system has reached better results than the operational manager. The paper is concluded with Sect. 6.

2 Related Work

The problem of process planning and scheduling in logistic warehouses described in this paper deals with complex optimization of logistic warehouses and distribution centers. The problems addressed within the logistic warehouse optimization

deal mostly with the optimization of some part of the logistic warehouse, such as design of warehouse layout, design of receiving and shipping areas and design of other parts of the logistic warehouse. The products handled in the warehouse are also quite often subject to optimization—the optimization deals with product grouping, classing, and zoning. For more information see [1, 2].

There are two basic approaches to solving the warehouse optimization problem regarding process planning and scheduling. The first approach, commonly used when the problem is not so complex, are methods of mathematical programming [3]. The second approach uses heuristic methods. In the past, a lot of heuristic methods were used to solve the considered problem such as shifting bottleneck, dispatching rules, simulated annealing [4], particle swarm optimization [5] and/or tabu search [6]. The biggest group of algorithms used is Evolutionary Algorithms (EA) [7]. Genetic algorithms (GA), one of the biggest part of EAs, have demonstrated their potential for solving difficult optimization problems, and they have proved to be very efficient and adaptive solutions for complex problem solving.

3 Optimization Method

The GA showed potential for problem solving in difficult and complex situations which require a certain demand of adaptability and robustness. The problem of using GA is that in the case of this work the structure of chromosome is not given by any prescription. Therefore, the Genetic Programming (GP) as an optimization method has been chosen instead of GAs. The GP has demonstrated the same or even better potential than GAs and in addition to that the GP algorithm is able to design the structure of chromosome automatically. In recent years, GP algorithms have been successfully applied in many problem domains, where the algorithms were creating relatively complex problem solutions or whole automated systems. For instance, in the paper [8] the GP was successfully used for creation of the image detector that is able to detect relatively complex objects in noisy ultrasound data, in another work [9] the GP was used to select optimal features to classify emotions in textual data, and in [10] the GP driven by the context-free grammar was used to design a non-cryptographic hash.

Figure 1 shows the proposed GP algorithm which represents the computational core of the proposed automated system. The input of the algorithm consists of two basic parts. The first part of input is a buffer of all tasks waiting to be processed. The task in this context is perceived as one complete assignment given to the employee by the manager, e.g. a process of commodity storing can be considered a task. This task consists of several independent activities, so-called jobs. The concrete jobs for the mentioned task are (a) folding (the commodity from truck), (b) transfer (the commodity to the target coordinates in the warehouse), and (c) storing (the commodity to the rack). The second input is a set of employees who are able to process the tasks and their assigned equipment. Logical structure of the system is depicted in Fig. 2.

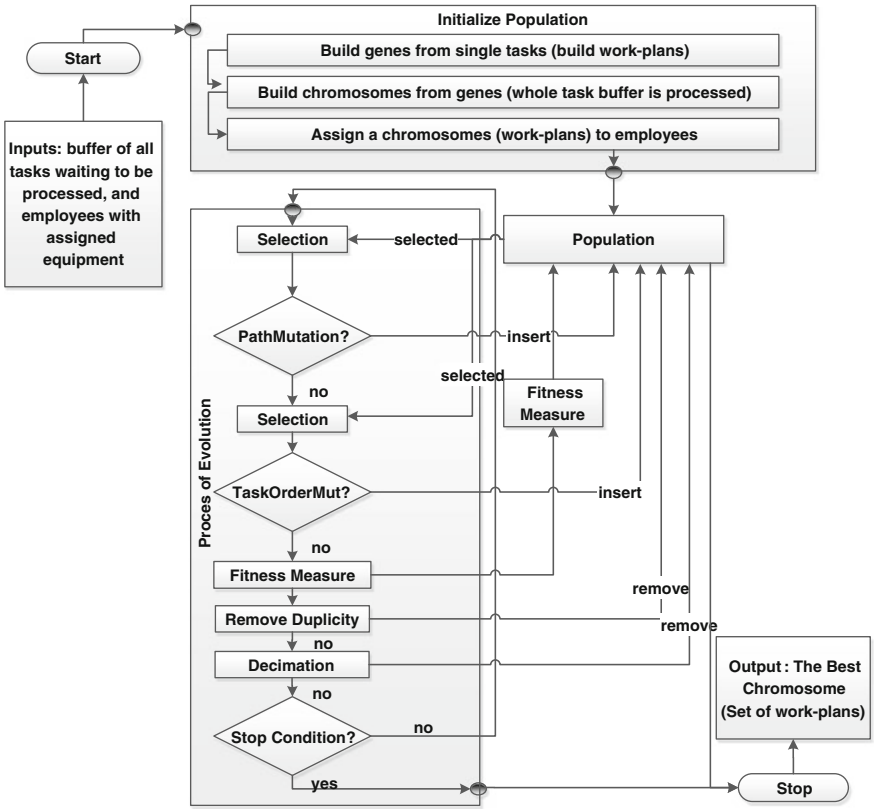


Fig. 1 Block scheme of genetic programming algorithm

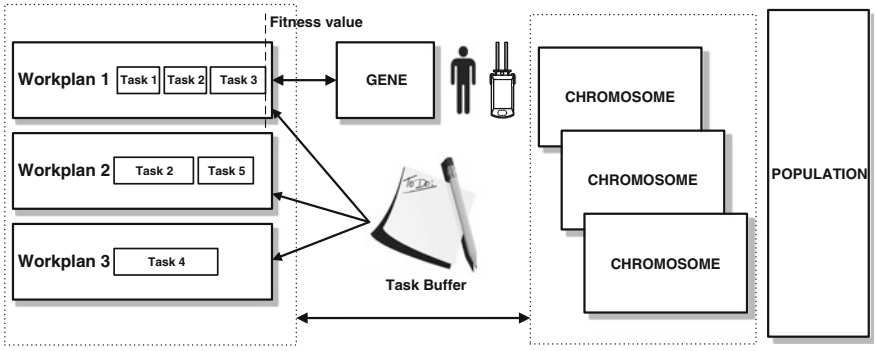


Fig. 2 Internal structure of algorithm core

Figure 2 represents inputs (task buffer, employees, and vehicles). The smallest logical structure of the proposed algorithm is a gene. A gene, in fact, is represented by a work-plan of employee. In the system there are as many work-plans as employees. The work-plans are filled automatically and randomly with tasks from the task buffer. All tasks waiting in the buffer to be processed have to be assigned to employee's work-plan. The work-plans form a chromosome and this is actually how the chromosomes are created. It is a completely random initialization process. Figure 2 also shows how the fitness function is calculated. The fitness value is determined as a finished time of the last task in a chromosome.

The evolution process is controlled by several parameters such as *population size*, *number of generation*, probabilities of evolutionary operators—a number of individuals who are copied to a new population *elitism*, probability of mutation operators application—*path mutation rate*, and *task order mutation rate*, a parameter which tells the evolution process to remove duplicities and the operator of decimation which holds the number of individuals under the prescribed level.

The whole evolution process is divided into several parts. Before the evolution process starts, the initial population has to be created as described in one of the previous paragraphs. When the initial population is created, the evolution process can run. The first step of evolution is to maintain the level of the best individuals in the population. This prevents the process from a decreasing tendency in the meaning of the best candidate solution. This process is called elitism, and a certain number of individuals R_e of the previous population is simply copied to a new population. In the case of this work $R_e = 1$. The second step is to apply genetic operators. First is the path mutation operator which is applied with the rate $R_{pm} = 30\,\%$. It means that 30 % of population will be mutated by path mutation. The second operator is the task order mutation which is applied with the rate $R_{to} = 30\,\%$. At this point the fitness value is calculated for all new individuals. When the genetic operators have been applied the algorithm can continue with duplicity removing and decimation of population, which secures the permanent number of individuals in the population. After this control processes the stop criterion is checked. If the stop condition is true, the evolution process is at the end and the best individual is stated as a solution. If the stop condition is false, the evolution process continues with next evolution step. The pseudo algorithm of the evolution process is described below.

```
method Output Evolution (PopulationSize, EvolutionSteps)
   var
     Double R_pm = 0.3;            %rate of path mutation
     Double R_to = 0.3;            %rate of task order mutations
   begin
     Population <- InitializePopulation(PS);
     EvaluatePopulation(Population);
     for (i = from 1 to ES)
       if (isPathMutationApplied) then
         n individuals <- Select(Population,R_pm);
         for (j = from 1 to n)
           Population <- PathMutation(get j from n);
         endfor
       endif
       if (isTaskOrderMutationApplied) then
         n individuals <- Select(Population,R_to);
         for (j = from 1 to n)
           Population <- TaskOrderMutation(get j from n);
         endfor
       endif
     endfor
     Return GetBestIndividual(Population);
   end
end.
```

3.1 Path Mutation

Path Mutation (Fig. 3a) is the first genetic operator designed for the purpose of this work. This kind of mutation is the simplest operator and its purpose is to change the path used to process a specific task (e.g. transportation of a pallet). The advantage of this operator lies in changing the path, especially when the collision of two vehicles is very probable or the lane between racks is under congestion.

The example in Fig. 3a shows how the operator works. R_{pm} percent of chromosomes is selected. First, the work-plan is selected at random in a chromosome, in this case it is work-plan no. 3. Second, the task is selected at random in the work-plan, in this case it is task no. 4, because it is only one in the work-plan. Then, the path mutation is applied and the transportation path is changed.

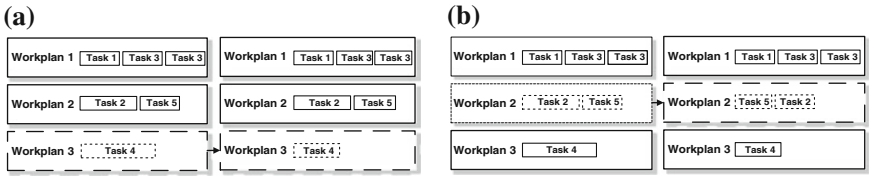


Fig. 3 Examples of path mutation operator and task order mutation operator

3.2 Task Order Mutation

The Task Order Mutation (Fig. 3b) is the second genetic operator designed. This operator is also quite simple, and its aim is to shuffle the tasks in the work-plan. This operator can show its advantage especially when the first task in the work-plan looks to be quite distant and it is more logic to process a closer task and then go further and further and process more distant tasks. The example in Fig. 3b shows how the operator works. R_{to} percent of chromosomes is selected. First, the work-plan is selected at random (work-plan no. 2). Then two tasks are randomly swapped, in this case there are only two tasks, so they are swapped.

4 Warehouse Layout

The reference warehouse described is based on a real-world situation. The warehouse is represented in Fig. 4 and it consists of several parts, such as: (a) trucks importing and exporting commodities; (b) receiving and shipping areas; (c) warehouse gates, in this example these gates are bi-directional (in/out traffic); (d) offices of employees; (e) hand pallet trucks (able to operate with shelves at level 0, level 0 represents the floor); (f) a low forklift truck (operates with shelves at levels 0–2); (g) a high forklift truck (operates with shelves at levels 0–9); $(h_1 - h_n)$ stationary racks in the warehouse, with shelves 0–9 for commodity storing, in this example n = 10; (i) a lane between racks and other warehouse space for commodity manipulation as receiving, packing, checking and others.

The warehouse is in fact described by three coordinates $\{x, y, z\}$. In the reference model, 10 columns of racks are in the warehouse. Each column has 19 racks standing next to each other and every rack has 10 shelves one above another to store pallets (0 indicates standing on the floor). The coordinate z which represents the level of shelves is not considered in this example, so the reference model of warehouse is represented as a two dimensional matrix. The warehouse space (i) is divided into x equal sized cells, where the cell size was chosen in view of the fact that it coincided with the largest dimensions of the truck (g). The speed is the most important parameter of vehicles and it is a central parameter of the time simulation when moving commodities through the warehouse. Time delay with

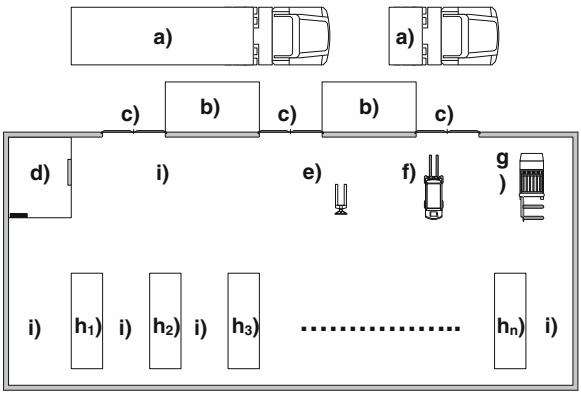


Fig. 4 Warehouse layout description

the imposition of the floor rack is now negligible. This implies that the speed of the vehicle has in this basic benchmark the most significant impact on the time of processing of the whole task buffer. The time of processing was chosen as the only fitness criterion in the work presented in this paper.

5 Results and Discussion

The benchmark was created as follows. First, the data from the warehousing company with which we cooperate under the terms of this project were obtained. Since the project is still in progress, the company must not be named because of license conditions of the contract. The data were selected according to the fully occupied time, which is the time before Christmas. During this period of time the operational manager who plans work for employees is influenced by stress conditions and is tired more than in any other season of the year. The data obtained were processed and simple scenarios were extracted. These simple scenarios put together two sets of benchmarks, 10 scenarios each. How these simple scenarios are processed is designed originally by operational manager as a reference point for results obtained by the proposed GP algorithm.

Each set of benchmarks consists of 10 simple warehouse scenarios. The first 10 scenarios contain from 2 to 4 employees with associated equipment. Special competences of employees are not considered in these scenarios. The first condition is that each employee is equipped only with a hand pallet truck. The speed of the hand pallet truck is set to 2 s.u. [speed units]. This type of truck was chosen with respect to that all kinds of warehouses contain this truck. The second condition is that each employee has his own simple task which has to be fulfilled from the very beginning to the end. And the third condition is that collisions of trucks, the performance of employees and the distance of employee and task are not taken into account when calculating the processing time. These scenarios are in fact only simplified scenarios from the second benchmark set.

Table 1 Results of benchmark set 1 and benchmark set 2

#	Manager	Optimization by GP algorithm			#	Manager	Optimization by GP algorithm		
		Op.1	Op.2	Op.1&2			Op.1	Op.2	Op.1&2
01	13.00	15.50 ↘	13.00 →	13.00 →	11	08.00	11.50 ↘	08.00 →	08.00 →
02	16.50	16.50 →	16.50 →	16.50 →	12	14.00	14.50 ↘	14.50 ↘	14.50 ↘
03	13.00	28.50 ↘	28.50 ↘	28.50 ↘	13	13.00	15.50 ↘	13.88 ↘	14.63 ↘
04	16.50	16.50 →	18.50 ↘	16.50 →	14	14.00	12.50 ↗	12.25 ↗	12.25 ↗
05	12.50	12.50 →	12.50 →	12.50 →	15	11.00	11.00 →	11.00 →	11.00 →
06	14.50	26.50 ↘	26.50 ↘	26.50 ↘	16	14.50	16.50 ↘	15.00 ↘	15.00 ↘
07	15.00	15.00 →	15.00 →	15.00 →	17	15.00	11.50 ↗	11.50 ↗	11.50 ↗
08	09.00	08.00 ↗	08.00 ↗	08.00 ↗	18	08.50	08.00 →	08.00 →	08.00 →
09	13.00	13.00 →	12.50 ↗	14.00 ↘	19	13.00	12.50 ↗	12.00 ↗	12.00 ↗
10	16.50	16.00 ↗	16.00 ↗	16.00 ↗	20	16.50	12.13 ↗	13.00 ↗	12.13 ↗

The second 10 scenarios consist of simple situations extracted from warehousing data, and at least one truck is a low forklift truck. The low forklift truck speed is set to 8 s.u. [speed units], so the scenarios are processed differently. The tasks in each scenario can be finished in different time, which implies that the collisions between the trucks can arise, e.g. Set 1 Scenario 1 was processed by employee with truck no. 1 (hand pallet), but now the truck no. 1 is low forklift truck. The speed units are used because the trucks have been standardized to two groups according to its type and employee performance.

The fitness function is calculated as follows. The final task of processing is given by the time when the last task in the scenario is finished. The time of processing each task is $T = S/R$, where T stands for the time of processing the task in t.u. [time units], S stands for the path length given in the number of cells in the warehouse which has to be exceeded when the truck is moving the commodity [cells], and R stands for the truck speed in s.u. [speed units].

5.1 Experimental Results

The first experimental results are shown in Table 1. While the arrow oriented to the upper row represents an increase in performance, the arrow oriented to the lower row represents a decrease in performance. The arrow oriented to the right shows the same performance level as that of the warehouse operational manager.

Table 1 left part, shows that the automated process of optimization reached comparable results to the operational manager. It is obvious that neither both the operators nor their combination reached the same or better results than the manager in all cases, but the first results can be considered comparable. Scenarios 1, 2, 4, 5, and 7 show the same level of performance as the manager, scenarios 8, 9, and 10 show better performance, and scenarios 3 and 6 show that the proposed algorithm failed in these cases. Table 1, right part provides little more interesting

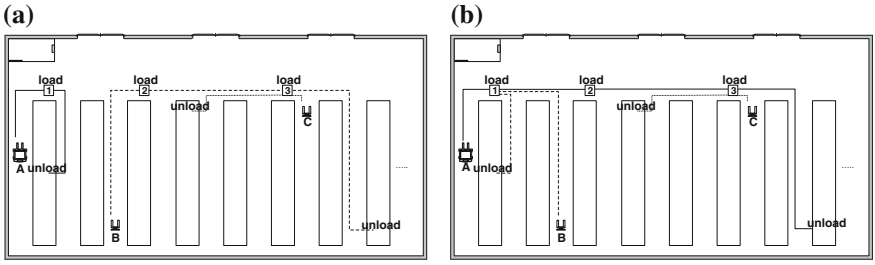


Fig. 5 Example—design by operational manager

results, because these scenarios also use a low forklift truck. Scenarios 11, 15, and 18 show the same level of performance as the manager, scenarios 14, 17, 19, and 20 show the performance increase, and scenarios 12, 13, and 16 show the performance decrease, but not that significantly as in the first set of benchmarks. Especially scenarios 3 and 6 from the first set of benchmarks will be subject to further testing and improving of the designed system.

5.2 Concrete Example of Scenario

The benchmark scenarios represent pieces of work-plans from historical data of the real logistic warehouse. One of these pieces is described in more details in the following text. The scenario is from the Christmas period when the operational manager is influenced by stress conditions and even a quite simple example can be designed in a non-optimal way. The scenario as it was designed by the operational manager is shown in Fig. 5a and the scenario designed by automated system is depicted in Fig 5b. The scenario contains three trucks (two hand trucks, one low forklift truck) associated to employees, and three pallets to store.

Figure 5a shows how the operational manager solve the situation in warehouse. The operational manager sends each employee (marked as A, B, and C) to manage one task (pallets are marked as 1, 2, and 3). Employee A processes pallet 1, paths of employee A are depicted by solid lines. Employee B processes pallet 2, paths of employee B are depicted by dashed lines, and employee C processes pallet 3, paths of this employee are depicted by dotted lines. The total time of this scenario is 16.50 t.u. (see Table 1 scenario 20). Total time is computed as a time when the last task (pallet in this case) has been processed.

Figure 5b shows how the proposed automated method solve the situation in warehouse. The resulting system distributed tasks also into three work-plans. In this case, the tasks of employee A and employee B have been switched. Employee A processes pallet 2, paths of employee A are depicted by solid lines. Employee B processes pallet 1, paths of employee B are depicted by dashed lines, and employee C processes pallet 3 with no change. Because the employee A has

associated truck with higher speed, it is better to process the more distant tasks. Therefore, the whole scenario is processed in 12.13 t.u. (see Table 1 column Op. 1&2). This looks like almost unimportant optimization, but it is only a small part of the huge and complex unit of work (e.g. a few minutes of the day in a warehouse). So, if the scenario like this is optimized just a little bit every few minutes, the total cost and the time of processing can be reduced.

6 Conclusion

In this paper a novel job-shop scheduling problem has been introduced, including the benchmark definition and baseline results reached by the genetic programming algorithms with support of the operational manager domain knowledge. This work was encouraged by a demand from the logistic distribution and warehousing industry and was created with the help and intensive consultation with experts involved in the logistic warehouse process optimization for many years. In total, 20 benchmark scenarios were created, each of them has multiple tasks with different optimization problems. The aim of this work is to set a common platform for collaborative research of the logistic warehouse process scheduling and to help solve the problem automatically or to simplify decision process.

Acknowledgments This research work is funded by projects SIX CZ.1.05/2.1.00/03.0072, MPO FR-TI1/444, and project FEKT-S-11-17.

References

1. de Koster, R., Le-Duc, T., Roodbergen, K.J.: Design and control of warehouse order picking: a literature review. Eur. J. Oper. Res. **182**(2), 481–501 (2007)
2. Geraldes, C.A.S., Sameiro, M., Carvalho, F., Pereira, G.A.B.: A warehouse design decision model case study. In: IEEE International Engineering Management Conference, IEMC Europe, pp. 397–401 (2008)
3. Būlbūl, K., Kaminsky, P.: A linear programming-based method for job shop scheduling. J. Sched. **16**(2), 161–183 (2013)
4. van Laarhoven, P.J.M., Aarts, E.H.L., Lenstra, J.K.: Job shop scheduling by simulated annealing. Oper. Res. **40**(1), 113–125 (1992)
5. Tasgetiren, M.F., Liang, Y-Ch., Sevkli, M., Gencyilmaz, G.: A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. Eur. J. Oper. Res. **177**(3), 1930–1947 (2007)
6. Nowicki, E., Smutnicki, C.: An advanced tabu search algorithm for the job shop problem. J. Sched. **8**(2), 145–159 (2005)
7. Kōskolan, M., Keha, A.B.: Using genetic algorithm for single-machine bicriteria scheduling problems. Eur. J. Oper. Res. **145**(3), 543–556 (2003)
8. Benes, R., Karasek, J., Burget, R., Riha, K.: Automatically designed machine vision system for the localization of CCA transverse section in ultrasound images. Comput. Methods Programs Biomed. **109**(1), 92–103 (2013)

9. Burget, R., Karasek, J., Smekal, Z.: Recognition of emotions in Czech newspaper headlines. Radioengineering **20**(1), 39–47 (2011)
10. Karasek, J., Burget, R., Morsky, O.: Towards an automatic design of non-cryptographic hash function. In: 34th International Conference on Telecommunications and Signal Processing, pp. 19–23 (2011)

A New Approach to Solve the Software Project Scheduling Problem Based on Max–Min Ant System

Broderick Crawford, Ricardo Soto, Franklin Johnson, Eric Monfroy and Fernando Paredes

Abstract This paper presents a new approach to solve the Software Project Scheduling Problem. This problem is NP-hard and consists in finding a worker-task schedule that minimizes cost and duration for the whole project, so that task precedence and resource constraints are satisfied. Such a problem is solved with an Ant Colony Optimization algorithm by using the Max–Min Ant System and the Hyper-Cube framework. We illustrate experimental results and compare with other techniques demonstrating the feasibility and robustness of the approach, while reaching competitive solutions.

Keywords Software engineering · Software project scheduling problem · Project management · Ant colony optimization · Max–Min ant system

B. Crawford · R. Soto · F. Johnson (✉)
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: franklin.johnson.p@mail.ucv.cl; franklin.johnson@upla.cl

R. Soto
e-mail: ricardo.soto@ucv.cl

B. Crawford
e-mail: broderick.crawford@ucv.cl

F. Johnson
Universidad de Playa Ancha, Valparaíso, Chile

B. Crawford
Universidad Finis Terrae, Santiago, Chile

R. Soto
Universidad Autónoma de Chile, Temuco, Chile

E. Monfroy
CNRS, LINA, Université de Nantes, Nantes, France
e-mail: eric.monfroy@univ-nantes.fr

F. Paredes
Universidad Diego Portales, Santiago, Chile
e-mail: fernando.paredes@udp.cl

R. Silhavy et al. (eds.), *Modern Trends and Techniques in Computer Science*,
Advances in Intelligent Systems and Computing 285, DOI: 10.1007/978-3-319-06740-7_4,

1 Introduction

In this paper, we present a new approach to the Software Project Scheduling Problem (SPSP), which consists in finding a worker-task schedule that minimizes cost and duration for the whole project, so that task precedence and resource constraints are satisfied [2]. This problem is known to be NP-hard, being difficult to solve it by a complete search method in a limited amount of time. We propose then to solve the problem with Ant Colony Optimization (ACO) [10], in particular with the Max–Min Ant System and the Hyper-Cube framework [12, 13]. ACO is a probabilistic method, inspired from the behavior of real ant colonies searching for food. Ants initially explore at random, but once food is found they return to the colony leaving a pheromone trail, which can therefore be followed by other ants to reach the food quickly. The Max–Min Ant System is a popular variation of the classic ACO algorithm which we tune with the ACO Hyper-Cube (ACO-HC). This combination allows one to automatically handle the limits of pheromone values by a modification in the update pheromone rule, resulting in a more robust and easier algorithm to implement [14]. We illustrate encouraging experimental results where our approach noticeably competes with other well-known optimization methods reported in the literature.

This paper is organized as follows. In Sect. 2 presents the definition of SPSP, in Sect. 3 presents a description ACO-HC for SPSP. In its subsection presents the construction graph, pheromone update rules, and the heuristic information. In Sect. 4 presents the experimental results, The conclusions are outlined in Sect. 5.

2 The Software Project Scheduling Problem

The software project scheduling problem is one of the most common problems in managing software engineering projects [16]. It consists in finding a worker-task schedule for a software project [3, 18]. The most important resources involved in SPSP are; the tasks, which is the job needed for completing the project, the employees who work in the tasks, and finally the skills.

Description of Skills: As mentioned above, the skills are the abilities required for completing the tasks, and the employees have all or some of these abilities. These skills can be for example: design expertise, programming expert, leadership, GUI expert. The set of all skills associated with software project is defined as $S = \{s_1, \ldots, s_{|S|}\}$, where $|S|$ is the number of skills.

Description of Tasks: The tasks are all necessary activities for accomplishing the software project. These activities are for example, analysis, component design, programming, testing. The software project is a sequence of tasks with different precedence among them. Generally, we can use a graph called task-precedence-graph (TPG) to represent the precedence of these tasks [5]. This is a non-cyclic directed graph denoted as $G(V, E)$. The set of tasks is represented by

$V = \{t_1, t_2, \ldots, t_{|T|}\}$. The precedence relation of tasks is represented by a set of edges E. An edge $(t_i, t_j) \in E$, means t_i is a direct predecessor task t_j. Consequently, the set of tasks necessary for the project is defined as $T = \{t_1, \ldots, t_{|T|}\}$, where $|T|$ is the maximum number of tasks. Each task has two attributes: t_j^{sk} is a set of skills for the task j. It is a subset of S and corresponds to all necessary skills to complete a task j, t_j^{eff} is a real number and represents the workload of the task j.

Description of Employees: The problem is to create a worker-task schedule, where employees are assigned to suitable tasks. The set of employees is defined as $EMP = \{e_1, \ldots, e_{|E|}\}$, where $|E|$ is the number of employees working on the project. Each employee has tree attributes: e_i^{sk} is a set of skills of employee i. $e_i^{sk} \subseteq S$, $e_i^{\max d}$ is the maximum degree of work. it is the ratio between hours for the project and the workday. $e_i^{\max d} \in [0, 1]$, if $e_i^{\max d} = 1$ the employee has total dedication to the project, if the employee has a $e_i^{\max}$ less that one, in this case is a part-time job, e_i^{rem} is the monthly remuneration of employee i.

Model Description: The SPSP solution can be represented as a matrix $M = [E \times T]$. The size $|E| \times |T|$ is the dimension of matrix determined by the number of employees and the number of tasks. The elements of the matrix $m_{ij} \in [0, 1]$, correspond to real numbers, which represent the degree of dedication of employee i to task j. If $m_{ij} = 0$, the employee i is not assigned to task j. If $m_{ij} = 1$, the employee i works all day in task j.

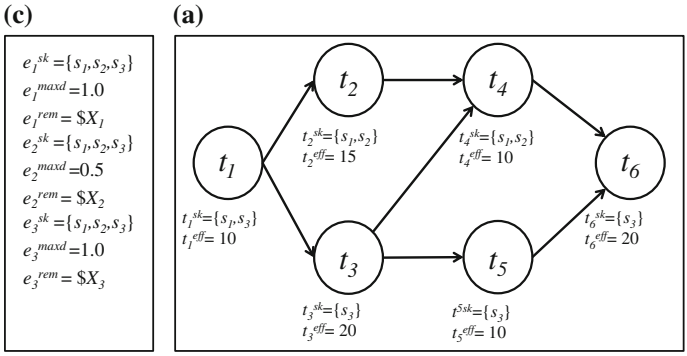
The solutions generated in this matrix M are feasible if they meet the following constraints. Firstly, all tasks are assigned at least one employee as is presented in Eq. 1. Secondly, the employees assigned to the task j have all the necessary skills to carry out the task, it is presented in Eq. 2.

$$\sum_{i=1}^{|E|} m_{ij} > 0 \; \forall j \in \{1, \ldots, T\} \tag{1}$$

$$t_j^{sk} \subseteq \bigcup_{i \mid m_{ij} > 0} e_i^{sk} \; \forall j \in \{1, \ldots, T\} \tag{2}$$

We represent in Fig. 1a an example for the precedence tasks TPG and their necessary skills t^{sk} and effort t^{eff}. For the presented example we have a set of employees $EMP = \{e_1, e_2, e_3\}$, and each one of these have a set of skills, maximum degree of dedication, and remuneration. A solution for problem represented in Fig. 1a, c is depicted in Fig. 1b.

First, it should be evaluated the feasibility of the solution, then using the duration of all tasks and cost of the project, we appraise the quality of the solution. We compute the length time for each task as t_j^{len}, $j \in \{1, \ldots, |T|\}$, for this we use matrix M and t_j^{eff} according the following formula:



(b)

	t_1	t_2	t_3	t_4	t_5	t_6
e_1	1.00	0.50	0.00	0.25	0.00	0.00
e_2	0.25	0.50	1.00	0.50	0.00	0.25
e_3	0.50	1.00	0.00	0.50	1.00	0.25

Fig. 1 **a** Task precedence graph TPG. **b** A possible solution for matrix M. **c** Employees information

$$t_j^{len} = \frac{t_j^{eff}}{\sum_{i=1}^{|E|} m_{ij}} \tag{3}$$

Now we can obtain the initialization time t_j^{init} and the termination time t_j^{term} for task j. To calculate these values, we use the precedence relationships, that is described as TPG $G(V, E)$. We must consider tasks without precedence, in this case the initialization time $t_j^{init} = 0$. To calculate the initialization time of tasks with precedence firstly we must calculate the termination time for all previous tasks. In this case t_j^{init} is defined as $t_j^{init} = \max\{t_l^{term} \| (t_l, t_j) \in E\}$, the termination time is $t_j^{term} = t_j^{init} + t_j^{len}$.

Now we have the initialization time t_j^{init}, the termination time t_j^{term} and the duration t_j^{len} for task j with $j = \{1, \ldots, |T|\}$, that means we can generate a Gantt chart. For calculating the total duration of the project, we use the TPG information. To this end, we just need the termination time of last task. We can calculate it as $p^{len} = \max\{t_l^{term} \| \forall l \neq j(t_j, t_l)\}$. For calculating the cost of the whole project, we need firstly to compute each cost associate to task us t_j^{cos} with $j \in \{1, \ldots, |T|\}$, and then the total cost p^{cos} is the sum of costs according to the following formulas:

$$t_j^{cos} = \sum_{i=1}^{E} e_i^{rem} m_{ij} t_j^{len} \tag{4}$$

$$p^{cos} = \sum_{j=1}^{T} t_j^{cos} \tag{5}$$

The target is to minimize the total duration p^{len} and the total cost p^{cos}. Therefore a fitness function is used, where w^{cos} and w^{len} represent the importance of p^{cos} and p^{len}. Then, the fitness function to minimize is given by $f(x) = (w^{cos} p^{cos} + w^{len} p^{len})$.

An element not considered is the overtime work that may increase the cost and duration associated to a task, consequently increase p^{cos} and p^{len} of the software project. We define the overtime work as e_i^{overw} as all work the employee i less $e_i^{\max d}$ at particular time.

To obtain the project overwork p^{overw}, we must consider all employees. We can use the following formula:

$$p^{overw} = \sum_{i=1}^{|E|} e_i^{overw} \tag{6}$$

With all variables required, we can determine if the solution is feasible. In this case, it is feasible when the solution can complete all tasks, and there is no overwork, that means the $p^{overw} = 0$.

3 ACO Hyper-Cube Framework for Schedule Software Project

The ACO algorithm exploits an optimization mechanism for solving discrete optimization problems in various engineering domains [11]. This framework implements ACO algorithms, which explicitly defines the multidimensional space for the pheromone values as the convex hull of the set of 0–1 coded feasible solutions of the combinatorial optimization problem under consideration. The Hyper-Cube was proposed by Blum and Dorigo [4, 8]. This framework makes a modification in the pheromone update rule, which is obtained through a normalization of the original pheromone update equation. This allows a more robust and autonomous handling of pheromone values to improve the exploration of the solution space.

To adapt ACO to SPSP using the Hyper-Cube Framework [13] must establish an appropriate construction graph and define the use of pheromone for Max–Min Ant System (MMAS) as well as heuristic information associated with the specified problem [1, 6]. The Max–Min Ant System is an Ant Colony Optimization algorithm [17], which establishes a minimum and maximum value for the pheromone, and provides that only the best ant can update the pheromone trail.

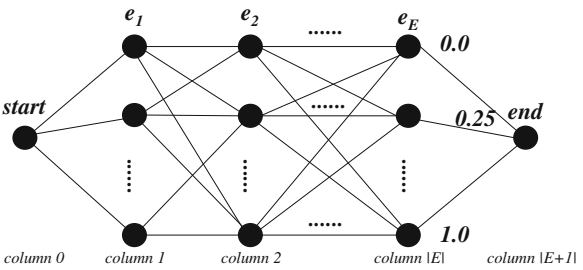


Fig. 2 Construction graph is a matrix $CG = [den \times E]$ with $mind = 0.25$ for a task

Construction Graph: For constructing a solution the ants travel through the construction graph. The ants start from an initial node and then select the nodes according to a probability function. This function is given by the pheromone and heuristic information of the problem, their relative influence is given by α and β respectively.

The proposed construction graph represents the association of employee and their dedication to a task. This representation is constructed for each task in the TGP; it is split into a graph with node and edge. The construction graph consists of each employee and their ratio of dedication contributions for the task. It is defined as *den*, this variable is density of nodes and it is defined as $den = \frac{1}{mind} + 1$, where *mind* is the lowest degree of dedication to a task. This structure is presented in Fig. 2. The employees dedication to a task can be 0 or integer multiple of *mind*.

The ants travel to the start node to the end node choosing edges from the column 1 to column |*E*| without returning. The ants choose only one node per each column. When the ant completes a tour, the dedication distribution of employees to the task is complete. To calculate the dedication of the employee *i* to the task, we just need the node *j*, with column *i* and the calculation is *j* * *mind*. These activities of ants must be done for each task in SPSP model.

The ants travel through the construction graph selecting ways of probabilistically way, using the following function:

$$p_{ij}^{t} = \frac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l=0}^{den}\left[\tau_{il}\right]^{\alpha}\left[\eta_{il}\right]^{\beta}}, \quad j \in \{1, \ldots, den\} \tag{7}$$

where τ_{ij} is the pheromone and η_{ij} is the heuristic information of the problem on the path between node *i* to *j* in the graph *CG* for de *t* task. α and β are two fixed parameters, which are used to determine the pheromone and heuristics influences.

Pheromone Update: In the hyper-cube framework the pheromone trails are forced to stay in the interval [0, 1] and the Max–Min Ant System the pheromone stay in the interval $[\tau_{\min}, \tau_{\max}]$. To adapt the Hyper-Cube framework to MMAS we define a $\tau_{\min} = 0$ and, $\tau_{\max} = 1$. In MMAS only one single ant is used to update the pheromone trails after each iteration, that ant can be the *iteration—best* ant or *global—best* ant. In this case we used only *iteration—best* ant.

We computationally represent the evaporation of pheromone and in addition the amount of pheromone in the ant path through the graph once a tour is completed using the following formula:

$$\tau_{ij} = \rho\tau_{ij} + (1 - \rho)\Delta\tau^{upd}, \tag{8}$$

where ρ is a rate of evaporation $\rho \in [0, 1]$. If ρ is high, the new pheromone value is less influenced by $\Delta\tau^{upd}$, but much influenced by the previous pheromone value, vice versa. And $\Delta\tau^{upd}$ it is associated with quality of the current solution of *upd* ant. *upd* ant is *iteration—best* ant [13]. We can use an updating pheromone strategy considering the duration, cost, and overwork of the project as follows:

$$\Delta\tau^{upd} = (w^{cos}p^{cos} + w^{len}p^{len} + w^{overw}p^{overw})^{-1}, \tag{9}$$

where w^{cos}, w^{len}, and w^{overw} are values that weight the importance of p^{cos}, p^{len}, and p^{overw} of the software project. The Δupd is the amount of pheromone added based on the quality of solution generated by *upd* ant.

Heuristic Information: We need to represent the heuristic information, that information is used to enhance the search ability of ants. The ants need to find the proper nodes using the problem information. The ants travel for a matrix m_{ij} as the node at column i and row j. To obtain the dedication of an employee e_i to a task which has been selected, we must calculate $j * minded$. We use as heuristic information the dedication of employee e_i to other task. If an employee works more in the previous tasks, that employee will has less dedication available for subsequent tasks. Consequently, the employee has less probability to be assigned to the current task. The heuristic information $h[i]$ to select node i for task t_k can be calculated as follows:

$$h[i] = \begin{cases} \frac{tmp[den-i-1]}{sum}, & \text{if} \quad allocD[k] > 0.5, \\ \frac{tmp[i]}{sum}, & else \end{cases} \tag{10}$$

where $tmp = \{1, \ldots, den\}$ is an array of temporal values generated with summation of possible dedication and allocated dedications for employees ($allocD[k]$, for k employee). *sum* is the summation of all values of the array *tmp*.

ACO-HC algorithm: The algorithm firstly reads the problem instance. That instance provides all the necessary data to generate the SPSP, such as number of tasks, and their required skills, number of employees, task precedence information to generate the TGP, the set of skills of every employee, and their remunerations. Then, we have to split operation to the task and then using the ACO-HC to generate solutions. To determine the quality of solutions, we use the fitness function. That function minimizes the cost and duration for whole project.

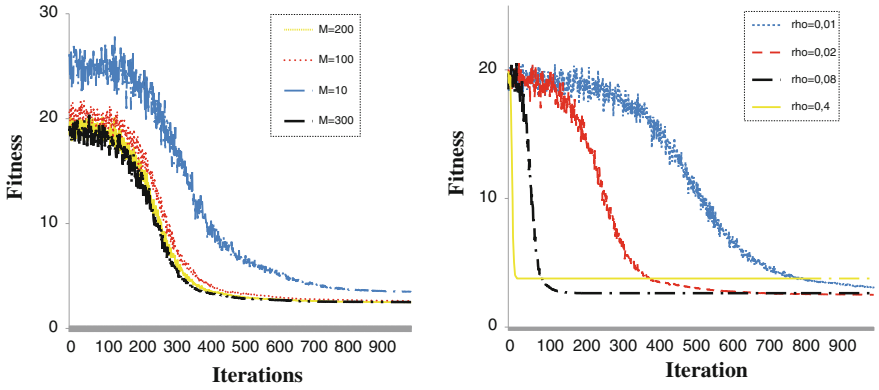


Fig. 3 Avg fitness using different values for m and ρ

Algorithm 1 ACO-HC for SPSP Algorithm

```
initialize feromone values τ_max and τ_min
repeat
    for g = 1 to G do
        for a = 1 to M do
            for t = 1 to T do
                the a ant travel on the matrix
            end for
            compute the feasibility of a ant and fitness of the solution
        end for
        select the best solution
        update pheromone values
    end for
until (iterations or time) is complete
```

4 Experimental Results

In this section we present the experimental results. The algorithm was ran 10 trials for each instance and we report the average value from those 10 trials. For the experiments, we use a random instances created by a generator.[1] The instances are labelled as <tasknumber> t<employeesnumber> e<skillsnumber> s. To compare the different results we use the *hit rate*: feasibles solution in 10 runs, *cost*: average cost of feasible solutions in 10 runs, *duration*: average duration of feasible solutions in 10 runs, and *fitness*: average fitness of feasible solutions in 10 runs.

[1] http://tracer.lcc.uma.es/problems/psp/generator.html

Table 1 Comparison with other techniques

Instance	Algorithms	Hit rate	Fitness	p^{len}	p^{cos}
10t5e5s	ACO-HC	100	**3.136531**	23	836531
	ACS	100	3.42039	22	1220390
	GA	95	3.52431	23	1224310
10t10e5s	ACO-HC	100	**2.134546**	13	834546
	ACS	100	2.55633	14	1156330
	GA	97	2.81331	16	1213310
20t10e5s	ACO-HC	30	6.741111	45	2241111
	ACS	67	6.36584	39	2465840
	GA	19	**6.28734**	38	2487340
10t5e10s	ACO-HC	100	**2.90579**	21	805790
	ACS	100	3.39316	22	1193160
	GA	90	3.51354	23	1213540
10t10e10s	ACO-HC	100	2.612948	17	912948
	ACS	100	2.62331	14	1223310
	GA	100	**2.51203**	13	1212030
20t10e10s	ACO-HC	50	6.249782	42	2049782
	ACS	65	6.31455	38	2514550
	GA	71	**6.19601**	37	2496010

4.1 Parameter Tunning and Convergence Analysis

It is known that the ACO results may vary depending on the parameters used. For this reason is important to make a simple but significant test. In this case we conducted a series of experiments to find the best parameter values for number of ants m and evaporation rate ρ. We used 10t10e10 s instance with $mind = 0.25$ and some parameters are constant which are $\alpha = 1$, $\beta = 2$ number of iterations $N_{it} = 1{,}000$ the results is presented in Fig. 3.

We can observe (left chart in Fig. 3) that the best fitness (low fitness) is obtained with $m = 200$ and $m = 300$ and the worst fitness is obtained with $m = 100$ and $m = 10$. To obtain the best results in shortest time, we used $m = 200$. In the right chart in Fig. 3 we observe the different fitness obtained with different values for ρ. For this parameter the best fitness is obtained with $\rho = 0.02$. When $\rho = 0.01$ the fitness converge too slow, if the ρ is very large (0.08 or 0.4) the fitness converges very fast to a suboptimal value, with $\rho = 0.02$ converges smoothly to fitness better value.

In order to analyse the convergence to feasible solutions of the algorithm, we can observe in Fig. 3 how to obtain better solutions from iteration 300 and converges slowly to a best solution.