2018 International Conference on Mathematical Applications

July 9-12, 2018

Madeira-Portugal

m·iti
Madeira Interactive
Technologies Institute

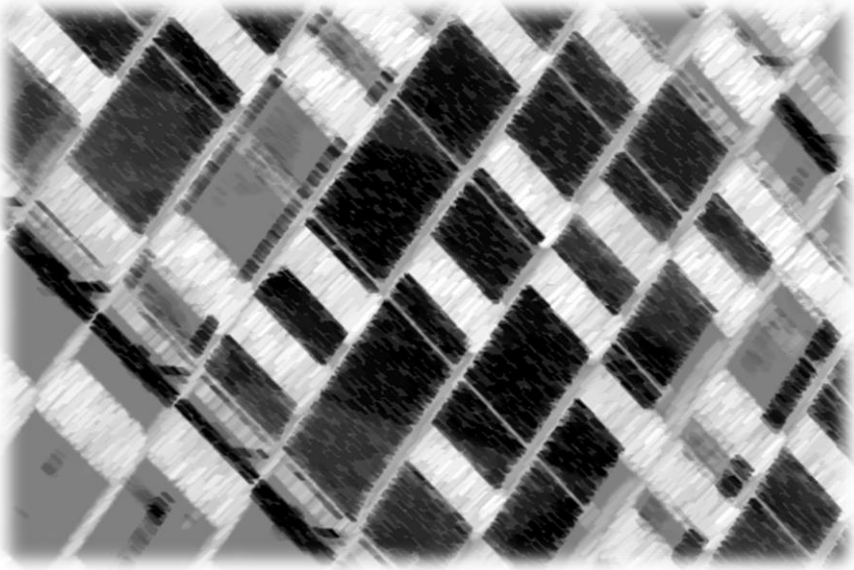
UNIVERSIDADE da MADEIRA

IKnowD

Editors:
Fernando Morgado Dias
Filipe Quintal

ISSN 2184-3945

International Conference on Mathematical Applications



PROCEEDINGS

July 9-12, 2018
Madeira-Portugal

Organized by

Madeira Interactive Technologies Institute

m·iti
Madeira Interactive
Technologies Institute

Universidade da Madeira

UNIVERSIDADE da MADEIRA

Institute of Knowledge and Development

IKnowD

General Information

Official Language

The official language of the conference is English. All presentations, including discussions and submissions, must be made in the official language. No translation will be provided.

Proceedings

Each accepted paper reaching the secretariat in time will be published in the proceedings.

Opening Hours of the Registration Desk

July 9, Monday: 08:00 – 10:00
July 10, Monday: 08:00 – 10:00
July 11, Tuesday: 8:00 – 17:30
July 12, Wednesday: 8:00 – 14:00

Presentation

Presentations can be done using a data projector. All authors are kindly asked to take their presentations in a flashdrive. All conference rooms are supplied with data projector, PC and internet.

Smoking

Please, be so kind to your lungs and your colleagues by not smoking during the sessions and social events.

WELCOME MESSAGE FROM THE GENERAL CHAIR

On behalf of the Institute of Knowledge and Development, it is our pleasure to welcome you to the International Conference on Mathematical Applications 2018 (ICMA18) in Funchal (Portugal).

Mathematics is at the core of Sciences and Engineering and is still the key to modeling and characterizing systems and processes, whether natural or artificial.

At ICMA we are looking for cross fertilization between areas that need Mathematics tools and that can provide the applications for different theoretical approaches.

With this conference the organization expects to contribute to this development and foster the integration of Mathematics with application areas.

The Organization would like to acknowledge the efforts of all the people and agents which have collaborated in the event.

The Chairmans:

Petrica Pop Sitar,
Technical University of Cluj-Napoca, Romania
Morgado-Dias,
University of Madeira and Madeira Interactive Technology Institute

Committees

GENERAL CHAIRS

Fernando Morgado-Dias, UMA – M-ITI, Portugal
Petrica Pop Sitar, Technical University of Cluj-Napoca, Romania

SCIENTIFIC PROGRAM COMMITTEE

Abdelhameed , Nagy Benha University
Alberto , Proenca University of Minho
Alberto , Cardoso DEI/CISUC – University of Coimbra
Ana , Moura Santos Instituto Superior Técnico. Universidade de Lisboa
Anca , Andreica Babes-Bolyai University
André , Souto University of Porto
António Mário , Almeida University of Minho
Antonio , Antunes University of Coimbra
Carlos , Barreira Faculdade de Psicologia e de Ciências da Educação Universidade de Coimbra
Cruz , Vargas De León Escuela Superior de Medicina, Instituto Politécnico Nacional
Daniela , Zaharie West University of Timisoara
Dragan , Simic University of Novi Sad, Faculty of Technical Sciences
Eduardo , Fermé Universidade da Madeira
Eugénio A.M. , Rocha University of Aveiro
Fernanda , Costa University of Minho
Fernando , Morgado-Dias UMa-M-ITI
Fernando A C C , Fontes University of Porto
Filomena , Soares Polytechnic of Porto – School of Hospitality and Tourism/School of Media Arts and Design
Francisco , Moreira University of Minho
Gabriela , Czibula Department of Computer Science, Babes-Bolyai University
Günther , Raidl Vienna University of Technology
Horia , Pop Babes-Bolyai University, Cluj
Ines , Sousa University of Minho
Ioana , Zelina Technical University of Cluj Napoca, North Center in Baia Mare, Romania
Jaime , Silva University of Coimbra
João , Clímaco Faculty of Economics University of Coimbra
Joaquim , Pinto ESMC
Jorge , Ferreira Universidade da Madeira
Jorge , Tiago CEMAT – Center for Computational and Stochastic Mathematics, at IST, ULisboa
José , Ferreira University of Coimbra
Jose Alfredo Ferreira , Costa Federal University, UFRN
José Ramón , Villar University of Oviedo
Luís , Grilo Instituto Politécnico de Tomar
Manuel , Grana University of Basque Country
Marc , Demange RMIT University
Marco , Garapa Universidade da Madeira

Margarida , Camarinha University of Coimbra
Maria Antónia , **Forjaz** University of Minho
Maria Da Graça , **Temido** University of Coimbra
Mario , Koeppen Kyushu Institute of Technology
Marius , Paun Transilvania University of Brasov
Marko , Beko Universidade NOVA de Lisboa
Marta , Ferreira University of Minho
Marta , Pascoal Department of Mathematics, University of Coimbra, and INESC-Coimbra
Maurício , Reis Universidade da Madeira
Morgado , Dias Centro de Competências de Ciências Exactas e Engenharias, Universidade da Madeira
Natalia , Bebiano University of Coimbra
Paulo , Mateus Instituto Superior Tecnico
Pedro , Quaresma Department of Mathematics, School of Science and Technology, University of Coimbra
Petrica , Pop Technical University of Cluj-Napoca, North University Center at Baia Mare
Rafael , Luís University of Madeira and CAMGSD (Centro de Análise Matemática, Geometria e Sistemas Dinâmicos)
Raquel , Barreira Polytechnic Institute of Setúbal
Reggie , Davidrajuh University of Stavanger
Sandra , Mendonça University of Madeira
Sílvia , Barbeiro University of Coimbra
Susana , Faria University of Minho
Teresa , Gomes University of Coimbra
Vítor , Vasconcelos ATP Group – Universidade de Lisboa

LOCAL ORGANIZING COMMITTEE

Morgado Dias, University of Madeira and Madeira Interactive Technology Institute
Filipe Quintal, University of Madeira and Madeira Interactive Technology Institute
Herlander Mata-Lima, Universidade Federal da Integração Latino-Americana
Lucas Pereira, University of Madeira and Madeira Interactive Technology Institute
Mary Barreto, University of Madeira and Madeira Interactive Technology Institute
Tiago Meireles, University of Madeira and Madeira Interactive Technology Institute
Dario Baptista, Madeira Interactive Technology Institute
Fábio Mendonça, Madeira Interactive Technology Institute
Luis Rodolfo Sousa, University of Madeira and Madeira Interactive Technology Institute
Roham Torabi, Madeira Interactive Technology Institute
Sandy Rodrigues, Madeira Interactive Technology Institute
Sheikh Mostafa, Madeira Interactive Technology Institute

UMa: University of Madeira
M-ITI: Madeira Interactive Technologies Institute
UNILA: Universidade Federal da Integração Latino-Americana

Keynote Abstracts

Prof. D. Rafael Luís

Rafael Luís received the PhD degree in Mathematical Analysis at University of Madeira in 2011. He has been study both autonomous and nonautononous (periodic) discrete dynamical systems applied in population dynamics and economics. He is teacher of Mathematics at University of Madeira and researcher at Center for Mathematical Analysis, Geometry and Dynamical Systems, Instituto Superior Técnico, University of Lisbon, Portugal.

Title: Paraconsistent Annotated Logic Programs and its Application to Intelligent Control/Safety Verification

Abstract

Nowadays a lot of data are treated automatically in various artificial intelligent systems by using computers though, those data include various kinds of contradiction and inconsistency and usual computer logics are not so good at dealing with contradiction in the same system. Paraconsistent annotated logic is well known as a formal logic that can deal with contradiction in the framework of consistent logical systems. One of its logic programs called Extended Vector Annotated Logic Program with Strong Negation (EVALPSN) has been developed for dealing with non-monotonic reasoning such as defeasible reasoning, etc. by Kazumi Nakamatsu and applied to conflict resolving, various intelligent control systems such as traffic signal control, railway interlocking safety verification, etc. One of these applications of EVALPSN, traffic signal control at an intersection will be introduced with visual simulation. Moreover, a special EVALPSN that can deal with a sort of temporal reasoning, before-after relations between processes (time intervals), which has been developed and named Bf(before-after)–EVALPSN by Kazumi Nakamatsu, and its application to real-time process order control will be introduced based on a small pipeline processing example.

Prof. Dr. Sc. Kazumi Nakamatsu

Kazumi Nakamatsu received the Ms. Eng. and Dr. Sci. from Shizuoka University, and Kyushu University, Japan, respectively. He is a full Professor at School of Human Science and Environment, University of Hyogo, Japan since 2005. His research interests encompass various kinds of logic and their applications to Computer Science, especially paraconsistent annotated logic programs and their applications. He has developed some paraconsistent annotated logic programs called ALPSN(Annotated Logic Program with Strong Negation), VALPSN(Vector ALPSN), EVALPSN(Extended VALPSN) and bf-EVALPSN (before-after EVALPSN) recently, and applied them to various intelligent systems such as a safety verification based railway interlocking control system and process order control. He is an author of over 150 journal papers, book chapters and conference papers, and edited 12 books published by prominent publishers such as Springerverlag. He has chaired various international conferences, workshops and invited sessions, and he has been a member of numerous international program committees of workshops and conferences in the area of Artificial Intelligence and Computer Science. He serves as Editor-in-Chief of the International Journal of Reasoning-based Intelligent Systems by Inderscience Publishers (UK), and as an Associate Editor of the Journal of Intelligent Technologies by IOS Press, International Journal of Hybrid Intelligence by Inderscience Publishers (UK), and Vietnamese Journal of Computer Science by Springerverlag. He also serves as an editorial board member of many international journals. He has contributed numerous invited talks at international workshops, conferences, and academic organizations such as universities. He also is a recipient of some conference and paper awards. He is a member of Japan AI Society, etc.

Title: Brain Computer Interfaces and Immersive Virtual Reality for Post Stroke Motor Rehabilitation

Abstract

Stroke is one of the most common causes of acquired disability, leaving numerous adults with cognitive and motor impairments, and affecting patients' capability to live independently. In recent years, novel rehabilitation paradigms have been proposed to address the life-long plasticity of the brain to regain motor function. Among them, the use of a hybrid brain–computer interface (BCI)—virtual reality (VR) approach can combine a personalized motor training in a VR environment, exploiting brain mechanisms for action execution and observation, and a neuro-feedback paradigm using mental imagery as a way to engage secondary or indirect pathways to access undamaged cortico-spinal tracts. I will present the development and validation experiments of the proposed technology. More specifically, I will discuss the underlying neuroscientific principles, use of low cost EEG acquisition systems, the integration in immersive VR and the use of haptic technology. I will show how the proposed motor imagery driven BCI-VR system is usable, engaging and able to engage the desired brain motor areas. This novel technology enables stroke survivors without active movement to engage in more effective rehabilitation paradigms.

Professor Ari Aharari

Ari Aharari received M.E. and PhD in Industrial Science and Technology Engineering and Robotics from Niigata University and Kyushu Institute of Technology, Japan in 2004 and 2007, respectively.

In 2004, he joined GMD-JAPAN as a Research Assistant. He was Research Scientist and Coordinator at FAIS- Robotics Development Support Office from 2004 to 2007. He was a Postdoctoral Research Fellow of the Japan Society for the Promotion of Science (JSPS) at Waseda University, Japan from 2007 to 2008.

He served as a Senior Researcher of Fukoka IST involved in the Japan Cluster Project from 2008 to 2010. In 2010, he became an Assistant Professor at the faculty of Informatics of Nagasaki Institute of Applied Science.

Since 2012, he has been Associate Professor at the department of Computer and Information Science, Sojo University, Japan. He has served as a main researcher and Principal Investigator in more than 25 projects and is working closely with more than 50 Japanese companies, Local government laboratories and Universities. His research interests are IoT, Robotics, IT Agriculture, Image Processing and Data Analysis (Big Data) and their applications. He is a member of IEEE (Robotics and Automation Society), RSJ (Robotics Society of Japan), IEICE (Institute of Electronics, Information and Communication Engineers), IIEEJ (Institute of Image Electronics Engineers of Japan).

Title: Advanced Mathematics for Designing IoT System

Abstract

The Internet of Things may be a hot topic in the society but it’s not a new concept especially in industry. In this talk, we introduce the fundamental concepts of Internet of Things (IoT) and critical points about how we can design an IoT system. In follow, we introduce Society 5.0 and Industry 4.0 and explain about two projects which was designed based on advanced mathematics for IoT systems.

Professor Vasile Berinde

Vasile Berinde is a Full Professor and Head of Department in the Department of Mathematics and Computer Science at Technical University of Cluj-Napoca, North University Centre at Baia Mare (Romania). His research interests include nonlinear analysis, iterative methods for solving nonlinear functional equations and numerical analysis, research areas in which he has published a significant number of papers in prestigious scientific journals with a high impact and visibility. He has been included in the 2016 and 2017 list of Web of Science Highly Cited Researchers and has been elected as a Honorary Doctor of National Technical University Donetsk, Ukraine. He is Vice-President of Romanian Mathematical Society and has been invited as keynote, plenary or invited speaker to many international conferences in different countries and continents.

Title: Pompeiu-Hausdorff metric and its wide spreading role in science and technology

Abstract

The distance between two sets has been introduced at the beginning of the XXth century by the successive contribution of D. Pompeiu (1873-1954), M. Frechet (1878-1973) and F. Hausdorff (1878-1942). The importance of this fundamental concept came rather late in mathematics (about 1940) and even later in applied sciences and technology but nowadays it is widely used in almost all research areas. The list of applications of what is generally known as Hausdorff metric and less often as Pompeiu-Hausdorff metric is really impressive and comprises more than 100 Web of Science Categories of research areas, of which we mention: radiology nuclear medicine medical imaging, medicine research experimental, clinical neurology, imaging science photographic technology, transportation science technology, engineering biomedical, automation control systems, remote sensing, green sustainable science technology, transportation science technology, environmental sciences, engineering

ocean etc. The main aim of this talk is to give a brief account on the role of Pompeiu-Hausdorff metric and its ubiquity in applied sciences and technology by means of some sample applications.

Table of papers

This page is intentionally left blank.

Comparing model performances applied to fall detection*

Samad Barri Khojasteh[1,2], José R. Villar[2], Enrique de la Cal[2], Víctor M. González[3] and Camelia Chira[4]

Abstract— **This study focuses on the comparison of techniques for modelling and classifying data gathered from wearable sensors, in order to detect fall events of elderly people. Although the vast majority of studies concerning fall detection place the sensory on the waist, in this research the wearable device must be placed on the wrist because it's usability. A first pre-processing stage is carried out as stated in [1], [2]; this stage detects the most relevant points to label. This study analyses the suitability of different models in solving this classification problem: a feed-forward Neural Network and a decision tree based on C5.0. A discussion about the results and the deployment issues is performed according to whether the models are to be exploited in edge/cloud computing or in the wearable device.**

I. INTRODUCTION

Fall Detection (FD) is a very active research area, with many applications to healthcare, work safety, etc. Even though there are plenty of commercial products, the best rated products only reach a 80% of success[3]. There are basically two types of FD systems: contex-aware systems and wearable devices [4]. FD has been widely studied using context-aware systems, i.e. video systems [5]; nevertheless, the use of wearable devices is crucial because the high percentage of elderly people and their desire to live autonomously in their own house [6].

Wearables-based solutions include, mainly, tri-axial accelerometers (3DACC) either alone or combined with other sensors. Several solutions incorporate more than one sensory element; for instance, Sorvala et al [7] proposed two sets of a 3DACC and a gyroscope, one on the wrist and another on the ankle, detecting the fall events with two defined thresholds. The use of 3DACC and a barometer in a necklace was also reported in [8]; similar approaches have been developed in several commercial products.

Several solutions using wearable devices combining 3DACC have been reported, i.e., identifying the fall events using Support Vector Machines [9]. In [10] several classifiers are compared using the 3DACC and the inertial sensor within a smartphone to sample the data. A similar solution is proposed in [11], using some different transformations of the 3DACC signal. A main characteristic in all these solutions is that the wearable devices are placed on the wrist. The reason of this location is that it is by far much easier to detect a fall using the sensory system in this placement. Nevertheless, this type of devices lacks in usability and the people trend to dismiss them in the bedside table. Thus, this research limits itself to use a single sensor -a marketed smartwatch- placed on the wrist in order to promote its usability.

Interestingly, the previous studies do not focus on the specific dynamics of a falling event: although some of the proposals report good performances, they are just machine learning applied to the focused problem. There are studies concerned with the dynamics in a fall event [12], [13], establishing the taxonomy and the time periods for each sequence. Additionally, Abbate et al proposed the use of these dynamics as the basis of the FD algorithm [1]. A very interesting point of this approach is that the computational constraints are kept moderate, although this solution includes a high number of thresholds to tune. In citekhojaste-HAIS2018, this solution was analysed with data gathered from sensors placed on the wrist, using the Abate solution plus a SMOTE balancing stage and a feed-forward Neural Network. In this research, an alternative based on decision trees and C5.0 is proposed.

II. ADAPTING FALL DETECTION TO A WRIST-BASED SOLUTION

Abate et al [1] proposed the following scheme to detect a candidate event as a fall event (refer to Fig. 1). A time t corresponds to a **peak time** (point 1) if the magnitude of the acceleration a is higher than $th_1 = 3 \times g, g = 9.8m/s$. After a peak time there must be a period of 2500 ms with relatively calm (no other a value higher than th_1). The **impact end** (point 2) denotes the end of the fall event; it is the last time for which the a value is higher than $th_2 = 1.5 \times g$. Finally, the **impact start** (point 3) denotes the starting time of the fall event, computed as the time of the first sequence of an $a <= th_3$ ($th_3 = 0.8 \times g$) followed by a value of $a >= th_2$. The impact start must belong to the interval $[impact\ end - 1200\ ms, peak\ time]$. If no impact end is found, then it is fixed to peak time plus 1000 ms. If no impact start is found, it is fixed to peak time.

Whenever a peak time is found, the following transformations should be computed:

- Average Absolute Acceleration Magnitude Variation, $AAMV = \sum_{t=is}^{ie} \frac{|a_{t+1} - a_t|}{N}$, with is being the impact start, ie the impact end, and N the number of samples in the interval.

*This work was supported by Spanish Ministry of Science and competitiviness (MINECO).

[1]Barri Khojasteh is with Sakarya University, Department of Industrial Engineering, Sakarya, Turkey samad.khojasteh@ogr.sakarya.edu.tr

[2]Barri Khojasteh, Villar and de la Cal are with the Computer Science Department, University of Oviedo, Oviedo, 33004, SPAIN UO267536, villarjose, delacal at uniovi.es

[3]González is with the Department of Electrical, Electronic, Automatica and Computer Engineering, University of Oviedo, Gijon, 33204, SPAIN vmsuarez at uniovi.es

[4]Chira is with the Computer Science Department, Technical University of Cluj-Napoca, Cluj-Napoca, Romania camelia.chira at cs.utcluj.ro

- Impact Duration Index, $IDI = impact\ end - impact\ start$. Alternatively, it could be computed as the number of samples.
- Maximum Peak Index, $MPI = max_{t\in[is,ie]}(a_t)$.
- Minimum Valley Index, $MVI = min_{t\in[is-500,ie]}(a_t)$.
- Peak Duration Index, $PDI = peak\ end - peak\ start$, with peak start defined as the time of the last magnitude sample below $th_{PDI} = 1.8 \times g$ occurred before peak time, and peak end is defined as the time of the first magnitude sample below $th_{PDI} = 1.8 \times g$ occurred after peak time.
- Activity Ratio Index, ARI, measuring the activity level in an interval of 700 ms centred at the middle time between impact start and impact end. The activity level is calculated as the ratio between the number of samples not in $[th_{ARIlow}0.85 \times g, th_{ARIhigh} = 1.3 \times g]$ and the total number of samples in the 700 ms interval.
- Free Fall Index, FFI, computed as follows. Firstly, search for an acceleration sample below $th_{FFI} = 0.8 \times g$ occurring up to 200 ms before peak time; if found, the sample time represents the end of the interval, otherwise the end of the interval is set 200 ms before peak time. Secondly, the start of the interval is simply set to 200 ms before its end. FFI is defined as the average acceleration magnitude evaluated within the interval.
- Step Count Index, SCI, measured as the number of peaks in the interval $[peak\ time - 2200, peak\ time]$. SCI is the step count evaluated 2200 ms before peak time. The number of valleys are counted, defining a valley as a region with acceleration magnitude below $th_{SCIlow} = 1 \times g$ for at least 80 ms, followed by a magnitude higher than $th_{SCIhigh}1.6 \times g$ during the next 200 ms. Some ideas on computing the time between peaks [14] were used when implementing this feature.

Evaluating this approach was proposed as follows. The time series of acceleration magnitude values are analysed searching for peaks that marks where a fall event candidate appears. When it happens to occur, the *impact end* and the *impact start* are determined, and thus the remaining features. As long as this fall events are detected when walking or running, for instance, a Neural Network (NN) model is obtained to classify the set of features extracted.

In order to train the NN, the authors made use of an Activities of Daily Living (ADL) and FD dataset, where each file contains a Time Series of 3DACC values corresponding to an activity or to a fall event. Therefore, each dataset including a fall event or a similar activity -for instance, running can perform similarly to falling- will generate a set of transformation values. Thus, for a dataset file we will detect something similar to a falling, producing a row of the transformations computed for each of the detected events within the file. If nothing is detected within the file, no row is produced. With this strategy, the Abbate et al obtained the training and testing dataset to learn the NN.

A. The modifications on the algorithm

As stated in [15], [16], the solutions to this type of problems must be ergonomic: the users must feel comfortable using them. We considered that placing a device on the waist is not comfortable, for instance, it is not valid for women using dresses. When working with elder people, this issue is of main relevance. Therefore, in this study, we placed the wearable device on the wrist. This is not a simple change: the vast majority of the literature reports solutions for FD using waist based solutions. Moreover, according to [17] the calculations should be performed on the smartwatches to extend the battery life by reducing the communications. Therefore, these calculations should be kept as simple as possible.

A second modification is focused on the training of the NN. The original strategy for the generation of the training and testing dataset produced a highly imbalanced dataset: up to 81% of the obtained samples belong to the class FD, while the remaining belong to the different ADL similar to a fall event.

To solve this problem a normalization stage is applied to the generated imbalanced dataset, followed by a SMOTE balancing stage [18]. This balancing stage will produce a 60%(FALL)-40%(no FALL) dataset, which would allow to avoid the over-fitting of the NN models. As usual, there is a compromise between the balancing of the dataset and the synthetic data samples introduced in the dataset.

These above mentioned changes have already been studied in [2]. In this research we proposed to analyse the performance of decision trees in this context: the decision trees represent very simple models that can be easily deployed in wearable devices and with a very reduced computational complexity. Therefore, they could represent a very interesting improvement, either if they work similarly to the NN or just similarly to them.

III. Experiments and results

A ADL and FD dataset is needed to evaluate the adaptation, so it contains time series sample from ADL and for falls. This research made use of the UMA-FALL dataset [19] among the publicly available datasets. This dataset includes data for several participants carrying on with different activities and performing forward, backward and lateral falls. Actually, this falls are not real falls -demonstrative videos have been also published-, but they can represent the initial step for evaluating the adapted solution problem. Interestingly, this dataset includes multiple sensors; therefore, the researcher can evaluate the approach using sensors placed on different parts of the body.

The thresholds used in this study are exactly the same as those mentioned in the original paper. All the code was implemented in R[20] and caret[21]. The parameters for SMOTE were perc.over set to 300 and perc.under set to 200 -that is, 3 minority class samples are generated per original sample while keeping 2 samples from the majority class- . These parameters produces a balanced dataset that moves from a distribution of 47 samples from the minority class and

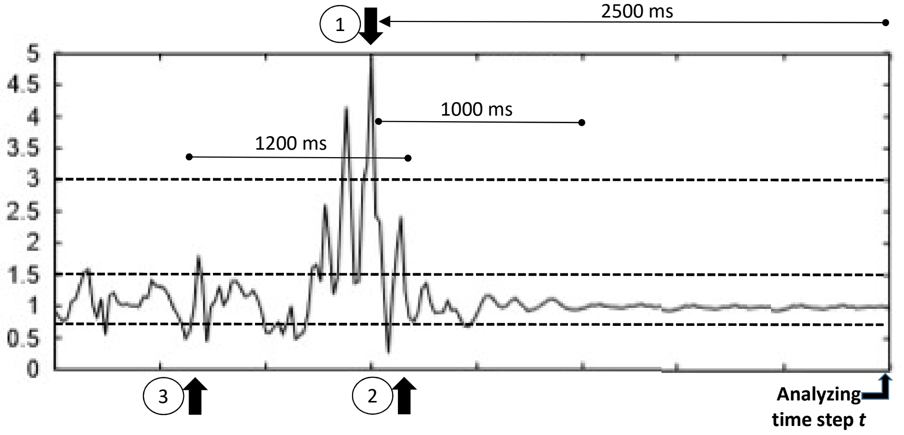


Fig. 1. Evolution of the magnitude of the acceleration -y-axis, extracted from [1]

200 from the majority class to a 188 minority class versus 282 majority class (40%/60% of balance).

To obtain the parameters for the NN a grid search was performed; the final values were size set to 20, decay set to 10^{-3} and maximum number of iterations 500, the absolute and relative tolerances set to 4×10^{-6} and 10^{-10}, respectively. In this research, we use the C5.0 implementation of the C4.5 that is included in the R package to obtain the decision trees. The parameters found optimum for this classification problem are cf set to 0.25, bands set to 2, the fuzzy Threshold parameter set to TRUE, the number of trials set to 15, and winnow set to FALSE.

Both 5x2 cross validation (cv) and 10-fold cv were performed to analysed the robustness of the solution. The latter cv would allow us to compare with existing solutions, while the former shows the performance of the system with an increase in the number of unseen samples. The results are shown in Table I and Table II for 10-fold cv and 5x2 cv, respectively. The boxplots for the statistical measurements Accuracy, Kappa factor, Sensitivity, Specificity, Precision and Recall are shown in Fig. 2 for 10-fold cv and Fig. 3 for 5x2 cv, respectively.

A. Discussion on the results

From the tables it can be seen that both modelling techniques perform exceptionally well once the SMOTE is performed and using test folds from 10-fold cv: the models even perform ideally for several folds. And more importantly, the two models are interchangeable with no apparent loss in the performance. Actually, these results are rather similar to those published in the original work [1]. However, when using 5x2 cv the results diverts from those previously mentioned.

With 5x2 cv, the size of the train and test datasets are of similar number of samples, producing a worse training and, what is more interesting, introduces more variability in the test dataset. Therefore, the results are worse. The point is that these results suggest the task is not solved yet as the number of false alarms increased unexpectedly.

This problem is important because in this experimentation we used the UMA-Fall dataset [19]. This dataset used was generated with young participants using a very deterministic protocol of activities. The falls were performed with the participants standing still and letting them fall in the forward/backward/lateral direction. Therefore, the differences with real falls might be relevant; even if they are not so different, the variability that might be introduced will severely punish the performance of the obtained models.

There are more publicly available datasets, the majority of these datasets have been gathered with healthy volunteers [22], [23]. However, a real-world fall and activity of daily living dataset is published in [24], where a comparison of the different methods published so far is also included. Therefore, the method described in this study needs to be validated with more datasets, more specifically, with data from real fall events.

In apart, the solution proposed analysed and extended in this work includes far too many thresholds. These thresholds have been manually set by the authors for the sensory system placed on the waist; consequently, these values must be tuned for the sensor in a different location as long as the acceleration values are not the same. Even if the thresholds are valid, perhaps the classification models must be specific for groups of people according to their movement characteristics [25], [26].

Besides, the eHealth and wearable applications deployment issues have been study in the literature [17]. According to the published results, there is a trade off between the mobile computation and the communication acts to extend the battery charge as long as possible. Consequently, it has been found that moving all the pre-processing and modelling issues to the mobile part could be advantageous provided the computational complexity of the solution is kept low. The consequences of these findings shall be reflected in the transformations and in the models, reducing complex floating point operations as much as possible [27], [26].

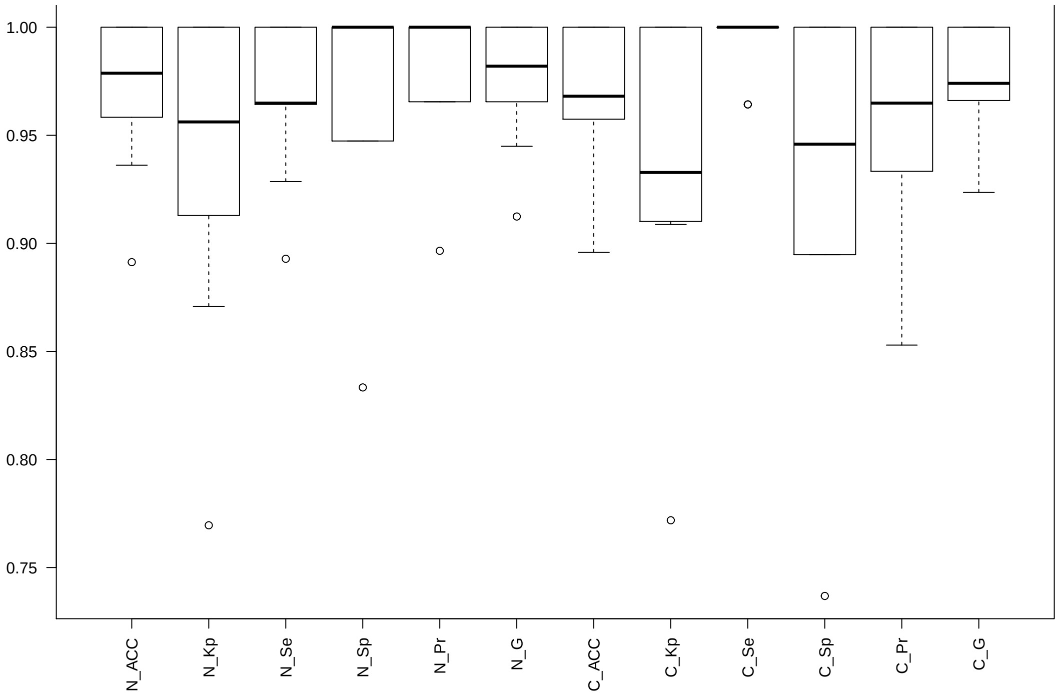


Fig. 2. 10 fold cv Boxplot for the different measurements -Accuracy (Acc), Kappa (Kp), Sensitivity (Se) and Specificity (Sp), Precision (Pr) and the geometric mean of the Acc and Pr, $G = \sqrt[2]{Pr \times Acc}$-, both for the feed-forward NN (six boxplots to the left, with the N_ prefix) and C5.0 (six boxplots to the right, with the C_ prefix).

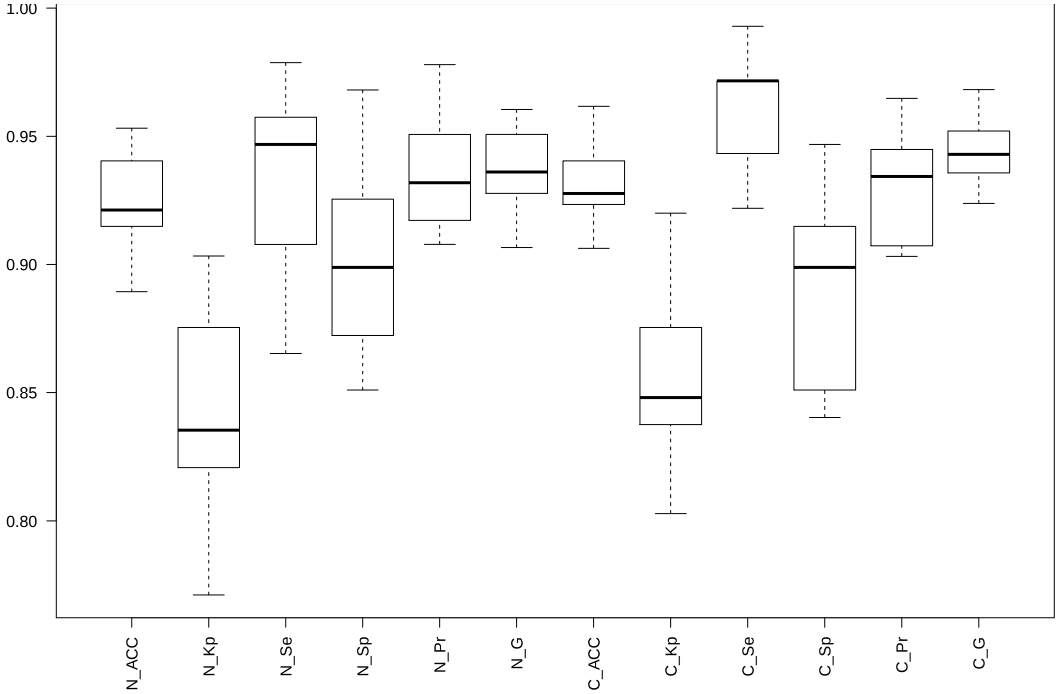


Fig. 3. 5x2 cv Boxplot for the different measurements -Accuracy (Acc), Kappa (Kp), Sensitivity (Se) and Specificity (Sp), Precision (Pr) and the geometric mean of the Acc and Pr, $G = \sqrt[2]{Pr \times Acc}$-, both for the feed-forward NN (six boxplots to the left, with the N_ prefix) and C5.0 (six boxplots to the right, with the C_ prefix).

	Feed forward NN						C5.0 decision tree					
Fold	Acc	Kp	Se	Sp	Pr	G	Acc	Kp	Se	Sp	Pr	G
1	0.97872	0.95620	0.9643	1.00000	1.00000	0.98198	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
2	1.00000	1.00000	1.0000	1.00000	1.00000	1.00000	0.95745	0.91013	1.00000	0.89474	0.93333	0.96609
3	0.97872	0.95620	0.9643	1.00000	1.00000	0.98198	0.97872	0.95620	0.96429	1.00000	1.00000	0.98198
4	0.95833	0.91289	0.9655	0.94737	0.96552	0.96552	0.89583	0.77186	1.00000	0.73684	0.85294	0.92355
5	0.93617	0.87076	0.8929	1.00000	1.00000	0.94491	0.95745	0.91013	1.00000	0.89474	0.93333	0.96609
6	1.00000	1.00000	1.0000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
7	1.00000	1.00000	1.0000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
8	0.89130	0.76954	0.9286	0.83333	0.89655	0.91242	0.95652	0.90873	0.96429	0.94444	0.96429	0.96429
9	0.97872	0.95545	1.0000	0.94737	0.96552	0.98261	0.95745	0.91013	1.00000	0.89474	0.93333	0.96609
10	0.97872	0.95620	0.9643	1.00000	1.00000	0.98198	0.97872	0.95545	1.00000	0.94737	0.96552	0.98261
mean	0.97007	0.93772	0.9680	0.97281	0.98276	0.97514	0.96821	0.93226	0.99286	0.93129	0.95827	0.97507
median	0.97872	0.95620	0.9649	1.00000	1.00000	0.98198	0.96809	0.93279	1.00000	0.94591	0.96490	0.97404
std	0.03412	0.07177	0.0355	0.05367	0.03351	0.02787	0.03158	0.06882	0.01506	0.08242	0.04716	0.02353

TABLE I

10 fold cv results obtained for hte NN and C5.0. From left to right, the main statistical measurements are shown: accuracy (Acc), Kappa factor (Kp, sensitivity (Se), the specificity (Sp), the precision (Pr) and the geometric mean of the Acc and Pr, $G = \sqrt[2]{Pr \times Acc}$.

	Feed forward NN						C5.0 decision tree					
Fold	Acc	Kp	Se	Sp	Pr	G	Acc	Kp	Se	Sp	Pr	G
1	0.92766	0.84740	0.9645	0.87234	0.91892	0.9415	0.92340	0.83755	0.97163	0.85106	0.90728	0.93891
2	0.95319	0.90334	0.9433	0.96809	0.97794	0.9604	0.92340	0.84155	0.92199	0.92553	0.94891	0.93535
3	0.91489	0.82079	0.9504	0.86170	0.91156	0.9308	0.90638	0.80287	0.94326	0.85106	0.90476	0.92381
4	0.88936	0.77113	0.8936	0.88298	0.91971	0.9066	0.93191	0.85455	0.99291	0.84043	0.90323	0.94701
5	0.89362	0.78336	0.8652	0.93617	0.95312	0.9081	0.96170	0.92007	0.97163	0.94681	0.96479	0.96820
6	0.94468	0.88455	0.9574	0.92553	0.95070	0.9541	0.94043	0.87544	0.95745	0.91489	0.94406	0.95073
7	0.92766	0.84629	0.9787	0.85106	0.90789	0.9426	0.94043	0.87410	0.97872	0.88298	0.92617	0.95209
8	0.91489	0.82143	0.9433	0.87234	0.91724	0.9302	0.92340	0.83755	0.97163	0.85106	0.90728	0.93891
9	0.91489	0.82456	0.9078	0.92553	0.94815	0.9278	0.92340	0.84099	0.92908	0.91489	0.94245	0.93574
10	0.94043	0.87544	0.9574	0.91489	0.94406	0.9507	0.94894	0.89286	0.97163	0.91489	0.94483	0.95814
mean	0.92213	0.83783	0.9362	0.90106	0.93493	0.9353	0.93234	0.85775	0.96099	0.88936	0.92938	0.94489
median	0.92128	0.83543	0.9468	0.89894	0.93188	0.9361	0.92766	0.84805	0.97163	0.89894	0.93431	0.94296
std	0.02085	0.04243	0.0357	0.03821	0.02301	0.0182	0.01585	0.03346	0.02274	0.03859	0.02245	0.01291

TABLE II

5x2 cv results obtained for hte NN and C5.0. From left to right, the main statistical measurements are shown: accuracy (Acc), Kappa factor (Kp, sensitivity (Se), the specificity (Sp), the precision (Pr) and the geometric mean of the Acc and Pr, $G = \sqrt[2]{Pr \times Acc}$.

IV. Conclusions

This study compares the performances of two classification techniques when tackling the problem fall detection with data gathered from accelerometers located on one wrist. The original proposal detected fall events and performed a feature extraction which was classified with a feed-forward NN. A SMOTE stage is included to balance the transformed dataset previous modelling. Two different techniques are compared: the feed-forward NN and C5.0 decision trees. A publicly available dataset with falls has been used in evaluating the proposal. Interestingly, the two modelling techniques performed similarly, which suggest that in real world applications with the solution embedded in smartwatches perhaps the decision tree is more likely to be used.

Although exceptional results have been found using 10 fold cv, the 5x2 cv results suggest that still a high number of false alarms is obtained. Although the percentages are better that those reported for commercial devices, some design aspects must be analyzed in depth: the robustness to the variability in the behaviour of the user, or the tuning of the threshold to fit specific populations like the elderly.

Acknowledgment

This research has been funded by the Spanish Ministry of Science and Innovation, under project MINECO-TIN2014-56967-R and MINECO-TIN2017-84804-R. Also, this research has been funded by the research contract no. 1996/12.07.2017, Internal Competition CICDI-2017 of Technical University of Cluj-Napoca, Romania.

References

[1] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and AlessioVecchio, "A smartphone-based fall detection system," *Pervasive and Mobile Computing*, vol. 8, no. 6, pp. 883–899, Dec. 2012.

[2] S. B. Khojasteh, J. R. Villar, E. de la Cal, V. M. González, J. Sedano, and H. R. YAZĞAN, *submitted to the 13th International Conference on Soft Computing Models in Industrial and Environmental Applications*, 2018, ch. Evaluation of a Wrist-based Wearable Fall Detection Method.

[3] Purch.com, "Top ten reviews for fall detection of seniors," http://www.toptenreviews.com/health/senior-care/best-fall-detection-sensors/, 2018.

[4] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering OnLine*, vol. 12, no. 66, 2013. [Online]. Available: http://www.biomedical-engineering-online.com/content/12/1/66

[5] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, , and Z. Li, "A review on human activity recognition using vision-based method," *Journal of Healthcare Engineering*, vol. 2017, 2017.

[6] P. Kumari, L. Mathew, and P. Syal, "Increasing trend of wearables and multimodal interface for human activity monitoring: A review," *Biosensors and Bioelectronics*, vol. 90, no. 15, pp. 298–307, Apr. 2017.

[7] A. Sorvala, E. Alasaarela, H. Sorvoja, and R. Myllyla, "A two-threshold fall detection algorithm for reducing false alarms," in *Proceedings of 2012 6th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2012.

[8] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 6, pp. 619–627, 2010.

[9] T. Zhang, J. Wang, L. Xu, and P. Liu, "Fall detection by wearable sensor and one-class svm algorithm," in *Intelligent Computing in Signal Processing and Pattern Recognition*, ser. Lecture Notes in Control and Information Systems, I. G. Huang DS., Li K., Ed. Springer Berlin Heidelberg, 2006, vol. 345, pp. 858–863.

[10] A. Hakim, M. S. Huq, S. Shanta, and B. Ibrahim, "Smartphone based data mining for fall detection: Analysis and design," *Procedia Computer Science*, vol. 105, pp. 46–51, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917302065

[11] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, "Development of a wearable-sensor-based fall detection system," *International Journal of Telemedicine and Applications*, vol. 2015, p. 11, 2015. [Online]. Available: https://www.hindawi.com/journals/ijta/2015/576364/

[12] S. Abbate, M. Avvenuti, P. Corsini, J. Light, and A. Vecchio, *Wireless Sensor Networks: Application - Centric Design*. Intech, 2010, ch. Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: a survey, p. 22.

[13] Y. S. Delahoz and M. A. Labrador, "Survey on fall detection and fall prevention using wearable and external sensors," *Sensors*, vol. 14, no. 10, pp. 19 806–19 842, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/10/19806/htm

[14] J. R. Villar, S. González, J. Sedano, C. Chira, and J. M. Trejo-Gabriel-Galán, "Improving human activity recognition and its application in early stroke diagnosis," *International Journal of Neural Systems*, vol. 25, no. 4, pp. 1 450 036–1 450 055, 2015.

[15] S. González, J. Sedano, J. R. Villar, E. Corchado, Á. Herrero, and B. Baruque, "Features and models for human activity recognition," *Neurocomputing*, vol. in press, 2015.

[16] J. R. Villar, S. González, J. Sedano, C. Chira, and J. M. Trejo, "Human activity recognition and feature selection for stroke early diagnosis," in *Hybrid Artificial Intelligent Systems*, ser. Lecture Notes in Computer Science, J.-S. Pan, M. M. Polycarpou, M. Wozniak, A. de Carvalho, H. Quintián, and E. Corchado, Eds. Springer Berlin Heidelberg, 2013, vol. 8073.

[17] P. M. Vergara, E. de la Cal, J. R. Villar, V. M. González, and J. Sedano, "An iot platform for epilepsy monitoring and supervising," *Journal of Sensors*, vol. 2017, p. 18, 2017.

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, pp. 321–357, 2002.

[19] E. Casilari, J. A. Santoyo-Ramn, and J. M. Cano-Garca, "Umafall: A multisensor dataset for the research on automatic fall detection," *Procedia Computer Science*, vol. 110, no. Supplement C, pp. 32 – 39, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917312899

[20] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[21] M. Kuhn, "The caret package," http://topepo.github.io/caret/index.html, 2017, last checked 15-1-2018.

[22] R. Igual, C. Medrano, and I. Plaza, "A comparison of public datasets for acceleration-based fall detection," *Medical Engineering and Physics*, vol. 37, no. 9, pp. 870–878, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1350453315001575

[23] S. S. Khan and JesseHoey, "Review of fall detection techniques: A data availability perspective," *Medical Engineering and Physics*, vol. 39, pp. 12–22, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1350453316302600

[24] F. Bagala, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk, "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PLoS ONE*, vol. 7, no. 5, p. e37062, 2012.

[25] S. González, J. R. Villar, J. Sedano, J. Terán, M. L. Alonso-Álvarez, and J. González, "Heuristics for apnea episodes recognition," in *accepted for Proceedings of the International Conference on Soft Computing Models in Industrial and Environmental Applications*. Springer, 2015.

[26] J. R. Villar, M. Menéndez, E. de la Cal, V. M. González, and J. Sedano, "Identification of abnormal movements with 3d accelerometer sensors for its application to seizure recognition," *accepted for publication, International Journal of Applied Logic*, 2016.

[27] J. R. Villar, P. Vergara, M. Menéndez, E. de la Cal, V. M. González, and J. Sedano, "Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition," *accepted for publication, International Journal of Neural Systems*, 2016.

Efficient Use of Parallel PRNGs on Heterogeneous Servers

André Pereira* and Alberto Proença*
*Algoritmi Centre, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
{ampereira, aproenca}@di.uminho.pt

***Abstract*—Scientific code often requires very large amounts of independent streams of random numbers with a Gaussian distribution for stochastic simulations. This code typically uses high statistical quality Pseudo-Random Number Generators (PRNGs) and distribution transformation algorithm. Scientific data analyses developed for the ATLAS Experiment at CERN were used to test and validate our approaches to an efficient parallel PRNG, with a Gaussian distribution in a multicore server with a GPU accelerator. This paper evaluates the performance of our approaches in ATLAS data analyses and presents a comparative performance evaluation among different implementations of popular PRNG algorithms available in ROOT, MKL, and PCG libraries, on an heterogeneous compute server, showing the positive impact of a GPU accelerator to generate large amounts of PRN streams.**

***Index Terms*—Pseudo-Random Number Generation, Efficient Parallel PRNG, Scientific Computing, Code Execution Efficiency, Performance Analysis, High Performance Computing.**

I. Introduction

Scientific data analysis usually aim to test hypothesis and theories or simulate phenomena. Computing tasks of these applications often require sampling of measured values within their margin of error, Monte Carlo algorithms or randomness associated with specific scientific phenomena, which may account for a significant portion of the overall execution time. This need for randomness in the deterministic environment of computer science created the demand for algorithms that provide seemingly random numbers.

Pseudo-random number generation, PRNG, the process of generating apparently random numbers on digital chips, is a well studied topic, with the first computer based algorithms being suggested as early as 1951 [1]. There are several PRNGs available with excellent statistical quality, as well as implementations on various programming environments with reasonable performance. However, the generator performance is often overlooked by non-computer scientists, which may lead to a significant application performance degradation.

Three main aspects should be considered when selecting a PRNG for a scientific application: the statistical quality, which is out of the scope in this work, the computational performance of the algorithm/implementation and the way that a given implementation is used in the code. These aspects are critical specially for parallel code executing on multicore and manycore compute servers, where algorithmic and computational inefficiencies may lead to significant performance and scalability bottlenecks.

This paper presents a performance evaluation of different implementations of a popular PRNG, the Mersenne Twister [2], as well as Gaussian distribution transformation algorithms, such as Inverse Transform Sampling and Box-Muller [3], available in popular scientific libraries. It also provides an insight on the best way to use these implementations, comparing three different approaches on two real parallel applications with different amounts of pseudo-random numbers (PRNs), related to the search of the Higgs boson [4]. A PRNG from the permuted congruential generator (PCG) family [5] was also evaluated, as the authors claim that it performs better than any other algorithm, although it is not yet fully accepted by the scientific community.

This paper is structured as follows: section II presents the two case studies used to evaluate the different PRNGs and their implementations; section III contextualises the generation of random numbers, presenting the most popular PRNGs, the distribution transformations and the different approaches to use them in a parallel environment; section IV evaluates the different PRNG implementations in the case studies; section V makes a critical analysis of the developed work with suggestions for further improvements.

II. Pipelined Scientific Data Analyses

A scientific data analysis is a process that converts raw scientific data (often from experimental measurements) into useful information to answer questions, test hypotheses or prove theories. When dealing with large amounts of experimental data, data is read from one or more files in variable sized chunks or datasets, and placed into an adequate data structure.

Parallel implementations of these analyses, where concurrent threads process different dataset elements, are often used in pipelined scientific data analyses, with few data dependencies among dataset elements. The overall analysis performance can be improved with an adequate balance between reading data from disk and data processing, exploiting some pipeline features and identifying and minimising code bottlenecks. Among these latter, the use of PRNGs often play a significant role in performance degradation and this is the key subject of this paper.

High energy physics scientists at the ATLAS Experiment [6] at CERN developed a pipelined scientific data analysis code, the $t\bar{t}H$ analysis, to study the associated production of top quarks with the Higgs boson, following head-on proton-proton collisions (known as events) at the Large Hadron Collider (LHC). The final state of an event is recorded by the ATLAS particle detector, which measures the characteristics of the bottom quarks (detected as jets of particles due to a hadronization process) and leptons (both muons and electrons), but not the neutrinos, as they do not interact with the detector sensors. This final state is presented in figure 1.

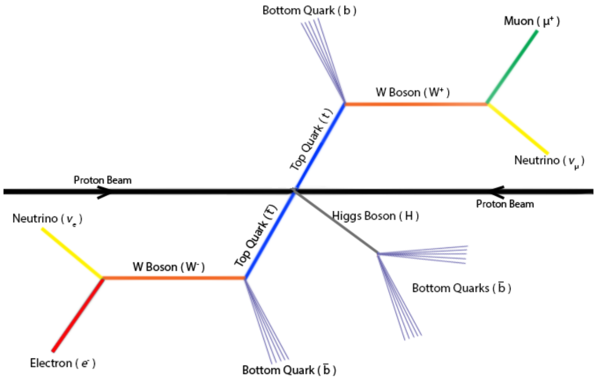


Fig. 1. Schematic representation of the $t\bar{t}$ system and Higgs boson decay.

$t\bar{t}H$ analytically computes the characteristics of the neutrinos with known information, to reconstruct at the end of the data processing pipeline both top quarks and the Higgs boson [4]. This process, known as kinematic reconstruction, tests every combination of bottom quarks and leptons, which are stored in a specific structure in predefined files provided by the experiments at the LHC.

Two compute-bound variants of the $t\bar{t}H$ analysis were considered as representative case studies:

- one reconstructs the top quarks and the Higgs boson, the `ttH_sci` (*sensors with a confidence interval*), assuming $\pm 1\%$ accuracy of the sensors in the ATLAS detector; it performs an extensive sampling within the 99% confidence interval in the kinematic reconstruction, from which only the best reconstruction is considered; this version works with 1024 samples, where each requires the generation of 30 different PRNs, to a total of 30Ki PRNs per event;
- the other is similar to the previous, but where two pipeline stages were replaced, `ttH_scinp` (`sci` *with a new pipeline*), to perform different operations on each data element, maintaining the same overall inter-stage dependencies and a similar sampling of the confidence interval of `ttH_sci`; this version requires less PRNs, 10Ki per event.

The PRNG used by default by these data analyses is the Mersenne Twister implementation provided by the ROOT framework [7], transforming the uniform distribution of the PRNs into a Gaussian distribution by the Box-Muller algorithm.

III. Random Number Generation

Random numbers are used in a wide spectrum of applications where unpredictability is required, including statistical data sampling, scientific computing, gaming and cryptography. Different applications often require specific properties from a random number, for which different random number generators may be used. In the context of computer science, these can be broadly classified as True Random Number Generators (TRNGs) or Pseudo-Random Number Generators (PRNGs).

TRNGs are based in physical random processes to generate random bit strings, which may have to be preprocessed to remove possible bias. The most common example of a TRNG is the coin toss of a symmetrical coin, where one can expect either heads or tails with a 50% certainty. A set of coins, or a series of coin tosses, can be used to generate a random sequence of bits. However, coins are not perfectly balanced and there is a small probability of landing on its side, slightly deviating the 50-50 chances of expecting heads or tails. Post-processing may be used to remove the bias of these processes. There are no correlations among generated numbers but these generators are usually slow, not suited for large scale computing and their results cannot be replicated, which makes debugging code harder.

PRNGs attempt to approximate the properties of truly random numbers, such as no repetition of sequence of values for a long period and no correlation between generated numbers. The generated values are not truly random as they are determined by an initial value (seed). A proper mathematical analysis of the generator algorithm is required to assess its quality and if they are close enough to truly random for the specific use that they were designed for. The main benefit of this type of random generator is the performance, which, depending on the algorithm, may be able to speedup with the amount of available cores. The use of a seed also eases the process of debugging code. With an proper algorithm, this type of generator is mostly used for scientific applications due to its higher performance and adequate mathematical properties. However, most PRNGs can only generate sets of uniformly distributed PRNs, which may require a transformation algorithm to convert them into another PRN distribution.

A short introduction to the most popular PRNGs and distribution transformations follows through the next subsections, as well as the most used libraries by the scientific community.

A. Popular PRNG Algorithms

There is a wide range of algorithms to generate PRNs currently available, each with strengths and weaknesses that may make them best suited for different uses. The quality of a PRNG randomness is usually evaluated by a set of benchmarks, such as the Diehard [8] and TestU01 [9] suites. An ideal PRNG has an infinite period, covers the entire range of possible PRNs (usually 32/64-bit numbers), and has no correlation between generated PRNs. Other mathematical characteristics may be equally important, but are not as relevant in the scientific community when choosing a PRNG.

The scientific community has been using several PRNG algorithms, such as the r1279 and Wichmann-Hill PRNG available in GSL [10], MKL [11], and NAG [12], but one stands above all other in popularity: the Mersenne Twister [2]. This algorithm was developed in 1997 and features a period of $2^{19337} - 1$, passes most statistical tests, and it is extremely fast to generate both 32 and 64-bit numbers. This generator is also implemented in most languages and available in most scientific computing libraries. There are limitations, such as low throughput, but they are often overcome by alternative implementations of this algorithm, which take advantage of vector/SIMD instructions, GPU architectures and multithreaded environments.

Recently, a PCG family of PRNGs was proposed [5], claiming better statistical quality and performance, for both single and multithreaded environments. Even though it is not yet fully accepted by the scientific community, the PCG RXS-M-XS 64 generator (a Linear Congruential Generator, LCG) will be included in our performance evaluation, alongside Mersenne Twister, in section IV, as the authors claims it is one of the best performing PRNGs currently available. Since the PCG generators only generate uniformly distributed numbers, they will be paired with an efficient implementation of the Box-Muller algorithm available in the ROOT framework [7].

B. Transforming Uniformly Distributed PRNs

PRNs are usually generated in an uniform distribution, but other distributions may be required. Gaussian distributed PRNs are often used in scientific computing, so having PRNG implementations that support that functionality is crucial.

Since most algorithms only generate uniformly distributed PRNs, this distribution may require post processing. One of the most common algorithms is the Box-Muller transformation [3], which generates a pair of independent Gaussian distributed PRNs based on a set of uniformly distributed numbers. It is not one of the most computationally efficient transformations, due to its iterative nature and reliance on square roots, logarithmic and trigonometric functions.

The Inverse Transform Sampling[1] is a method that transforms uniformly distributed numbers into any distribution, given its Cumulative Distribution Function (CDF). The CDF maps a PRN into a probability between 0 and 1 and then inverts this function, providing the final non-uniformly distributed number. This number can be adjusted to a specific mean and standard deviation afterwards, as required by a Gaussian distribution. The lack of an analytic CDF for the Gaussian distribution may affect the algorithm performance, favouring other transformations such as the Box-Muller. However, current implementations, widely accepted by the scientific community, use an extremely accurate approximation of the Gaussian CDF, which is faster than most transformations.

The computational performance of both Box-Muller and Inverse Transform Sampling methods (with the CDF approximation) will be assessed and evaluated on real scientific case studies. Other methods could be used, such as the Ziggurat transformation [13], but are not in the scope of this work as they are not used as often by the scientific community.

C. PRNG Libraries

Most scientific computing libraries and frameworks provide efficient implementations of a wide variety of PRNGs.

In the context of the particle physics community, related to the case study presented in section II, the most popular scientific libraries are provided in the ROOT framework. This framework only offers the Mersenne Twister PRNG with the Box-Muller transformation and is used by default in the two case study variants.

MKL is one of the most popular scientific computing libraries which offers a wide range of relevant functionalities. It features several PRNGs, from which only the Mersenne Twister will be considered, as it would be the most likely to be used by the scientific community. The Box-Muller and ICDF (Inverse Transform Sampling) transformations are available in this library and will be used to convert uniformly distributed PRNs into a Gaussian distribution. MKL also provides the option to generate a batch of PRNs, which will also be tested in section IV.

The fastest PRNG available in the PCG family, the RXS-M-XS 64 (LCG), will be coupled to the Box-Muller implementation available in ROOT to provide Gaussian distributed pseudo-random numbers.

To offload the PRNG to the CPU accelerator the NVidia CUDA toolkit includes a library of PRNGs, cuRAND [14]. It provides an efficient implementation of the Mersenne Twister algorithm and the Box-Muller transformation.

A request for a new PRN in a parallel environment can follow several approaches:

- a single PRNG to feed all concurrent threads, where each PRNG execution is atomic; results would not be reproducible as PRNs consumed by each thread varies between runs [15]; it does not support concurrent execution of the PRNG;
- a single PRNG to feed each stream request using a transition function to guarantee that there are no correlations among streams, known as leapfrog; used in the cuRAND implementation of Mersenne Twister [16]; it supports concurrent execution;
- a single PRNG to feed all concurrent threads with a different a precomputed seed for each stream, causing the generated PRNs to be equally spaced in the overall PRN sequence, which may be slow as shown in [17], known as splitting; it supports concurrent execution;
- an independent PRNG per compute thread initialised with different sets of parameters; if these parameters are not adequate, streams may not be truly independent, as referenced in [18]; the most common and portable approach used in scientific code.

An implementation of a PRNG on a library is as important to the overall scientific application performance as the approach used to interact with the PRNG itself in the

[1] Available at https://en.wikipedia.org/wiki/Inverse_transform_sampling.

code. For instance, one can generate all PRNs upfront, or request a PRN when needed, and these approaches will have a different performance impact depending on the application and execution environment, specially in parallelized code where there may be multiple threads accessing shared PRNs and/or PRNGs.

The parallel implementation of the two variants of our case study was modified to evaluate three different approaches to manage the generation of PRNs:

- to call the PRNG whenever a single PRN is needed;
- to generate a batch of PRNs and store the result in a thread private buffer; when a PRN is needed the compute thread removes it from the buffer; when the buffer is empty, a new batch is requested;
- to generate a batch of PRNs and store the result in a thread private dual-buffer; while the one buffer is being consumed, the other is being filled.

Preliminary results showed that sharing buffers among compute threads degrades performance, due to contention when accessing shared resources.

This dual-buffer approach minimises the overhead of the PRNGs. Figure 2 illustrates this approach. In the case of the PRNG on GPU, this approach also hides the costs of memory transfers between the CPU and GPU memory. The case studies that were implemented with a single thread do not need a PRNG management thread.

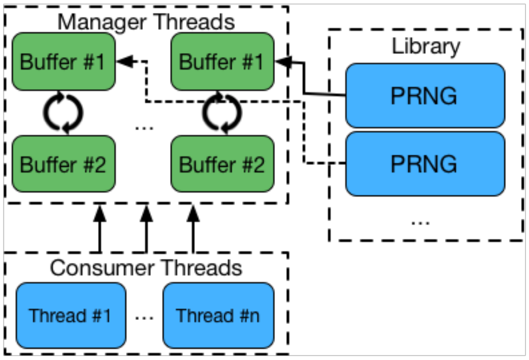


Fig. 2. Dual buffer implementation in the PRNG management threads.

To generate a single PRN at a time the cuRAND PRNG was not used, since the lack of parallelism and the overhead on memory transfers would greatly affect the performance. In the parallel implementations, each PRNG management thread in the multicore devices uses a different stream to launch kernels on the GPU and perform the memory transfers, ensuring that concurrent management threads can simultaneously generate and receive PRNs. Preliminary tests showed that for 24 computing threads (and associated management threads) the GPU device was not fully utilised, meaning that it could scale for a greater number of multicore threads.

IV. Results and Discussion

The testbed used for the quantitative evaluation of the PRNGs was a dual socket server with 12-core Intel Xeon E5-2695v2 Ivy Bridge devices, at 2.4 GHz [19] with 64 GiB RAM, coupled with one NVidia Tesla K20 with 2496 CUDA cores and 5GB of GDDR5 memory (Kepler architecture).

The two variants of the case study code, as described in section II, were `ttH_sci` and `ttH_scinp`. A k-best measurement heuristic was used to ensure that the results can be replicated, with $k = 5$ with a 5% tolerance, a minimum/maximum of 15/25 measurements. The multithreaded tests used 24 cores in the Xeon multicore devices, with 1 computing thread per core. The $t\bar{t}H$ analyses were tested with 128 files, each with $\pm 6,000$ events (dataset elements). In multithreaded tests each PRNG management thread uses an independent PRNG per compute thread.

The `ttH_sci` application, which is the most PRNG intensive, spent around 90% of the execution time calling the ROOT framework PRNG, while the application that required less PRNs, `ttH_scinp`, spent around 50% of the execution time.

Figure 3 compares the execution times to generate 10^6 PRNs of various implementations of the Mersenne Twister with the ROOT and MKL libraries, the former coupled with the Box-Muller (*BM*) transformation and the latter with the two available Box-Muller implementations and the Inverse Transform Sampling (*BM*, *BM2*, and *I*). MKL also offers implementations optimised to generate batches of PRNs (*A*). The PCG was coupled with the Box-Muller transformation.

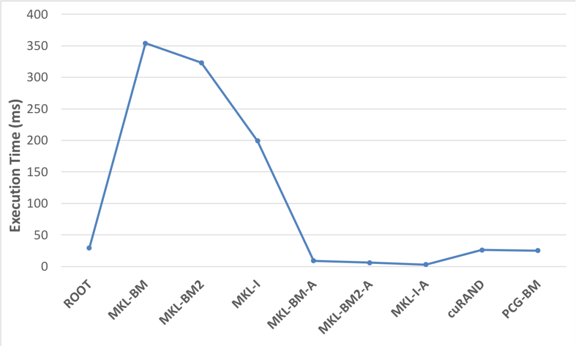


Fig. 3. Execution time of each PRNG to generate 10^6 PRNs.

This test shows that there is a small difference between the ROOT and PCG generator, with 29 and 25 milliseconds respectively, but the MKL batch generator using the Inverse Transform Sampling was able to generate these numbers in 3 milliseconds. This generator will be used as the default MKL generator on the next tests with the case studies. Offloading the PRNG to the GPU accelerator led to a similar performance to the PCG generator, considering PRNG initialisation, generation and memory allocation and transfers between the GPU and the multicore memories.

Figure 4 shows the speedup of the two encoded sequential versions of the data analyses (`ttH_sci` and `ttH_scinp`) with the selected PRNG algorithms and the different approaches detailed in subsection III-C compared to the original ROOT single number PRNG. For the `ttH_sci` application, approaches that use a single or dual PRN buffers are noted as *SB* and *DB*, respectively. In all Mersenne Twister PRNGs in multicore devices, the single and dual buffer approaches provide a slight performance improvement over generating a single pseudo-random number at a time. Offloading the PRNG to GPU provided speedups up to 3.8x, similar to the PCG PRNG. This approach benefits from a dual buffer approach the most as it hides the cost of memory transfers between CPU and GPU devices. The `ttH_scinp` application has a similar behaviour to `ttH_sci`, as it requires a high quantity of pseudo-random numbers. However, the PCG algorithm has a higher performance improvement when larger amounts of PRNs are required, and the same applies when offloading the PRNG into the GPU.

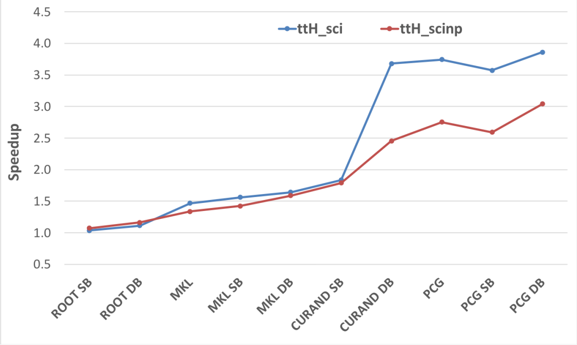


Fig. 4. Speedup of the sequential 2 data analyses with different PRNG algorithms and approaches *vs* the original ROOT single number PRNG.

Figure 5 shows the speedup of the two 24-threaded versions of the data analyses with the selected PRNG algorithms (one per physical core of the Xeon devices) and the different approaches for PRNG concurrent execution, compared to the original ROOT single number PRNG.

For the `ttH_sci`, which requires the most PRNs, the use of single or dual buffers approaches provide larger performance improvements than on the sequential code using only the multicore devices, specially for the PCG generator with a speedup improvement from 42x to 48x. The offload of the PRNG to the GPU devices, with dual buffers to hide the PRNG execution time and memory transfers, provides a performance improvement up to 70x over the original application with the ROOT single number PRNG. This speedup is due to the efficiency of the batch generation of pseudo-random numbers on GPUs, but also due to the higher availability of the Xeon cores since they were freed from generating PRNs, and in a compute-bound code this makes a difference. Worth mentioning is the fact that the GPU is not being fully used, which means that the codes can reach larger speedups if they require larger amounts of PRNs.

The `ttH_scinp` behaves similarly to `ttH_sci` but with smaller speedups, up to 20x using only the multicore devices. The use of a GPU device as a computing accelerator improves the performance up to 10x, but the overhead of the memory transfers and the less amount of pseudo-random numbers required by this application restricts the efficiency of this approach, when compared to the PCG PRNG with a dual buffer.

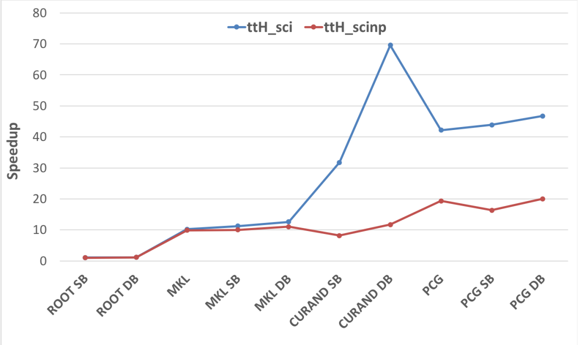


Fig. 5. Speedup of the 24-threaded data analyses with different PRNG algorithms and approaches *vs* the original ROOT single number PRNG.

The overhead of using the Box-Muller transformation with the PCG PRNG accounts for only 30.1% of the overall PRNG time for the `ttH_sci` application with 24 computing threads. However, it was not possible to profile MKL as the available libraries were not compiled with debugging symbols.

Both figures 4 and 5 prove that applications that require a huge amount of PRNs can greatly benefit by the use of efficient implementations of PRNG approaches. While the efficient use of the MKL library can provide significant performance improvements, the PCG PRNG tested was the best performing on multicore devices by a large margin. However, this may be considered an unfair comparison, since this PRNG algorithm is faster than the Mersenne Twister. It is the responsibility of the end user to assess if this PRNG should be used over other traditional PRNGs, which are well accepted and extensively tested by the mathematics's community.

V. Conclusions and Future Work

This paper presents an evaluation of the computational performance of different versions of a popular PRNG, the Mersenne Twister, available on MKL and ROOT libraries, comparing different implementations available in those libraries. It aims to provide an insight into the best way to use these generators with real software codes, evaluating the performance of their implementations. A PRNG of the PCG family was also included in this comparison with Mersenne Twister implementations, since the authors claim it has the best statistical quality and performance.

Two variations of a real scientific data analysis were used as case studies: `ttH_sci` and `ttH_scinp`, both compute-bound codes. `ttH_scinp` and `ttH_sci` require 10Ki and 30Ki PRNs per event (dataset element), with the tested dataset containing around 800K events. The PRNGs were used in these case studies as a single number generator, a batch generator to a thread-private buffer, and to a thread-private dual buffer, where one is filled while the other is being consumed. In both single and dual buffer approaches, the PRNG is managed by an additional thread per computing thread, so that data processing and PRNG can be concurrently executed.

An initial test of generating 10^6 PRNs with a Gaussian distribution of the various PRNG implementations available on MKL, ROOT, and PCG libraries, showed a clear disadvantage of using MKL PRNGs that only return a single number, rather than a batch generator. The batch MKL generator with the Inverse Transform Sampling method was 8x faster than the PCG and cuRAND generators, and 10x faster than ROOT. This implementation was used as the PRNG representative of the MKL library in the tests with the scientific applications. There was no significant performance difference between using GNU or Intel compilers.

The first set of tests only used sequential versions of the real scientific data analyses. The data analysis with the PCG PRNG with the dual buffer approach was the faster one, almost 4x faster than the original single number ROOT generator for the sequential version of the `ttH_sci` application, closely followed by cuRAND. The code with MKL PRNG did no beat any of the other codes. A similar behaviour is observed for `ttH_scinp`, but with smaller speedups, as this code variant requires fewer PRNs. For `ttH_sci` with 24 computing threads, the use of the GPU accelerator provided a speedup up to 70x with the dual buffer implementation, due to its faster PRNG and by freeing CPU resources to be used by the computing threads. Code with the PCG PRNG was again the faster one, 47x faster than ROOT, while MKL only displayed a 11x performance improvement.

Three main conclusions can be extracted from this analysis:

- the choice of an efficient implementation of a given algorithm is a key issue: both ROOT and MKL implement the Mersenne Twister but MKL is, at least, 10x faster;
- a performance analysis of these algorithms should not be made with a synthetic code, but rather with real code that requires these generators; although this may change with the applications, the fastest PRNGs on the initial tests were not the best when used in our real scientific case studies;
- the way these PRNGs are used in each application may have a significant impact on performance: the cuRAND dual buffer was 2.3x faster than the single buffer implementation.

This work focused mainly on the popular Mersenne Twister algorithm, but others could benefit from this analysis, such as cryptographycally secure PRNGs. The SIMD-oriented Mersenne Twister [20] could also be tested, but it is expected that an efficient SIMD implementation for Intel CPU devices is provided by MKL, as proved by the initial benchmark results. Other, possibly faster, distributions and distribution transformations could be tested, such as the Ziggurat (which was not available in the tested libraries), since the measured Box-Muller execution time takes up to 30% of the overall PCG PRNG execution time.

Acknowledgments

This work has been supported by COMPETE POCI-01-0145-FEDER-007043 and FCT - Fundação para a Ciência e a Tecnologia within the Project Scope UID/CEC/00319/2013 and by Search-ON2: Revitalization of HPC infrastructure of UMinho, (NORTE-07-0162-FEDER-000086), co-funded by ON.2-O Novo Norte under NSRF through ERDF. The first author is also sponsored by FCT grant SFRH/BD/119398/2016.

References

[1] J. Von Neumann, "Various Techniques Used in Connection With Random Digits," *Appl. Math Ser*, vol. 12, no. 36-38, p. 3, 1951.

[2] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[3] E. Golder and J. Settle, "The Box-Muller Method for Generating Pseudo-Random Normal Deviates," *Applied Statistics*, pp. 12–20, 1976.

[4] ATLAS Collaboration, "Observation of a New Particle in the Search for the Standard Model Higgs Boson with the ATLAS Detector at the LHC," *Phys.Lett.*, 2012.

[5] M. E. O'Neill, "PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation," Harvey Mudd College, Claremont, CA, Tech. Rep. HMC-CS-2014-0905, Sep. 2014.

[6] T. A. Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider," *Journal of Instrumentation*, 2008.

[7] F. Rademakers, P. Canal, B. Bellenot, O. Couet, A. Naumann, G. Ganis, L. Moneta, V. Vasilev, A. Gheata, P. Russo, and R. Brun, "ROOT." [Online]. Available: http://root.cern.ch/drupal/

[8] G. Marsaglia, "The Marsaglia Random Number CDROM Including the Diehard Battery of Tests of Randomness," *http://www.stat.fsu.edu/pub/diehard/*, 2008.

[9] P. L'Ecuyer and R. Simard, "TestU01: AC Library for Empirical Testing of Random Number Generators," *ACM Transactions on Mathematical Software (TOMS)*, vol. 33, no. 4, p. 22, 2007.

[10] B. Gough, *GNU Scientific Library Reference Manual*. Network Theory Ltd., 2009.

[11] E. Wang, Q. Zhang, B. Shen, G. Zhang, X. Lu, Q. Wu, and Y. Wang, "Intel Math Kernel Library," in *High-Performance Computing on the Intel® Xeon Phi^{TM}*. Springer, 2014, pp. 167–188.

[12] N. A. Group and N. A. G. L. (Oxford), *Fortran Library Manual*. Numerical Algorithms Group, 1988, vol. 3.

[13] G. Marsaglia, W. W. Tsang *et al.*, "The Ziggurat Method For Generating Random Variables," *Journal of Statistical Software*, vol. 5, no. 8, pp. 1–7, 2000.

[14] C. Nvidia, "Curand library," 2010.

[15] D. R. Hill, C. Mazel, J. Passerat-Palmbach, and M. K. Traore, "Distribution of Random Streams for Simulation Practitioners," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 10, pp. 1427–1442, 2013.

[16] M. Saito and M. Matsumoto, "A Deviation of CURAND: Standard Pseudorandom Number Generator in CUDA for GPGPU," in *Proceedings of 10th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, 2012.

[17] T. Bradley, J. du Toit, R. Tong, M. Giles, and P. Woodhams, "Parallelization Techniques for Random Number Generators," in *GPU Computing Gems Emerald Edition*. Elsevier, 2011, pp. 231–246.

[18] K. Claessen and M. H. Pałka, "Splittable Pseudorandom Number Generators Using Cryptographic Hashing," in *ACM SIGPLAN Notices*, vol. 48, no. 12. ACM, 2013, pp. 47–58.

[19] Intel, "Intel Xeon Processor E5 v2 Family: Datasheet," Intel Corporation, Tech. Rep., 2013.

[20] M. Saito and M. Matsumoto, "SIMD-Oriented Fast Mersenne Twister: A 128-bit Pseudorandom Number Generator," in *Monte Carlo and Quasi-Monte Carlo Methods 2006*. Springer, 2008, pp. 607–622.

Models for Optimization Decision of Capital And Investment Strategy of Life Insurer with Stochastic Assets and Liabilities

Hong Mao
Shanghai Second Polytechnic University
Shainghai, China
hmaoi@126.com

Krzysztof Ostaszewski
Department of Mathematics
Illinois State University
Normal, Illinois, U.S.A.
krzysio@ilstu.edu

Abstract — **In this article, we establish three models for determination of optimal capital and investment strategies of life insurance companies based on minimizing total frictional cost without constraint, with constraints of Solvency II and Swiss Solvency Test respectively. We also establish stochastic models of assets and liabilities of insurance companies.**

Keywords— Life Insurer Capital; Reinsurance; Regulation; Solvency

I. INTRODUCTION

In [11] determination of the optimal capital, investment and reinsurance of property and liability insurance companies is modeled, and comparisons are made with different frameworks of capital regulation (e.g., minimizing frictional cost of capital, Solvency II and Swiss Solvency Test). Other relevant works include [1], [2], [4], [5], [6], [7], [8], [9], [12] . The structures of assets and liabilities of life insurance firms is very different from those of property and liability insurers. For example, the mortality, longevity risk, investment risk, and surrender risk are important risks, which must be paid great attention to in the design of risk management strategy for a life insurance form, and for solvency regulation. In particular, rational selection of investment portfolio, as well as capital level, is more important to life insurers because of their characteristic of long term business. [4] presents a discrete time Asset-Liability Management (ALM) model for the simulation of simplified balance sheet of life insurance products. In this paper, we focus on establishing the stochastic model of assets and liabilities of life insurance and determine the optimal investment and capital strategy simultaneously.

II. METHODOLOGY

We now describe our model, which is based on minimizing the total friction cost.

Definition of Variables:

$A(t)$: assets of an insurer at time *t*;

$L(t)$: liabilities of an insurer at time *t*;

X_0 : amount of initial capital;

X_t : amount of capital or surplus at time *t*;

r_c : ratio of frictional capital cost;

Pr : probability of insolvency;

π : amount invested in risky asset;

r : risk-free interest rate;

P_{1t}^{xi} : premium rate of *i*th insurance contract of life insurance at time *t*, $i = 1,2...m_t$;

P_{2t}^{xj} : premium rate of *j*th insurance contract of annuity insurance at time *t*, $i = 1,2...m_t$;

n_t^i : number of *i*th insurance contract in force at time *t*, $i = 1,2...m_t$;

n_{1t}^{xi} : number of life insurance contracts at time *t* with death payment T^{xi}, $i = 1,2...m_1^t$;

n_{2t}^{xi} : number of annuity insurance contracts at time *t*, with survival payment E^{xi}, $i = 1,2...m_2^t$;

σ_1 : volatility of risky asset;

T_t : expected death payments at time *t*;

E_t : expected survival payment at time *t*;

S_t : expected surrender payment at time *t*;

R_t : reserve at time t ;

${}_tq_x$: probability that the insured will die between x and $x+t$ given that he is alive at age x.

${}_tp_x$: probability that the insured survive to age $x+t$ given that he is alive at age x .

s_t^E : expected surrender ratio of annuity insurance at time t ;

s_t^T : expected surrender ratio of life insurance at time t ;

α : ratio of the surrender cost to the reserve;

$R_{t(xi)}^E$: reserve of annuity insurance at time *t* with survival payment E^{xi} ;

$R_{t(xi)}^T$: reserve of life insurance at time *t* with death payment T^{xi} ;

$\sigma_{t(xi)}^T$: volatility of death payment of *i*th life insurance at time *t*;

$\sigma_{t(xi)}^E$: volatility of survival payment of *i*th annuity insurance at time *t*;

$\sigma_{t(xi)}^{ST}$: volatility of surrender payment of *i*th life insurance at time *t*;

$\sigma_{t(xi)}^{SE}$: volatility of surrender payment of *i*th annuity insurance at time *t*;

v : maximum issue age;

β : the ratio of indirect bankruptcy cost to direct bankruptcy cost.

III. OPTIMIZATION MODEL WITH NO CONSTRAINTS (MODEL 1)

Since the model including reinsurance would become too complicated to be carried out quantitative analysis, we assume that there is no reinsurance. Different from the approaches on the optimal decision of investment and capital level which uses backward dynamic programming, we here use the proper order dynamic programming since we assume that the total frictional cost, $FC(T)\geq 0$, that is, the boundary condition at the end of insurance term is known. Since the difference between the decision in economic and financial environment and engineering is that the former is much more uncertain than the later. It is difficult to estimate the states in all stages exactly in uncertain environment of economy and finance, especially, in the late stages of whole term when the decision term is rather long. Therefore, we believe the order dynamic programming is a better optimal decision method in the uncertain environment. $FC_t,\ t=0,1,2,....,T$ is defined as the sum the frictional cost of capital, and the expected cost of bankruptcy[12] at time *t*, that is:

$$FC_t=\sum_{i=1}^{m_1^t} n_{1t}^{xi}P_{1t}^{xi}+\sum_{j=1}^{m_2^t} n_{2t}^{xj}P_{2t}^{xj}+c_cK(t)$$
$$-c_fE\big(X(t)/X(t)<0\big)+C_a \tag{1}$$

where *K*(*t*) is the capital needed to be raised or to be reduced at time *t*, and we refer to it as the adjustment capital required, C_a, where $C_a\geq 0$, is the adjustment cost associated with raising or shedding external capital[3], *c_c*, is the ratio of frictional capital cost and c_f is the ratio of total bankruptcy cost to firm value (also called the coefficient of bankruptcy cost).

By establishing the objective function of minimizing the sum of the frictional cost of capital, and the expected cost of bankruptcy, we can find the optimal amount of risky asset to invest, the optimal capital level, and therefore the optimal risk-based capital.

The objective function is: Minimize

$$FC_t=\sum_{i=1}^{m_1^t} n_{1t}^{xi}P_{1t}^{xi}+\sum_{j=1}^{m_2^t} n_{2t}^{xj}P_{2t}^{xj}+$$
$$+c_cK(t)-c_fE\big(X(t)/X(t)<0\big)+C_a \tag{2}$$

where *K*(*t*) is the capital needed to be raised or to be reduced at time *t*. When *K*(*t*) > 0, it means that the insurer raises additional external capital of *K*(*t*); otherwise, the insurer reduces capital by *K*(*t*) either by paying dividends or repurchasing shares. By solving the objective function for each year, we can determine the optimal capital level, reinsurance, and investment strategy.

IV. STOCHASTIC DIFFERENTIAL EQUATIONS FOR CALCULATING SURPLUS OF LIFE INSURERS

The difference between property-liability insurers and life insurers is that for life insurers, we need to consider different kinds of risks including mortality risk, longevity risk, surrender risk, and investment risk.

Assume that the surplus of life insurance satisfies with the following stochastic differential equation:

[1] Bankruptcy costs can broadly be defined as either direct bankruptcy costs or indirect bankruptcy costs. Direct bankruptcy costs are those explicit costs paid by the debtor in reorganization/liquidation process including legal, accounting, filing and other administrative costs related to the liquidation of the firm's assets. Indirect bankruptcy costs are the opportunity costs of lost management energies [which could lead to] lost sales, lost profits, the higher cost of credit, or possibly the inability of the enterprise to obtain credit or issue securities to finance new opportunities (see [1]).

[2] We assume that there are no costs associated with adjusting to the optimal level of capital.

[3] For multi-period optimization models, it is important to consider the adjustment cost because the adjustment cost will affect the interval of adjusting the capital to the target value (There are significant works on this issue by Leary and Roberts 2005; Flannery and Rangan, 2006; Strebulaev, 2007).

$$dX(t) = d\left(A(t) - L(t)\right) =$$
$$\left(\sum_{i=1}^{m_1^t} n_{1t}^{xi} P_{1t}^{xi} + \sum_{j=1}^{m_2^t} n_{2t}^{xj} P_{2t}^{xj} + \pi X(t)\left(\mu - r\right)\right) dt$$
$$+\left(X(t)r - T_t - E_t - S_t - R_t - X_t r_c\right) dt$$
$$+\pi X(t)\sigma_1 dW_1 + \sum_{x=1}^{v}\sum_{i=1}^{m_1^t} T^{xi}\sqrt{n_{1t}^{xi}}\sigma_{t(xi)}^{T} dW_2^{xi} + \qquad (3)$$
$$+\sum_{x=1}^{v}\sum_{j=1}^{m_2^t} E^{xi}\sqrt{n_{2t}^{xi}}\sigma_{t(xj)}^{E} dW_3^{xi} + \sum_{x=1}^{v}\sum_{i=1}^{m_1^t}\sigma_{t(xi)}^{ST} dW_4^{xi} +$$
$$+\sum_{x=1}^{v}\sum_{j=1}^{m_2^t}\sigma_{t(xj)}^{SE} dW_5^{xi}$$

with the boundary condition $X(0) = X_0$ with Model 1, $X(0) = SCR_0$ (referring to Solvency Capital Requirement) with Model 2 and $X(0) = TC_0$ (referring to Target Capital) with Model 3, where W_1, W_2^{xi}, W_3^{xi}, W_4^{xi} and W_5^{xi} are independent Brownian Motions, $m_t = m_1^t + m_2^t$. We have

$$T_t = \sum_{x=1}^{v}\sum_{i=1}^{m_1^t}\left({}_t q_x - {}_{t-1}q_x\right) n_{1t}^{xi}, \qquad (4)$$

$$E_t = \sum_{x=1}^{v}\sum_{i=1}^{m_2^t} {}_t p_x n_{2t}^{xi}, \qquad (5)$$

$$S_t = \sum_{x=1}^{v}\sum_{i=1}^{m_1^t} {}_t p_x s_t^E R_{t(xi)}^{E}(1-\alpha) + \sum_{x=1}^{v}\sum_{i=1}^{m_2^t} {}_t p_x s_t^T R_{t(xi)}^{T}(1-\alpha), \quad (6)$$

and

$$R_t = \sum_{x=1}^{v}\sum_{i=1}^{m_1^t} R_{t(xi)}^{E} + \sum_{x=1}^{v}\sum_{i=1}^{m_2^t} R_{t(xi)}^{T}. \qquad (7)$$

Based on [7], we let the random variable D_{xt} denote the number of deaths in a population at age *x* in period between *x* and $x+t$. Let the random variable L_{xt} denote the number of survivors in a population at age *x* in period *t*, and let ω_{xt} be a dummy variable with $\omega_{xt} = 1$, when $e_{xt}^j > 0$ and $\omega_{xt} = 0$, when $e_{xt}^j = 0$, then volatility of mortality is

$$E(D_{xt}) = e_{xt}^j {}_t q_x,$$
$$\sigma_{t(xi)}^{T} = \frac{\sqrt{Var(D_{xt})}}{\omega_{xt}},$$
$$Var(D_{xt}) = V(E(D_{xt})),$$
$$V(u) = u\left(1 - \frac{u}{e_{xt}^j}\right). \qquad (8)$$

where the estimated life expectancy is:

$$\hat{e}_{xt}^j = \frac{\sum_{j>0} L_{xj}(t+j)\left(1 - \frac{1}{2}\hat{q}_{x+j}(t+j)\right)}{L_x(t)}.$$

We also have

$$E(L_{xt}) = e_{xt}^j - e_{xt}^j {}_t q_x,\ \sigma_{t(xi)}^{E} = \frac{\sqrt{Var(L_{xt})}}{\omega_{xt}},$$
$$Var(L_{xt}) = V(E(L_{xt})),\ V(u) = u\left(1 - \frac{u}{e_{xt}^j}\right)$$
$$(9)$$

$$R_{t(xi)}^{T} = \left(R_{t-1(xi)}^{T} + n_{1t}^{xi} P_{1t}^{i}\right)\frac{L_{x(t+1)}}{L_{xt}}(1+r) - \frac{D_{xt}}{L_{xt}}, \qquad (10)$$

$$R_{t(xi)}^{E} = \left(R_{t-1(xi)}^{E} + n_{2t}^{xi} P_{2t}^{i}\right)\frac{L_{x(t+1)}}{L_{xt}}(1+r) - \frac{L_{x(t+1)}}{L_{xt}}. \qquad (11)$$

V. OPTIMIZATION MODEL BASED ON SOLVENCY II (MODEL 2)

We use Value at Risk VaR with $1-\alpha =$ 99.5% of the net asset as the Solvency Capital Requirement (*SCR*) defined by Solvency II. Given confidence level $\alpha \in (0,1)$, the VaR of the net assets at the confidence level $1-\alpha$ is given by the smallest number l such that the probability of the loss X exceeding l is not larger than α. The SCR at time *t* is:

$$SCR_t = VaR_\alpha(\Delta X_t) =$$
$$-\inf\{\Delta X_t \in \Re : P\left(\Delta X_t > l\right) \le \alpha\} = \qquad (9)$$
$$-\inf\{l \in \Re : F_{\Delta X_t}(l) \ge \alpha\}$$

where $X(t)$ satisfies stochastic differential equation (3).

In a fashion to the one discussed above, we establish the objective function of minimizing the total frictional cost with the constraint that $\Pr\left(\Delta X_t \le -SCR_t\right) = \alpha$, that is, we minimize

$$FC_t = \sum_{i=1}^{m_1^t} n_{1t}^{xi} P_{1t}^{xi} + \sum_{j=1}^{m_2^t} n_{2t}^{xj} P_{2t}^{xj} +$$
$$c_c K(t) - c_f E\left(X(t) / X(t) < 0\right) + C_a$$

subject to:

$$\Pr\left(\Delta X_t \le -SCR_t\right) = \alpha,\ t = 1,2,...,T.$$

Note that times considered start with *t* = 1, because the company is assumed to satisfy the regulatory capital requirements at time 0, otherwise it would not be able to continue its existence from that point.

VI. OPTIMIZATION MODEL BASED ON SWISS SOLVENCY TEST (MODEL 3)

Swiss Solvency Test proposes the concept of target capital, which equals the one-year risk capital defined as the expected shortfall of the change of risk-bearing capital during a one-year period. The risk-bearing capital is defined as the difference between the market-consistent value of the assets and the best-estimate of the liabilities.

Based on the definitions above, we establish the formula for calculating the Target Capital (TC) at time *t*:

$$TC_t = ES_\alpha = \frac{1}{\alpha}\int_0^\alpha VaR_\gamma(\Delta X_t)d\gamma \qquad (10)$$

where X_t satisfies stochastic differential equation (3), $\Delta X_t = X_t - X_{t-1}$ and ES_α is the expected shortfall with confidential level of α . We establish the objective function of minimizing the total frictional cost with the following constraint:

$$TC_t = ES_\alpha = \frac{1}{\alpha}\int_0^\alpha VaR_\gamma(\Delta X_t)d\gamma, t = 1,2,...,T \ .$$

Note again that times considered start with *t* = 1, because the company is assumed to satisfy the regulatory capital requirements at time 0, otherwise it would not be able to continue its existence from that point.

The model presented seeks to minimize

$$\begin{aligned} &FC_t = \\ &= \sum_{i=1}^{m_1^t} n_{1t}^{xi} P_{1t}^{xi} + \sum_{j=1}^{m_2^t} n_{2t}^{xj} P_{2t}^{xj} + \qquad (11) \\ &c_c K(t) - c_f E\left(X(t) / X(t) < 0\right) + C_a \end{aligned}$$

subject to: $K(t) \geq TC_t$, where X_t satisfies the appropriate stochastic differential equation and $\Delta X_t = X_t - X_{t-1}$.

VI. CONCLUSION

In this paper, we establish three models of integrated optimization of capital, investment strategies for life insurance companies based on minimizing the total friction cost with no constraint, with the constraint of Solvency II and Swiss Solvency Test respectively. We consider the risks of mortality, investment and surrender, and establish stochastic assets and liabilities models. Further study may focus on the simulation of assets and liabilities of insurance companies and numerical determination of optimal capital and investment strategies and make some comparison among the established three models with examples.

REFERENCES

[1] Chandra, V. and M. Sherris, 2006, "Capital Management and Frictional Costs in Insurance," *Australian Actuarial Journal*, 12(4), 2006, pp. 344-399.

[2] Cummins, J. D., and R. D. Phillips, "Capital Adequacy and Insurance Risk-Based Capital Systems," *Journal of Insurance Regulation*, 28(1), 2009, pp. 25–72.

[3] Eling, M. and I. Holzmüller, "An Overview and Comparison of Risk-Based Capital Standards," *Journal of Insurance Regulation*, 26(4), 2008, pp. 31-60.

[4] Fier, S., K. McCullough, J. M. Carson, "Internal Capital Markets and the Partial Adjustment of Leverage," 2011 working paper available online (accessed June 5, 2017): http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1850488.

[5] Gatzert, N., and H. Wesker, "A Comparative Assessment of Basel II/III and Solvency II," *The Geneva Papers*, 37, 2012, pp. 539-570.

[6] Gerstner, M., M. Griebel, M. Holtz, R. Goschnick and M. Haep, A general asset-liability management model for the efficient simulation of portfolios of life insurance policies. *Insurance: Mathematics and Economics*, 42(2), 2008, pp. 704-716.

[7] Haberman, S. and A. Renshaw, "On Simulation-Based Approaches to Risk Measurement in Mortality with Specific Reference to Binomial Lee-Carter Modeling," 2008 working paper.

[8] Holzmüller, I., "The United States RBC Standards, Solvency II and the Swiss Solvency Test: A Comparative Assessment," *Geneva Papers*, 34, 2009, 56–77.

[9] Luder, T., "Swiss Solvency Test in Non-Life Insurance," 2005 working paper available online (accessed June 5, 2017): http://www.finma.ch/archiv/bpv/download/e/SST_Astin_colloquium_Luder_Thomas.pdf

[10] Mao, H. and K. Ostaszewski, "Pricing insurance contracts and determining optimal capital of insurers," *The Proceedings of International Conference of Industrial Engineering and Industrial Management*, 2010, pp. 1-5.

[11] Mao, H., J. Carson, K. Ostaszewski and W. Hao, "Integrated determination optimal economic capital, investment, and reinsurance strategies, *Journal of Insurance and Regulation*, 34(6), 2015, pp.1-29.

[12] Schmeiser, H and C. Siegel, "Regulating Insurance Groups: a Comparison of Risk-Based Solvency Models," *Journal of Financial Perspectives*, 1, 2013, pp. 119-131.

[13] Smith, M. J.-H., "Solvency II: The Ambitious Modernization of the Prudential Regulation of Insurers and Reinsurers Across the European Union (EU)," *Connecticut Insurance Law Journal*, 16, 2010, pp. 357-398.

[14] Staking, Kim, and David Babbel, "The Relation Between Capital Structure, Interest Rate Sensitivity, and Market Value in the Property-Liability Insurance Industry," *Journal of Risk and Insurance*, 62, 1995, pp. 690-718.

The Origin-Destination matrix estimation for large transportation models in an uncongested network

1st Frantisek Kolovsky[1]
Department of Geomatics
University of West Bohemia
Plzen, Czech Republic
kolovsky@kgm.zcu.cz

2nd Jan Jezek
Department of Geomatics
University of West Bohemia
Plzen, Czech Republic
jezekjan@kgm.zcu.cz

3rd Ivana Kolingerova[2]
NTIS - New Technologies for the Information Society
Department of Computer Science, University of West Bohemia
Plzen, Czech Republic
kolinger@kiv.zcu.cz

***Abstract*—To model the ever-increasing traffic volume is an important problem. The traditional transport model utilizes Origin-Destination matrix to describe how many vehicles travel between the given zones. Estimation of this matrix is usually done using a desktop-based software and so limited by memory size and CPU performance. As computation of real models containing millions of edges and tens of thousand zones is computationally very demanding, there is a big challenge to optimize it and thus enable faster calculation. Our proposed approach contributes to the solution of this problem by distributing the computation on multiple computers with the so-called Map-Reduce model and improving the convergence rate of one existing method for the matrix calibration using Conjugate gradient method. The presented approach was developed as the tool for Apache Spark and successfully tested on the transport data of the whole European Union.**

***Index Terms*—Origin-Destination matrix estimation, traffic assignment, distributed environment, Map-Reduce, traffic volume**

I. Introduction

In recent years, the traffic volume in road network has been increasing. This puts higher demands on the traffic management using intelligent systems. One of the tools for the traffic management is traffic modeling. A state-of-the-art of traffic modeling approach consists of four consecutive independent steps (trip generation, trip distribution, mode choice, route assignment), where each step models one aspect of the transport.

Nowadays these transport models are mostly created using proprietary software tools that work in single-computer desktop environment. This way of creating the transport models is limited by the computational speed and memory size. As the size of the model and computational complexity increases, there is a rise of the demand for a scalable solution that will utilize the benefits of cloud computing. Such a solution might open many new possibilities to model large-scale networks.

This work is focused on the trip distribution - the second step of the traditional transport model. The aim is to create the Origin-Destination matrix (T) that says how many vehicles travel from the place (zone) i to the place j. The travel demand in each zone and measured calibration data on the selected edges (traffic counts) are used for the T estimation.

This paper presents an improved estimation for the Origin-Destination matrix [1] so that it is suitable for large models. Our method was developed for an uncongested network, because a large network (e.g., the important roads in Europe) is usually not congested.

Our solution is based on the improvement of the convergence properties of the method by Spiess [2] by the well-known Conjugate Gradient method (CGM) and an estimate of some parameters of a deterrence function using the traffic counts. The developed algorithms were implemented on the base of Map-Reduce computing model [3]. This model is one of the most widely used for the distributed computing. The whole solution was tested on real data with good results.

Contents of the paper are as follows. First the problem is formulated and description of the state-of-the-art methods that solve this problem is given. A detailed description of the developed algorithms is presented next. After that, experiments and results are shown.

II. Background

The problem to be solved can be more precisely defined in the following way. The goal is to estimate the number of trips between all places (zones) in the area of interest. The set of all zones is called Z. The available input information for determination of the number of the trips between the zones i and j for a time unit (e.g., a day, an hour) T_{ij} is:

- a road network (graph) $G = (V, A)$; V is the set of vertices, A is the set of edges,
- the number of trips starting at zone i (O_i),
- the number of trips ending at zone j (D_j),
- the measured traffic volume (traffic count) $\widehat{v}_a$ on edges $a \in \widehat{A}$, where $\widehat{A}$ is the set of edges such that the value of $\widehat{v}_a$ is available (edges with the traffic count) and $\widehat{A} \subset A$.

The number of trips (T_{ij}) have to be determined for all i,$j \in Z$ and $i \neq j$. There are two basic methods of the Origin-Destination matrix (T) estimation:

[1]This work has been supported by the Project SGS-2016-004 ("Vyuzit matematiky a informatiky v geomatice III"). [2]This work was supported by Ministry of Education, Youth and Sports of the Czech Republic, the project PUNTIS (LO1506) under the program NPU I. Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

- accurate transport data, e.g., using license plate,
- a trip distribution model, e.g., Gravity, Gravity-Opportunity [1].

The created matrices can be calibrated using the traffic count. This calibration is labeled as the third method for T estimation in some literature [4].

The first method is the most accurate, but the equipment for it is too expensive and impractical for a large area of interest.

A. Trip distribution model

The detailed socio-economic data and information about local habits (e.g., how far people travel to work) for the area of interest are needed. The most important part of the model is a deterrence function (f), which describes the trip distribution depending on the travel cost (c_{ij}). Further, the matrix T has to satisfy the following constraints:

$$\sum_j T_{ij} = O_i \qquad \sum_i T_{ij} = D_j \tag{1}$$

The fundamental equation to determine the matrix T is

$$T_{ij} = O_i D_j A_i B_j f(c_{ij}) \tag{2}$$

where A_i and B_j are balancing factors that ensure fulfillment of the conditions 1. These factors might not be quantified, but are important for the upcoming theory [1]. As can be seen, the matrix T is dependent on itself, it is necessary to use an iterative algorithm. The first approximation of the matrix is computed using $A_i = B_j = 1$ according to 2. Every k-th iteration the elements of T are computed as

$$T_{ij}^{row} = T_{ij}^{k-1} \frac{O_i}{\sum_i T_{ij}^{k-1}} \tag{3}$$

$$T_{ij}^{k} = T_{ij}^{row} \frac{D_j}{\sum_j T_{ij}^{row}} \tag{4}$$

The algorithm is terminated when 1 is valid with a sufficient accuracy. This method for balancing the matrix is called the iterative proportional fitting (IPF) [5].

The parameters of the function f are usually determined empirically by a domain expert or using an accurate research in the area of interest. Another approach published by [6] uses the traffic counts to estimate the parameters of the deterrence function. The original model was extended for a trip purpose (p). For example the set of trip purposes contains shopping, sports, hobbies (Let us note that p is an index, not a power). Thus for all p there is one O_i^p, D_j^p, A_i^p, B_j^p, f^p. The relationship between the traffic volume (v_a) on the edge a and the travel demand (O_i, D_j) is [6]:

$$v_a = \sum_p \sum_i \sum_j O_i^p D_j^p A_i^p B_j^p f^p(c_{ij}) \delta_{ij}^a \tag{5}$$

The function δ_{ij}^a represents the shortest paths between the zones and is defined as:

$$\delta_{ij}^a = \begin{cases} 1 \text{ if the path from } i \text{ to } j \text{ contains the edge } a \\ 0 \text{ otherwise} \end{cases}$$

The least square method was used by [6] for the estimation of the parameters of the function f^p according to 5.

The creation of the matrix T in our proposed algorithm is based on the method by [6], because this method estimates the parameters of the function f (using the traffic count) unlike the original method (equation 2, 3, 4).

B. Origin-Destination Matrix calibration

This method only calibrates the existing matrix. Let us call the matrix created in the previous step $\widehat{T}$. For such a purpose, less accurate statistical data are usually sufficient as impact on the final model is insignificant. This problem can be formulated as an optimization of the objective function $F(v, T)$ to find the optimized matrix T [7]:

$$F(v, T) = \gamma_1 F_1(T, \widehat{T}) + \gamma_2 F_2(v, \widehat{v}) \tag{6}$$

where F_1 and F_2 are some matrix distance measures, $\widehat{v}$ is a vector of the traffic counts that contains $\widehat{v}_a$ for all $a \in \widehat{A}$, v is a vector of the traffic volume that is determined using the assignment function (the vector contains v_a for all $a \in A$). All-or-nothing (AON) or User Equilibrium (TAP) [8] can be used for the assignment. The parameters γ_1 and γ_2 express the importance of each function (such that $\gamma_1, \gamma_2 \in \mathbb{R}^+$ and $\gamma_1 + \gamma_2 = 1$). Further, it is necessary to set constraints for the minimization as

$$T_{ij}, v_a \geq 0$$

There are a lot of methods available in the literature solving this problem [9], split into five categories. Information Minimization (IM) and Entropy Maximization (EM) approaches are based on maximizing information entropy [10], [11]. Maximum likelihood approach maximizes the likelihood of observing T and the traffic counts [12]. The Generalized least squares method was used by [13]. This method is not suitable for large models because the matrix inversion of an approximate size $|Z|^2$ takes a lot of time. The Bayesian Inference approach provides a method for combining two sources of information [14], [15], [16], [9].

The Gradient based solution (Bi-level programing approach) is the most general optimization approach. The matrix is adjusted in each iteration using a suitable numerical method. The solution (T) converges to a local minimum. The minimization problem is formulated as [7]:

$$\min_{T_{ij} \geq 0} F(T) = \gamma_1 F_1(T, \widehat{T}) + \gamma_2 F_2(v(T), \widehat{v}) \tag{7}$$

Spiess [2] sets the parameters γ_1 and γ_2 as follows:

$$\gamma_1 = 0, \quad \gamma_2 = 1$$

then F_1 is irrelevant and the function F_2 is defined as:

$$F_2 = \frac{1}{2} \sum_{a \in \widehat{A}} (v_a - \widehat{v}_a)^2 \tag{8}$$

[1] The value of the factors can be calculated as $A_i = \prod_k \frac{O_i}{\sum_j T_{ij}^k}$ and $B_j = \prod_k \frac{D_j}{\sum_i T_{ij}^k}$, where k is the number of the iteration.

The steepest descent with a long step (gradient descent) was used for optimization of the function F. This approach is the most appropriate for large models [7], because the function F is very simple, therefore the method takes less time then other methods. One of the main problems of this method is a bad convergence rate.

Our proposed solution, which will be described in the next section, tries to solve the problem with a bad convergence using a better method for the numerical optimization. Further, all algorithms were modified for the use in the distributed computing environment.

III. The proposed algorithm

Our approach consists of two steps. The first step computes the matrix using the deterrence function and the second step calibrates it using the method based on [2]. Fig. 1 shows the computational workflow including all input and output datasets.

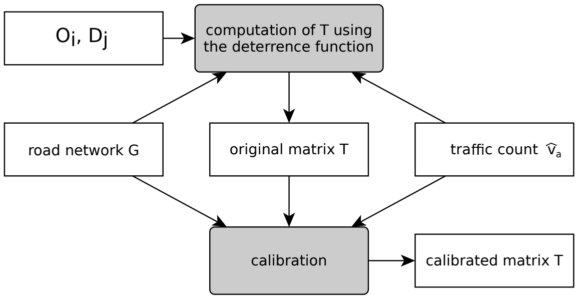


Fig. 1: Computational workflow

A. Computing the matrix T

The proposed method for determining the matrix T is derived from [6] with some adjustments. The used simplified approach has only one trip purpose. Therefore, the fundamental equation 5 can be rewritten as

$$v_a = \sum_i \sum_j T_{ij} \delta^a_{ij} \tag{9}$$

where

$$T_{ij} = \kappa O_i D_j A_i B_j f(c_{ij}) \tag{10}$$

and the constraints for the matrix T are

$$O_i = \sum_j O_i D_j A_i B_j f(c_{ij}) \tag{11}$$

$$D_j = \sum_i O_i D_j A_i B_j f(c_{ij}) \tag{12}$$

and the deterrence function was chosen as

$$f(c_{ij}) = c^{\alpha}_{ij} e^{-\beta c_{ij}} \tag{13}$$

More details about the deterrence function and a proper set of parameters can be seen in [1]. The same iterative method was used for balancing the matrix as in the previous section, c_{ij} represents the travel cost between the zone i and j. The travel costs c_{ij} were computed using Dijkstra's algorithm [17].

In the described model (equations 9 to 13) there are three unknown parameters (α, β, κ). These parameters should be estimated first. The least squares method (LSM) was used for this purpose. The problem can be written as:

$$\min_{\alpha,\beta,k} \sum_{a\in\widehat{A}} (v_a - \widehat{v}_a)^2 \tag{14}$$

Unfortunately, the problem 14 has an analytical solution only for the parameter κ, while the balancing factors A_i and B_j have to be computed iteratively. The solution is:

$$\kappa = \frac{\sum_{a\in\widehat{A}} \widehat{v}_a v'_a}{\sum_{a\in\widehat{A}} v'^2_a} \tag{15}$$

where v'_a is the derivative of v_a with respect to κ:

$$v'_a = \frac{dv_a}{d\kappa} = \sum_i \sum_j O_i D_j A_i B_j f_{ij} \delta^a_{ij} \tag{16}$$

The other two parameters were estimated by domain experts in our team, because the numerical computation of these values (e.g., using the simplex algorithm) is infeasible for large models (as verified by the authors of this paper).

The computation of the T matrix can be efficiently done by the Map-Reduce algorithm [3]. It consists of two basic parts. The first part computes the shortest paths between all zones (using Dijkstra's algorithm) and stores the identifiers of these edges $a \in \widehat{A}$, which lie on the shortest path between the zones i and j (the function δ^a_{ij}). The travel cost (c_{ij}) computed in the previous step is used to determine the matrix T according to 10, where the parameter $\kappa = 1$. The second part estimates κ according to 15.

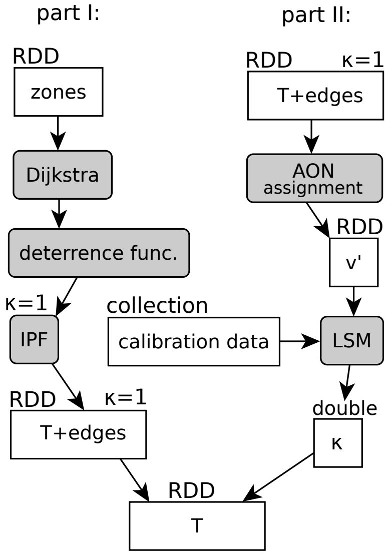


Fig. 2: Data flow of the target ODM computation

The calibration data collection contains the identifiers of the edges and measured values of the traffic on these edges. In Fig. 2 you can see the entire data flow of the T computation. The white box indicates a dataset and the light gray box indicates a process. The label RDD (Resilient Distributed Dataset) means that the dataset is distributed among cluster nodes.

B. Calibration

The method based on the approach by [2] was chosen for T calibration. Unlike Spiess' approach the CGM was used for the minimization of the objective function F. Spiess supposes that the assignment is the Wardrop's (User) equilibrium assignment (TAP). In this case the drivers use r paths with an equal travel cost between a pair of zones. It is assumed that the network is uncongested (the travel cost is independent of the traffic volume), therefore the AON assignment method is used. Therefore, the mathematical model has to be modified. The objective function F is the same as F_2 in 8. The modeled values of the traffic volume v_a are calculated according to 9 (AON). One iteration of the optimization algorithm is composed of three steps, which are:

1) determining the search direction,
2) searching a minimum of the objective function in the search direction (line search),
3) updating the T.

There are a lot of methods for determining the search direction available in the literature. The main measure of performance is the convergence rate (a speed).

Search direction as a negative value of the gradient: The simplest approach computes the search direction as the negative value of the gradient ($-\nabla F(T)$). This method is called the steepest descent. So the gradient of the objective function 8 is:

$$g_{ij} = \frac{\partial F(T)}{\partial T_{ij}} = \sum_{a \in \widehat{A}} \delta_{ij}^{a} (v_a - \widehat{v}_a) \tag{17}$$

In this case, the convergence rate is poor, because the zig-zag effect occurs. This effect is significant in a narrow valley. Fig. 3 shows the zig-zag effect (the dashed line), x_0 is the initial solution, which converges to the local minimum x.

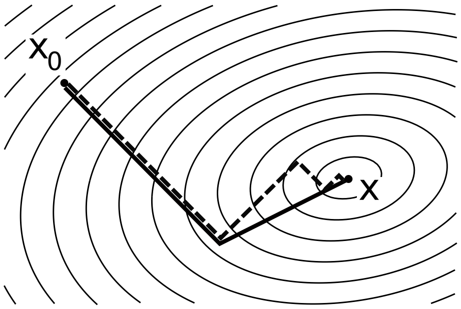


Fig. 3: Steepest descent with long step (dashed line) and CGM (solid line)

Search direction using Conjugate gradient method: The Conjugate gradient method provides a better approach for determining the search direction. In Fig. 3 there is a comparison of the Conjugate gradient with the method of the steepest descent in 2D. The method of the steepest descent (dashed line) slowly converges to the minimum while the Conjugate gradient method (solid line) finds the minimum of the quadratic problem after two steps.

The search direction of the k-th iteration can be written using the Conjugate gradient method as:

$$\mathbf{d}_k = \mathbf{g}_k + \beta_k \mathbf{d}_{k-1} \tag{18}$$

where $\mathbf{d}_k$ is the search direction and $\mathbf{g}_k = \nabla F(\mathbf{T}_k)$, where $\mathbf{T} = (T_{12}, T_{13}, \cdots, T_{|Z||Z|-1})$. The linear coefficient β_k can be determined using several techniques. The well-know PolakRibire method [18] was used. β_k can be expressed as:

$$\beta_k = \frac{(\mathbf{g}_k - \mathbf{g}_{k-1})^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \tag{19}$$

The Conjugate gradient method depends on the vectors $\mathbf{g}_{k-1}$ and $\mathbf{d}_{k-1}$, therefore it requires more memory than the method of the steepest descent.

Now the matrix T can be updated using the search direction $\mathbf{d}_k$ as

$$\mathbf{T}_{k+1} = \mathbf{T}_k \circ (\mathbf{1} - \lambda_k \mathbf{d}_k) \tag{20}$$

where $\circ$ is the element-wise multiplication and $\mathbf{1}$ is all-ones vector. The step $\lambda_k \in \mathbb{R}$ is determined using a minimization of the subproblem, which is defined as:

$$\min_{\lambda} F(\mathbf{T}_k \circ (\mathbf{1} - \lambda_k \mathbf{d}_k)) \tag{21}$$

subject to

$$\lambda_k d_{ij} \leq 1 \tag{22}$$

where d_{ij} is one element of $\mathbf{d}$ (the same construction as the vector $\mathbf{T}$). This subproblem has an analytical solution equal to

$$\lambda^* = \frac{\sum_{a \in \widehat{A}} v'_a (\widehat{v}_a - v_a)}{\sum_{a \in \widehat{A}} {v'_a}^2} \tag{23}$$

where v'_a is the derivative of v_a with respect to λ_k

$$v'_a = \frac{dv_a}{d\lambda_k} = -\sum_{ij} T_{ij} d_{ij} \delta_{ij}^{a} \tag{24}$$

The coefficient λ^* must obey the constraint 22.

The Map-Reduce algorithm, which calibrates T, can be split into three parts, initialization, main loop and line search (Fig. 4). The initialization part computes the shortest paths between all zones and associates the edges $a \in \widehat{A}$ with the pair of zones ij (the function δ_{ij}^{a}). This is the same as in the previous algorithm.

At the beginning of each iteration the algorithm computes the gradient $\mathbf{g}_k$ using 17. The matrix T, function δ_{ij}^{a} and the calibration data were used for this purpose. The gradient $\mathbf{g}_k$ and the old gradient $\mathbf{g}_{k-1}$ are used for determining the value of β_k, which together with $\mathbf{g}_k$ and the old direction $\mathbf{d}_{k-1}$ determine the new search direction $\mathbf{d}_k$.

The last part of the algorithm called line search searches for the optimal step λ_k of the CGM. As follows the step must be bounded according to constraints 22. First the interval for bounding must be computed.

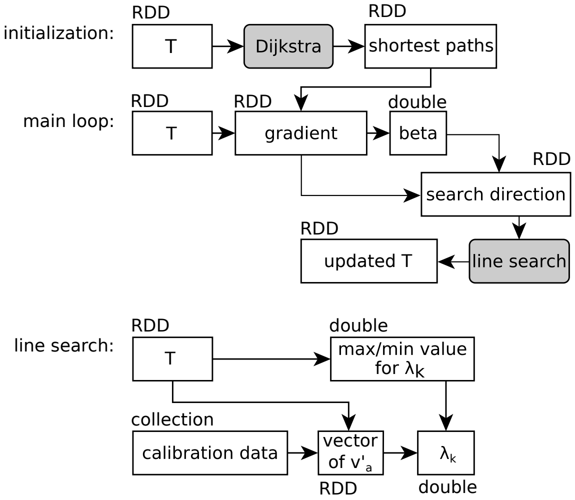


Fig. 4: Data flow of the ODM calibration

At the end of the iteration, the T is updated according to 20. The algorithm terminates when reaching a sufficiently small difference of the objective function values between two iterations or when reaching the maximum number of iterations (used for a large network).

IV. Experiments and Results

All algorithms that were described in the previous text were implemented in the Apache Spark (https://spark.apache.org). This is a framework for large-scale cluster data processing, whose priority is generality and speed. Spark's abstraction is a distributed collection of data called the RDD, which is distributed among cluster nodes. The correctness of the proposed method was tested manually on small datasets. Next we will present tests oriented to speed measurements.

Three datasets were created for testing the developed algorithms (computing and calibration of the T). In Tab. I there are parameters of these datasets. The model of city of Pilsen is small-sized and was used primarily for the development because all algorithms take less time to compute this model. The second model represents the Czech Republic. The set of all zones contains every village and all city districts. The road network consists of all roads, including all streets. The last model covers all the Europe. It includes all roads from motorways to the 3rd class roads. The set Z of the zones is represented by LAU 2 (Local Administrative Units) according to the European Union. The parameters of the deterrence function α, β were set to values 0 and 0.1.

TABLE I: Datasets for testing

dataset name	# edges $\|A\|$	# zones $\|Z\|$	$\|\widehat{A}\|$
City of Pilsen	12 207	115	60
Czech Republic (CR)	2 596 030	22 492	6 407
Europe	4 946 493	156 812	6 381

Testing was realized on two types of hardware. The dataset City of Pilsen was tested on a laptop (Intel(R) Core(TM) i5-4300M CPU @ 2.60GHz, 8GB RAM). The other two models were calculated on the cluster, which runs in YARN mode and has 24 nodes. Every node contains 4 cores (Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz) and has 64 GB RAM. So the program can use 1.5TB of memory and 96 cores.

TABLE II: Performance results for all datasets

dataset name	size of T	comp. the T time [h:m:s]	calibration time [h:m:s]	# iter
City of Pils.	10 302	00:00:05	00:00:15	30
CR	421 686 118	00:41:00	06:00:00	20
Europe	829 051 938	01:36:00	31:06:00	30

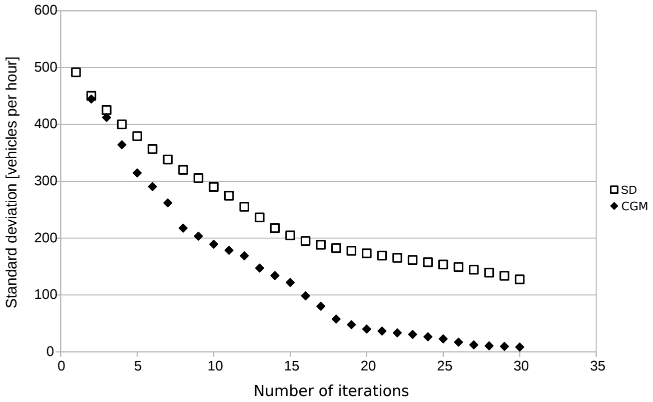


Fig. 5: Convergence rate for the implemented minimization algorithms for the Pilsen model (SD - steepest descent)

In Tab. II there are performance results. The size of T represents the number of the zone pairs ij (the number of cells in the T). For the Europe model the distance constraint was set for the Dijkstra search, because the influence of relationship between far zones is insignificant and the matrix would be too big to be computed. As can be seen in Tab. II, the T computation and calibration by our approach is reachable even for big data.

In Fig. 5 there is a comparison between two methods for determining the search direction for the Pilsen model. As can be seen, the CGM has a better convergence rate for the Pilsen model than the classic steepest descent.

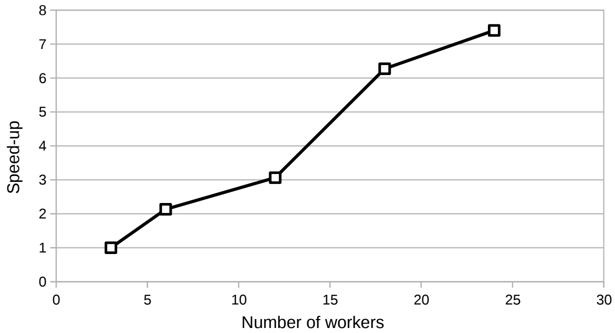
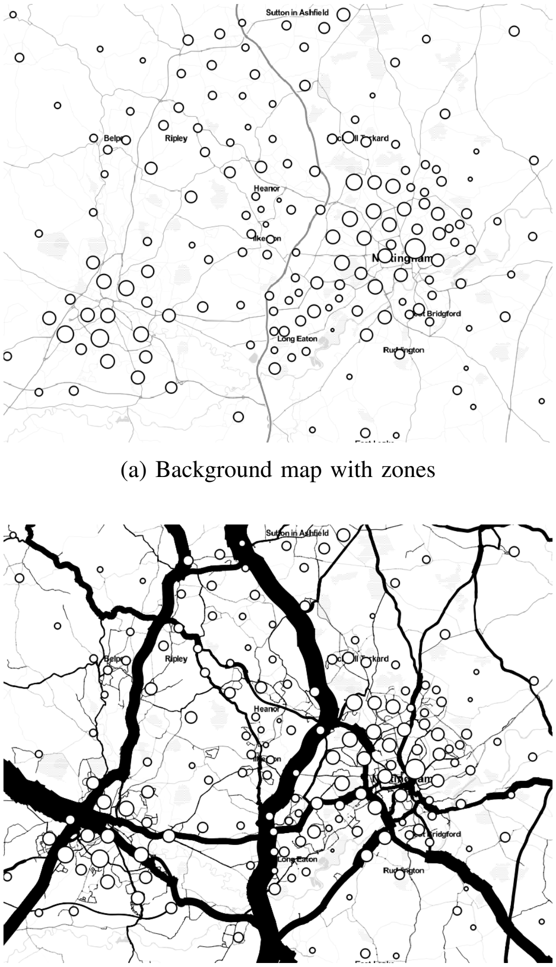


Fig. 6: Application runtime dependent on the number of computing nodes

The last performed test examines the dependence between the speed-up and the number of workers (n). The speed-up



(a) Background map with zones

(b) Traffic volumes

Fig. 7: Part of Europe transport model around Nottingham

(S) is defined as

$$S = \frac{t_3}{t_n} \tag{25}$$

where t_3 is the runtime for three workers and t_n is the runtime for n workers. This definition was chosen because the computation was unusably slow on less than three workers. In Fig. 6 there are results of this test. As can be seen, the dependence between the speed-up and the number of workers is nearly linear.

The results show that the Map-Reduce computing model is suitable for such a problem and that the proposed approach can estimate a huge matrix T. In Fig. 7 you can see part of the Europe model around Nottingham in England. The area of the circles represents O_i and the width of lines represents the traffic volume on the roads.

V. Conclusion

The tests and benchmarks show that the proposed method, where CGM was used to determine the search direction, is suitable to estimate a huge matrix using the traffic count. The solution can create larger models than the standard software for the traffic modeling based on a desktop environment and is scalable. This property was used to create the model of the whole Europe. The model contains approximately 150 000 zones and the road network has 5 000 000 links. This model was used to calculate the traffic volume for an open dataset OpenTransportMap (OTM), which is part of the project OpenTransportNet (OTN).

In the future, the more sophisticated transport model, the Wardrop's equilibrium traffic assignment should be implemented and tested in the distributed environment. It would also be appropriate to examine convergence of the Conjugate gradient method.

References

[1] J. de Dios Ortuzar and L. G. Willumsen, *Modelling Transport*, L. John Wiley & Sons, Ed. John Wiley & Sons, Ltd, 2011, iSBN 978-0-470-76039-0.

[2] H. Spiess, "A gradient approach for the od matrix adjustment problem," vol. 1, p. 2, 1990.

[3] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[4] J. Doblas and F. G. Benitez, "An approach to estimating and updating origin–destination matrices based upon traffic counts preserving the prior structure of a survey matrix," *Transportation Research Part B: Methodological*, vol. 39, no. 7, pp. 565–591, 2005.

[5] W. E. Deming and F. F. Stephan, "On a least squares adjustment of a sampled frequency table when the expected marginal totals are known," *The Annals of Mathematical Statistics*, vol. 11, no. 4, pp. 427–444, 1940.

[6] O. Z. Tamin and L. G. Willumsen, "Transport demand model estimation from traffic counts," *Transportation*, vol. 16, no. 1, pp. 3–26, 1989. [Online]. Available: http://dx.doi.org/10.1007/BF00223044

[7] J. T. Lundgren and A. Peterson, "A heuristic for the bilevel origin–destination-matrix estimation problem," *Transportation Research Part B: Methodological*, vol. 42, no. 4, pp. 339–354, 2008.

[8] J. G. Wardrop, "Road paper. some theoretical aspects of road traffic research." *Proceedings of the institution of civil engineers*, vol. 1, no. 3, pp. 325–362, 1952.

[9] S. Bera and K. Rao, "Estimation of origin-destination matrix from traffic counts: the state of the art," 2011.

[10] H. J. Van Zuylen and L. G. Willumsen, "The most likely trip matrix estimated from traffic counts," *Transportation Research Part B: Methodological*, vol. 14, no. 3, pp. 281–293, 1980.

[11] H. J. van Zuylen and D. M. Branston, "Consistent link flow estimation from counts," *Transportation Research Part B: Methodological*, vol. 16, no. 6, pp. 473–476, 1982.

[12] H. Spiess, "A maximum likelihood model for estimating origin-destination matrices," *Transportation Research Part B: Methodological*, vol. 21, no. 5, pp. 395–412, 1987.

[13] E. Cascetta, "Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator," *Transportation Research Part B: Methodological*, vol. 18, no. 4-5, pp. 289–299, 1984.

[14] M. Maher, "Inferences on trip matrices from observations on link volumes: a bayesian statistical approach," *Transportation Research Part B: Methodological*, vol. 17, no. 6, pp. 435–447, 1983.

[15] S. S. Dey and J. D. Fricker, "Bayesian updating of trip generation data: combining national trip generation rates with local data," *Transportation*, vol. 21, no. 4, pp. 393–403, 1994.

[16] T. Abrahamsson, "Estimation of origin-destination matrices using traffic counts-a literature survey," 1998.

[17] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[18] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *Revue française d'informatique et de recherche opérationnelle, série rouge*, vol. 3, no. 1, pp. 35–43, 1969.

Fake p-Values and Mendel Variables: Testing Uniformity and Independence

Maria de Fátima Brilhante
Universidade dos Açores
Centro de Estatística e Aplicações
Ponta Delgada, Portugal
maria.fa.brilhante@uac.pt

Sandra Mendonça
Universidade da Madeira
Centro de Estatística e Aplicações
Funchal, Portugal
smfm@uma.pt

Dinis Pestana
Universidade de Lisboa
Centro de Estatística e Aplicações
Instituto Bento da Rocha Cabral
Lisboa, Portugal
ddpestana@fc.ul.pt

Maria Luísa Rocha
Universidade dos Açores
Centro de Estatística e Aplicações
Ponta Delgada, Portugal
rocha_maria@sapo.pt

Fernando Sequeira
Universidade de Lisboa
Centro de Estatística e Aplicações
Lisboa, Portugal
ferseque@fc.ul.pt

Sílvio Velosa
Universidade da Madeira
Centro de Estatística e Aplicações
Funchal, Portugal
silviov@staff.uma.pt

***Abstract*—Combining p-values assuming independence and uniformity can be misleading, namely since the "Mendel temptation" to repeat experiments and then to use the most convenient p-value does exist. We discuss Mendel variables, which are mixtures of standard uniform random variables with the minimum or the maximum of two independent standard uniform random variables, that can model the presence of such fake p-values. We also consider variables X that are extremes and products of independent standard uniform random variables in $V=\min(X/Y,(1-X)/(1-Y))$, X and Y independent with support [0,1]. The stability result that X is a Mendel variable implies that V is also a Mendel variable is useful in testing autoregressive serial correlation vs. independence, and also to test non-uniform Mendel vs. uniformity using computationally augmented samples.**

***Index Terms*—uniform random variables, Mendel random variables, fake p-values, combined p-values, independence, uniformity.**

I. Introduction

The classical theory of combined p-values (Pestana, 2011) [12] assumes that those are observations of independent and identically distributed (iid) standard uniform random variables (rvs). But uniformity is solely a consequence of assuming that the null hypothesis is true, and this far-fetched assumption led Tsui and Weerahandi (1989) [15] to introduce the concept of generalized p-values, cf. also Weerahandi (1995) [16], Hung, O'Neill, Bauer and Kohne (1997) [9], and Brilhante (2013) [1].

On the other hand, Pires and Branco (2010) [13] shed some light on the Mendel-Fisher controversy showing that the suspiciously good Mendel results could be explained by assuming that experiments were repeated and only the most convenient result was reported, whenever the result of the original experiment didn't fit what the researcher wished to establish.

Under the validity of the null hypothesis a p-value is a standard uniform observation, due to the integral transform theorem. Therefore, whenever a researcher repeats the experiment and reports what he considers the most convenient of two observed p values he is in fact recording a *fake p-value* with either a Beta(2,1) or a Beta(1,2) distribution, according to whether he considers more convenient the maximum or the minimum of the two observed values.

The combination of p-values $p_1, p_2, \ldots, p_n$ must therefore consider the possibility that some of those are Beta(2,1) or Beta(1,2) fake p-values. This implies a drastic change in the underlying distribution theory, by considering that the p_k's are observations from random variables P_k that are mixtures X_m of $1-\frac{|m|}{2}$ independent uniform and of $\frac{|m|}{2}$ Beta(2,1) or Beta(1,2) random variables, that we shall call Mendel(m) random variables and denote by $X_m \sim Mendel(m)$. These mixtures have been thoroughly studied in Gomes, Pestana, Sequeira, Mendonça and Velosa (2009) [8] in the context of sample computational augmentation to test uniformity.

In Section 2 we discuss Mendel random variables, $X_m \sim Mendel(m)$, $m \in [-2,2]$, in the context of tilting the standard uniform probability density function (thus no tilting, $m=0$, leaves the uniform distribution unchanged).

Section 3 is devoted to the investigation of extensions of Deng and George (1992) [6] characterization of the standard uniform random variable, using $V=\min\left(\frac{X}{Y},\frac{1-X}{1-Y}\right)$ when X and Y are independent random variables with support [0,1]. In particular we show that:

- The class of Mendel random variables is closed for the above operation, namely $V=\min\left(\frac{X_m}{X_p},\frac{1-X_m}{1-X_p}\right)\overset{\mathrm{d}}{=}X_{\frac{mp}{6}}$.
- More generally, if X and Y with support [0,1] are

Research financed by FCT, Portugal, Project UID/MAT/00006/2013.

independent, with $X \sim Mendel(m)$,

$$V = \min\left(\frac{X_m}{Y}, \frac{1-X_m}{1-Y}\right) \overset{\mathrm{d}}{=} X_{m(2\mathbb{E}[Y]-1)}.$$

Therefore $V \overset{\mathrm{d}}{\approx} X$ if $\mathbb{E}[Y] \approx 1$. Observe that Deng and George (1992) [6] characterization of the standard uniform $X \sim Mendel(0)$ shows that in that case $V \sim Uniform(0,1)$ with Y and V independent.

When meta-analyzing p-values, it is of the utmost importance to test independence — we address in particular the investigation of independence vs. autoregressive serial correlation in Section 4 — and uniformity vs. a $Mendel(m),\ m \neq 0$, setup.

The estimation of the proportion $\frac{|m|}{2}$ of fake p-values is quite complex and we don't have so far a full-proof methodology to achieve it, cf. Brilhante, Pestana, Semblano and Sequeira (2015) [4].

II. Tilting the Uniform and Mendel Variables

Let U denote a standard uniform random variable. When tilting the probability density function of U with pole (0.5,1), for $m \in [-2,2]$, we obtain a probability density function of a variable X_m given by

$$f_m(x) = \left(mx + 1 - \frac{m}{2}\right) \mathbb{I}_{(0,1)}(x). \qquad (1)$$

We shall say that $X_m \sim Mendel(m)$. It is obvious that $X_0 \sim Uniform(0,1)$, $X_{-2} \sim Beta(1,2)$ is the minimum of two independent standard uniform rvs, and $X_2 \sim Beta(2,1)$ is the maximum of two independent standard uniform rvs.

For intermediate values of $m \in (-2,0)$, X_m is a mixture of a standard uniform, with weight $1-\frac{|m|}{2}$, and Beta(1,2) rvs, and for $m \in (0,2)$ it is a mixture of standard uniform and Beta(2,1) rvs:

$$X_m = \left\{ \begin{array}{cc} U & U_{i:2} \\ \\ 1-\frac{|m|}{2} & \frac{|m|}{2} \end{array} \right. , \quad i = 1,2, \qquad (2)$$

where $i=1$ if $m \in [-2,0]$ and $i=2$ if $m \in (0,2]$, and $U_{1:2}$ and $U_{2:2}$ denote, respectively, the minimum and maximum of two independent standard uniform random variables.

Gomes, Pestana, Sequeira, Mendonça and Velosa (2009) [8] used this family of variables in the context of testing uniformity using augmented samples, and the reason to call them Mendel random variables stems out from the interesting explanation devised by Pires and Branco (2010) [13] for the outstanding performance of Mendel experiments, that Fisher accused of being too good to be true. In fact, a possible explanation is the repetition of experiments whose results weren't in accordance with Mendel theory, reporting the "best" of the two p-values obtained.

Observe also that $X_m \sim Mendel(|m|)$ is $N(p)$-max-infinitely divisible, with $N(p) = \left\{ \begin{array}{cc} 1 & 2 \\ p & 1-p \end{array} \right.$, and $p = 1-\frac{|m|}{2} \in [0,1]$, since X_m may be interpreted as a random maximum with $N(p)$ subordinator (cf., e.g., the work by Mendonça, Pestana and Ivette (2015) [11]). In fact, considering a sequence of iid rvs $\{W_n\}_{n\in\mathbf{N}}$, identically distributed to a standard uniform rv W and independent from $N(p)$ we have

$$X_m \overset{d}{=} W_{N(p):N(p)|N(p)\geq 1}.$$

III. On $\min\left(\frac{X}{Y}, \frac{1-X}{1-Y}\right)$ when X and Y are Independent Beta or Mendel Variables

Let X and Y be independent random variables with support $\mathcal{S} = [0,1]$, and define

$$V = \min\left(\frac{X}{Y}, \frac{1-X}{1-Y}\right). \qquad (3)$$

Deng and George (1992) [6] established an useful characterization of the standard uniform distribution using the random variable V in (3):

$$X \sim Uniform(0,1) \Longleftrightarrow V \sim Uniform(0,1) \qquad (4)$$

with Y, V independent.

Observe that $X \sim Uniform(0,1)$ is the $Mendel(0)$ random variable, which is a $Beta(1,1)$ random variable, or a $BetaBoop(1,1,1,1)$ of the more general $BetaBoop(p,q,\pi,\rho)$ family of random variables with probability density function given by

$$f_{p,q,\pi,\rho}(x) = c_{p,q,\pi,\rho}$$
$$x^{p-1}(1-x)^{q-1}(-\ln(1-x))^{\pi-1}(-\ln x)^{\rho-1}\mathbb{I}_{(0,1)}(x),$$

$c_{p,q,\pi,\rho} = \int_0^1 x^{p-1}(1-x)^{q-1}(-\ln(1-x))^{\pi-1}(-\ln x)^{\rho-1}\mathrm{d}x$ and $p,q,\pi,\rho > 0$, introduced by Brilhante, Gomes and Pestana (2011) [3]. Further observe that the $BetaBoop(1,1,1,n)$ random variable is the product of n independent standard uniform random variables.

In what follows we investigate the distribution of the random variables V in (3) when X is a Mendel random variable, an extreme of a sequence of independent standard uniform rvs, or the product of independent uniform rvs.

The distribution function of V is

$$\begin{aligned} F_V(z) &= \mathbb{P}(X \leq Yz) + \mathbb{P}(X \geq 1-(1-Y)z) \\ &= 1 - \int_0^1 \left[F_X(1-(1-y)z) - F_X(yz)\right] f_Y(y)\mathrm{d}y \end{aligned}$$

and its probability density function in the support [0,1] is

$$\int_0^1 \left[y f_X(yz) + (1-y) f_X(1-z+zy)\right] f_Y(y)\mathrm{d}y, \qquad (5)$$

where F_X denotes the distribution function of X and f_X and f_Y are the probability density functions of X and Y, respectively.

(a) If X is the maximum of two independent standard uniform rvs and Y is the product of two independent standard uniform rvs, X and Y independent, then $f_V(z) = \left(\frac{3}{2} - z\right)\mathbb{I}_{(0,1)}(z)$.

More generally, if X is the maximum of two independent standard uniform rvs and Y with support [0,1] has expectation $\mathbb{E}[Y]$, with X and Y independent, the probability density function of V is

$$f_V(z) = [(4\mathbb{E}[Y]-2)z + 2(1-\mathbb{E}[Y])]\,\mathbb{I}_{(0,1)}(z). \quad (6)$$

For instance, if Y is the product of n independent standard uniform rvs, then $V \sim Mendel\left(\frac{1-2^{n-1}}{2^{n-2}}\right)$. If $Y \sim Beta(p,q)$, $V \sim Mendel\left(2\,\frac{p-q}{p+q}\right)$, and in particular

- if Y is the maximum of n independent standard uniform rvs, then $V \sim Mendel\left(\frac{2(n-1)}{n+1}\right)$,
- if Y is the minimum of n independent standard uniform rvs, then $V \sim Mendel\left(\frac{2(1-n)}{n+1}\right)$,
- more generally if Y is the k-th ascending order statistic in a sequence of n independent standard uniform rvs, then $V \sim Mendel\left(\frac{4k-2n-2}{n+1}\right)$.

Also, observe that if the expectation of Y is $\frac{1}{2}$, then $V \sim Uniform(0,1)$.

Observe however that this doesn't contradict Deng and George characterization of the uniform, since in this (4) case V is not independent of Y.

In this context, it seems worthwhile to quote from Johnson, Kotz and Balakrishnan (1995, p. 286) [10]:

> "*These results provide a partial answer to the important problem of determining the family of functions g for which the uniformity of U and V implies* [...] *uniformity of* $g(U,V)$ *if U and V are independent random variables having support* (0,1). (*This is relevant to construct methods for improving pseudorandom number generators to make them give results closer to standard uniform distributions.*)"

cf. also Gomes, Pestana, Sequeira, Mendonça and Velosa (2009).

Similarly, if X is the minimum of two independent standard uniform rvs and $Y \sim Beta(p,q)$, $f_V(z) = \frac{2}{p+q}\left[p+(q-p)\,z\right]\mathbb{I}_{(0,1)}(z)$.

(b) The above results are particular cases obtained when $X \sim Mendel(m)$.

Theorem:
If X and Y are independent random variables with $X \sim Mendel(m)$, then

$$V = \min\left(\frac{X}{Y}, \frac{1-X}{1-Y}\right) \sim Mendel\left((2\mathbb{E}[Y]-1)m\right).$$

Proof:
As $f_X(x) = \left(mx+1-\frac{m}{2}\right)\mathbb{I}_{(0,1)}(x)$, from

$$f_V(z) = \int_0^1 f_Y(y)\left[y\left(mzy+1-\frac{m}{2}\right)+ \right.$$
$$\left. + \;(1-y)\left(m-mz(1-y)+1-\frac{m}{2}\right)\right] dy$$

for $z \in (0,1)$, we obtain

$$f_V(z) = (2\mathbb{E}[Y]-1)mz+1-\frac{m(2\mathbb{E}[Y]-1)}{2}.$$

□

Corollary:
Let X and Y be independent random variables, $X \sim Mendel(m)$ and $Y \sim Mendel(p)$. Then

$$V = \min\left(\frac{X}{Y}, \frac{1-X}{1-Y}\right) \sim Mendel\left(\tfrac{mp}{6}\right). \quad (7)$$

Proof:
As $\mathbb{E}[Y] = \frac{1}{2}+\frac{p}{12}$, it follows that the Mendel parameter is $(2\mathbb{E}[Y]-1)m = \frac{mp}{6}$.

□

Observe that, as for $m,p \in [-2,2]$ we have $1-\frac{|mp|}{12} \geq \max\left(1-\frac{|m|}{2}, 1-\frac{|p|}{2}\right)$, it follows that the uniform component of $V \overset{d}{=} X_{\frac{mp}{6}}$ weights more than the uniform component of either X_m or X_p.

(c) If $X \sim Beta(3,1)$ and $Y \sim Beta(n,1)$

$$f_V(z) = \frac{3}{n+1}\left[1-\frac{4z}{n+2}+\frac{(n^2+2)z^2}{n+2}\right]\mathbb{I}_{(0,1)}(z).$$

In particular, if $Y \sim Uniform(0,1)$ then $f_V(z) = \left(\frac{3}{2}-2z+\frac{3z^2}{2}\right)\mathbb{I}_{(0,1)}(z)$; and if Y is the maximum of two independent standard uniform rvs then $f_V(z) = \left(1-z+\frac{3z^2}{2}\right)\mathbb{I}_{(0,1)}(z)$.

(d) If $X \sim Beta(2,2)$ and $Y \sim Beta(n,1)$, V has probability density function

$$6\left(\left(1-\frac{2n}{n+1}+\frac{2n}{n+2}\right)z+\left(\frac{3n}{n+1}-\frac{3n}{n+2}-1\right)z^2\right)$$

in the support [0,1].

For the simple cases $Y \sim Uniform(0,1)$ and $Y \sim Beta(2,1)$ — i.e., the maximum of two independent uniform rvs — we get the same result, $f_V(z) = \left(4z-3z^2\right)\mathbb{I}_{(0,1)}(z)$.

(e) More generally, if $X \sim Beta(p,q)$ we get, $f_V(z) = \int_0^1 A(z,y)f_Y(y)\mathrm{d}y$ for $z \in (0,1)$, where $A(z,y) =$

$$\frac{y^p z^{p-1}(1-zy)^{q-1}+[1-z(1-y)]^{p-1}(1-y)^q z^{q-1}}{B(p,q)}.$$

Therefore

$$f_V(z) = \frac{1}{B(p,q)}\left[\sum_{r=0}^{q-1}\binom{q-1}{r}(-1)^r z^{p+r-1}\mathbb{E}[Y^{p+r}]\right.$$
$$\left. +\sum_{s=0}^{p-1}\binom{p-1}{s}(-1)^s z^{q+s-1}\mathbb{E}[(1-Y)^{s+q}]\right]$$

and in particular if $p=q$ we get $f_V(z) =$

$$\frac{z^p}{B(p,q)}\sum_{r=0}^{p-1}\binom{p-1}{r}(-1)^r z^{r-1}\mathbb{E}\left[Y^{p+r}+(1-Y)^{p+r}\right].$$

For $p = q = 2$, $f_V(z) =$

$$6\left[2z\left(\mathbb{E}[Y^2 - Y + \frac{1}{2}\right) - 3z^2\left(\mathbb{E}[Y^2 - Y + \frac{1}{3}\right)\right],$$

of which examples from (d) are special cases.
If $X \overset{\mathrm{d}}{=} Y \sim Beta(2,2)$, $\mathbb{E}[Y] = \frac{1}{2}$ and $\mathbb{E}[Y^2] = \frac{3}{10}$, and therefore $f_V(z) = \left(\frac{6}{5}\, z\,(3-2z)\right)\mathbb{I}_{(0,1)}(z)$.

For $p = q = 3$,

$$f_V(z) = \frac{z^3}{B(3,3)}\left\{\frac{\mathbb{E}[Y^3 + (1-Y)^3]}{z}\right.$$
$$\left. - 2\mathbb{E}\left[Y^4 + (1-Y)^4\right] + z\left[\mathbb{E}[Y^5 + (1-Y)^5\right]\right\}.$$

If $Y \sim Beta(\alpha,\alpha)$ as $\mathbb{E}[Y^k] = \mathbb{E}[(1-Y)^k]$ the above expression is very easy to compute. For instance, if $Y \sim Beta(2,2)$

$$f_V(z) = \left(12z^2 - \frac{120}{7}z^3 + \frac{45}{7}z^4\right)\mathbb{I}_{(0,1)}(z).$$

(f) The probability density function of V may also be computed conditioning on the value of X:

$$f_V(z) = \frac{1}{z^2}\int_0^1 f_X(x)$$
$$\left[x f_Y\left(\frac{x}{z}\right) - (x-1) f_Y\left(\frac{z+x-1}{z}\right)\right]\mathrm{d}x.$$

So, if $Y \sim Mendel(m)$,

$$f_V(z) = \frac{1}{z^2}\left\{\frac{m}{z}\int_0^z x^2\left[f_X(x) - f_X(1-x)\right]\mathrm{d}x\right.$$
$$+ \left(1 - \frac{m}{2}\right)\int_0^z x\left[f_X(x) + f_X(1-x)\right]\mathrm{d}x$$
$$\left. + m\int_0^z x f_X(1-x)\mathrm{d}x\right\}.$$

Thus if $f_X(x) = f_X(1-x)$ and $Y \sim Mendel(m)$ the density of V doesn't depend on the Mendel parameter:

$$f_V(z) = \frac{\int_0^z 2x\, f_X(x)\mathrm{d}x}{z^2}.$$

For instance, if $X \sim Beta\left(\frac{1}{2},\frac{1}{2}\right)$, $f_V(z) =$

$$\left(\frac{1}{z^2} + \frac{\sqrt{z-1}\,\mathrm{arcsinh}\sqrt{z-1} - \sqrt{z(1-z)}}{\pi z^2}\right)\mathbb{I}_{(0,1)}(z),$$

and if $X \sim Beta(2,2)$

$$f_V(z) = \left(4z - 3z^2\right)\mathbb{I}_{(0,1)}(z).$$

(g) If $X \overset{\mathrm{d}}{=} 1 - X$ and $Y \overset{\mathrm{d}}{=} 1 - Y$ the expression (5) may be simplified:

$$f_V(z) = 2\int_0^1 f_X(yz) f_Y(y)\mathrm{d}y.$$

For instance:

- If $X \sim Beta(3,3)$ and $Y \sim Uniform(0,1)$,
$$f_V(z) = \left(15z^2 - 24z^3 + 10z^4\right)\mathbb{I}_{(0,1)}(z).$$
- If $X \overset{d}{=} Y \sim Beta(3,3)$,
$$f_V(z) = \left(\frac{75}{7}z^2 - \frac{100}{7}z^3 + 5z^4\right)\mathbb{I}_{(0,1)}(z).$$

IV. Testing the Independence of $\boldsymbol{p}$-values

Testing independent standard uniform rvs vs. autoregressive Mendel is relevant in the context of meta-analyzing p-values.

Let $\{X_{m,i}\}$, $i \geq 0$ be a sequence of replicas of independent Mendel variables X_m, $m \in [-2,2]$. Define

$$Y_{m,i} = \rho\, Y_{m,i-1} + (1-\rho)\, X_{m,i}, \quad Y_{m,0} = X_{m,0},$$

$1 \leq i \leq n$, $\rho \in [0,1)$.

If $\rho = 0$, the sequence $\{Y_{m,i}\}$, $i \geq 0$, is the initial one. But if $\rho > 0$ there is serial correlation.
The inverse transformation, $X_{m,i} = \frac{Y_{m,i} - \rho\, Y_{m,i-1}}{1-\rho}$, with $1 \leq i \leq n$, and $J = \left(\frac{1}{1-\rho}\right)^n$, leads to,

$$f_{Y_{m,1},\ldots,Y_{m,n}}(y) = \prod_{i=1}^{n}\left(m\,\frac{y_i - \rho\, y_{i-1}}{1-\rho} + \frac{2-m}{2}\right) J\,\mathbb{I}_{\boldsymbol{S}}(y),$$

where $(y) = (y_1, ..., y_n) \in [0,1]^n$ and

$$\boldsymbol{S} = \bigcap_{i=1}^{n}\left\{(y_1,\ldots,y_n) \in [0,1]^n : 0 < \frac{y_i - \rho\, y_{i-1}}{1-\rho} < 1\right\}.$$

As $\forall i \in \{1,\ldots,n\}, 0 < \frac{y_i - \rho\, y_{i-1}}{1-\rho} < 1$ is equivalent to

$$\rho < \min_{1\leq i\leq n}\min\left\{\frac{y_i}{y_{i-1}}, \frac{1-y_i}{1-y_{i-1}}\right\} =: A(y),$$

it follows that in the case $m = 0$ the joint density of $Y_1,\ldots,Y_n$ is

$$f_{Y_1,\ldots,Y_n}(y) = J\,\mathbb{I}_{\{(y)\in[0,1]^n :\, \rho < A(y)\}}(y),$$

and we have to solve

$$\min_{1\leq i\leq n}\min\left\{\frac{X_{0,i}}{X_{0,i-1}}, \frac{1-X_{0,i}}{1-X_{0,i-1}}\right\} = \min_{1\leq i\leq n}\left\{U_1,\ldots,U_n\right\},$$

where $\{U_1,\ldots,U_n\}$ is a sequence of independent standard uniform random variables, and therefore $\min_{1\leq i\leq n}\{U_1,\ldots,U_n\} \sim Beta(1,n)$.

More generally assuming $\rho > 0$, $m = 0$,

$$\min\left\{\frac{Y_{0,i}}{Y_{0,i-1}}, \frac{1-Y_{0,i}}{1-Y_{0,i-1}}\right\} =$$

$$\min\left\{\rho + (1-\rho)\frac{X_{0,i}}{Y_{0,i-1}}, \rho + (1-\rho)\frac{1-X_{0,i}}{1-Y_{0,i-1}}\right\}$$

is uniform with support $(\rho,1]$; denote

$$\min\left\{\rho + (1-\rho)\frac{X_{0,i}}{Y_{0,i-1}}, \rho + (1-\rho)\frac{1-X_{0,i}}{1-Y_{0,i-1}}\right\} = V_{i,\rho},$$

$V = \min_{1 \le i \le n} V_{i,\rho}$. V is the ML estimator of ρ, sufficient for ρ. The likelihood function is $L(\rho) = \left(\frac{1}{1-\rho}\right)^n \mathbb{I}_{\rho \le V}$.

Therefore, reject independence if $V > 1 - \alpha^{1/n}$, the power being

$$\begin{cases} \dfrac{\alpha}{(1-\rho)^n} & \text{if} \quad \rho \le 1 - \alpha^{1/n} \\ \\ 1 & \text{otherwise} \end{cases} .$$

Finally, for a general $m \in [-2, 2]$,

$$\frac{Y_{m,i}}{Y_{m,i-1}} = \rho + (1-\rho)\,\frac{X_{m,i}}{Y_{m,i-1}}$$

and

$$\frac{1 - Y_{m,i}}{1 - Y_{m,i-1}} = \rho + (1-\rho)\,\frac{1 - X_{m,i}}{1 - Y_{m,i-1}}$$

implying that

$$\min\left(\frac{Y_{m,i}}{Y_{m,i-1}}, \frac{1 - Y_{m,i}}{1 - Y_{m,i-1}}\right) =$$

$$\rho + (1-\rho)\,\min\left(\frac{X_{m,i}}{Y_{m,i-1}}, \frac{1 - X_{m,i}}{1 - Y_{m,i-1}}\right)$$

and from the independence of the $X_{m,i}$'s from the Corollary in Section 3 we get

$$\min\left(\frac{Y_{m,i}}{Y_{m,i-1}}, \frac{1 - Y_{m,i}}{1 - Y_{m,i-1}}\right) \overset{\mathrm{d}}{=} \rho + (1-\rho)\,X_{\frac{m^2}{6}} .$$

V. Testing the Uniformity vs. Mendel(m) Distribution of p-values

Let $p_1, p_2, \ldots, p_k$ be a sequence of p-values obtained when testing some null hypothesis H_0 in independent experiments; the rationale for combining p-values under the validity of the null hypothesis is well-established, for instance Fisher (1932) [7] used $-2\sum_{i=1}^{k} \ln P_i \sim \chi^2_{2k}$ and Tippett (1931) [14] used $\min\limits_{1 \le i \le k} \{P_i\} \sim Beta(1,k)$ to test the overall validity of H_0.

But under the validity of H_0 may we assume that they are observations from $P_i \sim Uniform(0,1)$, or is it possible that some of those recorded p_k's are fake p-values?

Maintaining uniformity using standard tests may be a weak decision, resulting from the fact that there exist very few p_i's.

We can however compute

$$v_i = \min\left(\frac{p_i}{b_i}, \frac{1 - p_i}{1 - b_i}\right),\ i = 1, ..., k, \qquad (8)$$

using for instance Beta(2,1) — and thus Mendel(2) — pseudo random numbers b_i, quite easy to generate.

If the P_i's are uniform, V_i will also be uniform and independent of the initial set of p_i's, otherwise if the P_i's are $Mendel(m)$ the V_i's will be Mendel($\frac{m}{3}$). Either way, we shall now have an augmented set $\{p_1, ..., p_k, v_1, ..., v_k\}$ to test uniformity.

This procedure may indeed be repeated to have an augmented set of size $3k$, and then of size $4k$, and so on. Observe however that the Mendel parameters decay from the original m to $\frac{m}{3}$, to $\frac{m}{9}$, to $\frac{m}{27}$, and so on, and thus the generated values will be from models closer and closer to the standard uniform, cf. Brilhante, Mendonça, Pestana and Sequeira (2010) [2] and Brilhante, Pestana and Sequeira (2010) [5], and therefore with very tiny contribution to collect evidence leading to rejection of the uniformity null hypothesis.

Thus this apparently appealing recursive procedure based on the Corollary presented in Section 3 can decrease drastically the power of the test, and must be used sparingly. In fact, using Mendel pseudo-random numbers b_i in the denominator of (8) and the corollary in section 3 to artificially increase the sample size will fatally decrease the power of the test, an apparently awkward result in Gomes, Pestana, Sequeira, Mendonça and Velosa (2009) [8] when these authors considered in their simulations Y a Mendel random variable.

On the other hand, the Theorem in Section 3 opens new possibilities, since we are no longer limited to use a Mendel variable in the denominator. Indeed, if we compute

$$v_i = \min\left(\frac{p_i}{y_i}, \frac{1 - p_i}{1 - y_i}\right),,\ i = 1, ..., k,$$

where the pseudo-random numbers y_i are from Y with a chosen $\mathbb{E}[Y] \approx 1$, the v_k will be from a $V \sim Mendel\,((2\mathbb{E}[Y] - 1)m)$ as close to the $Mendel(m)$ as we wish, even though the parameter m is unknown and being subject to testing.

We performed an elementary Monte Carlo study to compare the power of the Kolmogorov-Smirnov's goodness of fit test for the null hypothesis of uniformity for a sample $(p_1, \ldots, p_k)$ and for the computationally augmented sample $(p_1, \ldots, p_k, v_1, \ldots, v_k)$, where the p_i's are observations from a $X \sim Mendel(m)$ random variable and the y_i's, used to obtain the v_i's from the random variable V, are observations from a random variable $Y \overset{d}{=} 1 - \prod_{i=1}^{10} U_i$, with $U_1, \ldots, U_{10}$ independent standard uniform random variables. Observe that $\mathbb{E}(Y) = 1 - \left(\frac{1}{2}\right)^{10} \approx 1$, and thus the Mendel's parameter of V is approximately equal to m, the value of the parameter of X, which will not be known in pratice.

In Fig. 1 we show the results for the proportion of rejections of uniformity, for the significance level 0.05. As we can observe, the power of the test does increase as we augment the sample from a size k to a size $2k$, where $k = 5, 10, 15, 20$.

These results show that if there is some evidence against uniformity in the initial sample $(p_1, \ldots, p_k)$, this will also happen in the augmented sample, with the power of the test being higher for the larger samples, as desired.

Further, we have done the same type of study for the Fisher $-2\sum_{i=1}^{k} \ln P_i \sim \chi^2_{2k}$ test on the combined p-values to decide on the overall null hypothesis. The results are shown in Fig. 2, and as we can observe, we have the same type of conclusions as those for the Kolmogorov-Smirnov's test.

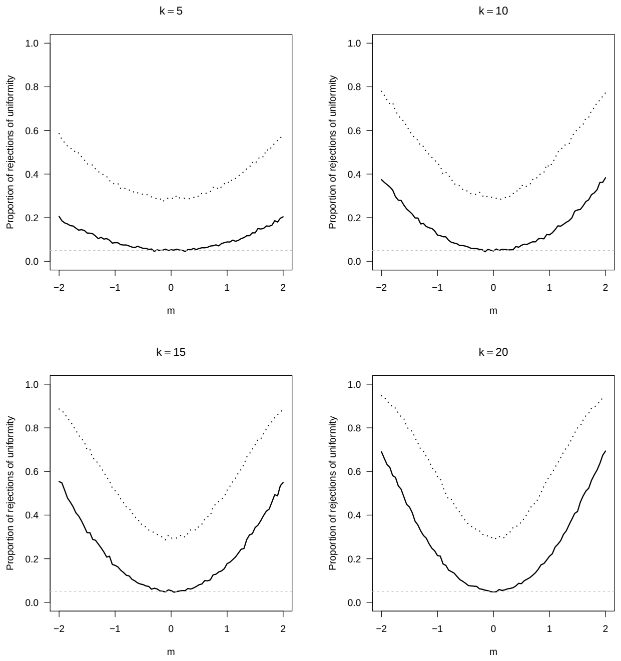


Fig. 1. Proportion of rejections of the null hypothesis of uniformity using Kolmogorov-Smirnov's test (dotted line corresponds to the augmented sample of size $2k$).

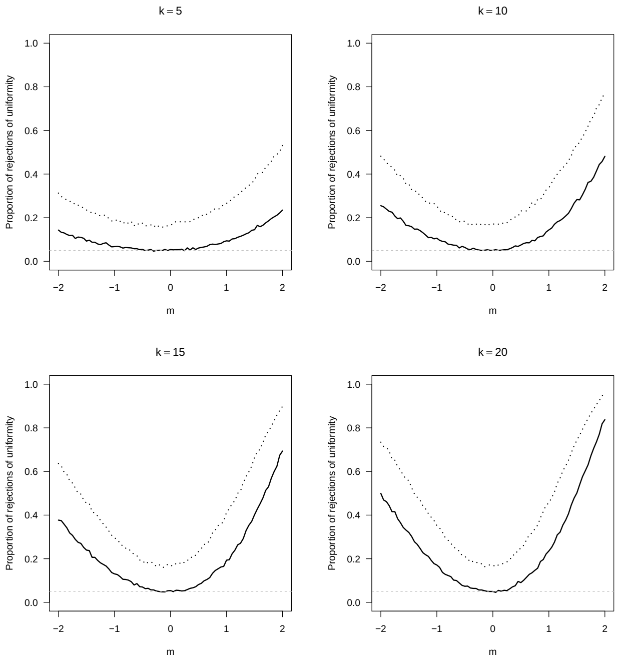


Fig. 2. Proportion of rejections of the overall null hypothesis using Fisher's test in a setting of combined p-values (dotted line corresponds to the augmented sample of size $2k$).

VI. Conclusion

The random variable V defined in (3) is very versatile, and useful namely in computational studies.

When $X \sim Mendel(m)$ in (3) the resulting V Mendel variable always has an heavier uniform component than X itself. This is drastically so when Y is also a Mendel variable, as shown in the Corollary of Section 3, and as a consequence using $Y \sim Mendel(p)$ in (8) to augment samples is pointless.

On the other hand, when the available sample of p-values is small and suspicious in the sense that some of them can be fake p-values, the use of the more general result in the Theorem of Section 3 to augment the sample size, requiring only that $\mathbb{E}[Y] \approx 1$, is sensible, since this can definitively increase power either in testing uniformity or the combined p-value.

References

[1] M. F. Brilhante, "Generalized p-values and random p-values when the alternative to uniformity is a mixture of a beta(1,2) and uniform". In Oliveira, P. et al. (eds), *Recent Developments in Modeling and Applications in Statistics*, Springer, Heidelberg, 159–167, 2013.

[2] M. F. Brilhante, S. Mendonça, D. Pestana and F. Sequeira, "Using products and powers of products to test uniformity", in Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces, IEEE CFP10498-PRT, V. Luzar-Stiffler, I. Jarec, and Z. Bekic, Eds., 2010, pp. 509–514.

[3] M. F. Brilhante, M. I. Gomes and D. Pestana, "BetaBoop Brings in Chaos", CMSim - Chaotic Modeling and Simulation Journal, vol. 1, pp. 39-50, 2011.

[4] M. F. Brilhante, D. Pestana, P. Semblano and F. Sequeira, "On the Proportion of Non Uniform Reported p-Values", in Numerical Analysis and Applied Mathematics ICNAAM 2014, AIP Conferecence Proceedings 1648, American Institute of Physics, T. E. Simos, G. Psihoyios, Ch. Tsitouras and Z. Anastassi, Eds., 2015.

[5] M. F. Brilhante, D. Pestana and F. Sequeira, "Combining p-values and random p-values", in *Proceedings of the 32nd International Conference on Information Technology Interfaces*, V. Luzar–Stiffler, I. Jarec and Z. Bekic, Eds. 2010, pp. 515–520.

[6] L.-Y. Deng and E. O. George, "Some characterizations of the uniform distribution with applications to random number generation", Annals of the Institute of Statistical Mathematics, vol. 44, pp. 379–385, 1992.

[7] R. A. Fisher, Statistical Methods for Research Workers, 4th ed., Edinburgh: Oliver and Boyd, 1932.

[8] M. I. Gomes, D. Pestana, F. Sequeira, S. Mendonça and S. Velosa, "Uniformity of offsprings from uniform and non-uniform parents", in Proceedings of the ITI 2009, 31st International Conference on Information Technology Interfaces, V. Luzar-Stiffler, I. Jarec, and Z. Bekic, Eds. 2009, pp. 243–248.

[9] H. Hung, R. O'Neill, R. Bauer and K. Kohne, "The behaviour of the p value when the alternative is true", Biometrics, vol. 53, pp. 11–22, 1997.

[10] N. L. Johnson, S. Kotz and N. Balakrishnan, Continuous Univariate Distributions, vol. 1, New York: Wiley, 1995.

[11] S. Mendonça, D. Pestana and M. I. Gomes, "Randomly Stopped *k*th Order Statistics". In *Theory and Practice of Risk Assessment*, Springer Proceedings in Mathematics & Statistics, vol 136, C. P. Kitsos, T. A. Oliveira, A. Rigas and S. Gulati, Eds., 2015, Cham: Springer, 2015.

[12] D. Pestana, "Combining p-values", in International Encyclopaedia of Statistical Science, M. Lovric, Ed. New York: Springer Verlag, 2011, pp. 1145–1147.

[13] A. M. Pires and J. A. Branco, "A statistical model to explain the Mendel-Fisher controversy", Statistical Science, vol. 25, pp. 545–565, 2010.

[14] L. H. C. Tippett, The Methods of Statistics, London: Williams & Norgate, 1931.

[15] K. Tsui and S. Weerahandi, "Generalized p-values in significance testing of hypothesis in the presence of nuisance parameters", The American Statistician, vol. 84, pp. 602–607, 1989.

[16] S. Weerahandi, Exact Statistical Methods for Data Analysis, New York: Springer, 1995.

A novel heuristic approach for solving the two-stage transportation problem with fixed-charges associated to the routes

Ovidiu Cosma
Department of Mathematics and Computer Science
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
ovidiu.cosma@cunbm.utcluj.ro

Petrica Pop
Department of Mathematics and Computer Science
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
petrica.pop@cunbm.utcluj.ro

Cosmin Sabo
Department of Mathematics and Computer Science
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
cosminsabo@gmail.com

***Abstract*—In this paper, we are addressing the two-stage transportation problem with fixed charges associated to the routes and propose an efficient heuristic algorithm for the total distribution costs minimization. Our heuristic approach builds several initial solutions by processing customers in a specific order and choosing the best available supply route for each customer. After each initial solution is built, a process of searching for better variants around that solution follows, restricting the way the transport routes are chosen. Computational experiments were performed on a set of 20 benchmark instances available in the literature. The achieved computational results show that our proposed solution approach is highly competitive in comparison with the existing approaches from the literature.**

Keywords—two-stage transportation problem, heuristic algorithms

I. Introduction

This paper focuses on a variant of the transportation problem, namely the two-stage transportation problem with fixed charges associated to the routes. The problem models a distribution network in a two-stage supply chain which involves: manufacturers, distribution centers and customers and its main characteristic is that a fixed charge is associated with each route that may be opened, in addition to the variable transportation cost which is proportional to the amount of goods shipped. The objective of the considered transportation problem is to identify and select the routes from manufacturers through the distribution centers to the customers satisfying the capacity constraints of the manufacturers in order to meet specific demands of the customers under minimal total distribution costs. In this form, the problem was introduced by Gen et al. [3]. This work deals with a variant of the transportation problem, namely the fixed-cost transportation problem in a two-stage supply chain network. In this extension, our aim is to identify and select the manufacturers and the distribution centers fulfilling the demands of the customers under minimal costs.

The two-stage transportation problem was first considered by Geoffrion and Graves [4]. Since then different variants of the problem have been proposed in the literature determined by the characteristics of the transportation system which models the real world application and several methods, based on relaxation techniques and on exact, heuristic and metaheuristics algorithms, have been developed for solving them.

Marin and Pelegrin [7] developed an algorithm based on Lagrangian decomposition and branch-and-bound techniques in the case when the manufacturers and the distribution centers have no capacity constraints and there are fixed costs associated to opening the distribution centers and the number of opened distribution centers is fixed and established in advance. Marin [8] proposed a mixed integer programming formulation and provided lower bounds of the optimal objective values based on different Lagrangian relaxations for an uncapacitated version of the problem when both manufacturers and distribution centers have associated fixed costs when they are used. Pirkul and Jayaraman [13] studied a multi-commodity, multi-plant, capacitated facility location version of the problem and proposed a mixed integer programming model and a solution approach based on Lagrangian relaxation of the problem. The same authors in [6] extended their model by taking into considerations the acquisition of raw material and described a heuristic algorithm that uses the solution generated by a Lagrangian relaxation of the problem. Amari [1] investigated a different version of the problem, allowing the use of several capacity levels of the manufacturers and distribution centers and developed an efficient solution approach based on Lagrangian relaxation for solving it.

Raj and Rajendran [18] proposed two scenarios for the two-stage transportation problem: the first scenario, called Scenario-1, takes into consideration fixed costs associated to the routes in addition to unit transportation costs and unlimited capacities of the distribution centers, while the second one, called Scenario 2, takes into consideration the opening costs of the distribution centers in addition to unit transportation costs.

They developed a genetic algorithm with a specific coding scheme suitable for two-stage transportation problems and as well they introduced a set of 20 benchmark instances. The same authors proposed in [17] a solution representation that allows a single-stage genetic algorithm to solve the considered problem. The major feature of these methods is a compact representation of a chromosome based on a permutation. A different genetic algorithm proposed for solving the two-stage transportation problem with fixed charge associated to the routes from manufacturers to customers was described by Jawahar and Balaji [5]. Recently, Pop et al. [16] presented, in the case of Scenario-1, a hybrid algorithm that combines a steady-state genetic algorithm with a powerful local search procedure. In the case of the two-stage transportation problem with fixed charges for opening the distribution centers, as introduced by Gen et al. [3] and called it Scenario-2 by Raj and Rajendran [18], the mentioned authors developed genetic algorithms based on sequentially getting first a transportation tree for the transportation problem from distribution centers to customers and secondly a transportation tree for the transportation problem from manufacturers to distribution centers. In both genetic algorithms, the chromosome contains two parts, each encoding one of the transportation trees. A different genetic algorithm was described by Calvete et al. [2], whose main characteristic is the use of a new chromosome encoding that provides information about the distribution centers that can be used within the distribution system.

A different variant was investigated by Molla et al. [9] in which it is considered only one manufacturer. They presented an integer programming mathematical model of the problem and proposed a spanning tree-based genetic algorithm with a Prüfer number representation and an artificial immune algorithm for solving the problem. Some remarks concerning the mathematical model of the problem were pointed out by El-Sherbiny [19]. For this variant of the two-stage transportation problem, Pintea et al. [10,12] developed some hybrid classical heuristic approaches and described an improved hybrid algorithm that combines the Nearest Neighbor search heuristic with a local search procedure for solving the problem. Recently, Pop et al. [15] proposed a novel hybrid heuristic approach which was obtained by combining a genetic algorithm based on a hash table coding of the individuals with a powerful local search procedure.

Another version of the two-stage transportation problem with one manufacturer takes into consideration the environmental impact by reducing the greenhouse gas emissions. This variant was introduced by Santibanez-Gonzalez et al. [19], dealing with a practical application occurring in the public sector. Considering this variant of the two-stage transportation problem, Pintea et al. [11] proposed a set of classical hybrid heuristic approaches and Pop et al. [14] developed an efficient reverse distribution system for solving the problem.

The variant addressed in this paper considers a two-stage transportation problem with fixed charge associated with each route that may be opened, as introduced by Gen et al. [3]. This transportation problem has been also studied by Raj and Rajendran [18], who called it Scenario-1, Jawahar and Balaji [5] and Pop et al. [16]. In all mentioned papers, the authors proposed genetic algorithms for solving the problem.

Our paper is organized as follows: in Section 2, we formally define the two-stage transportation problem with fixed-charge associated to the routes. The developed heuristic algorithm is presented in Section 3 and the computational experiments and the achieved results are presented, analyzed and discussed in Section 4. Finally, in the last section, we present the obtained results in this paper and propose some future research directions.

II. DEFINITION OF THE TWO-STAGE TRANSPORTATION PROBLEM WITH FIXED-CHARGES ASSOCIATED TO THE ROUTES

In order to provide a formal definition the considered two-stage transportation problem with fixed-charges associated to the routes, we begin by defining the sets, decision variables and parameters used in our paper:

p is the total number of manufacturers

q is the total number of distribution centers

r is the total number of customers

i is a manufacturer identifier, $i \in \{1,\ldots,p\}$

j is a distribution center identifier, $j \in \{1,\ldots, q\}$

k is a customer identifier, $k \in \{1,\ldots, r\}$

$D[k]$ is the demand of the customer k

$I[k]$ is the number of units delivered to customer k

$S[i]$ is the capacity of manufacturer i

$O[i]$ is the number of units delivered by manufacturer i

$F1[i,j]$ is the fixed transportation charge for the link from manufacturer i to distribution center j

$F2[j,k]$ is the fixed transportation charge for the link from distribution center j to customer k

$C1[i,j]$ is the unit cost of transportation from manufacturer i to distribution center j

$C2[j,k]$ is the unit cost of transportation from distribution center j to customer k

$X1[i,j]$ is the number of units transported from manufacturer i to distribution center j

$X2[j,k]$ is the number of units transported from distribution center j to customer k

The structure of the distribution system is presented in the next figure.

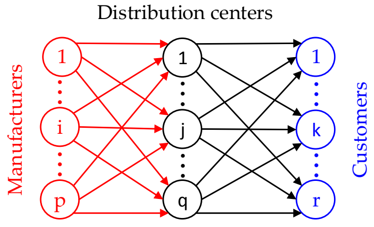


Fig. 1. The structure of the distribution system

Next we present the mathematical model of the two-stage transportation problem with fixed charges associated to the routes introduced by Jawahar and Balaji [5], based on integer programming.

Minimise

$$Z = \sum_{i=1}^{p}\sum_{j=1}^{q}(C1[i,j]X1[i,j] + F1[i,j]\delta 1[i,j]) + \\ + \sum_{j=1}^{q}\sum_{k=1}^{r}(C2[j,k]X2[j,k] + F2[j,k]\delta 2[j,k]) \tag{1}$$

Subject to

$$\sum_{j=1}^{q} X1[i,j] \leq S[i] \quad \forall i \in \{1,\dots,p\} \tag{2}$$

$$\sum_{j=1}^{q} X2[j,k] = D[k] \quad \forall k \in \{1,\dots,r\} \tag{3}$$

$$X1[i,j] \geq 0, \quad X2[j,k] \geq 0 \tag{4}$$

where

$$\delta 1[i,j] = \begin{cases} 0, if\ X1[i,j] = 0 \\ 1, if\ X1[i,j] > 0 \end{cases}, \quad \delta 2[j,k] = \begin{cases} 0, if\ X2[j,k] = 0 \\ 1, if\ X2[j,k] > 0 \end{cases}$$

The objective function (1) minimizes the total transportation cost: the fixed costs and transportation per-unit costs. Constraints (2) guarantee that the quantity shipped out from each plant does not exceed the available capacity and constraints (3) guarantee that the total shipment received from DCs by each customer is equal to its demand. Constraints 4 ensure the integrality and non-negativity of the decision variables.

III. THE HEURISTIC ALGORITHM FOR SOLVING THE TWO-STAGE TRANSPORTATION PROBLEM WITH FIXED CHARGES ASSOCIATED TO THE ROUTES

The operation of the proposed algorithm is shown in Fig. 2. It executes a fixed number of iterations that build several solution variants, of which the best is retained. The algorithm consists of the following two nested blocks:
A. Build variants,
B. Build distribution solution.

The *Build variants* block (A) calls the *Build distribution solution* block (B) to build a distribution solution, and then looks for better variants around it by applying a set of restrictions to the supply routes. The defined restrictions determine how the supply routes will be chosen within the new variants built by calling block B. The *Build distribution solution* block (B) builds a distribution solution, satisfying the demands of all customers, one by one. The resulting solution is saved only if it is better than all previous solutions.

The algorithm uses the following data structures:

– *Instance properties* (6) containing the fixed costs of opening the routes (*F1*, *F2*), the unit transport costs (*C1*, *C2*), the production capacities of the manufactures (*S*) and the demands of the customers (*D*).
– *Solution data (4)* containing the quantities transported on the routes from manufacturers to distribution centers (*X1*) and from distribution centers to customers (*X2*), the input quantities to customers (*I*) and the output quantities from manufacturers (*O*). This data structure will be updated during the construction of a solution. So at the end $I[k] = D[k], k \in \{1,\dots,r\}$ and $\sum_{i=1}^{n} O_i = \sum_{k=1}^{r} D_k$. Also this structure contains a series of restrictions for the routes from manufacturers to the distribution centers, which are applied in the route selection process (*R*).
– *Route* (5.1) that specifies a transport route for *a* units from manufacturer *i* through, distribution center *j*, to customer *k*.
– *Customers order* (2.2), which is an array containing the order in which customers will be processed by the algorithm.
– *Used permutations* (2.1), which is a hash set that contains all the permutations that were used in previous iterations of the algorithm.
– *Best solution (9)*, containing the quantities transported on the distribution routes within the optimum solution.

The algorithm works in the following way:
The *Shuffle customers* module (1) arranges the customers in random order, through the *Duplicate detector* module (2), which saves all permutations that were previously used in a hash set (*Used permutations* 2.1). Thus, any permutation that has been previously used is effectively detected and rejected. The operation ends when a permutation that was not previously used is generated.

Next, more iterations of the *Build variants* block (A) are processed. The total number of iterations is set at the initialization of the algorithm, based on the number of customers. This block builds a distribution solution by processing customers in the order given by the *Customers order* array (2.2), and then searches for more advantageous variants around this solution. For the construction of each solution, the *Reset solution* block (3) deletes the data corresponding to the previous solution, initializing the *X1*, *X2*, *I* and *O* arrays from the *Solution data* structure (4) with zeros. Thus, this structure is initialized for building a new solution.

The *Build distribution solution* block (B) builds a distribution solution, processing all customers, one by one, in the order given by the *Customers order array* (2.2). The *Route selection* module (5) seeks for each customer the most advantageous route of supply in the conditions created by meeting the demands of previous customers, resulting in the opening of some transport routes and consuming a part of the production capacity of manufacturers. The result returned by this module is a supply route (5.1). Each client's request can be satisfied in one or more steps, as the amount *a* of the route supports or not the customer's entire demand. The *Reserve route module* (7) reserves the route (5.1) by updating the *X1*, *X2*, *I* and *O* arrays from the *Solution data* structure (4). The processing of client k ends when $D[k] = I[k]$.

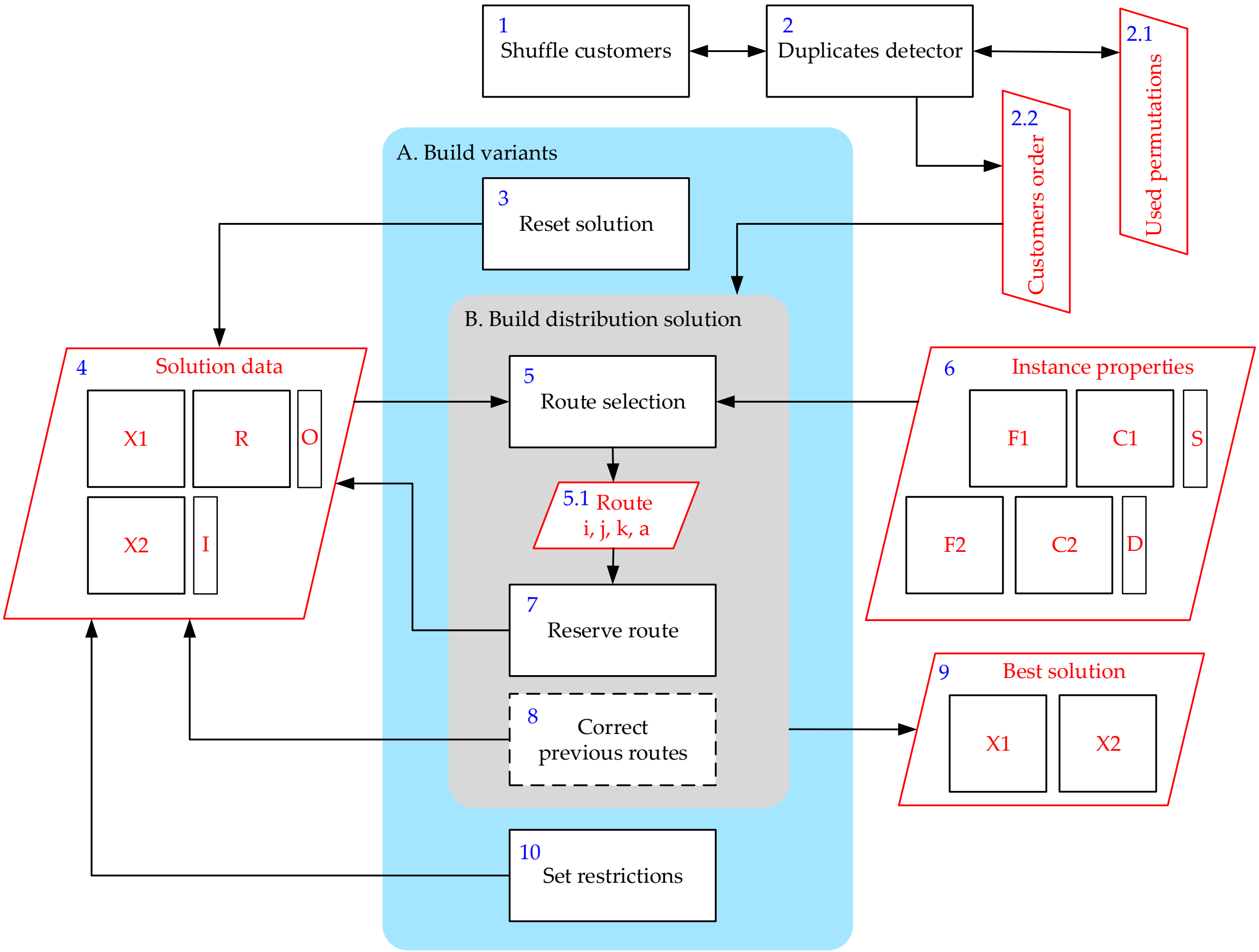


Fig. 2. The operation of the proposed heuristic algorithm

If there were no limitations on the capacities of the manufacturers and if there were no fixed costs for opening the transport routes, then modules 5 and 7 would build the optimal solution from the first attempt. But because of these restrictions, it is not certain that the optimal supply of a customer that can be found at any given time in the process of building a solution leads to the optimization of the entire distribution system. Any decision to supply a customer involves consuming a quantity of the production capacity of the manufacturers and possibly opening new transport routes. This will influence all the decisions to be taken within the algorithm. Consequently, the order in which customers are processed determines the final result. At the end of each customer's processing, an optional step of reviewing all previous decisions is applied, using the *Correct previous routes* module (8). This module deletes all previously reserved routes one after the other, and then attempts to replace them with variants that are more advantageous in the new conditions created by processing the last customer demand. The old routes are modified only if the change leads to a better solution. Using this module reduces on average the number of iterations needed to find the optimal solution, but the runtime of the algorithm increases as the complexity of the iterations increases. For comparison, the results obtained with and without this module will be presented in the *Computation Results* section (IV).

For certain distribution systems, it is not possible to reach the optimal solution simply by changing the customer processing order (using the *Shuffle customers* module 1) and the corrections performed by the *Correct previous routes* module (8). This is because the *Route selection* module (5) processes a single client at a time. This will always choose the optimal decision for each client, not the decisions optimizing the entire distribution system. Consequently, for these distribution systems, it is necessary to introduce new restrictions, which change the way decisions are made in the

Route selection module (5). These restrictions are fixed in the *Set restrictions* module (10). This module marks at each iteration one of the manufacturer-distribution center routes as mandatory. This route will be used with priority in the construction of the distribution solution until the manufacturer's production capacity is depleted. Thus, a search is made around the initial solution, by building other $p \cdot q$ variants of distribution systems.

In order to illustrate the operation of the algorithm, let's consider the example in figure 3. It represents a distribution system with one manufacturer (*M1*), two distribution centers (*DC1* and *DC2*) and two customers (*U1* and *U2*). The manufacturer's production capacity is 8 units, and *U1* and *U2* customer's demands are for 3 and 4 units, respectively. The transport costs are as follows: *C1 = {5, 3}, F1 = {10, 20}, C2[j, k] = 6, F2[j, k] = 7, j = {1, 2}, k = {1, 2}*. For this distribution system, there are two possible permutations for the customer set (*{U1, U2}*, and *{U2, U1}*), and the *Build Distribution Solution* module will attempt to build two variants of distribution systems.

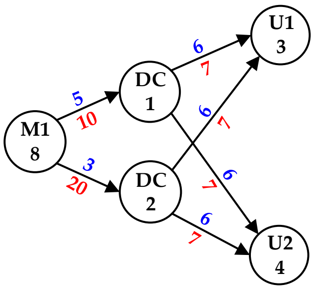


Fig. 3. A small example of a distribution system

In the first variant, the customer *U1* demand will be processed first. Of the two possible routes *M1-DC1-U1* and *M1-DC2-U1*, the first will be chosen, because it is more advantageous, resulting in a cost of (5 + 6) * 3 + 10 + 7 = 50. The cost of the second route is (3 + 6) * 3 + 20 + 7 = 54. Next the most advantageous route for the *U2* customer is chosen, which is *M1-DC1-U2*, resulting in a cost of (5 + 6) * 4 + 7 = 51. Thus, the total cost of the distribution solution is 50 + 51 = 101.

In the second variant, the most advantageous route for the U2 customer is chosen first. This is *M1-DC1-U2*, resulting in a cost of (5 + 6) * 4 + 10 + 7 = 61. Next the most advantageous route for the *U1* customer is chosen, which is *M1-DC1-U1*, resulting in a cost of (5 + 6) * 3 + 7 = 40. In conclusion, for both possible permutations, the same distribution solution is obtained in this example, which is suboptimal. This result is due to the fact that the choice of routes is always looking for the optimal solution for a particular customer in certain conditions, not the solution optimizing the whole distribution system.

The optimal solution can be reached through the action of the *Set restrictions module*, as follows: If the *M1-DC2* route is set as mandatory, it will have to be used with priority until the *M1* manufacturer's capacity is exhausted. Thus, if the customer *U1* is processed first, then the route *M1-DC2-U1* will be chosen, resulting in a cost of (3 + 6) * 3 + 20 + 7 = 54. Next the route *M1-DC2-U2* will be chosen for the *U2* customer, resulting in a cost of (3 + 6) * 4 + 7 = 43. Thus, the total cost of the distribution solution is 54 + 43 = 97.

IV. Computational Results

In order to analyze the performance of our proposed heuristic approach, we tested it on a set of 20 test instances that was generated by Gen et al. [3]. The files of the instances are available at the following address: https://sites.google.com/view/tstp-instances/.

Our algorithm was coded in Java 8 and we performed 30 independent runs for each instance on a PC with Intel Core i5-4590 processor at 3.3GHz, 4GB RAM and Windows 10 Education 64 bit operating system.

In Table 1, we show the computational results of our proposed heuristic algorithm in comparison with the genetic algorithm described by Jawahar and Balaji [4], called *JRGA*, the two genetic algorithms introduced by Raj and Rajendran [18], denoted by *TSGA* and *SSGA* and the hybrid genetic algorithm (*HGA*) described by Pop et al. [16]. The first column in the Table 1 gives the size of the instances, the next columns provide the solution achieved by the genetic algorithm described by Jawahar and Balaji [4], the two genetic algorithms introduced by Raj and Rajendran [18], the hybrid genetic algorithm described by Pop et al. [16] and the number of solution evaluations necessary to obtain it. The results written in bold represent cases for which the obtained solution is the best existing in literature.

The following columns show the results of our heuristic algorithm, obtained with and without the correction module (8). For each test instance, the running time and the number of solutions evaluated until the best solution is found are presented. Both the best values and the averages calculated for all 30 runs are presented.

Our algorithm finds the best known solution for each of the 20 test instances at each of the 30 runs, which demonstrates its robustness. In the variant without the correction block, all 20 instances are resolved in less than 1ms. In the version with the correction block, the average number of evaluated solutions is lower than in the case of the other known algorithms for all of the test instances. This is true also for the variant without the correction block, with only one exception.

References

[1] A. Amiri, "Designing a distribution network in a supply chain system: Formulation and efficient solution procedure," Eur. J. of Oper. Res., vol. 171(2), pp. 567-576, 2006.

[2] H. Calvete, C. Gale, and J. Iranzo, "An improved evolutionary algorithm for the two-stage transportation problem with fixed charge at depots," OR Spectrum, vol. 38, pp. 189-206, 2016.

[3] M. Gen, F. Altiparmak, and L. Lin, "A genetic algorithm for two-stage transportation problem using priority based encoding," OR Spectrum, vol. 28, pp. 337-354, 2006.

[4] A.M. Geoffrion and G.W. Graves, "Multicommodity distribution system design by Benders decomposition," Management Sci., vol. 20, pp. 822-844, 1974.

[5] N. Jawahar, and A.N. Balaji, "A genetic algorithm for the two-stage supply chain distribution problem associated with a fixed charge," Eur. J. of Oper. Res., vol. 194, pp. 496-537, 2009.

[6] V. Jayaraman, and H. Pirkul, "Planning and coordination of production and distribution facilities for multiple commodities," Eur. J. of Oper. Res, vol. 133(2), pp. 394-408, 2001.

[7] A. Marin, and B. Pelegrin, "A branch-and-bound algorithm for the transportation problem with locations p transshipment points," Comp. & Oper. Res, vol. 24(7), pp. 659-678, 1997.

[8] A. Marin, "Lower bounds for the two-stage uncapacitated facility location problem," Eur. J. of Oper. Res., vol. 179(3), pp. 1126-1142, 2007.

[9] S. Molla-Alizadeh-Zavardehi, M. Hajiaghaei-Kesteli, and R. Tavakkoli-Moghaddam, "Solving a capacitated fixed-cost transportation problem

by artificial immune and genetic algorithms with a Prüfer number representation,” Exp. Syst. with Appl., vol. 38, pp. 10462-10474, 2011.

[10] C.-M. Pintea, C. Pop Sitar, M. Hajdu-Macelaru, and P.C. Pop, “A Hybrid Classical Approach to a Fixed-Charge Transportation Problem,” In Proc. of HAIS 2012, Part I, Eds E. Corchado et al., Lecture Notes in Computer Science, vol. 7208, pp. 557-566, 2012.

[11] C.-M. Pintea, P.C. Pop, and M. Hajdu-Macelaru, “Classical Hybrid Approaches on a Transportation Problem with Gas Emissions Constraints” In Proc. of SOCO 2012, Advances in Intelligent Systems and Computing, vol. 188, pp. 449-458, 2013.

[12] C.M. Pintea, and P.C. Pop, “An improved hybrid algorithm for capacitated fixed-charge transportation problem,” Logic J. of IJPL vol. 23(3), pp. 369-378, 2015.

[13] H. Pirkul, and V. Jayaraman, “A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution,” Comp. & Oper. Res., vol. 25(10), pp. 869-878, 1998.

[14] P.C. Pop, C.-M. Pintea, C. Pop Sitar, and M. Hajdu-Macelaru, “An efficient Reverse Distribution System for solving sustainable supply chain network design problem,” J. of Appl. Logic, vol. 13(2), pp. 105-113, 2015.

[15] P.C. Pop, O. Matei, C. Pop Sitar, and I. Zelina, “A hybrid based genetic algorithm for solving a capacitated fixed-charge transportation problem,”Carpathian J. Math., vol. 32(2), pp. 225-232, 2016.

[16] P.C. Pop, C. Sabo, B. Biesinger, B. Hu, and G. Raidl, “Solving the two-stage fixed-charge transportation problem with a hybrid genetic algorithm,” Carpathian J. Math., vol. 33(3), pp. 365-371, 2017.

[17] K.A.A.D. Raj, and C. Rajendran, “A Hybrid Genetic Algorithm for Solving Single-Stage Fixed-Charge Transportation Problems,” Technology Operation Management, vol. 2(1), pp. 1-15, 2011.

[18] K.A.A.D. Raj, and C. Rajendran, “A genetic algorithm for solving the fixed-charge transportation model: Two-stage problem,” Comp. & Oper. Res., vol. 39(9), pp. 2016-2032, 2012.

[19] E. Santibanez-Gonzalez, R. Del, G. Robson Mateus, and H. Pacca Luna, “Solving a public sector sustainable supply chain problem: A Genetic Algorithm approach,” In Proc. of Int. Conf. of Artificial Intelligence (ICAI), Las Vegas, USA, pp. 507-512, 2011.

[20] M.M. El-Sherbiny, “Comments on "Solving a capacitated fixed-cost transportation problem by artificial immune and genetic algorithms with a Prüfer number representation" by Molla-Alizadeh-Zavardehi, S. et al. Expert Systems with Applications (2011),” Exp. Syst. with Appl., vol. 39, pp. 11321-11322, 2012.

TABLE I. COMPUTATIONAL RESULTS OBTAINED BY OUR PROPOSED APPROACH IN COMPARISON WITH OTHER ALGORITHMS FROM LITERATURE

Instance size	JRGA [4]		TSGA [18]	SSGA [18]		HGA [16]		Our solution approach								
								obj	with Correction block				without Correction block			
									run time		#eval		run time		#eval	
	obj	#eval	obj	obj	#eval	obj	#eval		best	avg.	best	avg.	best	avg.	best	avg.
2x2x3	**112600**	1444	**112600**	**112600**	637	**112600**	2	**112600**	<1	**<1**	1	**1.00**	<1	**<1**	1	**1.00**
2x2x4	**237750**	1924	**237750**	**237750**	857	**237750**	2	**237750**	<1	**<1**	1	**1.00**	<1	**<1**	1	**1.00**
2x2x5	**180450**	2404	**180450**	**180450**	1214	**180450**	319	**180450**	<1	**<1**	1	**1.37**	<1	**<1**	1	8.87
2x2x6	**165650**	2884	**165650**	**165650**	1354	**165650**	324	**165650**	<1	**<1**	1	**3.27**	<1	**<1**	1	30.93
2x2x7	**162490**	3364	**162490**	**162490**	1889	**162490**	335	**162490**	<1	**<1**	1	**5.60**	<1	**<1**	1	52.77
2x3x3	**59500**	2164	**59500**	**59500**	1503	**59500**	317	**59500**	<1	**<1**	1	**3.50**	<1	**<1**	1	11.03
2x3x4	**32150**	2884	**32150**	**32150**	1859	**32150**	339	**32150**	<1	0.53	1	**2.10**	<1	**<1**	1	6.97
2x3x6	69970	4324	67380	**65945**	2577	**65945**	356	**65945**	<1	0.53	1	**19.43**	<1	**<1**	1	22.00
2x3x8	263000	5764	**258730**	**258730**	5235	**258730**	546	**258730**	<1	5.17	1	**435.47**	<1	**<1**	36	728.77
2x4x8	80400	7684	84600	**77400**	5246	78550	1039	**77400**	<1	**<1**	1	**24.23**	<1	**<1**	10	661.00
2x5x6	94565	7204	80865	**75065**	3574	80865	430	**75065**	<1	**<1**	1	**1.00**	<1	**<1**	1	6.50
3x2x4	**47140**	2884	**47140**	**47140**	1429	**47140**	321	**47140**	<1	0.53	1	**1.37**	<1	**<1**	1	1.50
3x2x5	178950	3604	178950	**175350**	2061	178950	320	**175350**	<1	1.03	1	**20.50**	<1	**<1**	6	45.80
3x3x4	**57100**	4324	61000	**57100**	3060	**57100**	354	**57100**	<1	**<1**	1	**1.73**	<1	**<1**	1	18.93
3x3x5	**152800**	5404	156900	**152800**	4555	**152800**	335	**152800**	<1	**<1**	1	**1.00**	<1	**<1**	1	**1.00**
3x3x6	**132890**	6484	**132890**	**132890**	2981	**132890**	3	**132890**	<1	**<1**	1	**1.00**	<1	**<1**	1	**1.00**
3x3x7(a)	104115	7564	106745	**99095**	7095	103815	1330	**99095**	<1	2.07	11	175.70	<1	**<1**	1	**1.00**
3x3x7(b)	287360	7564	295060	**281100**	7011	**281100**	380	**281100**	<1	**<1**	1	9.93	<1	**<1**	1	**1.00**
3x4x6	77250	8644	81700	**76900**	7105	77250	373	**76900**	<1	**<1**	1	**8.73**	<1	**<1**	1	8.87
4x3x5	**118450**	7204	**118450**	**118450**	4227	**118450**	394	**118450**	<1	**<1**	1	**9.57**	<1	**<1**	1	30.93

Some Ky Fan type inequalities on time scales

1st Cristian Dinu
University of Craiova
Department of Computer Science, Research Center for Artificial Intelligence “Nicolae Ţăndăreanu”
Craiova RO-200585, Romania
c.dinu@yahoo.com

2nd Daniela Dănciulescu
University of Craiova
Department of Computer Science, Research Center for Artificial Intelligence “Nicolae Ţăndăreanu”
Craiova RO-200585, Romania

3rd Alexandru Ţugui
A. I. Cuza University of Iaşi
Iaşi RO-700506, Romania

***Abstract*—In this paper we present some improved versions of the Ky-Fan inequality for functions on time scales, in the framework of some weights that are allowed to take some negative values.**

***Index Terms*—Time scales, convex function, dynamic derivatives, Ky Fan inequality, Jensen inequality**

I. Introduction

The theory of dynamic derivatives on time scales provides an unification and also an extension of traditional differential and difference equations. It is an unification of the discrete theory with the continuous theory, that was introduced by Stefan Hilger in [2]. Using Δ (delta) and ∇ (nabla) dynamic derivatives, a combined dynamic derivative, so called $\Diamond_\alpha$ (diamond-α) dynamic derivative, was introduced as a linear combination of Δ and ∇ dynamic derivatives on time scales. The diamond-α dynamic derivative reduces to the Δ derivative for $\alpha = 1$ and to the ∇ derivative for $\alpha = 0$. Throughout this paper, it is assumed that the basic notions of the time scales calculus are well known and understood. For these, we refer the reader to [1], [2], [9], [11], [12].

The inequality of Ky Fan can be considered a counterpart to the arithmetic-geometric mean inequality. It can be stated as

Theorem 1: If $0 < x_i \leq \frac{1}{2}$, for $i = 1, ..., n$, then

$$\left[\prod_{i=1}^{n} x_i / \prod_{i=1}^{n} (1 - x_i)\right]^{\frac{1}{n}} \leq \sum_{i=1}^{n} x_i / \sum_{i=1}^{n} (1 - x_i), \tag{1}$$

with equality only if $x_1 = ... = x_n$.

For a given n-tuple of numbers $x = (x_1, ..., x_n)$, the arithmetic, geometric and harmonic means of weight $w = (w_1, ..., w_n)$, (where $w_k \geq 0$ for each k and $\sum_{k=1}^{n} w_k = 1$) are defined as follows

$$\begin{aligned} A_n(x, w) &= \sum_{k=1}^{n} w_k x_k, \\ G_n(x, w) &= \prod_{k=1}^{n} x_k^{w_k}, \\ H_n(x, w) &= \frac{1}{\sum_{k=1}^{n} w_k / x_k}. \end{aligned} \tag{2}$$

It was proved that the inequality (1) works also in the weighted case. Using the above definitions, that is:

$$\frac{G_n(x, w)}{G_n(1 - x, w)} \leq \frac{A_n(x, w)}{A_n(1 - x, w)} \tag{3}$$

and also the complementary inequality (see [15]),

$$\frac{H_n(x, w)}{H_n(1 - x, w)} \leq \frac{G_n(x, w)}{G_n(1 - x, w)}. \tag{4}$$

Dragomir and Scarmozzino have improved the above inequalities in [7]. S. Simić gave a converse version of Ky Fan inequality in [14].

A complete weighted version of the Jensen inequality for weights that are allowed to take some negative values was presented by C. Dinu in [5].

Theorem 2: [Theorem 2 in [5]]. Let $a, b \in \mathbb{T}$ and $m, M \in \mathbb{R}$. If $g \in C([a, b]_{\mathbb{T}}, [m, M])$ and $w \in C([a, b]_{\mathbb{T}}, \mathbb{R})$ with $\int_a^b w(t) \Diamond_\alpha t > 0$, then the following assertions are equivalent:

(i) w is an α-SP weight for g on $[a, b]_{\mathbb{T}}$;
(ii) for every $f \in C([m, M], \mathbb{R})$ convex function, we have

$$f\left(\frac{\int_a^b g(t) w(t) \Diamond_\alpha t}{\int_a^b w(t) \Diamond_\alpha t}\right) \leq \frac{\int_a^b f(g(t)) w(t) \Diamond_\alpha t}{\int_a^b w(t) \Diamond_\alpha t}. \tag{5}$$

For the concave functions, the above inequality is reversed. Using this version, we improve the inequalities from [7] and [14].

In section II, we give our main results, regarding the extension of the Ky Fan inequality to a larger class of weights.

II. Extension of the Ky Fan Inequality

For the rest of this paper, let $\mathbb{T}$ be a time scale and $a, b \in \mathbb{T}$. We define the weighted means of a function on time scales, that extend the definitions given in (2). For that, we say that a continuous function $w : \mathbb{T} \to \mathbb{R}$ is a α-weight on $[a, b]_{\mathbb{T}}$, provided that $\int_a^b w(t)\Diamond_\alpha t > 0$, where $\alpha \in [0, 1]$.

Definition 3: Let $x : [a, b]_{\mathbb{T}} \to \mathbb{R}_+$ be a continuous positive function and w an α-weight on $[a, b]_{\mathbb{T}}$. We define:

- the generalized weighted arithmetic mean of the function x on the time scale interval $[a, b]$ of weight w:

$$A_{[a,b]}(x, w) = \frac{\int_a^b w(t)x(t)\Diamond_\alpha t}{\int_a^b w(t)\Diamond_\alpha t}; \tag{6}$$

- the generalized weighted geometric mean of the function x on the time scale interval $[a, b]$ of weight w:

$$G_{[a,b]}(x, w) = \exp\left(\frac{\int_a^b w(t)\ln(x(t))\Diamond_\alpha t}{\int_a^b w(t)\Diamond_\alpha t}\right); \tag{7}$$

- the generalized weighted harmonic mean of the function x on the time scale interval $[a, b]$ of weight w:

$$H_{[a,b]}(x, w) = \frac{\int_a^b w(t)\Diamond_\alpha t}{\int_a^b w(t)/x(t)\Diamond_\alpha t}. \tag{8}$$

Example 4:

(i) If $\mathbb{T} = \mathbb{R}$ then, for the α-weight $w : \mathbb{R} \to \mathbb{R}$, (that is $\int_a^b w(t)dt > 0$) we have

$$A_{[a,b]}(x, w) = \frac{\int_a^b w(t)x(t)dt}{\int_a^b w(t)dt};$$

$$G_{[a,b]}(x, w) = \exp\left(\frac{\int_a^b w(t)\ln(x(t))dt}{\int_a^b w(t)dt}\right);$$

$$H_{[a,b]}(x, w) = \frac{\int_a^b w(t)dt}{\int_a^b w(t)/x(t)dt}.$$

(ii) If $\mathbb{T} = \mathbb{Z}$, $a = 1$, $b = n + 1$ and $\alpha = 1$, we define $w(i) = w_i$ and $x(i) = x_i$. The condition of 1-weight for w means that $\sum_{i=1}^n w_i > 0$. Then, we have

$$A_{[a,b]}(x, w) = A_n(x, w) = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i};$$

$$G_{[a,b]}(x, w) = G_n(x, w) = \sqrt[w]{\prod_{i=1}^n x_i^{w_i}} \text{ where } w = \sum_{i=1}^n w_i;$$

$$H_{[a,b]}(x, w) = H_n(x, w) = \frac{\sum_{i=1}^n w_i}{\sum_{i=1}^n w_i/x_i}.$$

Remark 5: The generalized means inequality is also true for the generalized weighted means. That is,

$$H_{[a,b]}(x, w) \leq G_{[a,b]}(x, w) \leq A_{[a,b]}(x, w). \tag{9}$$

For the proof of the right hand of this inequality, we use Theorem 2 for the concave function $f(t) = ln(t)$ and $g = x$. For the left side, we use the same function f and $g = 1/x$.

We can give now our main result.

Theorem 6: Let $x : [a, b]_{\mathbb{T}} \to [m, M]$ be a continuous positive function such that $0 < m \leq x(t) \leq M \leq \frac{\gamma}{2}$, $\gamma > 0$ and w be an α-weight on $[a, b]_{\mathbb{T}}$. Then

$$\begin{aligned} \frac{A_{[a,b]}(x, w)}{G_{[a,b]}(x, w)} \geq \left(\frac{A_{[a,b]}(x, w)}{G_{[a,b]}(x, w)}\right)^{M^2/(\gamma-M)^2} &\geq \frac{A_{[a,b]}(\gamma - x, w)}{G_{[a,b]}(\gamma - x, w)} \\ \geq \left(\frac{A_{[a,b]}(x, w)}{G_{[a,b]}(x, w)}\right)^{m^2/(\gamma-m)^2} &\geq 1. \end{aligned} \tag{10}$$

Particulary,

$$\frac{A_{[a,b]}(x, w)}{A_{[a,b]}(\gamma - x, w)} \geq \frac{G_{[a,b]}(x, w)}{G_{[a,b]}(\gamma - x, w)}.$$

Proof:

We will adapt an idea used by Dragomir and Scarmozzino in [7] in the context of α-positive weights.

We begin by noticing that $\frac{A_{[a,b]}(x,w)}{G_{[a,b]}(x,w)} \geq 1$, and so, using the fact that $m, M \in (0, \frac{\gamma}{2}]$, we have the first and the last inequality in (10).

Let $f : (0, \gamma) \to \mathbb{R}$ be a function defined by $f(t) = \ln\frac{\gamma - t}{t} + c\ln t$. An easy computation shows that

$$f'(t) = -\frac{\gamma}{t(\gamma - t)} + \frac{c}{t}, \quad t \in (0, \gamma),$$

and

$$f''(t) = \frac{\gamma(\gamma - 2t)}{[t(\gamma - t)]^2} - \frac{c}{t^2} = \frac{1}{t^2}\left[\frac{\gamma(\gamma - 2t)}{(\gamma - t)^2} - c\right], \quad t \in (0, \gamma).$$

Considering the function $g : (0, \gamma) \to \mathbb{R}$, $g(t) = \frac{\gamma(\gamma-2t)}{(\gamma-t)^2}$, we have $g'(t) = \frac{-2\gamma t}{(\gamma-t)^3} < 0$. This shows that the function g is monotonically strictly decreasing on $(0, \gamma)$ and so, for any $m \leq t \leq M$, we have

$$\frac{\gamma(\gamma - 2M)}{(\gamma - M)^2} = g(M) \leq g(t) \leq g(m) = \frac{\gamma(\gamma - 2m)}{(\gamma - m)^2}. \tag{11}$$

Using (11), the function f is strictly convex on (m, M) if $c \leq \frac{\gamma(\gamma-2M)}{(\gamma-M)^2}$.

Now, we will apply generalized Jensen theorem 2 for the function $f : (m, M) \to \mathbb{R}$, $f(t) = \ln\frac{\gamma-t}{t} + c\ln t$, with $c \leq \frac{\gamma(\gamma-2M)}{(\gamma-M)^2}$. We deduce

$$\ln\left(\frac{\gamma-\frac{\int_a^b w(t)x(t)\Diamond_\alpha t}{\int_a^b w(t)\Diamond_\alpha t}}{\frac{\int_a^b w(t)x(t)\Diamond_\alpha t}{\int_a^b w(t)\Diamond_\alpha t}}\right)+c\ln\left(\frac{\int_a^b w(t)x(t)\Diamond_\alpha t}{\int_a^b w(t)\Diamond_\alpha t}\right)$$
$$=f\left(\frac{\int_a^b w(t)x(t)\Diamond_\alpha t}{\int_a^b w(t)\Diamond_\alpha t}\right)\leq\frac{\int_a^b w(t)f(x(t))\Diamond_\alpha t}{\int_a^b w(t)\Diamond_\alpha t}$$
$$=\frac{1}{\int_a^b w(t)\Diamond_\alpha t}\left(\int_a^b w(t)\ln\left(\frac{\gamma-x(t)}{x(t)}\right)\Diamond_\alpha t\right.$$
$$\left.+c\int_a^b w(t)\ln(x(t))\Diamond_\alpha t\right).$$

That is,

$$\ln\left(\frac{A_{[a,b]}(\gamma-x,w)}{A_{[a,b]}(x,w)}\right)+c\ln(A_{[a,b]}(x,w))$$
$$\leq\ln\left(\frac{G_{[a,b]}(\gamma-x,w)}{G_{[a,b]}(x,w)}\right)+c\ln(G_{[a,b]}(x,w)),$$

which is equivalent to

$$\ln\left(\frac{G_{[a,b]}(x,w)}{A_{[a,b]}(x,w)}\right)^c\geq\ln\left(\frac{A_{[a,b]}(\gamma-x,w)}{A_{[a,b]}(x,w)}/\frac{G_{[a,b]}(\gamma-x,w)}{G_{[a,b]}(x,w)}\right),$$

or,

$$\left(\frac{G_{[a,b]}(x,w)}{A_{[a,b]}(x,w)}\right)^{c-1}\geq\frac{A_{[a,b]}(\gamma-x,w)}{G_{[a,b]}(\gamma-x,w)}. \tag{12}$$

The best possible choice of c in (12) is its maximal value, that is $c=\frac{\gamma(\gamma-2M)}{(\gamma-M)^2}$, which yields

$$\left(\frac{G_{[a,b]}(x,w)}{A_{[a,b]}(x,w)}\right)^{\frac{\gamma(\gamma-2M)}{(\gamma-M)^2}-1}\geq\frac{A_{[a,b]}(\gamma-x,w)}{G_{[a,b]}(\gamma-x,w)}$$

and now, the second inequality in (10) is obvious.

For the third inequality, we define the function $h(t)=d\ln t-\ln\left(\frac{\gamma-t}{t}\right)$. This function is convex on (m,M) if $d\geq\frac{\gamma(\gamma-2m)}{(1-m)^2}$ and the proof goes in the same manner. ■

The inequality (10) generalizes the main results from [8] and [7].

Remark 7:

(i) If $\mathbb{T}=\mathbb{Z}$, $a=1$, $b=n+1$, $\alpha=1$, defining $w(i)=w_i$ and $x(i)=x_i$ with $\sum_{i=1}^n w_i>0$ and $x_i\in[m,M]\subset(0,\gamma/2]$ for all $i\in\{1,...,n\}$, then, we have

$$\frac{A_n(x,w)}{G_n(x,w)}\geq\left(\frac{A_n(x,w)}{G_n(x,w)}\right)^{M^2/(\gamma-M)^2}\geq\frac{A_n(\gamma-x,w)}{G_n(\gamma-x,w)}$$
$$\geq\left(\frac{A_n(x,w)}{G_n(x,w)}\right)^{m^2/(\gamma-m)^2}\geq 1. \tag{13}$$

(ii) If $\mathbb{T}=\mathbb{R}$ then, for any $a,b\in\mathbb{R}$, any α-weight $w:\mathbb{R}\to\mathbb{R}$, and for any continuous function $x:\mathbb{R}\to\mathbb{R}$, with $x([a,b])\subset[m,M]\subset(0,\gamma/2]$ we have

$$\frac{A_{[a,b]}(x,w)}{G_{[a,b]}(x,w)}\geq\left(\frac{A_{[a,b]}(x,w)}{G_{[a,b]}(x,w)}\right)^{M^2/(\gamma-M)^2}$$
$$\geq\frac{A_{[a,b]}(\gamma-x,w)}{G_{[a,b]}(\gamma-x,w)} \tag{14}$$
$$\geq\left(\frac{A_{[a,b]}(x,w)}{G_{[a,b]}(x,w)}\right)^{m^2/(\gamma-m)^2}\geq 1.$$

The inequalities from (14) represent a continuous version of (13) and an improved continuous version of Theorem 2 from [8].

REFERENCES

[1] Agarwal R.P. and Bohner M., *Basic calculus on time scales and some of its applications*, Results Math. 35 (1999), 3-22

[2] Bohner M. and Peterson A., *Dynamic Equations on Time Scales, An introduction with Applications*, Birkhäuser, Boston, 2001

[3] Czinder P. and Páles Z., *An extension of the Hermite-Hadamard inequality and an application for Gini and Stolarsky means*, J. Inequal. Pure and Appl. Math, **5** (2004), Issue 2, Article no. 42.

[4] Dinu C., *Hermite–Hadamard inequality on time scales*, Journal of Inequalities and Applications, vol. **2008**, Article ID 287947, 24 pages, (2008)

[5] Dinu C., *A weighted Hermite–Hadamard inequality for Steffensen–Popoviciu and Hermite–Hadamard weights on time scales*, Analele Ştiinţifice ale Universităţii "Ovidius", Constanţa, Ser. Mat., **17** (2009)

[6] Dragomir S. S. and Mcandrew A., *Refinements of the Hermite-Hadamard inequality for convex functions*, J. Inequal. Pure and Appl. Math, **6**(5) (2005), Art. 140.

[7] Dragomir S. S. and Scarmozzino F. P., *On the Ky Fan inequality*, J. Math. Anal. Appl. **269** (2002), 129-136.

[8] Florea A. and Niculescu C. P., *A Note on the Ky Fan Inequality*, J. Ineq. Appl. (2005), Issue 5, 459-468

[9] Hilger S. *Analysis on measure chains – a unified approach to continuous and discrete calculus*, Results Math. **35** (1990), 18-56

[10] Niculescu C. P. and Persson L.-E., *Convex functions and their applications. A contemporary aproach*, Springer-Verlang, Berlin, 2005.

[11] Rogers Jr. J. W. and Sheng Q., *Notes on the diamond-α dynamic derivative on time scales*, J. Math. Anal. Appl., **326** (2007), no. 1, 228-241

[12] Sheng Q,, Fadag M., Henderson M. and Davis J. M., *An exploration of combined dynamic derivatives on time scales and their applications*, Nonlinear Anal. Real World Appl. **7** (2006), no. 3, 395-413

[13] Sidi Ammi M. R., Ferreira R. A. C. and Torres D. F. M., *Diamond-α Jensen's Inequality on Time Scales and Applications*, Journal of Inequalities and Applications, vol. **2008**, Article ID 576876, 13 pages, 2008

[14] Simić S., *On a converse of Ky Fan inequality*, Kragujevac J. Math., **33** (2010), 9599.

[15] Wang W.-L. and Wang P.-F., *A class of inequalities for symmetric functions*, Arta Math. Sinica, **27** (1984), 485-497.

An Approach of Non-cyclic Nurse Scheduling Problem

Svetlana Simić, University of Novi Sad
Faculty of Medicine
21000 Novi Sad, Hajduk Veljkova 1-9, Serbia
svetlana.simic@mf.uns.ac.rs

Dragan Simić, University of Novi Sad
Faculty of Technical Sciences
21000 Novi Sad, Trg Dositeja Obradovića 6, Serbia
dsimic@eunet.rs, dsimic@uns.ac.rs

Dragana Milutinović, University of Novi Sad
Faculty of Medicine
21000 Novi Sad, Hajduk Veljkova 1-9, Serbia
dragana.milutinovic@mf.uns.ac.rs

Svetislav D. Simić, University of Novi Sad
Faculty of Technical Sciences
21000 Novi Sad, Trg Dositeja Obradovića 6, Serbia
simicsvetislav@uns.ac.rs

Abstract— **Demand for healthcare is increasing due to ever growing and aging population. Choosing an adequate schedule for medical staff can present a difficult dilemma for managers. The goal of nurse scheduling is to minimize the cost of the staff while maximizing their preferences and the overall benefits for the unit. This paper is focused on a new hybrid strategy based on detecting the optimal solution in nurse scheduling problem. The new proposed hybrid non-cyclic nurse scheduling combines randomly selected nurse and variable neighborhood descent search. The model is tested and obtained from a well-known previous published data-set.**

Keywords— nurse scheduling problem; non-cyclic; variable neighborhood descent search; medical staff

I. Introduction

One of the most important factors that should be taken into consideration in organization management is human resource management within the organization for maximum efficiency at all times. Employee timetabling problems (ETPs) can naturally be represented by constraints networks for real world instances which are large and difficult.

For personnel management within hospitals, scheduling the nurses' responsibilities is an important factor which is difficult to manage for maximum efficiency due to unknown number of patients each day, which makes it difficult to adequately and appropriately provide services to patients. The largest single cost factor in the hospital budget, typically representing about one third of the total, is the salaries of the nursing staff. If it were possible to make any inroads on the current escalation of hospital costs, it should begin with the most efficient possible utilization of the nursing staff [1]. On the other hand, it is also necessary to constantly aim toward minimizing labor costs, maximizing quality indicators and outcomes, and maintaining patient and employee satisfaction. In order to make the best staffing decisions and truly impact patient care it is necessary to have access to real time data. Inadequate staffing and schedule management are contributing factors to poor work environment, burnout and eventually turnover in any healthcare organization.

Nurse Scheduling Problem (NSP) represents a subclass of employee timetabling problems that are difficult to solve for optimal results. Studies of nurse scheduling problems date back to the early 1960s. Nurse scheduling deals with assigning shifts to staff nurses subject to satisfying required workload and other constraints. The constraints are classified into hard constraints (compulsory) and soft constraints (should be satisfied as much as possible). A feasible solution is a solution that satisfies all hard constraints. However, the quality of the duty roster is measured by satisfying the soft constraints and presents the most import thing in choosing an adequate schedule for nursing staff, but on the other hand it can be a difficult dilemma for nurse managers. Complete search algorithms, even with good heuristics are unable to solve large enough instances of ETPs. In fact, several local search techniques have been proposed in the past decade for solving timetabling problems. It has been shown that local search can efficiently solve large ETPs [2].

This paper is focused on new strategy based on hybrid approach to detecting the optimal solution in NSP. The new proposed hybrid approach is obtained by combining random nurse choice and variable neighborhood descent search. Also, this paper continuous the authors' previous research in nurse decision-making, scheduling and rostering health-care organizations which are presented in [3] [4] [5] [6] [7].

The rest of the paper is organized in the following way: Section 2 provides an overview of the basic idea in NSP and related work. Section 3 presents the optimization problem and applied technique for solving nurse scheduling problem proposed in this paper. Experimental results are presented in Section 4, while Section 5 provides conclusions and some points for future work.

II. Nurse Scheduling Problem and Related Work

The basic problem of NSP is to provide patient 24/7 using nurses who generally work five days a week, one shift per day, and prefer to have weekends off. Scheduling is usually done by nursing supervisors for the units of floors for which they are responsible. They estimate patient care requirements and allocate the available nursing staff to the days of the week so that these requirements are satisfied in general, and hospital personnel regulations observed. They try to schedule the nursing staff so that each nurse gets her share of weekends off and none of the nurses is rotated to evenings or night shifts for an unduly long time, and they also try to accommodate individual nurses' requests for specific days off. And finally, preparation of the schedule is a time-consuming task for the nursing supervisor.

NSP is a well-known NP-hard scheduling problem that aims to allocate the required workload to the available staff nurses at healthcare organizations to meet the operational requirements and a range of preferences. The NSP is a two-dimensional employee timetabling problem that deals with the assignment of nursing staff to shifts across a scheduling period subject to certain constraints.

In general, there are two basic types of scheduling used for the NSP: cyclic and non-cyclic scheduling. In cyclic scheduling, each nurse works in a pattern which is repeated in consecutive scheduling periods, whereas, in non-cyclic scheduling, a new schedule is generated for each scheduling period: weekly, fortnightly or monthly. Cyclic scheduling was first used in the early 1970s due to its low computational requirements and the possibility for manual solution [8].

A. *Related Work in Nurse Scheduling Problem*

In the past decades, many approaches have been proposed to solve NSP as they are manifested in different models. In the 1990s, a number of papers provided classifications of nurse scheduling systems and reviews of methods for solving different classes of problems [9]. The three commonly used general methods are: mathematical programming (MP), heuristics and artificial intelligence (AI) approaches. Many heuristics approaches were straightforward automation of manual practices, which have been widely studied and documented [10] [11].

Further advances were made in applying linear and/or mixed integer programming and network optimization techniques for developing nurse rosters. Constraint programming (CP) methods were also used to model the complicated rules associated with nurse rosters. The methods were applied to problems involving cyclic and non-cyclic rosters. Typically, the problems contained roster rules applicable to a particular hospital. As such, these approaches may require substantial reformulation for use in a different hospital.

For combinatorial problems, exact optimization usually requires large computational times to produce optimal solutions. In contrast, metaheuristic approaches can produce satisfactory results in reasonably short times. In recent years, metaheuristics including: tabu search algorithm (TS), genetic algorithm (GA) and simulated annealing (SA) have all been proven as very efficient in obtaining near-optimal solutions for a variety of hard combinatorial problems including the NSP [12].

Some TS approaches have been proposed to solve the NSP. In TS, hard constraints remained fulfilled, while solutions move in the following way: calculate the best possible move which is not tabu, perform the move and add characteristics of the move to the tabu list. The TS with strategic oscillation used to tackle the NSP in a large hospital is presented in [13].

The basic idea is to find a genetic representation of the problem so that 'characteristics' can be inherited. Starting with a population of randomly created solutions, better solutions are more likely to be selected for recombination into novel GA solutions. In addition, these novel solutions may be formed by mutating or randomly changing the old ones [14].

III. Optimization Problem and Applied Technique

Optimization tools have greatly improved during the last two decades. This is due to several factors: (i) progress in mathematical programming theory and algorithmic design; (ii) rapid improvement in computer performances; (iii) better communication of innovative ideas and integration in widely used complex software. Consequently, many problems long viewed as out of reach are currently solved, sometimes in very moderate computing times. This success, however, has led researchers and practitioners to address much larger instances and more difficult classes of problems. Many of these may again only be solved heuristically.

A. *Deterministic Optimization Problem*

A deterministic optimization problem may be formulated as

$$\min \{f(x) \mid x \in X, X \subseteq \mathcal{S}\} \qquad (1)$$

where $\mathcal{S}$, X, x, and f denote the *solution space*, the *feasible set*, a *feasible solution*, and a real-valued *objective function*, respectively. If $\mathcal{S}$ is a finite but large set, a *combinatorial optimization problem* is defined. If $\mathcal{S} = \Re^n$, we refer to continuous optimization. A solution $x^* \in$ X is optimal if

$$f(x^*) \leq f(x)\,,\ x^* \in X \qquad (2)$$

An *exact algorithm* for problem (1), if one exists, finds an optimal solution x^*, together with the proof of its optimality, or shows that there is no feasible solution, or the solution is unbounded. Moreover, in practice, the time needed to do so should be finite and not too long. For continuous optimization, it is reasonable to allow for some degree of tolerance, to stop when sufficient convergence is detected.

Let is denote N_k ($k = 1, \dots, k_{\max}$), a finite set of pre-selected neighborhood structures, and with $N_k(x)$ the set of solutions in the k-th neighborhood of x. Most local search heuristics use only one neighborhood structure, $k_{\max} = 1$. Often successive neighborhoods N_k are nested and may be induced from one or more metric (or quasi-metric) functions introduced into a solution space S. An *optimal solution* x_{opt} (or global minimum) is a feasible solution where a minimum is reached. It is call $x' \in X$ a *local minimum* of (1) with respect to N_k, if there is no solution $x \in N_k(x') \subseteq X$ such that $f(x) < f(x')$. Metaheuristics, based on local search procedures, try to continue the search by other means after finding the first local minimum.

B. Variable Neighborhood Descent Search

Variable neighborhood search (VNS) is a metaheuristic proposed by some of the present authors a dozen years ago [15]. It is based on the idea of a systematic change of neighborhood both in a descent phase to find a local optimum and in a perturbation phase to get out of the corresponding valley [16]. Originally designed for approximate solution of combinatorial optimization problems, it was extended to address mixed integer programs, nonlinear programs, and recently mixed integer nonlinear programs.

VNS is based on three simple empirical facts: i) a local minimum with respect to one neighborhood structure is not necessarily so for another; ii) a global minimum is a local minimum with respect to all possible neighborhood structures; iii) local minima with respect to one or several N_k are relatively close to each other [17]. To solve (1) by using several neighborhoods, facts 1 to 3 can be used in three different ways: (i) deterministic, (ii) stochastic, (iii) both deterministic and stochastic.

Algorithm 1 Neighborhood change

```
Function NeighborhoodChange (x, x', k)
1 if f(x') < f(x) then
2     x ← x'  // Make a move
3     k ← 1   // Initial neighborhood
  else
4     k ← k + 1 // Next neighborhood
  end else
return x, k
```

The solution move and neighborhood change function is given in Algorithm 1. Function `NeighborhoodChange()` compares the incumbent value $f(x)$ with the new value $f(x')$ obtained from the k-th neighborhood (line 1). If an improvement is obtained, the new incumbent is updated (line 2) and k is returned to its initial value (line 3). Otherwise, the next neighborhood is considered (line 4).

Algorithm 2 Variable neighborhood descent

```
Function VND (x, k_max)
k ← 1
repeat
    x' ← argmin_{y∈N_k(x)} f(y) // Find the best neighbor in N_k(x)
    x, k ← NeighborhoodChange (x, x', k) // Change neighborhood
until k = k_max
return x
```

The **variable neighborhood descent** (VND) method is obtained if a change of neighborhoods is performed in a deterministic way. It is presented in Algorithm 2, where neighborhoods are denoted as N_k, $k = 1, \ldots, k_{\max}$.

Most local search heuristics use a single or sometimes two neighborhoods for improving the current solution ($k_{\max} \leq 2$). Note that the final solution should be a local minimum, all $k_{\max}$ neighborhoods, and thus a global optimum is more likely to be reached than with a single structure.

IV. Modeling the Nurse Scheduling Problem

Modeling the NSP is the process of ensuring that there are always enough nurses present, it comprises of numerous decisions based on different time horizons and different levels of details. These decisions can be divided into three planning phases, as illustrated in Fig. 1.

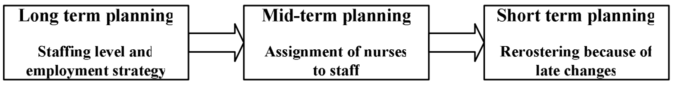

Long term planning
Staffing level and employment strategy
Mid-term planning
Assignment of nurses to staff
Short term planning
Rerostering because of late changes

Fig. 1. The three phases of nurse scheduling

The long-term planning is a part of the overall strategic planning process for each ward. First the ward managers must estimate how many nurses with each of the necessary skills are needed during all possible time periods of the day. When the staffing demand is known and there is a given workforce of nurses, each nurse is assigned to a schedule specifying which shifts she should work, usually for a scheduling period of four to ten weeks. This phase in the planning process can be referred to as the mid-term planning, or nurse rostering. Whenever there is a shortage of nurses for a shift, the short-term planning consists of deciding whether to use overtime, to call in a nurse on her day off, to call in a substitute nurse, or to try to manage despite the shortage.

Depending on the context in which the schedule is to be used and how the scheduling process is carried out, a definition of a good schedule can differ. The two main categories of scheduling strategies are cyclic and non-cyclic, which will both be described below. A third strategy, which will also be commented on, is self-scheduling, a kind of non-cyclic scheduling that has quickly grown in popularity during the past two decades.

A. Cyclic schedules

Using a traditional cyclic schedule means repeating the same schedule over and over again until the ward decides to change the schedule. Each schedule is typically made for a period of 4 to 10 weeks and is used for 6-12 months. Since the same schedule is used repeatedly it is very important that it is almost perfectly fair with respect to the score, the distribution of unpopular shifts, the number of hours, and quality aspects. This choice of scheduling strategy imposes the boundary restriction that the first week of the schedule can follow on from the last week. Further, because the nurses are bound to use the same schedule repeatedly, they typically have a low level of influence on the scheduling when this kind of strategy is used.

B. Non-cyclic schedules

Non-cyclic scheduling means that a new schedule, usually for a period of 2-8 weeks, is created for each scheduling period.

The advantage of a new schedule for each period is greater flexibility due to the possibility to take into account both changes on the ward and period specific requests from the nurses. A boundary condition is that the first week's schedule is affected by the last week's schedule from the previous scheduling period. Instead of making the schedule very fair in each period, it can be made reasonably fair and then information about for example the score, the distribution of unpopular shifts, and the number of hours worked can be passed on to next period, achieving a higher level of fairness in the long run.

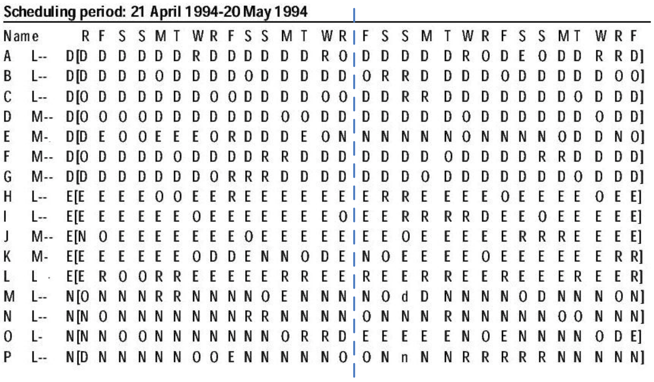
C. *Self-scheduling*

Self-scheduling is a general term used for the kind of scheduling processes where the nursing staff is jointly responsible for creating the schedule. This kind of scheduling exists in different forms around the world, but in general involve the following steps: i) Without taking into account the staffing demand and other nurses' preferences, each nurse individually proposes a schedule for herself; ii) An improved and more feasible schedule is created through informal negotiations between the nurses; iii) A scheduling group consisting of approximately four nurses makes further adjustments to the schedule; iv) The head nurse makes some final adjustments and approves the schedule.

D. *Input data set*

Table I gives the data set which is generated by the scheduling system presented in [18], where the normal shifts, eight hours a day, include the *Day shift* (*D*) (8 a.m. to 4 p.m.), the *Evening shift* (*E*) (4 p.m. to midnight), and the *Night shift* (*N*) (midnight to 8 a.m.).

TABLE I. STARTING DATA SET



Scheduling period: 21 April 1994-20 May 1994

Name			R	F	S	S	M	T	W	R	F	S	S	M	T	W	R	F	S	S	M	T	W	R	F	S	S	M	T	W	R	F
A	L--	D	[D	D	D	D	D	D	R	D	D	D	D	D	D	R	O	D	D	D	D	D	R	O	D	E	O	D	D	R	R	D]
B	L--	D	[D	D	D	D	O	D	D	D	D	O	D	D	D	D	D	O	R	R	D	D	D	D	O	D	D	D	D	D	O	O]
C	L--	D	[O	D	D	D	D	D	D	O	O	D	D	D	D	O	O	D	D	R	R	D	D	D	D	D	D	D	O	D	D	D]
D	M--	D	[O	O	O	O	D	D	D	D	D	D	D	O	O	D	D	D	D	D	D	D	O	D	D	D	D	D	D	O	D	D]
E	M-	D	[D	E	O	O	E	E	E	O	R	D	D	D	E	O	N	N	N	N	N	N	O	N	N	N	N	O	D	D	N	O]
F	M--	D	[O	D	D	D	D	O	D	D	D	D	R	R	D	D	D	D	D	D	D	O	D	D	D	D	R	R	D	D	D	D]
G	M--	D	[D	D	D	D	D	D	D	O	R	R	R	D	D	D	D	D	D	D	O	D	D	D	D	D	D	D	O	D	D	D]
H	L--	E	[E	E	E	E	O	O	E	E	R	E	E	E	E	E	E	E	R	R	E	E	E	E	O	E	E	E	E	O	E	E]
I	L--	E	[E	E	E	E	E	E	O	E	E	E	E	E	E	E	O	E	E	R	R	R	R	D	E	E	O	E	E	E	E	E]
J	M--	E	[N	O	E	E	E	E	E	E	E	O	E	E	E	E	E	E	E	O	E	E	E	E	E	R	R	R	E	E	E	E]
K	M-	E	[E	E	E	E	E	E	O	D	D	E	N	N	O	D	E	N	O	E	E	E	E	O	E	E	E	E	E	E	R	R]
L	L ·	E	[E	R	O	O	R	R	E	E	E	E	E	R	R	E	E	R	E	E	R	R	E	E	R	E	E	E	R	E	E	R]
M	L--	N	[O	N	N	N	R	R	N	N	N	N	O	E	N	N	N	N	O	d	D	N	N	N	N	O	D	N	N	N	O	N]
N	L--	N	[N	O	N	N	N	N	N	N	N	R	R	N	N	N	N	O	N	N	N	R	N	N	N	N	N	O	O	N	N	N]
O	L-	N	[N	N	O	O	N	N	N	N	N	N	N	O	R	R	D	E	E	E	E	E	N	O	E	N	N	N	N	O	D	E]
P	L--	N	[D	N	N	N	N	N	O	O	E	N	N	N	N	N	O	O	N	n	N	N	R	R	R	R	R	N	N	N	N	N]

Special shifts, twelve hours a day, include the *Daytime 8 o'clock shift* (*d*) (8 a.m. to 8 p.m.) and the *Nighttime 8 o'clock shift* (*n*) (8 p.m. to 8 a.m.) The use of daytime and night-time 8 hour shifts is to make up for nursing shortages in exceptional cases. The *Requested Day off* by (*R*), and the *Ordinary day off* by (*O*). In addition, the second column denotes *Shift Leader* (*L*), and *Member* (*M*), respectively.

Based on the approximate ratio the total numbers of nursing personnel are 16, but 12 or 13 nurses should be on duty each day. According to percentages for three shifts, the department should have six, four, and three nurses on duty for the day, evening, and night shifts, respectively.

It is very easy to recognize that implemented data set [18] is very unbalanced when viewed in different shifts which is used in nurse scheduling. It can be shown for nurse **A**{*D*, *E*, *N*} = {21, 1, 0} and 8 *Day off*; contrary to nurse **E**{*D*, *E*, *N*} = {6, 5, 11} and 8 *Day off*, or for nurse **O**{*D*, *E*, *N*} = {2, 7, 14} and 7 *Day off*.

V. EXPERIMENTAL RESULTS

The focus of this research is to propose hybrid non-cyclic nurse scheduling model which combines randomly selected nurse and variable neighborhood descent search. Non-cyclic scheduling period is usually for a period of 2 weeks and, therefore the starting data-set is divided in two parts for 15 days each. For first half of a month (~ 2 weeks) one nurse schedule is generated and for second half of a month (~ 2 weeks) next nurse schedule which does not carry any consequence from the previous scheduling part is generated. On the other hand, the proposed hybrid scheduling model is at least very fair in each shift period and maintains high level of fairness during the timetabling in considering type on nurse shifts.

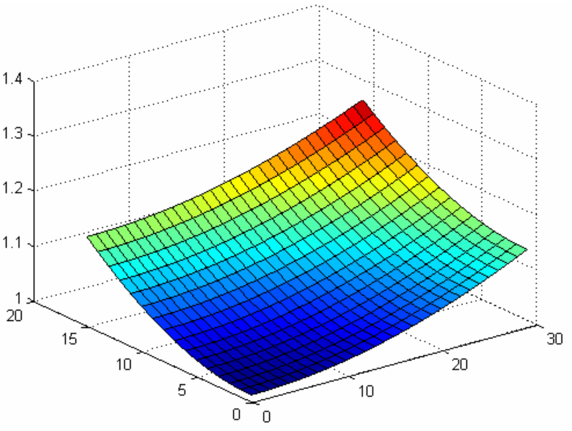


Fig. 2. The variable neighborhood descent search

On May 8th ”*n*” and ”*d*” are shown in Table I., and every the *Requested Day off*, and the *Ordinary day off* will be respected in creating new hybrid nurse scheduling timetable. It is typical for May 11th where 3 nurses are on *Requested Day off* and 2 nurses are on *Ordinary day off*.

Also, what is interesting to note is that typical work dynamic is: 5 nurses are in *Day shift*, 4 nurses are in *Evening shift* and 3 nurses are in *Night shift*. But, when necessary, this suggested schema is changed, therefore, for example, on May 14th is: 5 nurses in *D*, 5 nurses in *E*, and 3 nurses in *N*; and on May 15th is: 5 nurses in *D*, 3 nurses in *E*, and 3 nurses in *N*.

In order to balance the *deficit between shifts*, a score is calculated using the following formula:

$$D_{shift}=|D_{\mathrm{night}} - D_{\mathrm{day}}| + |D_{\mathrm{night}} - D_{\mathrm{evening}}| + |D_{\mathrm{day}} - D_{\mathrm{evening}}| \qquad (3)$$

where D_{shift} represents the monthly gap between the quota and the number of nurses assigned by shift.

TABLE II. EXPERIMENTAL RESULTS WITH NON-CYCLIC SCHEDULE APPROACH

Non-Cyclic Schedule Shift	
Nurse	***Day***
	RFSSMTWRFSSMTWR FSSMTWRFSSMTWRF
A	DDDDDDRDDDDDDRO DDDDDRONNODERRD
B	DDDDODDDDODDDDD ORRNNEEOEDDDEOO
C	ODDDDDDOODDDDOO NERRDDNEENEONNN
D	OOOODDDDDDDOODD EDEEDODDEDEDODE
E	DEOOEEEORDDDEON NNEDEODDDEONENO
F	ODDDDODDDDRRDDD DENEODEEDRREDEN
G	DDDDDDDORRRDDDD EDDOEEDDDEDODDD
H	EEEEOOEEREEEEEE DRREDENODNNEONN
I	EEEEEEOEEEEEEEO DERRRRDNNOEDDDE
J	NOEEEEEEEOEEEEE NEODEEDNRRRDDDD
K	EEEEEEODDENNODE EODNNNODDDNENRR
L	EROORREEEEERREE RDDRRNEONEDRNER
M	ONNNRRNNNNOENNN DO**d**DDNEDODNNEOE
N	NONNNNNNNRRNNNN ONENRDNEEDOODED
O	NNOONNNNNNNORRD EDNEEDOEENEDOEE
P	DNNNNNOOENNNNNO ON**n**DNRRRRRDNEDD

TABLE III. NURSE WORKLOAD

Nurse	**Shifts**			**Shifts (%)**			***Dshift***	**Dshift [18]**
	D	*E*	*N*	*D (%)*	*E (%)*	*N (%)*		
A	19	1	2	0.863	0.045	0.090	36	42
B	16	4	2	0.727	0.181	0.090	28	44
C	12	4	6	0.545	0.181	0.272	16	44
D	16	6	0	0.727	0.454	0.000	32	44
E	8	9	5	0.363	0.409	0.227	**8**	12
F	15	6	2	0.652	0.260	0.087	26	46
G	20	4	0	0.833	0.166	0.000	40	48
H	3	15	5	0.130	0.652	0.217	24	46
I	5	16	2	0.217	0.695	0.087	28	44
J	6	15	3	0.250	0.625	0.125	24	46
K	7	10	7	0.291	0.416	0.291	**6**	30
L	3	11	3	0.176	0.647	0.176	16	34
M	5	4	13	0.227	0.181	0.590	18	36
N	4	4	15	0.173	0.173	0.652	22	46
O	4	8	11	0.173	0.347	0.478	**14**	24
P	5	2	13	0.250	0.100	0.650	22	34

The Table II. presents experimental results for nurse scheduling timetable when the proposed hybrid non-cyclic nurse scheduling which combines randomly selected nurse and variable neighborhood descent search is used. The experimental results present scheduling for period of 30 days with the attempt to balance between, *Day*, *Evening* and *Night* shifts with respect to every the *Requested Day off*, and the *Ordinary day off* as is given in the original input data [18].

The Table III. presents the *deficit between shifts*, which shows a score of imbalance between shifts calculated *Dshift* first for the hybrid non-cyclic nurse scheduling system proposed in this paper and second Dshift[18] which is calculated with original input data-sat. In the table **red bold** presents the best experimental results, **blue bold** presents the second best experimental results, while **pink bold** presents the third best experimental results.

It is very easy to note that all *Dshift* are much less then Dshift[18], which guaranties better balance which is generated with the proposed novel hybrid *Nurse Scheduling System*. The average value of *deficit between shifts* for novel hybrid *Nurse Scheduling System* is 22.50, while Dshift[18] has much higher value of 38.75.

Also, it is very important to mention that schedule which is used as the input data-set is generated as ”Scheduling nursing personnel on a microcomputer” [18] for hospital, and that developed software and implemented system is promoted and used at a leading hospital in Taiwan.

VI. CONCLUSION AND FUTURE WORK

One challenge when working with a new ward is to understand what is essential in their scheduling, and to be able to successfully deliver a schedule, it is of crucial importance to understand their values and traditions. During this type of work, the responses from the nurses have usually been expectant and skeptical. Expectant because of the time-consuming work and difficulties associated with the manual scheduling process, and skeptical mainly because they are afraid of losing control over the scheduling. Because of the nurses’ skepticism, it is important to present the outcome of the automatic scheduling pedagogically and to emphasize that the optimization tool only offers a qualified suggestion for a schedule. If it is considered beneficial for the nurses to be allowed to make minor adjustments themselves.

The aim of this paper is to propose the new hybrid strategy, the novel hybrid non-cyclic nurse scheduling system for detecting the quality solution in nurse scheduling problem. The new proposed hybrid approach is obtained by combining randomly selected nurse and variable neighborhood descent search. The model is tested with original real-world data-set from leading hospital in Taiwan.

The great benefit of this approach should be the time and effort saved if the head nurse is handed a schedule that is both feasible and fair. Other benefits are the objectivity of a computerized planning tool and the decrease in lead time for constructing a schedule.

Preliminary experimental results of our research, compared with original input data-set from leading hospital in Taiwan, are better, which is shown in this paper. The experimental results encourage further research. Our future research will focus on creating new hybrid model combined some new soft-computing techniques, to solve problems logically, considering different options until the best solution is discovered, which will efficiently solve NSP. The new model will be tested with original real-world dataset for longer periods, including the data-set for 2018 obtained from the Oncology Institute of Vojvodina in Serbia.

REFERENCES

[1] C. M. Rothe and H. B. Wolfe, ”Cyclical scheduling and allocation of nursing staff,” Socio-Economic Planning Sciences, vol. 7, no. 5, pp. 471-487, 1973.

[2] A. Meisels and E. Kaplansky, ”Iterative restart technique for solving timetabling problems,” European Journal of Operational Research, vol. 153, pp. 41-50, 2004.

[3] D. Simić, ”Nursing logistics activities in massive services,” Journal of Medical Informatics & Technologies, vol. 18, pp. 77-84, 2011.

[4] D. Simić, D. Milutinović, S. Simić and V. Suknaja, ”Hybrid patient classification system in nursing logistics activities,” Springer, LNAI, vol. 6679, pp. 421-428, 2011.

[5] D. Simić, S. Simić, D. Milutinović and J. Đorđević, ”Challenges for nurse rostering problem and opportunities in hospital logistics,” Journal of Medical Informatics & Technologies, vol. 23, pp. 195-202, 2014.

[6] S. Simić, D. Simić, D. Milutinović, J. Đorđević and S. D. Simić, ”A hybrid approach to detecting the best solution in nurse scheduling problem,” Springer, LNAI, Vol. 10334, pp. 710-721, 2017.

[7] S. Simić, D. Simić, D. Milutinović, J. Đorđević and S. D. Simić, ”A fuzzy ordered weighted averaging approach to rerostering in nurse scheduling problem,” Springer, Advances in Intelligent Systems and Computing, vol. 649, pp 79-88, 2017.

[8] K. A. Dowsland, ”Nurse scheduling with tabu search and strategic oscillation,” European Journal of Operational Research, vol. 106, Issue 2-3, pp. 393-407, 1998.

[9] A. T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier, ”Staff scheduling and rostering: A review of applications, methods and models,” European Journal of Operational Research, vol. 153, pp. 3-27, 2004.

[10] M. W. Isken and W. M. Hancockm, ”A heuristic approach to nurse scheduling in hospital units with non-stationary, urgent demand, and a fixed staff size,” Journal of the Society for Health Systems, vol. 2, Issue 2, pp. 24-40, 1991.

[11] M. Warner, B. J. Keller and S. H. Martel, ”Automated nurse scheduling,” Journal of the Society for Health Systems, vol. 2, No. 2, pp. 66-80, 1990.

[12] B. Cheang, H. Li and B., Rodrigues, ”Nurse rostering problems - a bibliographic survey,” European Journal of Operational Research, vol. 151, Issue 3, pp. 447-460, 2003.

[13] H. Millar and M. Kiragu, ”Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming,” European Journal of Operational Research, vol. 104, No. 3, pp. 582-592, 1998.

[14] K. Leksakul and S. Phetsawat, ”Nurse scheduling using genetic algorithm,” Mathematical Problems in Engineering, Article ID 246543, http://dx.doi.org/10.1155/2014/246543, 2014.

[15] N. Mladenović and P. Hansen, ”Variable neighborhood search,” Computers & Operations Research, vol. 24, Issue 11, pp. 1097-1100, 1997.

[16] P. Hansen, N. Mladenović, J. Brimberg and J. A. Moreno Pérez, ”Variable Neighborhood Search,” M. Gendreau and J.-Y. Potvin, Eds. Handbook of Metaheuristics (Second Edition), pp. 61-96, 2010.

[17] N. Mladenović, M. Dražić, V. Kovačević-Vujčić and M. Čangalović, ”General variable neighborhood search for the continuous optimization,” European Journal of Operational Research, vol. 191, Issue 3, pp. 753-770, 2008.

[18] C.-J. Liao and C.-Y. Kao, ”Scheduling nursing personnel on a microcomputer,” Health Manpower Management, vol. 23, Issue 3, pp.100-106, 1997.

A Neural Network Based Approach for Approximating Real Roots of Polynomials

Diogo Freitas
University of Madeira
Master's Programme in Mathematics
Funchal, Portugal
2019214@student.uma.pt

Luiz Guerreiro Lopes
University of Madeira, Funchal,
CIMO/IPB, Bragança, and
ICAAM/UE, Évora, Portugal
lopes@uma.pt

Fernando Morgado-Dias
University of Madeira
Madeira Interactive Technologies Institute
Funchal, Portugal
morgado@uma.pt

***Abstract*—There are many iterative methods for finding all the zeros of a polynomial sequentially or simultaneously. However, the determination of all zeros of a given polynomial by one of the methods that find one zero at a time involves repeated deflations, which leads to the accumulation of rounding errors and inaccurate results. In turn, the simultaneous methods require very good starting approximations for all the zeros in order to converge. In view of these drawbacks, in this work we adopt a different approach based on neural networks for finding the zeros of real polynomials with only real zeros. This approach is tested with random polynomials of different degrees. The results obtained, although preliminary and limited, indicate that this approach seems to be quite robust and promising, and faster when compared with the well known Durand–Kerner method.**

***Index Terms*—Artificial Neural Networks, Polynomials, Roots, Durand–Kerner method.**

I. Introduction

Although there are many iterative methods to calculate one real zero or a pair of complex conjugate zeros of a polynomial, such as the well-known Laguerre's and Jenkins–Traub's methods [1], the determination of all zeros of a given polynomial by one of such methods involves repeated deflations, which can lead to very inaccurate results due to the problem of accumulating rounding errors when using finite accuracy floating-point arithmetic.

Iterative methods for finding all zeros of a polynomial simultaneously, such as the methods of Durand–Kerner and Ehrlich–Aberth (see, e.g., [2]–[4]), appeared in literature only in the 1960s. The simultaneous zero-finding algorithms, in addition to being inherently parallel, have the advantage of avoiding the polynomial deflation steps required by methods that determine only one real root or a pair of complex roots at a time. However, these simultaneous methods need very good initial approximations for all the zeros in order to converge.

Due to the drawbacks mentioned above, and since traditional Artificial Neural Networks (ANN) or shallow neural networks are well known for their capability to model data and to find good approximations for complex problems, in this work we try a different approach for finding real zeros of polynomials based on neural networks, in order to assess their potentiality and limitations in terms of efficiency and accuracy of the approximations when compared with traditional iterative methods for polynomial zero finding.

II. Datasets and methodology

In this section, the steps taken to build a training and a test dataset, and to train the ANNs to produce approximations for the real zeros of polynomials are described.

Even though the neural approach proposed in this paper can be extended to the case of approximating complex zeros of a polynomial, the ANNs are only used here for approximating the real zeros α_i $(i = 1, 2, \ldots, n)$ of a degree n real univariate polynomial, $P(x) = a_0 + a_1x + \ldots + a_nx^n$, with only real zeros, given its coefficients, as already mentioned in Section I. The block diagram of this approach is shown in Fig. 1.

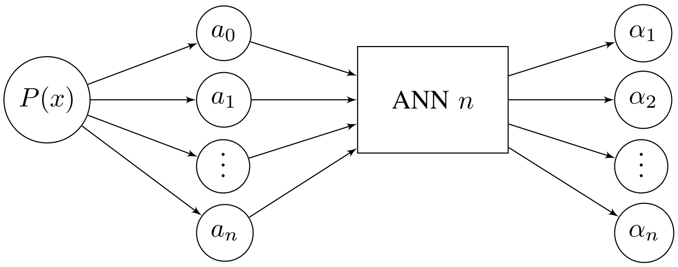


Fig. 1. Flowchart showing the inputs, the processing flow and the outputs of the proposed neural approach to polynomial root finding.

The use of a similar approach based on ANNs for finding all (real and complex) zeros of real polynomials, and also the application of such kind of approach to the most general problem of determining all zeros of complex polynomials are not considered here since they will be the subject of future papers currently in preparation.

Regarding the neural network structure, although there is not an exact method to determine the number of neurons that should be in its hidden layer, there are two common ways to do this: applying the well known Kolmogorov's mapping neural network existence theorem [5] or using a rule of thumb adequate for this purpose [6].

For this work, after several tests according to these two methods, we found that there is little variance resulting from a change in the number of hidden neurons of the neural network. In view of this, in this experimental study we used ten neurons in the hidden layer for all the final tests.

In this preliminary study, we used five neural networks with only three layers (input, hidden, and output layer), that were trained using as input the coefficients of a set of polynomials of degrees 5, 10, 15, 20 and 25, respectively. In Fig. 1, we denote by ANN n the neural network that can output the real zeros of a degree n real polynomial. Tables I and II show the head of the datasets (with 100 000 records) that were used with ANN 5. It is important to note here that, although coefficients and zeros are shown with only four decimal places, double precision values were used to generate the datasets. To generate these datasets, it was used two algorithms to: generates real zeros for any polynomial degree and, given a set of real zeros, compute the respective coefficients.

TABLE I
HEAD OF THE INPUT DATASET FOR TRAINING ANN 5

a_0	a_1	a_2	a_3	a_4	a_5
-4593.5077	3961.1594	-155.2456	-120.6867	3.6453	1
-5351.3845	3272.8352	251.6259	-125.4805	-2.3285	1
643.6638	701.0272	133.1762	-46.9399	-7.0419	1
0.8773	51.3427	28.4148	-15.6812	-2.9906	1
-0.2478	12.5678	55.9301	-66.4759	-2.5088	1
⋮	⋮	⋮	⋮	⋮	⋮

TABLE II
HEAD OF THE OUTPUT DATASET FOR TRAINING ANN 5

α_1	α_2	α_3	α_4	α_5
-9.2110	-8.9445	1.2858	6.0128	7.2117
-9.2925	-5.9211	1.5966	6.3456	9.5999
-4.2366	-1.9342	-1.5788	5.1719	9.6196
-3.1738	-1.2258	-0.0173	2.8990	4.5084
-7.4262	-0.1998	0.0183	1.0033	9.1133
⋮	⋮	⋮	⋮	⋮

Mathematically, an ANN is represented by a weighted, directed graph with nodes. For this study, a multilayer perceptron (MLP) feedforward neural network was chosen. The first layer of the ANN contains the input nodes, which have no incoming links attached to them. The last layer contains the output nodes, and the intermediate hidden layer consists of nodes connected to each of the nodes in the input and output layers [7].

The well-known Levenberg–Marquardt backpropagation algorithm (LMA) was used for ANN training, due to its efficiency and convergence speed, being one of the fastest methods for training feedforward neural networks, especially medium-sized ones. The application of the Levenberg–Marquardt algorithm to neural network training is described, e.g., in [8] and [9]. LMA is a hybrid algorithm that combines the efficiency of the Gauss-Newton method with the robustness of the gradient descent method, making one of these methods more or less dominant at each minimization step by means of a damping factor that is adjusted at each iteration [10].

The hyperbolic tangent sigmoid (tansig) function [11], defined in (1), has been chosen in this study as the activation function for the hidden and output layer nodes, in order to ensure that values stay within a relatively small range and to allow the network to learn nonlinear relationships [12] between coefficients and zeros.

$$\phi(x) = \frac{2}{1+e^{-2x}} - 1. \tag{1}$$

The use of this antisymmetric (S-shaped) function for the input to output transformation allows the output of each neuron to assume positive and negative values in the interval $[-1, 1]$ (see Fig. 2). A min-max normalization method [13] was used to scale all data into this interval.

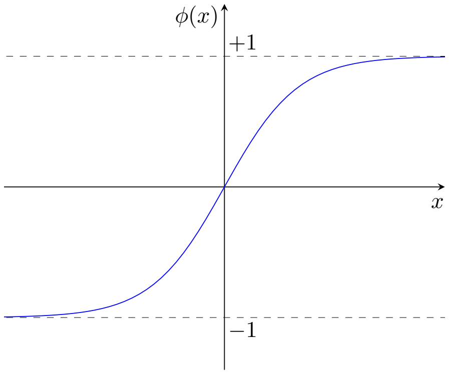


Fig. 2. Hyperbolic tangent sigmoid (tansig) transfer function.

III. DISCUSSION AND RESULTS

In this section, some results obtained with this approach are presented along with comparisons with the numerical approximations provided by the Durand–Kerner method in terms of execution time and accuracy.

The Durand–Kerner (D-K) method, also known as Weierstrass' or Weierstrass–Dochev's method [3], is a well-known iterative method for the simultaneous determination of all zeros of a polynomial that does not require the computation of derivatives, but has the drawback of requiring a good initial approximation to each of the zeros (which must be obtained using another numerical method) in order to converge and produce approximations to these zeros with the required accuracy.

Let $P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ $(a_n \neq 0)$ be a degree n univariate polynomial with real (or complex) coefficients. The D-K method is given by [14]

$$x_i^{(k+1)} = x_i^{(k)} - \frac{P(x_i^{(k)})}{a_n \prod\limits_{\substack{j=1\\ j\neq i}}^{n} (x_i^{(k)} - x_j^{(k)})}, \tag{2}$$

where $i = 1, \ldots, n$ and $k = 0, 1, \ldots$.

The convergence order of the Durand–Kerner method is quadratic for simple zeros but only linear in case of multiple zeros [15].

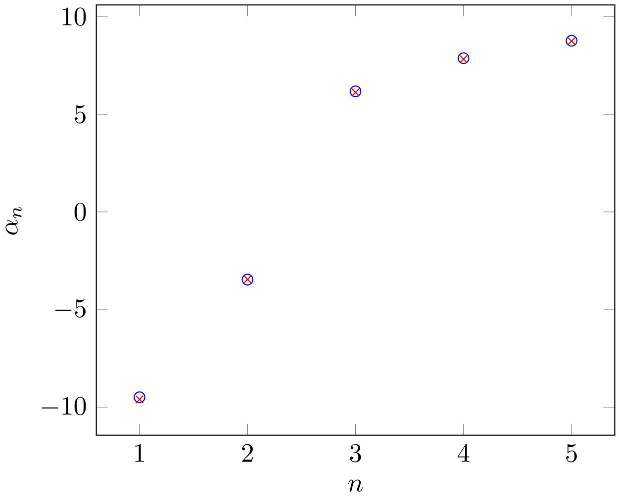


Fig. 3. Comparison between ANN (red) and Durand–Kerner approximations to the real zeros of a random real polynomial of degree 5.

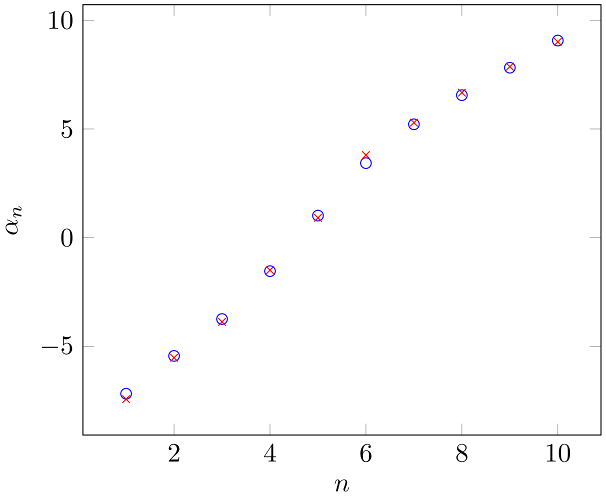


Fig. 4. Comparison between ANN (red) and Durand–Kerner approximations to the real zeros of a random real polynomial of degree 10.

Figs. 3 to 7 show the comparisons, in terms of accuracy, between the approximations to the real zeros of five random real polynomials of degrees 5, 10, 15, 20, and 25 produced by the ANN approach and the Durand–Kerner method.

Analysing the plots, it is clearly noticeable that our approach produces results relatively similar to those obtained with the D-K method. But, as the degree of the polynomial increases, it is possible to observe a slight increase of the differences between the approximations produced by both methods.

Table III shows the Mean Square Error (MSE) for each of the five examples, computed as follows, where D_i denotes the approximation to the i-th zero ($i = 1, \ldots, n$) computed by the D-K method and N_i the corresponding approximation obtained with our approach:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(D_i - N_i)^2. \tag{3}$$

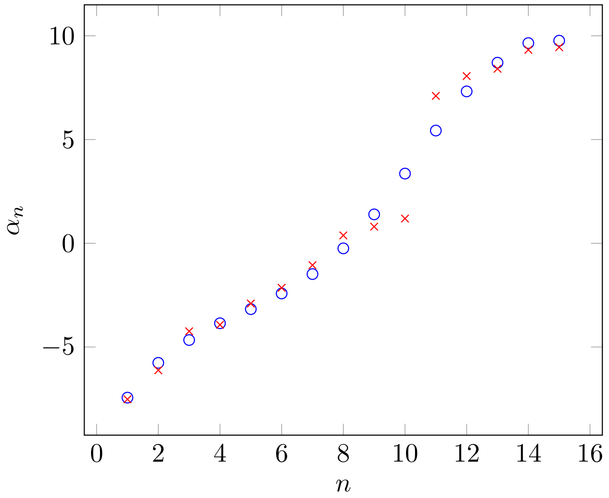


Fig. 5. Comparison between ANN (red) and Durand–Kerner approximations to the real zeros of a random real polynomial of degree 15.

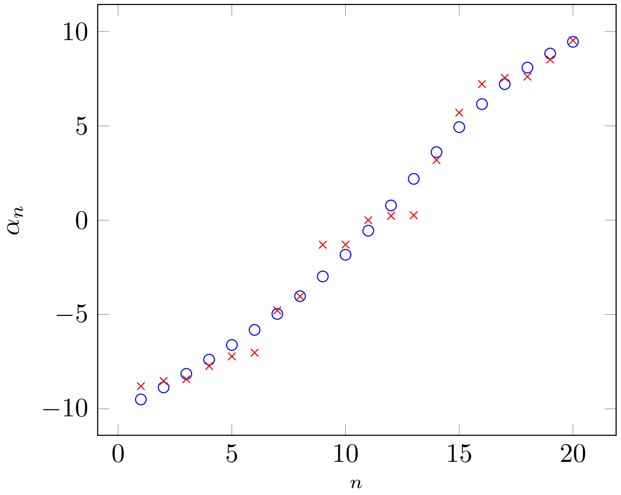


Fig. 6. Comparison between ANN (red) and Durand–Kerner approximations to the real zeros of a random real polynomial of degree 20.

TABLE III
MSE OF THE ANN BASED APPROACH

Degree	MSE
5	0.0036
10	0.0262
15	0.6474
20	0.6213
25	0.4731

The results obtained, although limited, are very encouraging and demonstrate the viability and potentiality of our ANN based approach for approximating real roots of polynomials.

The results on execution time for both methods, showed below, were obtained using a personal computer equipped with a 7th generation Intel Core i7 processor and 16 GB of RAM.

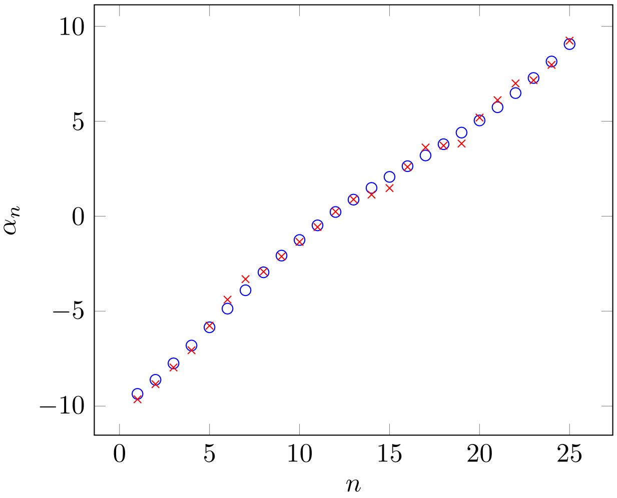


Fig. 7. Comparison between ANN (red) and Durand–Kerner approximations to the real zeros of a random real polynomial of degree 25.

TABLE IV
COMPARISON IN TERMS OF EXECUTION TIME

Degree	ANN approach	D–K method
5	0.004	0.010
10	0.002	0.009
15	0.005	0.011
20	0.003	0.019
25	0.005	0.026

Table IV shows that, when the degree of the polynomial increases, the execution time with ANN remains almost constant. The opposite happens with the Durand–Kerner method, with which an increase in the degree of the polynomial implies an increase of the execution time. Comparing the execution times of both methods, we can observe that the execution time required to compute the approximations to the zeros using ANN is significantly lower than that of the Durand–Kerner method. This result was already expected because computing polynomial zeros using this latter method, unlike ANN, is a pure iterative procedure.

We also assessed the capacity of the networks to generalize the outputs to other spaces of results. For this, new datasets were used with samples that were not employed to train the networks. The computed outputs and targets are compared in Table V.

Since all the MSE values presented in Table V are significantly less than one, we can infer that the networks have a good capacity to generalize the space of results. Thus, with some confidence, we can conclude that the networks can solve any real univariate polynomial of the respective degree with only real zeros.

IV. CONCLUSION

This short paper is a concise report of ongoing work about the use of artificial neural networks for finding numerical approximations to the zeros of polynomials.

TABLE V
CAPACITY OF THE NETWORKS TO GENERALIZE THE OUTPUTS

Degree	MSE
5	0.4087
10	0.6684
15	0.6666
20	0.4768
25	0.4766

Although the results presented here are preliminary and limited to a particular class of polynomials, namely polynomials with real coefficients and only real zeros, they are very promising and indicate the potential of this neural network based approach for determining the zeros of polynomials.

The proposed approach seems to be quite robust and also shows to be faster than the well known Durand–Kerner iterative method for simultaneous polynomial root finding.

V. ACKNOWLEDGMENTS

Acknowledgments to the Portuguese Foundation for Science and Technology (FCT) for their support through the Strategic Project LA 9 – UID/EEA/50009/2013.

REFERENCES

[1] J. M. McNamee, *Numerical methods for roots of polynomials, Part I.* Amsterdam: Elsevier, 2007.

[2] Bl. Sendov, A. Andreev, and N. Kjurkchiev, "Numerical solution of polynomial equations," in *Handbook of Numerical Analysis, Vol. III*, P. G. Ciarlet and J. L. Lions, Eds. Amsterdam: Elsevier, North-Holand, 1994, pp. 625–776.

[3] M. Petković, *Point estimation of root finding methods*. Berlin: Springer-Verlag, 2008.

[4] O. Cira, *The convergence simultaneous inclusion methods*. Bucureşti: Matrix ROM, 2012.

[5] R. Hecht-Nielsen, "Kolmogorov's mapping neural network existence theorem," in *Proceedings of the IEEE First Annual International Conference on Neural Networks*, M. Caudil and C. Butler, Eds. San Diego, CA: IEEE, 1987, pp. 609–618.

[6] D. Baptista, S. Abreu, C. Travieso-Gonzlez, and F. Morgado-Dias, "Hardware implementation of an artificial neural network model to predict the energy production of a photovoltaic system," *Microprocessors and Microsystems*, vol. 49, pp. 77–86, 2017.

[7] T. L. Fine, *Feedforward Neural Network Methodology*. New York: Springer-Verlag, 1999.

[8] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, pp. 989–999, 1994.

[9] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesús, *Neural Network Design*, 2nd ed. Stillwater, OK: Oklahoma State Univ., 2014.

[10] J. Heaton, *Artificial Intelligence for Humans, Vol. 3: Deep Learning and Neural Networks*. Chesterdfield, MO: Heaton Research, 2015.

[11] P. B. Harrington, "Sigmoid transfer functions in backpropagation neural networks," *Analytical Chemistry*, vol. 65, pp. 2167–2168, 1993.

[12] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, pp. 431–441, 1963.

[13] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction*. Bellingham, WA: SPIE Press, 2005.

[14] A. Terui and T. Sasaki, "Durand-Kerner method for the real roots," *Japan Journal of Industrial and Applied Mathematics*, vol. 19, pp. 19–38, 2002.

[15] P. Fraigniaud, "The Durand-Kerner polynomials roots-finding method in case of multiple roots," *BIT Numerical Mathematics*, vol. 31, pp. 112–123, 1991.

Authors' Index