

P O L I S H A C A D E M Y O F S C I E N C E S
C O M M I T T E E O F A U T O M A T I C C O N T R O L A N D R O B O T I C S

ARCHIVES OF CONTROL SCIENCES

VOLUME 27 (LXIII), No. 2

QUARTERLY

Special issue on Discrete Systems: Theory and Applications

Wojciech Bożejko, Guest Editor
Wrocław University of Science and Technology
Wrocław, Poland

Mieczysław Wodecki, Guest Editor
The University of Wrocław
Wrocław, Poland

Institute of Automatic Control
Silesian University of Technology
Gliwice 2017

Archives of Control Sciences welcomes for consideration papers on topics of significance in broadly understood control science and related areas, including: basic control theory, optimal control, optimization methods, control of complex systems, mathematical modeling of dynamic and control systems, expert and decision support systems and diverse methods of knowledge modeling and representing uncertainty (by stochastic, set-valued, fuzzy or rough set methods, etc.), robotics and flexible manufacturing systems. Related areas that are covered include information technology, parallel and distributed computations, neural networks and mathematical biomedicine, mathematical economics, applied game theory, financial engineering, business informatics and other similar fields.

Warunki prenumeraty: Roczna prenumerata *Archives of Control Sciences* może być rozpoczęta w dowolnym momencie. Opłata za prenumeratę krajową w roku 2017 wynosi 120 zł, a zagraniczna 160 \$. Głównym dystrybutorem jest Warszawska Drukarnia Naukowa PAN. Zamówienia należy składać przez e-mail, fax lub pocztą na adres: Warszawska Drukarnia Naukowa Polskiej Akademii Nauk 00-656 Warszawa, ul. Śniadeckich 8 tel./fax (0-22) 628-76-14, (0-22) 628-87-77, e-mail: wdnpan@home.pl lub dystrybucja@wdnpan.home.pl
Zamówienia z zagranicy można kierować: Ars Polona JSC, Foreign Trade Enterprise, Obrońców 25, 03-933 Warszawa, Fax: 022 509-86-48.

Subscription information: The promotional subscription price in 2017 is EUR 122,- plus surface postage EUR 12,-. At subscriber's request this journal will be air mailed at additional fee. Subscription for foreign customers is available through: Ars Polona JSC Foreign Trade Enterprise 25, Obrońców Street 03-933 Warsaw, Poland Fax: 004822/5098648

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owner.

Submission of a paper implies the author's irrevocable and exclusive authorization of the publisher to collect any sums or considerations for copying or reproduction payable by third parties.

Copyright ©Silesian University of Technology, 2017

The publication of this Journal is supported by funds from the Polish Ministry of Science and Higher Education.

Address of the Editor: Institute of Automatic Control, Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland. Tel: +(48-32) 237-23-98, fax +(48-32) 237-21-27

<http://acs.polsl.pl>

Typeset in T_EX at the Institute of Automatic Control, Silesian University of Technology

Printing: Centrum Poligrafii Sp. z o.o., ul. Łopuszańska 53,
02-232 Poland

CONTENTS

G. Bocewicz, Z. Banaszak, I. Nielsen: Delivery-flow routing and scheduling subject to constraints imposed by vehicle flows in fractal-like networks .	135
W. Bożejko, A. Gnatowski, R. Idzikowski, M. Wodecki: Cyclic flow shop scheduling problem with two-machine cells	151
W. Bożejko, J. Pempera, M. Wodecki: A fine-grained parallel algorithm for the cyclic flexible job shop problem	169
W. Bożejko, Z. Chaczko, M. Uchroński, M. Wodecki: Parallel patterns determination in solving cyclic flow shop problem with setups	183
J. Brodny, S. Alszer, J. Krystek, M. Tutak: Availability analysis of selected mining machinery	197
K. Chmielewska, D. Formanowicz, P. Formanowicz: The effect of cigarette smoking on endothelial damage and atherosclerosis development – modeled and analyzed using Petri nets	211
A. Galuszka, J. Krystek, A. Swierniak, C. Lungoci, T. Grzejszczak: Information management in passenger traffic supporting system design as a multi-criteria discrete optimization task	229
M. Kardynska, J. Smieja: Sensitivity analysis of signaling pathway models based on discrete-time measurements	239
J. Kasprzyk, P. Krauze, S. Budzan, J. Rzepecki: Vibration control in semi-active suspension of the experimental off-road vehicle using information about suspension deflection	251
M. Koryl, D. Mazur: Towards emergence phenomenon in business process management	263
M. Koryl: Active resources concept of computation for enterprise software . .	279
H. Krawczyk, M. Nykiel: Mobile devices and computing cloud resources allocation for interactive applications	293
W. Mitkowski, W. Bauer, M. Zagórowska: Discrete-time feedback stabilization	309
J. Pempera: An exact block algorithm for no-idle RPQ problem	323

K. Rżosińska, D. Formanowicz, P. Formanowicz: The study of the influence of micro-environmental signals on macrophage differentiation using a quantitative Petri net based model 331

K. Skrzypczyk, M. Mellado: Vehicle navigation in populated areas using predictive control with environmental uncertainty handling 351

Preface

In September 2016 a Jubilee 20th National Conference of Discrete Automatization Processes organized by the Institute of Automatic Control of the Silesian University of Technology in Gliwice and the Committee on Automatic Control and Robotics of the Polish Academy of Sciences took place in Zakopane. The papers from the conference were selected and, after extension, presented in the special issue entitled "Discrete Systems: Theory and Applications".

Discrete processes cover different spheres of human activity, especially connected with automatic control and production management.

Automatization of production processes constitutes a base of modern manufacturing systems, enabling obtaining high quality products in repeatable way and minimizing production costs. Today's basic automation device is the programmable logic controllers (PLC) used to automate almost all manufacturing processes, including mechanical, energetic, chemical, automotive, food and many others. Developing advanced low-level devices such as programmable PLCs and industrial robots, makes it possible to focus on a far-reaching goal – increasing productivity and reducing production costs by effective using of the resources: devices and employees, including robots. This goal may be the most spectacular in areas where mass production is the norm. In the longer term, Big Data and Industry 4.0 appear on the horizon, which are not only the subject of future work, but today's reality. The practice generates problems of enormous dimensions and the number of variables for which optimization should be made, and the existing methods are – unfortunately – insufficient. Therefore, there is a strong need to create new, more effective methods.

In this special issue selected topics in the field of automation and discrete process control are presented, including matters related to model building, algorithms design as well as creating and implementing of decision-making systems. In addition to works connected with classical discrete control problems related to the use of discrete methodologies in bioinformatics, robotics, transport and business processes management, the paper also includes articles covering broader understood automation control, i.e. discrete-time measurements, vibration control, control under uncertainty and computing cloud resources allocation.

All the papers have been peer-reviewed. We wish to use this opportunity to thank all the reviewers for their contributions to this edition.

Wojciech Bożejko and Mieczysław Wodecki

Delivery-flow routing and scheduling subject to constraints imposed by vehicle flows in fractal-like networks

GRZEGORZ BOCEWICZ, ZBIGNIEW BANASZAK and IZABELA NIELSEN

The problems of designing supply networks and traffic flow routing and scheduling are the subject of intensive research. The problems encompass the management of the supply of a variety of goods using multi-modal transportation. This research also takes into account the various constraints related to route topology, the parameters of the available fleet of vehicles, order values, delivery due dates, etc. Assuming that the structure of a supply network, constrained by a transport network topology that determines its behavior, we develop a declarative model which would enable the analysis of the relationships between the structure of a supply network and its potential behavior resulting in a set of desired delivery-flows. The problem in question can be reduced to determining sufficient conditions that ensure smooth flow in a transport network with a fractal structure. The proposed approach, which assumes a recursive, fractal network structure, enables the assessment of alternative delivery routes and associated schedules in polynomial time. An illustrative example showing the quantitative and qualitative relationships between the morphological characteristics of the investigated supply networks and the functional parameters of the assumed delivery-flows is provided.

Key words: transport network, fractal structure, declarative modeling, multimodal process, delivery flow, vehicles flow.

1. Introduction

Assuming that, just as any system, a supply network (SN) has a structure and behavior (Bocewicz and Banaszak 2015), one can distinguish elements of the structure of such a system (including transport roads, a fleet of vehicles, goods transfer facilities, etc.) and the processes which determine its behavior (e.g. the frequency and timeliness of deliveries, downtime, costs, etc.). This distinction is reflected in the researchers perspective take on SNs. Some researchers accentuating the context of analysis of behavior

G. Bocewicz and Z. Banaszak (corresponding author), are with Department of Computer Science and Management, Faculty of Electronics and Computer Science, Koszalin University of Technology, Poland, e-mails: bocewicz@ie.tu.koszalin.pl, zbigniew.banaszak@tu.koszalin.pl. I. Nielsen is with Department of Mechanical and Manufacturing Engineering, Aalborg University, Aalborg East, Denmark, e-mail: izabela@m-tech.aau.dk

Received 20.12.2016. Revised 30.04.2017.

(i.e. the execution of processes which guarantee the desired quality of supplies) reachable in networks with arbitrarily given structures, and others focusing on the synthesis of structures which enable processes executed in these structures to be completed in a satisfactory manner.

The approach proposed in this paper assumes the possibility of decomposing the above-discussed problems of synthesis and analysis. Such a decomposition would entail treating the flow of means of transport and flow of materials transported by those means separately. In other words, traffic and delivery flow problems observed within SN can be seen as decomposed into the two separate problems of:

- analysis/synthesis of processes related to the movement of means of transport and
- analysis/synthesis of processes of supply/distribution of the goods transported by those means.

If this type of decomposition is implemented, the above problems can be formulated either as problems of finding an optimum goods flow solution for a given structure of a given SN or as problems of scheduling the fleet (and/or configuring the transport structure) of the SN. The solution of the latter should guarantee the expected quality of goods flow (deliveries).

Assuming the SN structure encompasses a part of a transportation network (TN), the following SN analysis and synthesis problems are usually considered: Does an arbitrarily given SN following TN topology make it possible to carry out transport processes that meet user expectations? Is there a SN network structure which guarantees the execution of transport processes that match the expectations of its users? The distinction made between the above two classes of problems assumes that just as any structure of a system determines its admissible behavior, so too the behavior of a system can be determined by its different structures. The elements that determine the solution to these problems are the relationships between selected structural and functional parameters of the system. This means that declarative models of analysis and synthesis problems should incorporate decision variables specifying the topology of the transport networks, the fleet of vehicles that use it and the stations and loading/unloading stops across the network as well as the transport routes of the objects being moved and transport route schedules. The structure of the constraints which connect the decision variables in this type of models allows the formulation of suitably dedicated constraint satisfaction problems (CSPs) (Banaszak and Bocewicz 2014). These types of problems have the added advantage that they are easily implemented in constraint programming languages such as OzMozart, ILOG, ECLiPSe, (Banaszak and Bocewicz 2014; Dang et al. 2014; Sitek and Wikarek 2015).

In the above context, the class of TN analyzed here is limited to network structures with a regular, recursive morphology typical of tree or mesh (grid) topologies (Fig. 1). This category of topologies, which include urban transport systems, are the subject of intensive research (Bahrehdar and Moghaddam 2014; Buhl et al. 2006; Courtat 2012; Sandkuhl and Kirikova 2011).

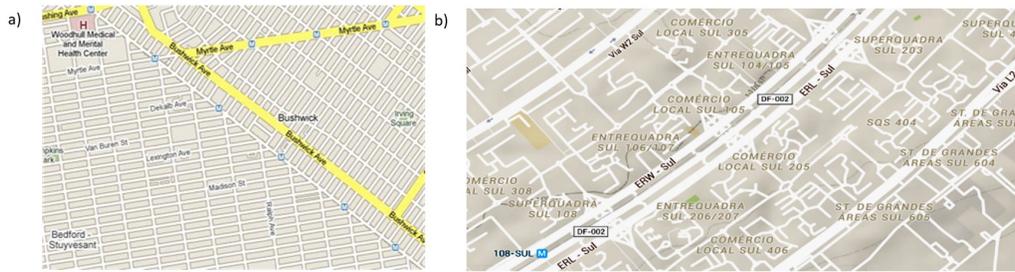


Figure 1: TN structures with a mesh topology (a), and routes with a tree-like structure in the city of Brasilia (b)

For the sake of further considerations let us assume that a SN encompasses all possible branches of transport and transport technologies, including road and rail (surface and underground) transport, e.g. buses, streetcars, subway lines etc.

Multimodal transport is by definition the transportation of freight performed with different, alternative modes of transport along the same transport route, during which goods can be transshipped between different transport modes (Susan 2014). Therefore the concept of a multimodal transport process (MTP) has been introduced (Bocewicz et al. 2015a, 2015b). According to this definition, a MTP involves the movement of objects using different modes of transport in a single, integrated transport chain on a given route. Examples of MTPs include the processes associated with daily commuting (bus – street car – subway), courier services (e.g. DHL), etc. A characteristic feature of such multimodal processes is that their routes are made up of local segments operated by one mode of transport and the objects are moved by suitable local means of transportation. A good illustration of this feature is a passenger travelling by subway who, in the course of his journey, changes from one line to another in accordance with an itinerary.

The considered problem is the simultaneous supply/distribution of different goods from suppliers to recipients, who are all located at various points of the transport network. Knowing the parameters of the local transport companies, one for example wishes to identify delivery routes which would minimize transport time or satisfy the constraints regarding the number of vehicles that can be simultaneously loaded/unloaded at a transshipment point. On the other hand, in solving a synthesis problem subject to the same assumptions and constraints, one builds on knowledge of routes and transport times to find the parameters of local transport companies which would guarantee the delivery of goods within a given time window. Both problem classes belong to the class of computationally hard problems.

In that context the aim of this research is to develop a methodology for the synthesis of regularly structured SNs that follow a fractal-like topology of a TN ensuring fixed cyclic execution of local transport processes. The proposed methodology, which implements sufficient conditions for the synchronization of local cyclic processes allows to develop a method for rapid prototyping of supply/distribution processes which satisfy

the time window constraints given. The adopted simplifying assumptions enable rapid prototyping of admissible solutions in polynomial time. The investigations presented in this paper and the results obtained in the course of this study are a continuation of previous studies collected in (Bocewicz et al. 2015a, 2015b; Bocewicz and Banaszak 2015).

The article is divided into seven sections. Section 2 contains a review of the most important trends in the research into Vehicle Routing Problems. Section 3 presents a declarative modelling driven approach to grid-like SN design while being a part of a TN fractal-like topology. Section 4 provides a problem statement while Section 5 describes a methodology for rapid prototyping of congestion free traffic flows. Computational experiments illustrating the proposed methodology and the scope of future work are presented in Section 6 and Section 7, respectively.

2. Related work

The Vehicle Routing Problem (VRP) is an optimization problem which encounters in logistics and transport and applies to supply chain management in the physical delivery of goods and services. In a simplest form VRP can be defined as the problem of designing least cost delivery routes from a depot to a set of geographically dispersed locations subject to a set of constraints. The VRP is an NP-hard problem (Kumar and Panneerselvam 2012) and one of the most widely studied topics in Operations Research.

A special instance of the VRP is the Periodic VRP (PVRP), in which delivery routes are constructed over a period of time (Francis and Smilowitz 2006; Coene and Arnout 2010). The objective of both the VRP and the PVRP is to minimize travel costs or the total travel distance required to visit all customers during the planning horizon. The difference is that in PVRPs the frequency of visits to customers within a given time horizon varies from customer to customer. Besides of the assumptions traditionally associate with the VRP, the PVRP has to take into account a time horizon, usually subdivided into regular periods, as well as a customer visit frequency stating how often within a particular period a customer must be visited. A solution to the PVRP consists of sets of routes which jointly satisfy demand constraints and frequency constraints. The objective is to minimize the sum of the costs of all routes over the planning horizon (Coene and Arnout 2010).

There are several variants to the PVRP as there are for the general VRP. Among special instances of the PVRPs the Service Choice (PVRP-SC) plays a pivotal role through its focus on more efficient vehicle tours and/or greater service benefit to customers. The PVRP-SC can be stated as follows: Given is a set of customers with known demand and minimum visit frequency requiring service over the given time horizon as well as a fleet of capacitated vehicles, a set of service schedules with headways and service benefits as well as a network with travel times. The goal is to find an assignment of nodes to service schedules and a set of vehicle routes for each period of the given time horizon that minimizes the total routing cost incurred net of the service benefit accrued (Francis and Smilowitz 2006).

Since the design of an optimal route for a fleet of vehicles to service a set of customers is the main objective of the VRP, most study efforts are focused on the problems of management of different modes of transport services rather than on the routing of delivery processes within a given TN. The majority of research focuses either on scheduling available transport modes so that they can service customers in given time windows or on designing supply networks taking into account the size and capacity of the planned fleet and the topology and traffic capacity of routes. Gąska et al. (2015)'s work synchronization of public transport services and urban network analysis by Sevtsuk and Mekonnen (2012) are classical examples of such research.

Relatively few studies are devoted to delivery-flow routing and scheduling in the existing supply networks; examples of this type of solutions are courier, express and parcel services (CEP), the rail-road freight services (Crisalli et al. 2013), and many other door-to-door multimodal supply chains. The vast majority of studies focus on the supply/distribution of goods. Besides vehicle routing other media routings are considered such as message routing (Socievole et al. 2015), energy routing, data routing (Gurakan 2016), cargo routing (Cargo Movement, Defense Transportation Regulation), and courier routing (Lan et al. 2007).

An alternative approach to traffic flow modelling the specific character of passenger traffic is found in the work of Gao and Wu (2011). Their concept of multimodal transport processes assumes that supply processes are executed in an environment of cyclic transport processes. An example of a serial multimodal process is the movement of passengers travelling by different metro lines along a route of arbitrarily selected stations. A characteristic feature of Gao and Wu's approach is that it offers the possibility of distinguishing levels that determine traffic flows of transportation means and traffic flows of multi-commodity flows.

3. Fractal-like supply networks

The numerous road network patterns deployed in public and/or freight transport systems range from tightly structured fractal networks with perpendicular roads in a regular raster pattern to hierarchical networks with sprawling secondary and tertiary roads feeding into arterial roads in a branch like system (Courtat 2012; Kelly and McCabe 2006; Levinson and 2012). The throughput of passengers and/or freight depends on geometrical and operational characteristics of public transportation and/or cargo supply networks.

Recent research (Courtat 2012; Sun et al. 2015; Zhang 2011) draws attention to the fact that the development of urban agglomerations, and in particular the morphology of urban regions, is subject to the laws of recursion, which are best modeled by fractal structures. The consequences of this fact can be used both in predicting the needs related to the expansion of the existing transport infrastructure, as well as planning new industrial and/or urban agglomerations. The main advantages following from the regular structure of supply network layout are flexibility and robustness which are vital to the improvement of robustness of supply/distribution networks (Haghani and Oh 1996;

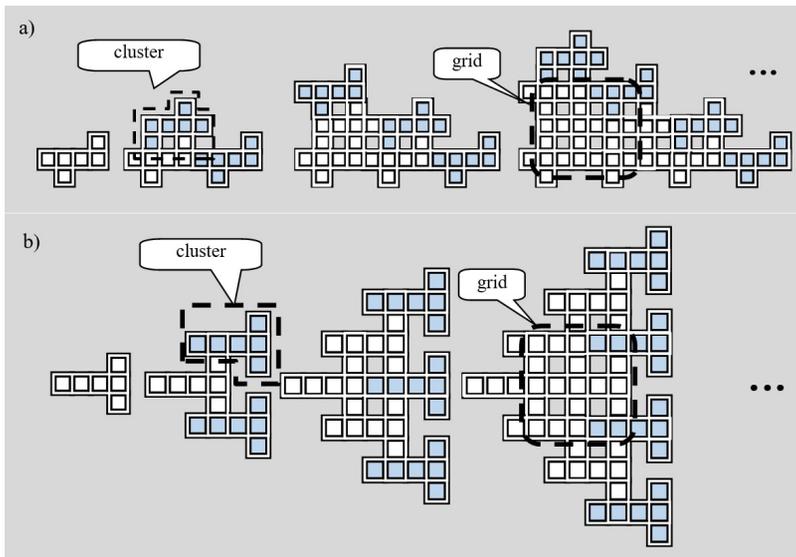


Figure 2: Examples of routes in transport networks with a fractal structure: generated by clusters of shape (a), generated by clusters of shape (b)

Susan 2014). Among numerous reports concerning mesh-like or grid-like as well as fractal-like structures of urban transportation networks the approach proposed in Buhl et al. (2006) should be mentioned. A positive aspect of this the approach is that it allows easy estimation of delivery schedules while taking into account the cyclic behavior of both local transportation modes and the structure of the whole supply network. This is because transportation processes executed by particular lines are usually cyclical and as a consequence multimodal transport processes supported by them also have a periodic character. In other words, these assumptions explicitly constrain the topology of a SN to TNs with fractal structures and implicitly make the efficiency of potential MTPs (e.g. regarding the possible delivery dates) conditional on the admissible flow of traffic (e.g. congestion-free traffic) operating under local transport processes.

For illustration, let us consider two types of TNs with fractal structures as those shown in Fig. 2. A graph theoretical model of SN mirroring the TN structure from Fig. 2 (a) with vertices (which represent network resources, i.e. stations and stops and shared route sections) and edges is shown in Fig. 3 (a). In general case the SN structure can be treated as a part mapped of TN topology. Further discussion, will focus on SNs of grid-like structures, i.e. structures obtained in the course of clusters aggregation following specific rules of fractal growth. This assumption implies that research can be limited to certain elementary structures, the combination of which define the whole transport network. Examples of such structures are marked with a dotted line in Figs. 3 (a), (b).

The considered grid-like structures are composed of clusters (involved in a fractal growth) which are identical repeating substructures, called Elementary Covering Struc-

tures (ECS). The proper example provides ECS from Fig. 3 (b), resources of which are denoted by kR_r that means the r -th resource in the k -th elementary transport network. Local transport processes are marked with labeled arcs whose orientations indicate the direction of flow of local traffic (transport modes). For example, the arc labeled P_i^j denotes that the considered traffic flow, in the local transport process, is comprised of " j " units of an i -th transport mode (Figs. 3 (b)).

Dispatching rules for synchronization of access of local transport processes to shared network resources are marked with labels ${}^k\sigma_r$, e.g. ${}^k\sigma_{10} = (P_4^1, P_5^1)$ describe the order of access of means of transport, e.g. P_4^1, P_5^1 to a shared resource kR_r , e.g. ${}^kR_{10}$; see Figs. 3 (b).

Models of MTP routes that show the sequences of the resources between which objects are moved are represented graphically with bold symbols of nodes and arcs; see Fig. 4 (a).

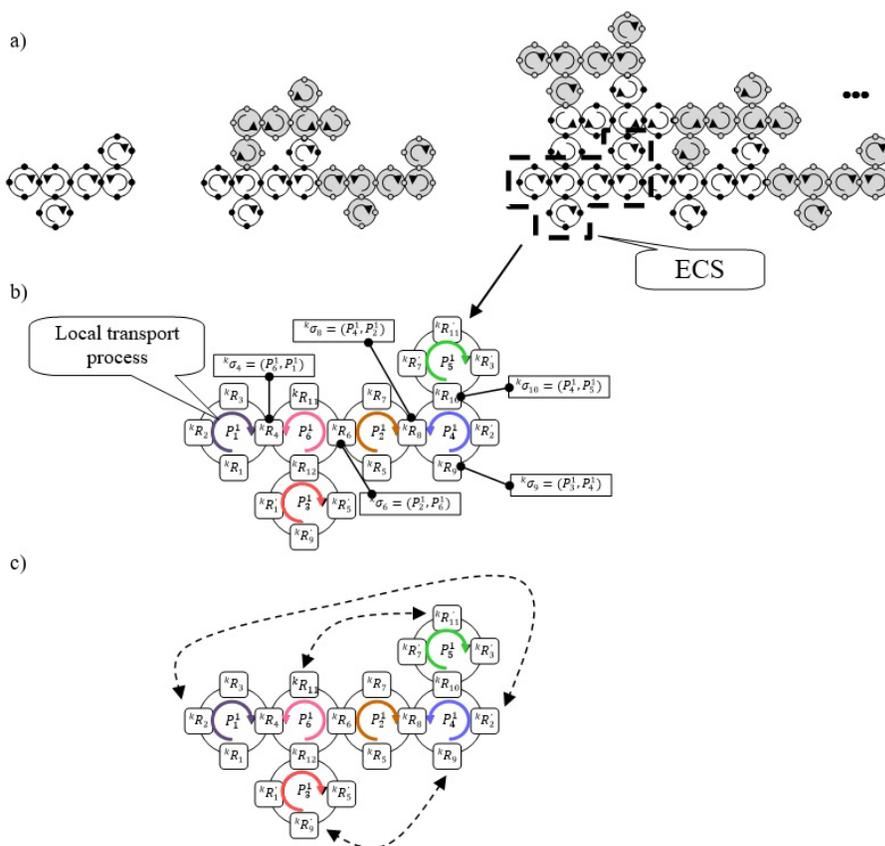
Travel and/or dwelling times of means of transport for various network resources are designated with $t_{i,j}$ which denotes the time of execution of the j -th operation of the i -th transport process. A graphic model which can be used to represent both travel/dwelling times and the waiting times of the means of transport used and the goods moved using those means is a Gantt chart (Fig. 4 (b)). Its graphical representation of timetables and/or supply schedules, allows one to assess waiting times associated with the fact that a means of transport has to wait for access to a requested but currently occupied resource, as well as waiting times of objects transported in MTP chains resulting from unavailability of the scheduled means of transport.

A further assumption is that the repetitive, cyclic behavior of an ECS described in above way implies cyclic, i.e. congestion-free flow of traffic across the network SN (Bocewicz and Banaszak 2015; Bocewicz et al. 2015a).

4. Problem statement

In order to determine the relationships between the structure of TNs along with the means of transport moving in these network and MTPs which determine the routes for the transport of objects, let us consider a reference TN model which integrates the models of a SN structure, local transportation processes and MTP. The reference model of a TN with a fractal structure presented in the previous section makes it possible to develop an appropriate dedicated declarative model allowing the formulation of the aforementioned analysis and synthesis problems as CSPs.

A constraint satisfaction problem: $SC = (X, D, C)$ is usually given by (Sitek and Wikarek 2015) a finite set of decision variables $X = \{x_1, x_2, \dots, x_n\}$, a finite family of finite domains of discrete decision variables $D = \{D_i \mid D_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,j}, \dots, d_{i,m}\}, i = 1..n\}$, and a finite set of constraints limiting the values of the decision variables $C = \{C_i \mid i = 1..L\}$, where: C_i is a predicate $P[x_k, x_l, \dots, x_h]$ defined on a subset of set X . What



Legend:

- ${}^k R_r$ – r -th resource in the k -th transport network,
- P_i^j – local transportation process performed by j -transport units of the i -th transport mode
- ${}^k \sigma_r$ – dispatching priority rule assigned to ${}^k R_r$.

Figure 3: Graph theoretical models of fractal structures corresponding to the route patterns from Fig. 2 (a) are shown in (a); uncovered and covered forms of ECS are shown in (b) and (c), respectively

is sought is an admissible solution, i.e. a solution in which the values of all decision variables X satisfy all constraints C .

Accordingly, the declarative TN model comprises:

- Sets of decision variables describing the structures of
 - local transport processes, i.e. the type and number of resources and modes of transport they use, as well as the associated travel/ dwelling times,
 - the MTP, i.e. the type and number of resources in a chain and the type and number of transport modes used, as well as the associated travel/dwelling times,

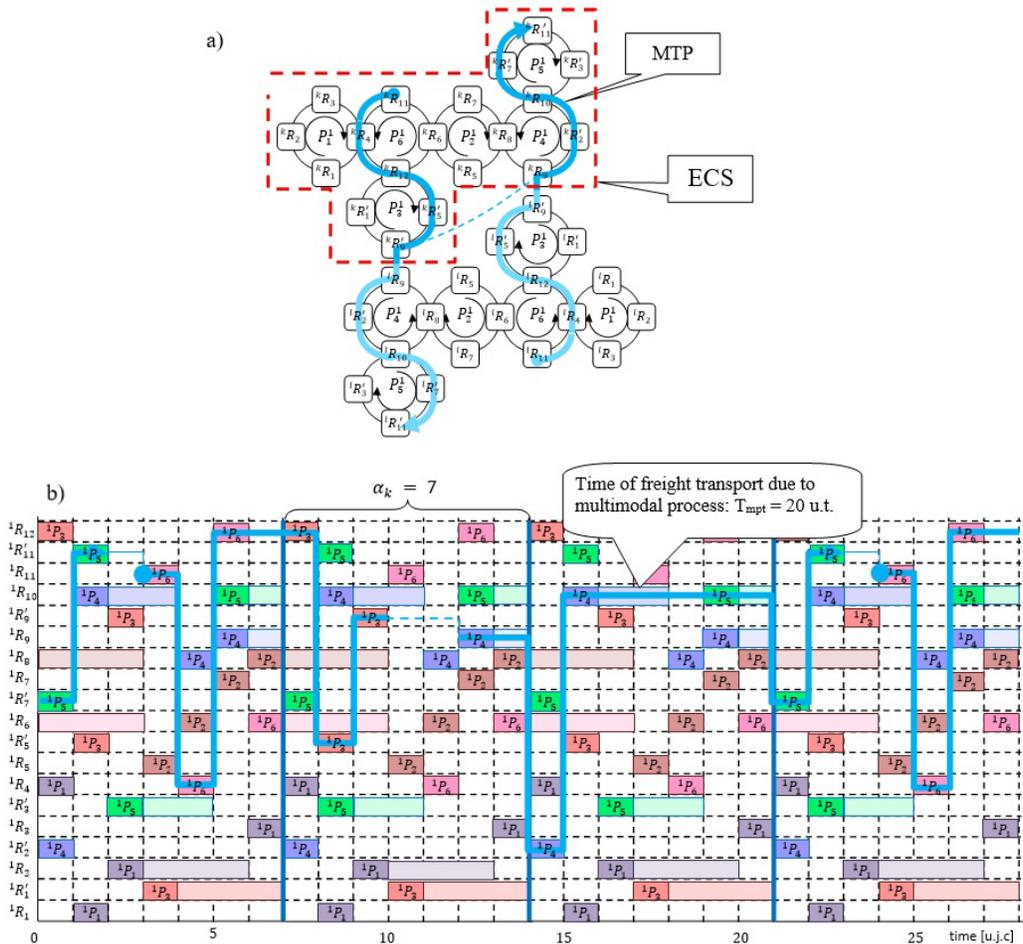


Figure 4: A representation of a MTP performed in a SN given in Fig. 3 (a), Gantt's chart of local transport processes and the MTP (b)

- Domains of decision variables,
- Sets of determining constraints:
 - sets of dispatching rules assigned to shared network resources,
 - transport schedules determining the periods (tacts) and dates of delivery of transported goods.

The set of decision variables found in this model includes:

- a set of resources R found in the structure of the SN and a set of resources \bar{R} determining MTPs,

- a set of local processes occurring in the SN and transportation routes of multimodal processes,
- a set of dispatching rules θ governing the access of local processes/streams to shared resources of the SN and a set of dispatching rules $\bar{\theta}$ governing the access of multimodal processes/streams to the resources of set \bar{R} ,
- initial states: S^0 in the SN (corresponding to the initial allocation of local processes) and \bar{S}^0 corresponding to the initial allocation of MTPs in the set of resources \bar{R} .

Sets of decision domains corresponding to the above-mentioned decision variables determine the integer values of resource availability, integer operation execution times, integer vectors specifying the number of local process streams (vehicles moving along shared routes), permutation vectors of local processes sharing the resources and binary-valued vectors which specify the values of initial states.

Sets of constraints imposing regularity of structure of an SN and the principle of mutual exclusion of local processes as well as constraints enforcing the principle of mutual exclusion of MTPs (Bocewicz et al. 2015) also include constraints which treat as identical the resources of these structures. For example, $\bar{R} \subset R$, and times of goods transport operations with the travel times of the means of transport used for this purpose (Bocewicz et al. 2014). Additionally, it is assumed that there are constraints which relate routes of multimodal processes mP_i with fragments of routes of local processes.

Let us consider the "covered" form of ECS from Fig. 3. (b). The so called "covered" form arises as a result of "gluing together" selected vertices of ECS. Which particular vertices are "glued" together in the "covered" form is determined by the choice of those resources of the elementary transport structure which are shared with the resources of neighboring structures of the TN. For example, a vertex corresponding to resource kR_9 is glued with a vertex corresponding to resource ${}^kR'_{11}$, because resource kR_9 is shared with resource lR_9 which is a counterpart of kR_9 , see Fig. 4 (a).

It can be shown that if the traffic flow in a given covered form of an ECS is free of congestion, i.e. the schedule which specifies it is a cyclic schedule, the flow of traffic in the entire transport network consisting of uncovered forms of ECSs also has a cyclic nature (Bocewicz and Banaszak 2015). This observation allows one to focus on formulating the following CSP, the solution to which is a structure (a set of dispatching rules) that guarantees a congestion-free flow of traffic. In other words, assuming that the behavior of each i -th ECS is represented by a cyclic schedule ${}^{(i)}X' = ({}^{(i)}X_k \mid k = 1, \dots, h, \dots, L_i)$, where: ${}^{(i)}X_h$ is a set of beginning moments of operation of the h -th local process of the i -th ECS and L_i denotes the cardinality of the set of local processes comprising the i -th ECS, the constraint satisfaction problem in question has the following form:

$$PS_i = (\{ {}^{(i)}X', {}^{(i)}\theta, {}^{(i)}\alpha \}, \{ D_X, D_\theta, D_\alpha \}, \{ C_L, C_M, C_D \}) \quad (1)$$

where: ${}^{(i)}X', {}^{(i)}\theta, {}^{(i)}\alpha$ – decision variables,

- ${}^{(i)}X'$ – cyclic schedule of the i -th ECS,
- ${}^{(i)}\theta$ – set of dispatching rules determining the order of operations competing for access to the common resources of the i -th ECS,
- ${}^{(i)}\alpha$ – set of values of periods of local processes occurring in the i -th ECS,

D_X, D_θ, D_α – domains of admissible values of discrete decision variables C_L, C_M, C_D – finite sets of constraints limiting the values of decision variables

- C_L, C_M – sets of conditions constraining the set of potential behaviors of the i -th ECS (Bocewicz et al. 2015a),
- C_D – a set of sufficient conditions that if satisfied guarantee congestion-free (i.e. deadlock-free and collision-free) flow of traffic in a transport network modeled by the i -th ECS and execution of transport operations and loading/unloading operations (i.e. operations competing for access to common resources).

The sought for solution to problem (1) is schedule ${}^{(i)}X'$ which satisfies all the constraints of the family of sets $\{C_L, C_M, C_D\}$. Constraints C_L, C_M (Bocewicz et al. 2015a) ensure that local processes in the "uncovered" form of an ECS are executed in a cyclic manner, i.e. the execution of operations is specified by an appropriate cyclic schedule. They do not guarantee, however, the same for the "covered" form of this structure. The additional constraints C_D , which follow from the match-up rule (Bocewicz and Banaszak 2015) that conditions the fit between cyclic schedules, guarantee that the local processes occurring in the structures which satisfy them are executed in a cyclic manner.

5. Methodology for congestion free traffic flows prototyping

The methodology proposed assumes a regular (grid-like or fractal-like) topology of a SN composed of a finite set of identical repeating substructures, called Elementary Covering Structures (ECSs). An ECS is a directed coherence graph comprised of elementary substructures (modelling local cyclic processes) which occur in the regular structure of an SN such that:

- the graph is composed of elementary cyclic digraphs which model local transportation processes,
- it can be used to tessellate a given regular structure.

The patterns do not exhaust all possible cases of ECSs which may differ in shape (i.e. in the way the elementary substructures are put together) and in scale (i.e. in the number of elementary structures they contain). However, their common feature is that they can be tessellated (like a mosaic) to form a pattern that recreates a given regular structure. Each ECS has a corresponding covered-form ECS which is formed by gluing

together the vertices of the ECS which in the regular structure of the SN are joined with the corresponding fragments of the elementary structures of an adjacent ECS.

The behavior of a given supply network can be predicted on the basis of the behavior of its ECSs. It is easy to observe that an initial allocation of local processes in a regular network (which maps the allocation of processes in the ECS), will be followed in each individual ECS by a next allocation of local processes in compliance with the same priority selection rules. That is because if one replicates the same initial process allocation in all remaining ECSs, the structure will be free of collisions between processes which use the same resources (both shared resources and those integrated/merged in the covered representation). The new allocation in the regular structure concerns the processes and resources which are "copies" of the processes and resources of the contemplated ECS. This means that the successive process allocations which occur in cycles after each initial allocation in the ECS will have their counterparts in the overall regular structure – allocations of all the local processes of the structure. Thus, period α of the processes executed in the regular structure is equal to the period of processes executed in the ECS. This means that cyclic behavior of the ECS implies cyclic behavior of the regular-structure SN.

In summary, a cyclic behavior of the covered form of the ECS entails a cyclic behavior of the supply network. It means that by solving a small-scale computationally hard problem (associated with an ECS), one can solve, in online mode, a large scale problem associated with a corresponding regular-structure supply network.

Moreover, it can be shown that every cyclic behavior of the supply network with a regular structure can be associated with an appropriate time-driven event system, whose period is determined by the period of the covered form of the ECS. Viewing those systems as time-driven discrete event systems, provides the possibility of quantitative analysis of the behavior of an SN oriented toward estimating traffic flow rates, routing of distribution streams, and scheduling of transport fleet and the material flows it supports.

6. Illustrative example

As an illustration of the approach, consider the grid-like networks shown in Fig. 2. Problem (1) considered for corresponding ECSs from Figs. 4 (a) and 5 (a), respectively was implemented and solved in the constraint programming environment Oz-Mozart (CPU Intel Core 2 Duo 3GHz RAM 4 GB). When the assumption was made that all operation times in local processes are the same and equal to $t_{i,j} = 1$ u.t. (unit of time), the first acceptable solution was obtained in less than one second. The sets of the dispatching rules obtained are collected in the Table 1.

An analysis of cyclic schedules ${}^kX'$ allows an easy deduction in both cases of the same value of the cycle length $\alpha_k = 7$ u.t. However, the time of freight $T_{mpi} = 20$ u.t. in the case from Fig. 4 (a) is significantly shorter, than in the case observed in Fig. 5 (a) i.e. $T_{mpi} = 52$ u.t. Therefore, in the considered case, the grid-like SNs obtained in the course

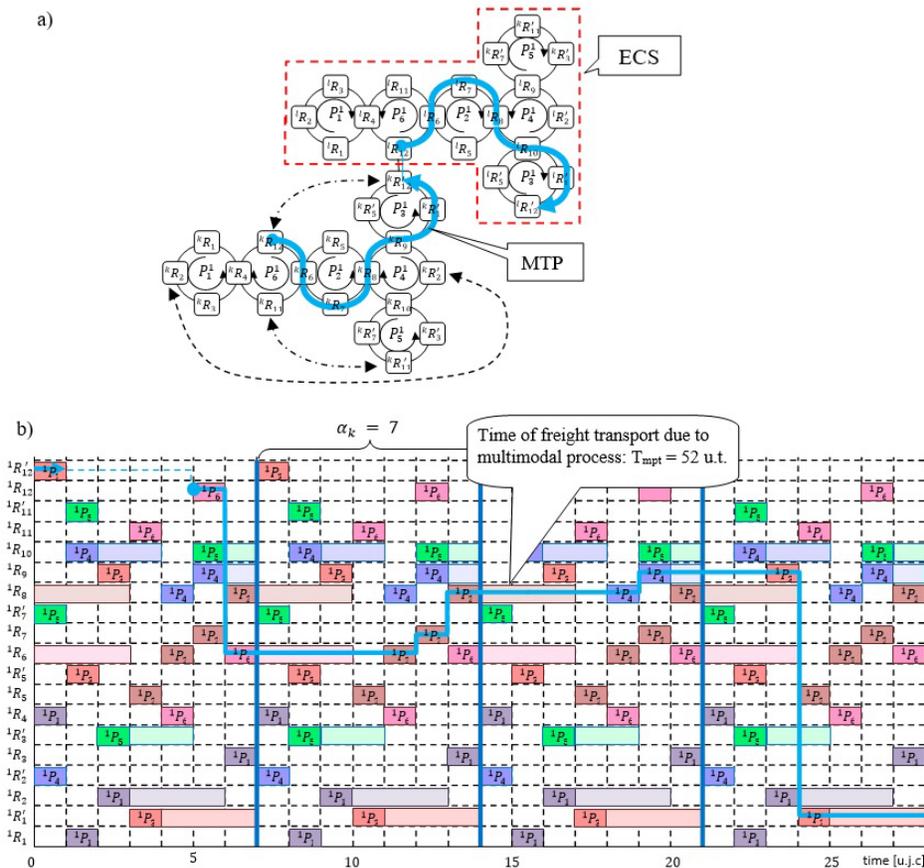


Figure 5: A representation of a MTP performed in a SN following TN from Fig. 2 (b), Gantt's chart of local transport processes and the MTP (b)

Table 1: The sets of the dispatching rules

ECS	Dispatching rules
ECS following SN from Fig. 4 (a)	${}^k\sigma_{11} = (P_5^1, P_6^1), {}^k\sigma_9 = (P_3^1, P_4^1),$ ${}^k\sigma_2 = (P_4^1, P_1^1), {}^k\sigma_4 = (P_1^1, P_6^1),$ ${}^k\sigma_6 = (P_6^1, P_2^1), {}^k\sigma_8 = (P_2^1, P_4^1)$
ECS following SN from Fig. 5 (a)	${}^k\sigma_{12} = (P_3^1, P_6^1), {}^k\sigma_{11} = (P_5^1, P_6^1),$ ${}^k\sigma_2 = (P_4^1, P_1^1), {}^k\sigma_4 = (P_1^1, P_6^1),$ ${}^k\sigma_6 = (P_6^1, P_2^1), {}^k\sigma_8 = (P_2^1, P_4^1)$

of growth of different fractal structures results in the same cyclic steady state of traffic flow guaranteeing congestion free delivery flows.

7. Conclusion

The declarative reference model of a transport system presented in this study enables an analysis of the relationships between the structure of the system and its potential behavior, thus allowing formulation and solving of analysis and synthesis problems corresponding to questions such as: Is it possible to make supplies which meet customer demands in a SN with a preset structure? Is there a SN structure that ensures deliveries which meet user expectations? Because this model focuses on TN with a fractal-like and/or grid-like structure, it allows one to formulate a constraint satisfaction problem and determine the constraints of this problem in the form of sufficient conditions, the satisfaction of which guarantees smooth execution of traffic and delivery flows in this type of networks. These conditions, when implemented in commercially available constraint programming platforms, allow rapid prototyping of alternative transport routes and associated schedules in polynomial time.

The issues of planning and/or prototyping of alternative structures and/or behavior of transport networks with fractal structures presented in this work are part of the broader topic of cyclic scheduling which includes problems occurring in tasks associated with determining timetables, telecommunications transmissions, production planning, etc. In future, while continuing along the line of inquiry related to preventing traffic flow congestion in transport networks, we plan to broaden the scope of our research to include the problems of robust scheduling and the related problem of preventing re-scheduling of timetables in urban transport networks.

References

- [1] S.A. BAHREHDAR and H.R.G. MOGHADDAM: A decision support system for urban journey planning in multimodal public transit network. *Int. J. of Advances in Railway Engineering*, **2**(1), (2014), 58-71.
- [2] G. BOCEWICZ and Z. BANASZAK: Multimodal processes scheduling in mesh-like network environment. *Archives of Control Sciences*, **25**(2), (2015), 237-261.
- [3] G. BOCEWICZ, Z. BANASZAK and I. NIELSEN: Multimodal processes prototyping subject to fuzzy operation time constraints. *15th IFAC Symp.on Information Control Problems in Manufacturing*, **48**(3), (2015), 2103-2108.
- [4] G. BOCEWICZ, W. MUSZYŃSKI and Z. BANASZAK: Models of multimodal networks and transport processes. *Bulletin of the Polish Academy of Sciences Techni-*

- cal Sciences*, **63**(3), (2015a), 636-650.
- [5] G. BOCEWICZ, I. NIELSEN and Z. BANASZAK: Automated guided vehicles fleet match-up scheduling with production flow constraints. *Engineering Applications of Artificial Intelligence*, **30** (2014), 49-62.
- [6] J. BUHL, J. GAUTRAIS, N. REEVES, R.V. SOL'E, S. VALVERDE, P. KUNTZ and G. THERAULAZ: Topological patterns in street networks of self-organized urban settlements. *The European Physical J. B*, **49** (2006), 513-522.
- [7] *Cargo Routing and Movement*, Chapter 202, Cargo Movement, Defense Transportation Regulation–Part II 4 March, 2016, http://www.ustranscom.mil/dtr/part-ii/dtr_part_ii_202.pdf
- [8] S. COENE and A. ARNOUT: On a periodic vehicle routing problem. *J. of the Operational Research Society*, **61**(12), (2010), 1719-1728.
- [9] T. COURTAT: Walk on city maps – Mathematical and physical phenomenology of the city, a geometrical approach. *Modeling and Simulation*, Université Paris-Diderot - Paris VII, (2012), <https://tel.archives-ouvertes.fr/tel-00714310>
- [10] U. CRISALLI, A. COMI and L. ROSATI: A methodology for the assessment of rail-road freight transport policies. *Procedia – Social and Behavioral Sciences*, **87** (2013), 292-305.
- [11] Q.-V. DANG, I. NIELSEN, K. STEGER-JENSEN and O. MADSEN: Scheduling a single mobile robot for part-feeding tasks of production lines. *J. of Intelligent Manufacturing*, **25**(6), (2014), 1271-1287.
- [12] P. FRANCIS and K. SMILOWITZ: Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B*, **40**(10), (2006), 872-884.
- [13] S. GAO and Z. WU Modeling passenger flow distribution based on travel time of urban rail transit. *J. of Transportation Systems Engineering and Information Technology*, **11**(6), (2011), 124-130.
- [14] D. GAŠKA, M. TRPISOVSKY and M. CIEŚLA: Comparison of public transport services organization in the Prague and Warsaw metropolitan regions. *Zeszyty Naukowe Politechniki Śląskiej, Transport* **86**(1926), (2015), 21-32.
- [15] B. GURAKAN, O. OZEL and S. ULUKUS: Optimal energy and data routing in networks with energy cooperation. *IEEE Trans. on Wireless Communications*, **15**(2), (2016), 857-870.
- [16] A. HAGHANI and S.C. OH: Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. *Transportation Research Part A, Policy Pract.*, **30** (1996), 231-250.

- [17] G. KELLY and H. MCCABE: A survey of procedural techniques for city generation. *ITB Journal*, *14* (2006), 87-130.
- [18] S.N. KUMAR and R. PANNEERSELVAM A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, *4* (2012), 66-74, <http://dx.doi.org/10.4236/iim.2012.43010>
- [19] W.W. LAN, C.-J. TING RM AND K.-C. WU: Ant colony system based approaches to the airexpress courier's routing problem. *Proc. of the Eastern Asia Society for Transportation Studies*, *6* (2007).
- [20] D. LEVINSON and A. HUANG: A positive theory of network connectivity. *Environment and planning B: Planning and design*, *39*(2), (2012), 308-325.
- [21] K. SANDKUHL and M. KIRIKOVA: Analysing enterprise models from a fractal organisation perspective – potentials and limitations. *Lecture Notes in Business Information Processing*, *92* (2011), 193-207.
- [22] A. SEVTSUK and M. MEKONNEN: Urban network analysis (A new toolbox for ArcGIS). *Revue Internationale de géomatique*, *2* (2012), 287-305.
- [23] A. SOCIEVOLE, E. YONEKI, F. DE RANGO and J. CROWCROFT: ML-SOR: Message routing using multi-layer social networks in opportunistic communications. *Computer Networks*, *81* (2015), 201-219.
- [24] P. SITEK and J. WIKAREK: A hybrid framework for the modelling and optimisation of decision problems in sustainable supply chain management. *Int. J. of Production Research*, *53*(21), (2015), 6611-6628.
- [25] J.P. SUSAN: Vehicle Re-routing Strategies for Congestion Avoidance. New Jersey Institute of Technology, 2014.
- [26] Y. SUN, M. MAOXIANG LANG and D. WANG: Optimization models and solution algorithms for freight routing planning problem in the multi-modal transportation Nnetworks: A Review of the state-of-the-art. *The Open Civil Engineering J.*, *9* (2015), 714-723.
- [27] J. ZHANG, F. LIAO, T. ARENTZE and H. TIMMERMANS: A multimodal transport network model for advanced traveler information systems. *Procedia Social and Behavioral Sciences*, *20* (2011), 313-322.

Cyclic flow shop scheduling problem with two-machine cells

WOJCIECH BOŻEJKO, ANDRZEJ GNATOWSKI, RADOSŁAW IDZIKOWSKI and MIECZYŚLAW WODECKI

In the paper a variant of cyclic production with setups and two-machine cell is considered. One of the stages of the problem solving consists of assigning each operation to the machine on which it will be carried out. The total number of such assignments is exponential. We propose a polynomial time algorithm finding the optimal operations to machines assignment.

Key words: job shop, cyclic scheduling, multi-machine, assignment.

1. Introduction

Cells equipped with machines of the same types but with different efficiencies are elements of various manufacturing systems. Workpieces flow through a cell in a determined order and are processed on an assigned machine. In cyclic scheduling problems, a determined set of jobs (called MPS, *Minimal Part Set*) is performed multiple times at constant intervals called the cycle time. In another words, each operation of the job is executed on the same machine cyclically. A comprehensive introduction to the problems of cyclic scheduling includes work of Kampmeyer [8].

Processing times of operations on an assigned machine have a direct influence onto the length of the cycle (which is usually minimized). Additionally, machine setups between adjacent operations are considered, therefore minimal cycle time determination constitutes an NP-hard optimization problem, as it comes down to solving a particular traveling salesman problem (Bożejko, Uchroński, Wodecki [1]).

In the paper, we consider the cycle time minimization problem in two-machine cells, which are elements of the non-permutational (the order of operations in each cell can be different) Cyclic Flow Shop (FSP) manufacturing system. The problem, which will be referred to as CFSAP in this paper, is obviously more complex then classic FSP,

W. Bożejko (corresponding author), A. Gnatowski and R. Idzikowski are with Department of Automatics, Mechatronics and Control Systems, Faculty of Electronics, Wrocław University of Technology, Janiszewskiego 11-17, 50-372 Wrocław, Poland. E-mails: {wojciech.bozejko, andrzej.gnatowski, radoslaw.idzikowski}@pwr.edu.pl. M. Wodecki is with Institute of Computer Science, University of Wrocław, Joliot-Curie 15, 50-383 Wrocław, Poland, e-mail: mieczyslaw.wodecki@uwr.edu.pl

The paper is supported by the funds for the development of young researchers.

Received 14.12.2016. Revised 27.04.2017.

as one must determine (i) an operations to machines assignment and also (ii) an order of operations execution on each machine; such that the cycle time (Bożejko, Wodecki [4]) is minimal. A review of computational complexity of the algorithms solving cyclic scheduling problems can be found in Levner i in. [9], similar makespan minimization problem is researched in Sawik [11].

The cycle time in each cell is minimized as it can have an impact onto the cycle time of the whole system. Because of the non-permutational aspect of the problem, each cell can be solved separately. On a top level a modified Tabu Search algorithm is used to determine the permutations in each cell. On the lower level, there are 2^o possible operations-to-machines assignments in each cell, where o is the number of operations. The algorithm with polynomial computational complexity, determining (for a given permutation of operations) the minimal cycle time in each two-machine cell is proposed.

2. Problem formulation

The considered problem consists of finding such permutations of operations in cells and such assignment of operations to one of two machines in each cell, that the cycle time of a manufacturing process is minimal. In the paper we consider non-permutational flow shop scheduling problem with two-machine cells (CFSAP), where (i) the order of operations in a cell is independent from other cells' operations orders and (ii) optimization criterion is a cycle time, which equals to the longest processing time of operations in a cell of the whole system. Due to these two properties, any instance of CFSAP can be divided into q independent, one-cell subproblems (Cyclic Permutation and Assignment Problems, CPAPs), where q is the number of cells. Let T_i denote the minimal cycle time obtained for the CPAP subproblem consisting of the cell i only. Then, the minimal cycle time obtained for CFSAP can be obtained from equation $T = \min\{T_i : i \in \{1, 2, \dots, q\}\}$. For the sake of notation simplicity, hereinafter a single two-machine cell is considered (CPAP).

CPAP can be formulated in a following way: a set of operations $O = \{1, 2, \dots, o\}$, which must be executed on machines from a set $\mathcal{M} = \{1, 2\}$ (constituting a cell) is given. The processing order of the operations (sometimes referred to later as *permutation*) can be represented as a tuple $\pi = (\pi(1), \pi(2), \dots, \pi(o)) \in \Pi$, where Π is a set of all the possible orders. Each operation $i \in O$ must be being executed uninterruptible on the assigned machine $l \in \mathcal{M}$ for p_i^l time, wherein in the cell at most one operation can be processed at the same time. The assignment $P = (Z_1, Z_2)$ is defined by the two disjoint sets Z_1, Z_2 of operations executed on the machines 1 and 2 respectively (of course $Z_1 \cup Z_2 = O$). The set of all the possible assignments

$$P = \bigcup_{I \in \wp(O)} \{(Z_1 = I, Z_2 = O \setminus \{I\})\}, \quad (1)$$

where $\wp(O)$ is an exponential set and has a cardinality of $|\mathcal{P}| = 2^o$. For a given assignment P , a tuple

$$\pi_P^l = (\pi_P^l(1), \pi_P^l(2), \dots, \pi_P^l(|\pi_P^l|)), l \in \mathcal{M}, \tag{2}$$

defines processing order of the operations a machine l , and $v_P(i)$ the machine on which operation i is executed. Between each two adjacent operations on a machine $l \in \mathcal{M}$, a setup with a duration of

$$s_{\pi_P^l(i), \pi_P^l(i+1)}^l, l \in \mathcal{M}, i \in \{1, 2, \dots, |\pi_P^l| - 1\}, \tag{3}$$

must be done, when no other setups can take place, nor operations can be executed. Additionally, due to the cyclic character of the problem, an initial setup

$$s_{\pi_P^l(|\pi_P^l|), 1}^l, l \in \mathcal{M}, \tag{4}$$

must be done before the first operations of each MPS on each machine.

The solution of CPAP consists of the operations-to-machines assignment and times of operations starts and finishes in consecutive MPSes. Starting times of operations in x -th MPS are denoted by $S^x = (S_1^x, S_2^x, \dots, S_o^x)$ and by $C^x = (C_1^x, C_2^x, \dots, C_o^x)$ finishing times. These sequences must fulfill following constrains:

$$\forall i \in \{1, 2, \dots, o\} \quad C_{\pi(i)}^x = S_{\pi(i)}^x + p_{\pi(i)}^{v_P(\pi(i))}, \tag{5}$$

$$\forall i \in \{1, 2, \dots, o\} \quad S_{\pi(i)}^{x+1} = S_{\pi(i)}^x + T, \tag{6}$$

$$\forall i \in \{2, 3, \dots, o\} \quad S_{\pi(i)}^x \geq C_{\pi(i-1)}^x + s_{\pi(i-1), \pi(i)}^{\pi(i-1)}, \tag{7}$$

$$S_{\pi(1)}^{x+1} \geq C_{\pi(o)}^x + s_{\pi(o), \pi(1)}^{\pi(o)}, \tag{8}$$

where x denotes the MPS number, T is the cycle time and

$$s_{\pi_P^l(i)}^{\pi_P^l(i)} = \begin{cases} s_{\pi_P^l(i), \pi_P^l(i+1)}^l & \text{for } i = \{1, 2, \dots, |\pi_P^l| - 1\} \\ s_{\pi_P^l(i), \pi_P^l(1)}^l & \text{for } i = |\pi_P^l| \end{cases}, l \in \mathcal{M}, \tag{9}$$

is the setup time before execution of an operation $\pi_P^l(i) \in Z_l$ for an assignment P and permutation π . The constrain (5) ensures the uninterrupted of operations execution and the equation(6) its cyclic character. The equation (7) represents setups within, and the equation (8) between MPSes.

For an assignment P and permutation π , let $T(P, \pi)$ denote the minimal time of a cell work (cycle time), for which sequences S^x and C^x fulfilling constrains (5)–(8) exist. It is easy to observe, that

$$T(P, \pi) = \sum_{i=1}^o \left(p_i^{v_P(i)} + s_{\pi(i)}^{\pi(i)} \right). \tag{10}$$

The Cyclic Permutation and Assignment Problem boils down to finding such $P^* \in \mathcal{P}$ and $\pi^* \in \Pi$ that minimizes equation (10).

3. Graph model

As discussed in the previous section, each cell constitutes a separate subproblem of finding the optimal order (permutation) and assignment of operations. In the paper, two-level problem solving approach is devised:

Level 1 Search for the optimal permutation by altering the operations order of execution only. Evaluate each solution, assuming that the optimal assignment for any given permutation is known.

Level 2 For a given permutation, calculate the optimal assignment minimizing equation (10).

In the following section, a graph model used in solving Level 2 is presented. Therefore, without loss of generality, following assumptions constituting Cyclic Assignment Problem (CAP) are taken: (i) there is only one cell (each cell is a separate subproblem); (ii) the permutation is natural $\pi = (1, 2, \dots, o)$ and therefore omitted (since operations in the cell can be renumbered).

The graph presented below cannot be used to model the assignments in which all the operations are executed on a single machine (constituting the set P^Z , the assignments from the set P^Z can be evaluated in $O(o)$ time). Therefore, from now on unless stated otherwise, only the assignments from the set $\mathcal{P} \setminus P^Z$ are considered.

Directed graph \mathcal{A} is defined as follows:

$$\mathcal{A} = (\mathcal{W} \cup \mathcal{W}', \mathcal{E} \cup \mathcal{E}'). \quad (11)$$

where \mathcal{W} i \mathcal{W}' are sets of vertices, \mathcal{E} and \mathcal{E}' are sets of arcs, such that:

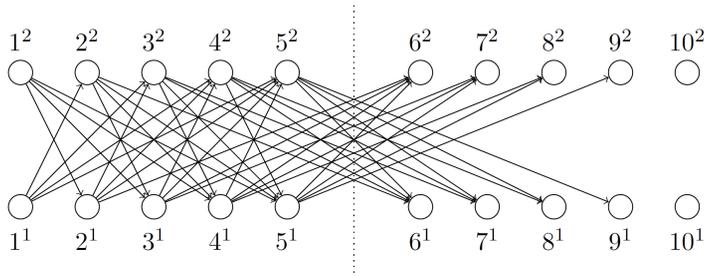
$$\mathcal{W} = \bigcup_{i \in \mathcal{M}} \bigcup_{j=1}^o \{j^i\}, \quad \mathcal{W}' = \bigcup_{i \in \mathcal{M}} \bigcup_{j=o+1}^{2o} \{j^i\}, \quad (12)$$

$$\mathcal{E} = \bigcup_{a \in \mathcal{M}} \bigcup_{i=1}^{o-1} \bigcup_{j=i+1}^o \{(i^a, j^b)\}, \quad \mathcal{E}' = \bigcup_{a \in \mathcal{M}} \bigcup_{i=2}^o \bigcup_{j=o}^{o-1+i} \{(i^a, j^b)\}, \quad (13)$$

where $b \in \mathcal{M} \setminus \{a\}$. Vertex $i^a \in \mathcal{W}$ matches the operation i executed on the machine a , while vertex $j^a \in \mathcal{W}'$, respectively, a copy of the operation $j - o$ from next MPS. Set \mathcal{E} consists of arcs between the vertices of \mathcal{W} ; \mathcal{E}' between the vertices of the set \mathcal{W} and \mathcal{W}' . An example of the graph \mathcal{A} for the number of operations $o = 5$ is presented in Figure 1.

Vertices in a graph \mathcal{A} have no weights. A weight of an arc $(i^a, j^b) \in \mathcal{E} \cup \mathcal{E}'$ is the sum of execution and setup times of the operations (or their copies from the next MPS) from i to $j - 1$. The weights can be calculated from the formula

$$d(i^a, j^b) = p_i^a + s_{i,j+1}^a + \sum_{k=i+1}^{j-1} (p_k^b + s_{k,k+1}^b) \quad (14)$$

Figure 1: Graph \mathcal{A} for $o = 5$.

where

$$s_{i,j}^{i^a} = s_{((i-1) \bmod o)+1, ((j-1) \bmod o)+1}^a, \quad a \in \mathcal{M}, i \in \mathcal{O}, j \in \mathcal{O} \setminus \{i\}, \quad (15)$$

is the setup time between operations matching the vertices i^a and j^b , while

$$p_i^{i^a} = p_{((i-1) \bmod o)+1}^a, \quad a \in \mathcal{M}, i \in \mathcal{O} \quad (16)$$

is the execution time of the operation represented by the vertex i^a .

For a given assignment $P \in \mathcal{P} \setminus \mathcal{P}^Z$, a tuple $\pi'_P = (\pi'_P(1), \pi'_P(2), \dots, \pi'_P(|\pi'_P|))$ denotes all the operations with following operations executed on a different machine

$$\forall i \in \mathcal{O} \setminus \{o\} \quad v_P(i) \neq v_P(i+1) \Rightarrow i \in \pi'_P, \quad (17)$$

$$v_P(o) \neq v_P(1) \Rightarrow o \in \pi'_P, \quad (18)$$

preserving the order of executed operations from π .

For example, for $P = (\{2, 3, 5\}, \{1, 4, 6\})$, $\pi'_P = (1, 3, 4)$. Then, let

$$v(P) = (v_1(P), v_2(P), \dots, v_{|\pi'_P|+1}(P)), \quad (19)$$

where:

$$v_k(P) = \begin{cases} \pi'_P(k)^{v_P(\pi'_P(k))} & \text{for } k \in \{1, 2, \dots, |\pi'_P|\}, \\ (\pi'_P(1) + o)^{v_P(\pi'_P(1))} & \text{for } k = |\pi'_P| + 1. \end{cases} \quad (20)$$

be a path defined by its vertices in the graph \mathcal{A} .

Definition 1 A path $(i^a, (i+1)^b, (i+2)^a, \dots, (i+o)^a)$, $a, b \in \mathcal{M}$, $a \neq b$, $i \in \mathcal{O} \setminus \{o\}$ in the graph \mathcal{A} is called the *highlighted path*. A set of all such paths is denoted by \mathcal{N} .

An example of the *highlighted path* for the assignment $P = (\{1, 4, 6\}, \{2, 3, 5\})$ and the number of operations $o = 5$ is presented in Figure 2. Black circles correspond to the assignment.

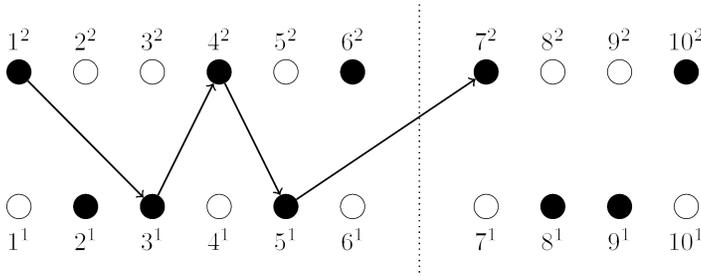


Figure 2: Exemplary highlighted path in graph \mathcal{A} .

Lemma 1 For any highlighted path $\mu \in \mathcal{N}$, corresponding assignment $P \in \mathcal{P} \setminus P^Z$, such that $v(P) = \mu$ exists; i.e.

$$\forall \mu \in \mathcal{N}, \exists P \in \mathcal{P} \setminus P^Z \quad v(P) = \mu. \tag{21}$$

Proof Let $\mu = (\mu_1 = i^a, \mu_2, \dots, \mu_{|\mu|} = (i+o)^a)$ be a highlighted path connecting vertices i^a and $(i+o)^a$, $1 \leq i \leq o$, $a \in \mathcal{M}$. Sequences φ and γ are defined in such a way, that any vertex from the path $\mu_k \in \mu$ can be expressed as $\mu_k = \varphi_k^{\gamma_k}$, $k \in \{1, 2, \dots, |\mu|\}$. It is easy to see that for the assignment $P = (Z_a, Z_b)$, where

$$Z_a = \{1, \dots, \varphi_1\} \cup \{\varphi_{|\varphi|-1} + 1, \dots, n\} \cup \bigcup_{k=1}^{|\varphi|/2} \bigcup_{l=\varphi_{2k-1}}^{\varphi_{2k}} \{l\}, \tag{22}$$

$$Z_b = O \setminus Z_a, \tag{23}$$

and $a = \gamma_1$, there is $v(P) = \mu$. □

Lemma 2 For any assignment $P \in \mathcal{P} \setminus P^Z$, a path $v(P)$ in the graph \mathcal{A} is a highlighted path, i.e.:

$$\forall P \in \mathcal{P} \setminus P^Z, \exists \mu \in \mathcal{N} \quad v(P) = \mu. \tag{24}$$

Proof The Lemma 2 results directly from the definition of path $v(P)$.

Theorem 1 Function $f : \mathcal{P} \setminus P^Z \rightarrow \mathcal{N}$, $f(P) = v(P)$ is bijective.

Proof Function f must fulfill properties of bijection:

$$\forall \mu \in \mathcal{N}, \exists P \in \mathcal{P} \setminus P^Z \quad f(P) = v(P) = \mu, \tag{25}$$

$$\forall P \in \mathcal{P} \setminus P^Z, \exists \mu \in \mathcal{N} \quad f(P) = v(P) = \mu, \tag{26}$$

$$\forall P_1, P_2 \in \mathcal{P} \setminus P^Z \quad f(P_1) = f(P_2) \Leftrightarrow P_1 = P_2. \tag{27}$$

From Lemmas 1 and 2, conditions (25) and (26) are met. Let us consider first $|v(P)| - 1$ vertices of a path $v(P)$. From the definition, $v_k(P) = \pi'_p(k)^{v_p(\pi'_p(k))}$,

$k \in \{1, 2, \dots, |\nu_P| - 1\}$. Since permutation π'_P includes all the operations followed by the changes in the machine assignments, it is easy to see that

$$\forall P_1, P_2 \in \mathcal{P} \quad (P_1 = (Z_1, Z_2) \wedge P_2 = (O \setminus Z_1, O \setminus Z_2) \vee P_1 = P_2) \Leftrightarrow \pi'_{P_1} = \pi'_{P_2}. \quad (28)$$

The case $P_1 = (Z_1, Z_2)$, $P_2 = (O \setminus Z_1, O \setminus Z_2)$ can be rejected because then: $\nu_{P_1}(k) \neq \nu_{P_2}(k)$, $k \in O$. Hence, the condition from the equation (27) is fulfilled. \square

The consequence of the Theorem 1 is an existence of the inverse function to f , $g(f(P)) = P$, $P \in \mathcal{P} \setminus P^Z$. Bijective function g assigns a highlighted path to an assignment from the set $\mathcal{P} \setminus P^Z$.

Lemma 3 For any assignment $P \in \mathcal{P} \setminus P^Z$, the sum of weights of arcs of highlighted path $\nu(P)$ is equal to the minimal cycle time $T(P)$

$$d(\nu(P)) = T(P). \quad (29)$$

Proof The proof is based on calculation of sum of weights of arcs of path $\nu(P)$

$$d(\nu(P)) = \underbrace{\sum_{k=1}^{|\pi'| - 1} d((\nu_k(P), \nu_{k+1}(P)))}_{X(P)} + \underbrace{d((\nu_{|\pi'|}(P), \nu_{|\pi'|+1}(P)))}_{Y(P)}, \quad (30)$$

where $X(P)$ is the sum of weights of arcs belonging to set \mathcal{E} and $Y(P)$ to set \mathcal{E}' . Values $X(P)$ and $Y(P)$ can be determined by Eq. (14). After transformations (described in detail in report [7])

$$\begin{aligned} d(\nu(P)) &= X(P) + Y(P) = \\ &= \sum_{k=\pi'_P(1)}^{\pi'_P(|\pi'_P|)-1} \left(p_k^{\nu_P(k)} + s_P^\alpha(k) \right) + \sum_{k=\pi'_P(|\pi'_P|)}^o \left(p_k^{\nu_P(k)} + s_P^\alpha(k) \right) + \\ &\quad + \sum_{k=1}^{\pi'_P(1)-1} \left(p_k^{\nu_P(k)} + s_P^\alpha(k) \right) = \\ &= \sum_{k=1}^o \left(p_k^{\nu_P(k)} + s_P^\alpha(k) \right). \end{aligned} \quad (31)$$

right sides of Eqs. (31) and (10) are equal, therefore $d(\nu(P)) = T(P)$. \square

Theorem 2 In graph \mathcal{A} , weight of highlighted path with the minimum weight is equal to the minimal cycle time $T(P)$ for $P \in \mathcal{P} \setminus P^Z$; i.e.

$$\arg \max_{\mu \in N} \{d(\mu)\} = \nu(\arg \max_{P \in \mathcal{P} \setminus P^Z} \{T(P)\}). \quad (32)$$

Proof From Lemma 3 and Theorem 1

$$\begin{aligned} \arg \max_{\mu \in \mathcal{N}} \{d(\mu)\} &\stackrel{\text{th. 1}}{=} \arg \max_{\mu \in \bigcup_{P \in \mathcal{P} \setminus P^Z} \{v(P)\}} \{d(\mu)\} = \\ &= v(\arg \max_{P \in \mathcal{P} \setminus P^Z} \{d(v(P))\}) \stackrel{\text{lm. 3}}{=} v(\arg \max_{P \in \mathcal{P} \setminus P^Z} \{T(P)\}). \end{aligned}$$

The obvious consequence of Theorem 2 is the equation □

$$\max_{\mu \in \mathcal{N}} \{d(\mu)\} = \max_{P \in \mathcal{P} \setminus P^Z} \{T(P)\}. \quad (33)$$

4. Solving CAP

In this section, two CAP solving algorithms are presented and their computational complexity is discussed.

4.1. The polynomial algorithm (PA)

Proposed algorithm utilizes graph \mathcal{A} (described in previous section) to determine the optimal assignment $P^* \in \mathcal{P}$, and therefore solve CAP. The algorithm is summarized in Algorithm 1.

Algorithm 1 The polynomial algorithm (PA)

- 1: Construct graph \mathcal{A} .
 - 2: **for all** $i \in \mathcal{O} \setminus \{o\}$ **do**
 - 3: **for all** $a \in \mathcal{M}$ **do**
 - 4: Find the path with minimum weight from vertex i^a to $(i+o)^a$.
 - 5: From paths obtained in steps 3–5, choose the path with minimal weight and determine the corresponding assignment $P_1 \in \mathcal{P} \setminus P^Z$.
 - 6: Calculate the minimal cycle time $T(P)$ of the individually analyzed cases $P_2 = (\emptyset, \mathcal{O})$ and $P_3 = (\mathcal{O}, \emptyset)$.
 - 7: **return** $P^* = \arg \min_{P \in \{P_1, P_2, P_3\}} \{T(P)\}$
-

In lines 2–5. the algorithm determines the highlighted path with minimal weight

$$\min_{P \in \mathcal{P} \setminus P^Z} \{d(v(P))\} = \min_{P \in \mathcal{P} \setminus P^Z} \{T(P)\} \quad (34)$$

and thus, by Theorem 2, the corresponding assignment from $\mathcal{P} \setminus P^Z$ with the minimal cycle time. In line 7., the values of $T(P)$ for $P \in P^Z$ are calculated, hence the algorithm determines the optimal solution.

Theorem 3 *For the Cyclic Assignment Problem, the optimal assignment P^* minimizing the cycle time can be determined in $O(o^3)$ time.*

Proof The proof is based on the analysis of the computational complexity of Alg. 1. Constructing graph \mathcal{A} from line 2. requires calculation of $O(o^2)$ weight of arcs, where each weight is the sum of $O(o)$ elements, hence the computational complexity is $O(o^3)$. Line 5. can be realized by sequentially determining the longest path from initial vertex to the following vertices (in topological order). Because there are $O(o)$ in- and out-arcs from each vertex, complexity of the lines 5. equals $O(o^2)$. The operations from line 5. are performed $O(o)$ times (lines 3–5.), resulting in $O(o^3)$ time complexity. Line 7. comes down to determining the minimal cycle time for separately evaluated assignments from P^Z . They can be calculated from the formula (10) in $O(o)$ time. Finally, the computational complexity of the algorithm equals $O(o^3) + O(o^3) + O(o) = O(o^3)$. \square

4.2. One Opt algorithm

One opt algorithm (1-Opt) is a heuristic for CAP. Pseudocode for the algorithm is presented in Algorithm 2.

Algorithm 2 One Opt

```

1: for all  $i \in O$  do
2:    $P' \leftarrow (Z_{v_P(i)} \setminus \{i\}, Z_{\{1,2\} \setminus v_P(i)} \cup \{i\})$ 
3:   if  $T(P) > T(P')$  then
4:      $P \leftarrow P'$ 
5:   goto line 1.
6: return  $P$ 

```

In each iteration, an assignment of each operation (one at a time) is temporarily changed. If the change lowers the minimal cycle time, it becomes permanent. The algorithm stops when no change in an assignment of a single operation can improve the minimal cycle time.

Since there are 2^o different assignments and $T(P)$ can be calculated in $O(o)$ time, the total computational complexity of 1-Opt cannot exceed $O(o \cdot o \cdot 2^o) = O(o^2 2^o) = O(2^o)$. It is worth noting, that as shown in the computational experiments, in practical applications, 1-Opt can be much faster than the PA algorithm (with $O(o^3)$ computational complexity).

5. Solving CFSAP with Tabu Search algorithm

CFSAP consists of q independent subproblems, one for each cell. Therefore one should focus on optimizing (which is done in this paper, with Tabu Search algorithm)

the subproblem determining the total cycle time of the problem (bottleneck subproblem). This strategy is implemented in Algorithm 3.

Algorithm 3 CFSAP solving strategy

- 1: For each cell, create a TS algorithm instance for solving one-cell subproblem.
 - 2: Run 1 iteration of TS algorithm solving the subproblem with the longest minimal cycle time $i = \arg \max_{i=1,2,\dots,q} \{T(P_i, \pi_i)\}$.
 - 3: If the time for calculations is not over yet, go to line 2.
-

The Tabu Search (TS) algorithm is a local search metaheuristic, proposed for the first time by Glover in [5]. Through years, the original idea has been modified repeatedly, creating multiple variants of the TS algorithm; applied to a wide range of scheduling problems (such as famous TSAB [10], neuro-tabu [2], or parallel TS [3]).

The algorithm used in the paper utilizes two types of a memory:

- TL** Tabu List, designed to avoid cycles and to leave local minimums. It consists of L last moves. Whenever the capacity is exceeded, the oldest move is removed from the list.
- LTM** Long-Term Memory, storing promising solutions (and associated TS states consisting of: current solutions, tabu lists and neighbourhoods) to provide diversification, each state can be used only once.

The algorithm (shown on Fig. 3) starts with generation of an initial solution. The solution is provided by a simple heuristic — operations are scheduled in an ascending order according to their number and assigned to a single machine. Then, a neighbourhood is generated (the neighbour is defined by the swap move, e.g. unordered pair of operations to be swapped in the permutation.) by the procedure described in Algorithm 4. Then, the

Algorithm 4 Neighbourhood creating procedure

- 1: Find a pair of consecutive operations with different machines assigned. Take the first operation.
 - 2: If line 1. provided no operations, take first and last operation from the permutation.
 - 3: Create neighbours, by swapping in the permutations operations from line 1. or 2. with α following and α preceding operations.
 - 4: Remove duplicates (neighbours with the same pairs of operations to be swapped).
-

neighbours are evaluated by one of the three methods:

- PA** For all the neighbours, compute the exact value of the minimal cycle time, using the polynomial algorithm.
- 1-Opt** For all the neighbours, estimate the value of the minimal cycle time, using the 1-Opt heuristic.

Hybrid perform the previously described **1-Opt** method and then compute the exact value of the minimal cycle time of the best β neighbours, using the polynomial algorithm.

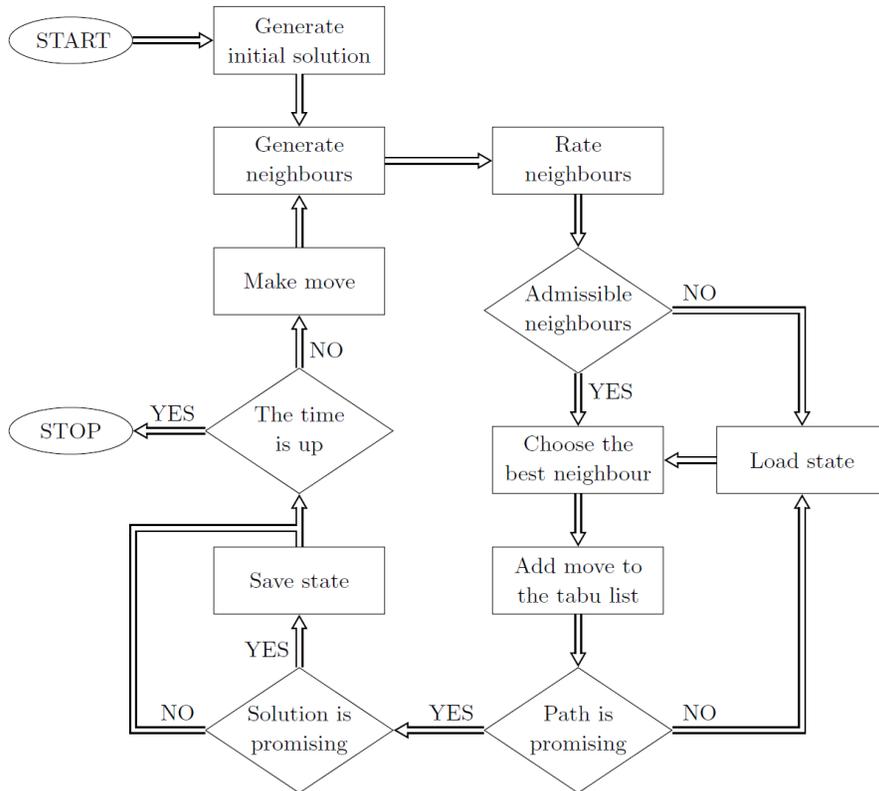


Figure 3: Schematic diagram of the Tabu Search algorithm.

The neighbours consisting of the moves from TL are removed, unless they improve the best known value of the minimal cycle time (aspiration criterion). If all the neighbours are removed, the last promising state is loaded from LTM. Then, the neighbour with the lowest value of the minimal cycle time is chosen and the associated move is added to TL. If the best solution found has not been improved since $1000 + 0.1 \cdot \textit{iteration number}$ iterations, the last promising state is loaded from LTM. If the neighbour fulfills at least one of the following conditions:

- there are less than 5 states memorized;
- there are less than 200 states memorized and $T \leq 1.1 \cdot T_{best}$;
- $T < T_{best}$;

the current state of TS is saved in LTM. If time for calculations is not over yet, the move is applied to the current solution, finishing an iteration.

6. Computational experiments

This section provides a description of an experimental evaluation of the proposed algorithms. First, the comparison of speed and quality of the results obtained in various scenarios by the CAP solving algorithms is presented. Then, effectiveness of the three Tabu Search variants on benchmark instances is tested.

The algorithms were implemented in C++ programming language, compiled with the default compiler of Microsoft Visual Studio 2015. The programs were executed on PC equipped with Intel Core i7-4930K CPU @3.4GHz, 32GB RAM and Windows 10 Education.

6.1. CAP solving algorithms

The two Cyclic Assignment Problem solving algorithms (namely 1-Opt and PS) were experimentally compared. Performance was measured on randomly generated instances of the following sizes: $o = 10, 20, 40, 80, 160, 320, 640, 1280, 2560$. For each size, 16 instances were generated (144 in total).

The algorithms were tested in three usage scenarios (as shown in Fig. 4), with different initial solutions for the experiments:

- experiments 1 and 2 – random solution;
- experiments 3 and 4 – optimal assignment, with a random swap move performed on the permutation;
- experiments 5 and 6 – random solution processed by 1-Opt algorithm, with a random swap move performed on the permutation.

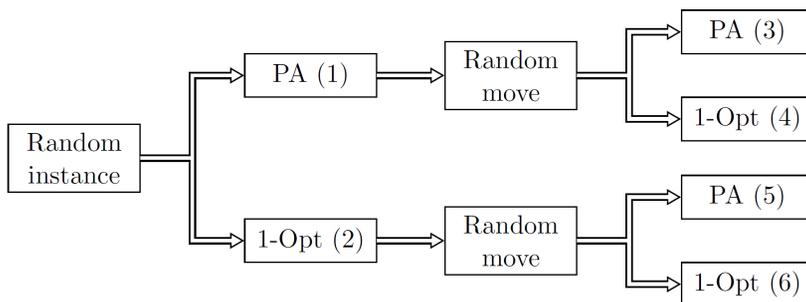


Figure 4: Setup for the experiments on CAP solving algorithms.

For each experiment and instance, computation times and the minimal cycle times T were measured. The 1-Opt algorithm is a heuristic, therefore a gap ΔT between the obtained cycle time T and the optimal cycle time T^* was also calculated

$$\Delta T = \frac{T - T^*}{T^*} \cdot 100\%.$$

The results of the experiments are presented in Fig. 5 and Tab. 1. The computation time of PA is almost unaffected by the initial solution and 2–4 times longer than 1-OPT starting from a random solution. As shown in the experiments 2, 4, 6; the quality of the results obtained by 1-OPT are dependent on the quality of the initial solution. With an initial solution close to the optimal, 1-OPT provides relatively good results up to about 1000 times faster than PA.

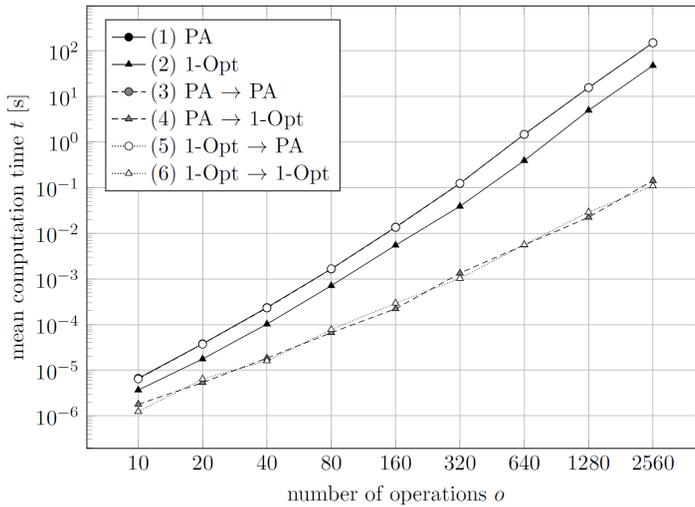


Figure 5: Computation times of the algorithms from experiments 1–6.

6.2. CFSAP solving algorithms

Since the problem has not yet been researched before, no available benchmarks existed. The test instances were therefore generated and published online [6]. More details on the data can be found in the report [7]. In the paper, the first 120 instances were used ($gi0001$ – $gi0120$, as shown in Tab. 4). The values of TS algorithm parameters were obtained experimentally (Tab. 2).

Three variants of the Tabu Search algorithm were tested with a time limit of 60 seconds for each instance. The minimal cycle time, mean neighbourhood size and a number of iterations were measured. The instances were divided according to their sizes into 12 groups. For each group, an average gap ΔT between obtained cycle time T and

Table 1: The results of the CAP solving algorithms tests. Columns correspond to experiments 1–6.

o	Mean computation time [s]						Mean ΔT [%]		
	1	2	3	4	5	6	2	4	6
10	6.66E-6	3.66E-6	6.50E-6	1.78E-6	6.41E-6	1.24E-6	2.48	1.35E+0	2.07
20	3.79E-5	1.76E-5	3.77E-5	5.34E-6	3.70E-5	6.39E-6	1.56	2.18E-1	1.08
40	2.37E-4	1.02E-4	2.31E-4	1.84E-5	2.32E-4	1.60E-5	1.71	3.96E-1	1.67
80	1.66E-3	7.11E-4	1.64E-3	6.63E-5	1.67E-3	7.76E-5	1.39	9.68E-2	1.23
160	1.37E-2	5.47E-3	1.34E-2	2.21E-4	1.35E-2	2.95E-4	1.47	8.10E-2	1.49
320	1.24E-1	3.90E-2	1.23E-1	1.33E-3	1.24E-1	1.03E-3	1.70	5.43E-2	1.68
640	1.47E+0	3.91E-1	1.48E+0	5.53E-3	1.47E+0	5.63E-3	1.54	2.59E-2	1.56
1280	1.56E+1	4.94E+0	1.56E+1	2.24E-2	1.56E+1	2.91E-2	1.45	1.24E-2	1.45
2560	1.50E+2	4.72E+1	1.49E+2	1.42E-1	1.49E+2	1.09E-1	1.45	4.48E-3	1.45

Table 3: An average gap to the best obtained result, grouped by an instance size.

Group $n \times q$	ΔT [%]		
	PA	1-OPT	Hybrid
10 × 10	0	0	0
10 × 15	0	0	0
10 × 20	0.1393	0	0
20 × 10	0.1851	0.1044	0
20 × 15	0	0.0093	0
20 × 20	0.0206	0.1016	0
50 × 10	4.4533	0.4277	0.2084
50 × 15	5.4054	0.4060	0.2277
50 × 20	4.9372	0.6615	0.0787
100 × 10	21.2598	0.2158	0.1161
100 × 15	20.7294	0.4415	0.0994
100 × 20	20.3415	0.6333	0.2731
MEAN:	6.4560	0.2501	0.0836

Table 2: The CFSAP solving algorithms parameters.

Algorithm	L	α	β
PA	50	15	-
1-OPT	100	10	-
Hybrid	75	15	1

the best cycle time across the three algorithms T_{min} was calculated

$$\Delta T = \frac{T - T_{min}}{T_{min}} \cdot 100\%.$$

The results of the experiments are summarized in Tables 3 and 4. The Hybrid TS algorithm provided better results for the majority of instances, followed by the 1-Opt. PA performance dropped significantly for the bigger instances, probably due to an insufficient number of TS iterations (for $n = 100$, an average of 36 iterations; compared to 597 for 1-OPT and 594 for Hybrid).

Table 4: The best results obtained for the instances *gi001–gi120*; 60 seconds of the computation time for each instance.

name	$n \times q$	T	name	$n \times q$	T	name	$n \times q$	T
<i>gi001</i>	10 × 10	608	<i>gi041</i>	20 × 15	992	<i>gi081</i>	50 × 20	2147
<i>gi002</i>	10 × 10	623	<i>gi042</i>	20 × 15	1097	<i>gi082</i>	50 × 20	2285
<i>gi003</i>	10 × 10	500	<i>gi043</i>	20 × 15	989	<i>gi083</i>	50 × 20	2270
<i>gi004</i>	10 × 10	585	<i>gi044</i>	20 × 15	1038	<i>gi084</i>	50 × 20	2264
<i>gi005</i>	10 × 10	549	<i>gi045</i>	20 × 15	998	<i>gi085</i>	50 × 20	2192
<i>gi006</i>	10 × 10	629	<i>gi046</i>	20 × 15	1077	<i>gi086</i>	50 × 20	2052
<i>gi007</i>	10 × 10	603	<i>gi047</i>	20 × 15	997	<i>gi087</i>	50 × 20	2130
<i>gi008</i>	10 × 10	506	<i>gi048</i>	20 × 15	921	<i>gi088</i>	50 × 20	2172
<i>gi009</i>	10 × 10	588	<i>gi049</i>	20 × 15	902	<i>gi089</i>	50 × 20	2530
<i>gi010</i>	10 × 10	493	<i>gi050</i>	20 × 15	1035	<i>gi090</i>	50 × 20	2498
<i>gi011</i>	10 × 15	597	<i>gi051</i>	20 × 20	914	<i>gi091</i>	100 × 10	4465
<i>gi012</i>	10 × 15	554	<i>gi052</i>	20 × 20	1019	<i>gi092</i>	100 × 10	4269
<i>gi013</i>	10 × 15	594	<i>gi053</i>	20 × 20	997	<i>gi093</i>	100 × 10	4201
<i>gi014</i>	10 × 15	505	<i>gi054</i>	20 × 20	928	<i>gi094</i>	100 × 10	4330
<i>gi015</i>	10 × 15	631	<i>gi055</i>	20 × 20	973	<i>gi095</i>	100 × 10	4158
<i>gi016</i>	10 × 15	626	<i>gi056</i>	20 × 20	1011	<i>gi096</i>	100 × 10	4355
<i>gi017</i>	10 × 15	585	<i>gi057</i>	20 × 20	943	<i>gi097</i>	100 × 10	4363
<i>gi018</i>	10 × 15	529	<i>gi058</i>	20 × 20	959	<i>gi098</i>	100 × 10	4268
<i>gi019</i>	10 × 15	649	<i>gi059</i>	20 × 20	1079	<i>gi099</i>	100 × 10	4143
<i>gi020</i>	10 × 15	561	<i>gi060</i>	20 × 20	940	<i>gi100</i>	100 × 10	4313
<i>gi021</i>	10 × 20	535	<i>gi061</i>	50 × 10	2176	<i>gi101</i>	100 × 15	4285
<i>gi022</i>	10 × 20	586	<i>gi062</i>	50 × 10	2100	<i>gi102</i>	100 × 15	4421
<i>gi023</i>	10 × 20	602	<i>gi063</i>	50 × 10	2147	<i>gi103</i>	100 × 15	4288
<i>gi024</i>	10 × 20	607	<i>gi064</i>	50 × 10	2289	<i>gi104</i>	100 × 15	4295
<i>gi025</i>	10 × 20	598	<i>gi065</i>	50 × 10	2171	<i>gi105</i>	100 × 15	4295
<i>gi026</i>	10 × 20	572	<i>gi066</i>	50 × 10	2185	<i>gi106</i>	100 × 15	4257
<i>gi027</i>	10 × 20	605	<i>gi067</i>	50 × 10	2152	<i>gi107</i>	100 × 15	4610
<i>gi028</i>	10 × 20	619	<i>gi068</i>	50 × 10	2321	<i>gi108</i>	100 × 15	4579
<i>gi029</i>	10 × 20	646	<i>gi069</i>	50 × 10	2173	<i>gi109</i>	100 × 15	4578
<i>gi030</i>	10 × 20	591	<i>gi070</i>	50 × 10	2242	<i>gi110</i>	100 × 15	4219
<i>gi031</i>	20 × 10	958	<i>gi071</i>	50 × 15	2222	<i>gi111</i>	100 × 20	4472
<i>gi032</i>	20 × 10	838	<i>gi072</i>	50 × 15	2403	<i>gi112</i>	100 × 20	4460
<i>gi033</i>	20 × 10	974	<i>gi073</i>	50 × 15	2305	<i>gi113</i>	100 × 20	4446
<i>gi034</i>	20 × 10	904	<i>gi074</i>	50 × 15	2279	<i>gi114</i>	100 × 20	4504
<i>gi035</i>	20 × 10	1002	<i>gi075</i>	50 × 15	2283	<i>gi115</i>	100 × 20	4417
<i>gi036</i>	20 × 10	998	<i>gi076</i>	50 × 15	2014	<i>gi116</i>	100 × 20	4503
<i>gi037</i>	20 × 10	988	<i>gi077</i>	50 × 15	2185	<i>gi117</i>	100 × 20	4442
<i>gi038</i>	20 × 10	872	<i>gi078</i>	50 × 15	2236	<i>gi118</i>	100 × 20	4411
<i>gi039</i>	20 × 10	1009	<i>gi079</i>	50 × 15	2223	<i>gi119</i>	100 × 20	4506
<i>gi040</i>	20 × 10	1000	<i>gi080</i>	50 × 15	2136	<i>gi120</i>	100 × 20	4556

7. Final remarks

Cyclic work of a two-machine production cell with setup times is considered in the paper. We proved, that the optimal operations to machines assignment (for a fixed order of operations), can be determined in the polynomial time $O(o^3)$, where o is the number of operations, despite the exponential number of all the possible assignments. In the further research we plan to extend our considerations onto cells with the number of machines greater than two.

References

- [1] W. BOŻEJKO, M. UCHROŃSKI and M. WODECKI: Parallel metaheuristics for the cyclic flow shop scheduling problem. *Computers & Industrial Engineering*, **95** (2016), 156-163.
- [2] W. BOŻEJKO, A. GNATOWSKI, T. NIŻYŃSKI and M. WODECKI: Tabu search algorithm with neural tabu mechanism for the cyclic job shop problem. In *Artificial Intelligence and Soft Computing: 15th International Conference, ICAISC 2016, Zakopane, Poland, June 12-16, 2016, Proceedings, Part II*, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, Eds. Springer International Publishing, Cham, 2016, 409-418.
- [3] W. BOŻEJKO, M. UCHROŃSKI and M. WODECKI: *Multi-GPU Tabu Search Metaheuristic for the Flexible Job Shop Scheduling Problem*. Springer International Publishing, Heidelberg, 2014, 43-60.
- [4] W. BOŻEJKO and M. WODECKI: Problemy cykliczne z równoległymi maszynami (in Polish). In *Automatyzacja procesów dyskretnych, Teoria i zastosowania (red. A. Świerniak, J. Krystek)*, A. Świerniak and J. Krystek, Eds. Gliwice, 2014, 27-36.
- [5] F. GLOVER: Future paths for integer programming and links to artificial intelligence. *Computer & Operational Research*, **13**(5), (1986), 533-549.
- [6] A. GNATOWSKI: Benchmarks for nonpermutational flowshop with 2-machine nests and cycle time criterion. 2016. http://wojciech.bozejko.staff.iia.pwr.wroc.pl/Benchmarks/F_cycle_nonperm/F_cycle_2machines_benchmarks.zip, Available online [2017-03-13].
- [7] A. GNATOWSKI: Finding properties of an assignment problem in application to cyclic scheduling problems (in Polish). Tech. Rep. PRE No. 5, Faculty of Electronics, Wrocław University of Science and Technology, 2016.
- [8] T. KAMPMEYER: *Cyclic Scheduling Problems*. PhD thesis, University Osnabrück, 2006.

-
- [9] E. LEVNER, V. KATS, D.A.L. DE PABLO and T. CHENG: Complexity of cyclic scheduling problems: A state-of-the-art survey. *Computers & Industrial Engineering*, **59**(2), (2010), 352-361.
- [10] E. NOWICKI and C. SMUTNICKI: A fast taboo search algorithm for the job shop problem. *Management science*, **42**(6), (1996), 797-813.
- [11] T. SAWIK: A mixed integer programming for cyclic scheduling of flexible flow lines. *Bulletin of the Polish Academy of Sciences, Technical Sciences*, **62**(1), (2014), 121-128.

A fine-grained parallel algorithm for the cyclic flexible job shop problem

WOJCIECH BOŻEJKO, JAROSŁAW PEMPERA and MIECZYŚLAW WODECKI

In this paper there is considered a flexible job shop problem of operations scheduling. The new, very fast method of determination of cycle time is presented. In the design of heuristic algorithm there was the neighborhood inspired by the game of golf applied. Lower bound of the criterion function was used in the search of the neighborhood.

Key words: job shop, cyclic scheduling, parallel algorithm.

1. Introduction

Universal globalization, and thus increasing competition forces lowering of the cost of production, which can be achieved through mass production. In practice, production, in which a set of products is manufactured in large quantities, is carried out in a cyclic manner. It is a very effective method because once fixed schedule is repeated over many periods of time. Such method enables delivery of the batch of products, at pre-determined intervals, resulting from the demand of consumers. It provides a systematic replenishment of small inventories and generates a systematic demand for raw materials and components from suppliers. In this way it is much easier to manage the logistics of supply. Moreover, it is relatively easy to detect certain anomalies that may indicate a deterioration of the operating parameters of the production system. New technologies, materials and rapidly changing customers' needs enable manufacturers to frequent modernization of the machinery. As a result of this process companies have machines with different parameters. In this case, production planning requires determining of not only the allocation of tasks to individual machines but also determining of the order of their execution. This leads to a complex, strongly NP-hard optimization problems, in particular to, known in the literature, flexible job shop problem. Due to the large number of decision variables and requirements of the above mentioned management methods it

W. Bożejko (corresponding author), and J. Pempera are with Department of Automatics, Mechatronics and Control Systems, Faculty of Electronics, Wrocław University of Technology, Janiszewskiego 11-17, 50-372 Wrocław, Poland. E-mails: {wojciech.bozejko, jaroslaw.pempera}@pwr.edu.pl. M. Wodecki is with Institute of Computer Science, University of Wrocław, Joliot-Curie 15, 50-383 Wrocław, Poland, e-mail: mieczyslaw.wodecki@uwr.edu.pl

Received 15.12.2016. Revised 28.04.2017.

is necessary to use advanced algorithms to support scheduling at the operational level. Flexible production planning systems enable the implementation of challenges posed by modern management methods such as JIT (just in time) or JIS (just in sequence). In the work there is considered a flexible job shop problem in which a set of tasks to be performed on machines grouped into production cells is given. One should assign tasks to the appropriate machines and determine the order of their execution on each machine so as to optimize some criterion (e.g. all tasks execution time C_{\max}). In most of the works on a flexible job shop problem, as an optimization criterion there is completion date of all tasks execution considered (Yan and Xu [11], Gonzales et al. [6], Pezella et al., [8], Bożejko et al. [2]). Much less papers were devoted to the cyclic version of this problem. In the works of Brucker and Kampmeyer [5] and Kampmeyer [7] there was presented a cyclical job shop problem (i.e. a special case of the considered in this work problem in which each slot contains only one machine). In turn, in the work by Bożejko et al. [4] there were not only certain properties proved but also algorithms for solving the cyclic job shop problem presented. The main problem that exists in the design of efficient algorithms to solve NP-hard cyclic scheduling problems is time-consuming determination of the cycle time. In this paper we present not only some properties of a cyclic job shop problem but also an efficient method of determining the cycle time in which parallel processing was used. They were applied in the algorithm which was described in Bożejko et al. [4] where significant reduction of computation time was obtained, without worsening of the quality of designated solutions.

2. Cyclic job shop problem

In the flexible job shop problem there are given: a set of tasks $\mathcal{J} = \{1, 2, \dots, n\}$ and a set of multi-functional machines $\mathcal{M} = \{1, 2, \dots, m\}$ grouped into production slots. Each machine at any time can execute at most one task. Task $j \in \mathcal{J}$ consists of o_j operation for a set $\mathcal{J}^j = \{l_j + 1, \dots, l_j + o_j\}$ where $l_j = \sum_{i=1}^{j-1} o_i$ is the number of operations of the first $j - 1$ jobs. Operations included in the tasks are performed according to the order of their numbering and form the so-called technological line. By $\mathcal{O} = \{1, 2, \dots, o\}$, where $o = \sum_{i=1}^{j-1} o_i$ we denote the set of all operations. For each operation $v \in \mathcal{O}$ there is defined a subset of machines $\mathcal{M}^v \subset \mathcal{M}$. The operation $v \in \mathcal{O}$ is to be executed on any k machine from the set \mathcal{M}^v in time $p_{v,k} \geq 0$. Execution of operations on the machine cannot be interrupted. In the cyclic production system a set of tasks (hereinafter referred to as *MPS-Minimal Part Set*), is executed repeatedly. In each of the MPS on each machine operations are performed in the same order. The problem consists in the allocation of jobs to machines from the adequate type and the schedule of jobs execution determination on each machine to minimize the cycle time. The following constrains have to be fulfilled:

- (i) each job has to be executed on only one machine of a determined type in each moment of time,

- (ii) machines cannot execute more than one job in each moment of time,
- (iii) there are no idle times (i.e. the job execution must not be broken),
- (iv) the technological line has to be obeyed,
- (v) each operation is performed in sequence after the cycle time is completed.

Constraints (i)-(iv) define known in the literature *Flexible Job Shop* problem (in short denoted by *FJS*). If in addition we assume that each socket contains exactly one machine, then it is a classical in task scheduling theory *Job Shop* problem (abbreviated to *JS*). Descriptions of these problems and metaheuristic algorithms solving them is presented in the works by Nowicki and Smutnicki [9], Barnes and Chambers [1], González et al. [6], Yuan and Xu [11] and Bożejko et al. [3] and [4].

By $\mu = (\mu_1, \dots, \mu_o)$ we denote the assignment of operations to machines where $\mu_a \in \mathcal{M}^a$ is a machine assigned to perform the operation $a \in O$. The set

$$O^l = \{a \in O : \mu_a = l\} \quad (1)$$

operations executed on the machine $l \in \mathcal{M}$, wherein $\cup_{i=1}^m O^l = O$.

Let permutation π_l be a certain sequence of operations from the set O^l on machine l ($|O^l| = n_l$) whereas Φ^l the set of all permutations of elements from O^l . The sequence of operations' execution on the machines is determined by the composition m of permutation $\pi = (\pi_1, \pi_2, \dots, \pi_m)$, where $\pi_i \in \Phi^i$, $i = 1, 2, \dots, m$. Let Φ be the set of all such permutations. Let us see that a permutation $\pi \in \Phi$ unambiguously determines the allocation of operations to machines and the order of operations' execution of individual machines. For the fixed sequence $\pi \in \Phi$ (of solution to *FJS*) problem, let $S^k = (S_1^k, S_2^k, \dots, S_o^k)$ be a sequence of beginning times of operations' execution in the k -th MPS, where S_i^k denotes the commencement date of operation i on machine μ_i in k -th cycle. We assumed that time schedule of the system is cyclic (constraint (v)). This means that there is a constant $T(\pi)$ (the so-called *cycle time*) such that

$$S_{\pi(i)}^{k+1} = S_{\pi(i)}^k + T(\pi), \quad i = 1, \dots, o, \quad k = 1, 2, \dots \quad (2)$$

or equivalently

$$S_{\pi(i)}^{k+1} = S_{\pi(i)}^1 + (k-1)T(\pi), \quad i = 1, \dots, o, \quad k = 1, 2, \dots \quad (3)$$

Equality (2) is an implementation of the constraint (v). Undoubtedly, in a feasible schedule there must be met also restrictions (i-iv) met, which can be written in the form of the following inequities:

$$S_{\pi(i)}^k \geq 0 \quad i = 1, 2, \dots, o, \quad k = 1, 2, \dots \quad (4)$$

$$S_{\pi(i)}^k + p_{\pi(i), \mu(\pi(i))} \leq S_{\pi(i+1)}^k, \quad \pi(i), \pi(i+1) \in \mathcal{J}^j, \quad j \in \mathcal{J}, \quad k = 1, 2, \dots \quad (5)$$

$$S_{\pi_l(j)}^k + p_{\pi_l(j), \mu(\pi_l(j))} \leq S_{\pi_l(j+1)}^k, \quad l \in \mathcal{M}, \quad j = 1, \dots, n_l - 1, \quad k = 1, 2, \dots \quad (6)$$

$$S_{\pi_l(n_l)}^k + p_{\pi_l(n_l), \mu(\pi_l(n_l))} \leq S_{\pi_l(1)}^{k+1}, \quad l \in \mathcal{M}, \quad k = 1, 2, \dots \quad (7)$$

Inequity (5) is a realization of the constraint (iv), (6) of constraint (ii), whereas (7) of constraint (v). For a fixed order of operations execution on machines π , the minimum value of $T(\pi)$, for which there is a feasible schedule that meets (2)–(7), will be called *minimal time cycle* and denoted by $T_{\min}(\pi)$. Since the order of operations for each MPS is the same, it is enough just to determine the beginning moments of execution of operations $S_1^1, S_2^1, \dots, S_o^1$ for the first MPS and make the shift by the size of $T(\pi)$. The considered in this work flexible cyclic job shop problem (in short denoted by *CFJS*) consists in determining a permutation $\pi^* \in \Phi$ with a minimum value of cycle time, i.e. such that

$$T_{\min}(\pi^*) = \min\{T_{\min}(\pi) : \pi \in \Phi\}. \quad (8)$$

3. Graph model

For a fixed sequence $\pi = (\pi_1, \pi_2, \dots, \pi_m)$ ($\pi \in \Phi$), of operations execution in a cyclic flexible job shop problem and the first k -cycle of production ($k = 1, 2, \dots, m + 1$) we define a directed graph $G(\pi, k) = (\mathcal{V}, \mathcal{T} \cup \mathcal{E}(\pi) \cup \mathcal{C}(\pi))$ consisting of a set of vertices \mathcal{V} and three sets of arcs: \mathcal{T} , $\mathcal{E}(\pi)$, $\mathcal{C}(\pi)$. The set \mathcal{V} includes $k \cdot o$ vertices numbered with successive natural numbers. Each operation is assigned to one vertex, wherein the operation $v \in O$ executed in x -th ($x = 1, \dots, k$) MPS corresponds to the vertex $v^x(v) = v + (x - 1)o$ (x -th copy of vertex v) of weight $p_{v, \mu(v)}$. In the further part we will identify vertices of the graph with operations executed within the corresponding MPS. Sets of arcs represent sequence constraints and are defined as follows:

$$(a) \quad \mathcal{T} = \bigcup_{x=1}^k \bigcup_{j=1}^n \bigcup_{v=l_{j-1}+o_{j-1}}^{l_{j-1}+o_{j-1}-1} \{(v^x(v), v^x(v+1))\}, \quad \text{contains arcs representing the technological line (constraint (5)),}$$

$$(b) \quad \mathcal{E}(\pi) = \bigcup_{x=1}^k \bigcup_{l=1}^m \bigcup_{i=1}^{n_l-1} \{(v^x(\pi_l(i)), v^x(\pi_l(i+1)))\}, \quad \text{arcs connecting the operations executed on the same machine (constraint (6)),}$$

$$(c) \quad \mathcal{C}(\pi) = \bigcup_{x=1}^{k-1} \bigcup_{l=1}^m \{(v^x(\pi_l(n_l)), v^{x+1}(\pi_l(1)))\}, \quad \text{arcs connecting the operations executed on the same machine between the MPS (constraint (7)).}$$

Theorem 1 *If $\pi \in \Phi$ is the order of operations' execution on the machines in CFJS problem, then the graph $G(\pi, k)$, $k = 1, 2, \dots, m + 1$ does not contain cycles.*

Proof Let permutation $\pi \in \Phi$ be the solution to *CFJS* problem. The first MPS is represented by a graph $G(\pi, 1)$. It is easy to see that this is also a graph solving the *FJS* problem, whose acyclicity is easy to prove. For a fixed k ($k = 2, 3, \dots, m + 1$) we consider the graph $G(\pi, k)$. It follows from definition of a set of vertices and arcs (a), (b) that the graph is k -fold copy of the graph (components) $G(\pi, 1)$. It also includes arcs between certain vertices belonging to the subsequent neighboring components (a set of $C(\pi)$). Because the components are acyclic graphs and it is not possible to return to the previous components (the definition of a set of arcs (c)), so the graph $G(\pi, k)$ does not contain cycles. \square

We consider the longest path in graph $G(\pi, m + 1)$ from a vertex $v \in O$ in the first MPS, to the same vertex in x -th MPS, i.e. vertex v^x . By L_v^x we denote the length of this path (the length does not include the weight of vertex v^x). If S_v^1 is the moment of beginning of execution of operation v in the first MPS, and S_v^x the moment of its commencement (i.e. operation v^x) in x -th MPS, then:

$$S_v^x \geq S_v^1 + L_v^x. \quad (9)$$

This inequity is a direct result of the constraints (i)-(v). Before the beginning of operation v^x all the operations lying on any path (including the longest) between v and v^x must be executed.

Let

$$\Lambda^*(\pi) = \max_{v \in O} \max_{x=2, \dots, m+1} \{\lambda_{v,x}\}, \quad (10)$$

where

$$\lambda_{v,x} = L_v^x / (x - 1), \quad v \in O, \quad x = 2, 3, \dots, m + 1. \quad (11)$$

Below we will prove two theorems showing the relationship between the minimum cycle time $T_{\min}(\pi)$, and the value $\Lambda^*(\pi)$.

Theorem 2 *If $\pi \in \Phi$ is allowable execution order of operations in a cyclic flexible job shop problem, then the minimum cycle time $T_{\min}(\pi) \leq \Lambda^*(\pi)$.*

Proof Let

$$T'(\pi) = \Lambda^*(\pi) = \max_{v \in O} \max_{x=2, \dots, m+1} \{\lambda_{v,x}\} = \lambda_{a,t} = L_a^t / (t - 1).$$

We will show that the so defined $T'(\pi)$ is the cycle time for solution π . Let $(S_{\pi(1)}^1, S_{\pi(2)}^2, \dots, S_{\pi(o)}^o)$ be a sequence of the commencement moment of operations of the first MPS. We will show that

$$S_{\pi(i)}^{k+1} = S_{\pi(i)}^k + (k - 1)T'(\pi), \quad i = 1, 2, \dots, o, \quad k = 2, 3, \dots, m + 1,$$

are the beginning moments of separate operations in subsequent MPS, i.e. they meet the constraints (3). By definition (10)

$$S_{\pi(i)}^{k+1} = S_{\pi(i)}^1 + (k - 1)T'(\pi) = S_{\pi(i)}^1 + (k - 1)\Lambda^*(\pi) = S_{\pi(i)}^1 + (k - 1)(L_a^t / (t - 1)) \geq S_{\pi(i)}^1 + (k - 1)(L_{\pi(i)}^k / (k - 1)) = S_{\pi(i)}^1 + L_{\pi(i)}^k.$$

The last inequity follows from the fact that $\lambda_{a,t} = L_a^t / (t - 1)$ is a maximum element, thus $\lambda_{a,t} \geq \lambda_{v,j}$ ($v = 1, 2, \dots, o, j = 2, 3, \dots, m + 1$). We have shown this way, that $T'(\pi)$ is cycle time (i.e. it satisfies the inequity (3)), which completes the proof of the theorem. □

Theorem 3 *If $\pi \in \Phi$ is allowable execution order of operations in a cyclic flexible job shop problem, then the minimum cycle time $T_{\min}(\pi) \geq \Lambda^*(\pi)$.*

Proof For the solution $T_{\min}(\pi)$, let $\pi \in \Phi$, be minimum cycle time, whereas $(S_{\pi(1)}^k, S_{\pi(2)}^k, \dots, S_{\pi(o)}^k)$ a sequence of beginning moments of operation in k -th MPS. According to (3)

$$S_{\pi(i)}^k = S_{\pi(i)}^1 + (k - 1) \cdot T_{\min}(\pi), \tag{12}$$

for $i = 1, 2, \dots, o, k = 2, 3, \dots, m + 1$.

Let $\pi(l)$ be any operation from the set O . Rusing from (9) and (12) we obtain

$$S_{\pi(i)}^1 + (k - 1) \cdot T_{\min}(\pi) \geq S_{\pi(i)}^1 + L_{\pi(l)}^k,$$

hence

$$T_{\min}(\pi) \geq L_{\pi(l)}^k / (k - 1) = \lambda_{k,\pi(l)}.$$

Since this inequality is valid for every $l = 1, 2, \dots, o$ and $k = 2, 3, \dots, m + 1$, then

$$T_{\min}(\pi) \geq \max\{\lambda_{k,\pi(l)} : k = 1, 2, \dots, m, k = 1, 2, \dots, o\} = \Lambda^*(\pi),$$

which completes the proof of the theorem. □

Designation of the minimum value of the cycle time $T_{\min}(\pi)$ (i.e. value $\Lambda^*(\pi)$) requires the calculation of $m \cdot o$ values of the coefficients $\lambda_{i,j}$. In the work of Bożejko et al. [4] there was a theorem proven enabling much faster calculation of minimum cycle time.

Let

$$\mathcal{A} = \{v : v = \pi_l(1), l \in \mathcal{M}\}$$

be the set of all operations executed as first on the individual machines in the first MPS.

Theorem 4 *For any solution $\pi \in \Phi$ minimum cycle time*

$$T_{\min}(\pi) = \Lambda^*(\pi) = \max_{v \in \mathcal{A}} \max_{x=2,\dots,m+1} \{\lambda_{v,x}\}.$$

Proof See theorem 2 and 3.

Using this theorem the number of determined coefficients $\{\lambda_{v,x}\}$ can be reduced from $m \cdot o$ to $m \cdot m$, where o is the number of operations and m the number of machines.

Determination of value $\lambda_{v,x}$ dla $x = 2, \dots, m + 1$, $v \in \mathcal{A}$ requires construction of a graph $G(\pi, m + 1)$ consisting of $(m + 1)o$ vertices and the same order of arcs. In turn, to calculate the value $\{\lambda_{v,x}\}$, for a given $v \in \mathcal{A}$, one must designate the length of the longest paths from v to other vertices which requires $O(mo)$ time. Ultimately, we get the computational complexity designation $\Lambda^*(\pi)$, $|\mathcal{A}|O(mo) = O(om^2)$.

A path in a graph $G(\pi, m + 1)$, whose length $L_v^x/(x - 1) = \Lambda^*(\pi)$ will be called a *critical path*. In turn, the maximum subsequence of vertices of the path representing the operations executed one after another on the same machine will be called a *block*. In case of the considered in the work *CJFS* tasks problem, one can use the so-called a *blocks eliminating properties*'. The theory was successfully used in the construction of the best optimization algorithms for a wide class of scheduling problems with the criterion C_{\max} , e.g. by Nowicki and Smutnicki [9] or Bożejko et al. [3].

Theorem 5 *If the solution $\beta \in \Phi$ was generated from $\pi \in \Phi$ and $T(\beta) < T(\pi)$ at least one operation of at least one block of tasks is executed*

- (a) *before the first operation of this block, or*
- (b) *after the last operation of this block, or*
- (c) *on another machine.*

This theorem will be used when generating elements of the neighborhood in the tabu search algorithm to solve the considered in the work problem.

4. Effective determination of cycle time

Currently the best optimization algorithms for a wide class of scheduling problems are based on iterative methods of local search solution space. Quality of solutions determined by these algorithms depends on the number of directly considered solutions, which with the limited time of the algorithm action depends on the computational complexity of the procedure for calculating the value of the criterion function. Described in the previous chapter method for determining the cycle time has a computational complexity of $O(om^2)$ and is $O(m^2)$ times bigger than the time of determination of the value C_{\max} . Acceleration of calculations determining cycle time is possible by use of a parallel processing. For this purpose, the method of parallel vector processing will be used.

We consider a graph $G(\pi, 1)$, $\pi \in \Phi$ corresponding to the first MPS. For any vertex (operation) $v \in O$, by $\tau(v)$ i $\eta(v)$ we successively denote two successors: technological and sequential (in permutation π executed on the same machine). If the operation v has not a corresponding successor, then after $\tau(v)$ or $\eta(v)$ we assume zero. Since the graph

$G(\pi, 1)$ is directed acyclic and weakly consistent (i.e. for each pair of distinct vertices x, y there exists the path from x to y or from y to x), so its vertices can be sorted topologically. We can therefore number the vertices in such a way that the beginning of a given arc has a smaller number than its end. In particular the successors $\tau(v)$ or $\eta(v)$ of vertex v have greater numbers than v . Sorting Algorithm topologically sorting vertices of the graph $G(\pi, 1)$ has a complexity $O(o)$.

If $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(o))$ is the topological order of the vertices of the graph $G(\pi, 1)$, then it is easy to extend it to any of the graphs $G(\pi, k)$, $k = 2, 3, \dots, m + 1$. In such a case vertex v^i (from i -th MPS) is given the number of $\sigma(v) + (i - 1)o$.

Procedure SeqTC

π - feasible solution;

σ - topological ordering of the vertices of the graph $G(\pi, 1)$;

1. For $k = 1, \dots, m$ do
2. Set $L_i^x = -\infty$ for $i \in O$, $x = 1, \dots, m + 1$.
3. Set $L_{\pi_k(1)}^1 = 0$.
4. For $x = 1, \dots, m + 1$ do
5. For $v = \sigma(1), \dots, \sigma(o)$ do
6. Set $L_v^x = L_v^x + p_v$, $L_{\tau(v)}^x = L_v^x$ **and** $L_{\eta(v)}^x = L_v^x$.
7. For $l = 1, \dots, m$ do $L_{\pi_l(1)}^{x+1} = L_{\pi_l(\eta_l)}^x$
8. For $k = 1, \dots, m$ **and** $x = 1, \dots, m + 1$ do
9. Set $\lambda_{k,x} = L_{\pi_k(1)}^{x+1} - L_{\pi_k(1)}^x$.

Figure 1: Sequential cycle time computing procedure.

Let $\pi \in \Phi$ be the order of operations' execution on the machines in *CFJS*, problem, whereas $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(o))$ topological order of vertices in the graph $G(\pi, 1)$. Figure 1 depicts **SeqTC** procedure of the sequential determining the length of the longest paths L_v^x in the graph $G(\pi, m + 1)$ used in the computation of coefficients $\lambda_{v,x}$ (formula(11)). On this basis we determine the value Λ^* (10), i.e. minimum cycle time $T_{\min}(\pi)$. In the description of the procedure the sources are vertices of the graph $G(\pi, 1)$, who are not the end of any arc (being the first operations executed on machines). The length of paths $L_v^{(x)}$, $x = 1, \dots, m + 1$, $v \in O$ are to be interpreted in two ways. Until step 6 they are lower estimate of the length of the longest path coming out of the vertex being the source to the vertex representing an operation v in x -th MPS (without the weight of the vertex). The source is determined in Step 3. Then, in step 6 the exact value of the length of the longest path to vertex v (with the weight of that vertex) are computed and lower estimate of the length of the longest paths to vertices being successors v , i.e. $\tau(v)$ and $\eta(v)$. are updated. Finally, in step 7, the lower estimate of the length of the longest paths to vertices representing the operations performed on the first machine, in the next MPS are updated.

Based on the analysis of the code it is easy to see that the computational complexity of the procedures for the designation of sequential cycle time **SeqTC** is $O(om^2)$. This

time can be significantly reduced by using techniques of parallel search based on the vector processing. Then, in a vector processor cycle there are performed logical, arithmetic operations or data movements, etc. on one or two multi-element vectors. Vector operations are implemented in hardware in all modern processors, both desktops and laptops, and above all, in programmable graphics cards.

Procedure ParTC

π - feasible solution;

σ - topological ordering of the vertices of the graph $G(\pi, 1)$;

1. **Set** $\vec{L}_v^x = -\infty$ **for** $v \in O, x = 1, \dots, m + 1$.
2. **For** $k = 1, \dots, m$ **do**
3. **Set** $L_{\pi_k(1)}^1(k) = 0$.
4. **For** $x = 1, \dots, m + 1$ **do**
5. **For** $v = \sigma(1), \dots, \sigma(o)$ **do**
6. **Set** $\vec{L}_v^x = \vec{L}_v^x + \vec{p}_v, \vec{L}_{\tau(v)}^x = \vec{L}_v^x$ **and** $\vec{L}_{\eta(v)}^x = \vec{L}_v^x$.
7. **For** $l = 1, \dots, m$ **do** $\vec{L}_{\pi_l(1)}^{x+1} = \vec{L}_{\pi_l(n_l)}^x$.
8. **For** $k = 1, \dots, m$ **and** $x = 1, \dots, m + 1$ **do**
9. **Set** $\lambda_{k,x} = \vec{L}_{\pi_k(1)}^{x+1}(k) - \vec{L}_{\pi_k(1)}^x(k)$.

Figure 2: Parallel cycle time computation.

In Figure 2 there is shown a diagram of **ParTC** procedure effectively determining the cycle time using a vector parallel processing. Designations are the same as in the sequential procedure. In the vector

$$\vec{L}_i^{(x)} = (\vec{L}_i^x(1), \vec{L}_i^x(2), \dots, \vec{L}_i^x(m)) \quad (13)$$

there is remembered the length of the longest path reaching to the vertex representing operation i in x -th MPS. For various elements of the vector the values differ from one another due to a different source vertex assigned to each vector element. The most time-consuming are iterative instructions in row 4 and 5. Assuming that operations on vectors are executed in one tact we get superior computational complexity $O(om)$, in case of CPU processors transforming m - element vectors.

5. Neighborhood viewing

In the algorithms of local search the adjacent solutions (neighborhood) can be viewed in two ways, ie. by generating: (i) all neighboring solutions (ii) subset containing only some solutions. Undoubtedly, the first method is much more time-consuming. However, it usually enables determination of good solutions with fewer iterations of the whole

algorithm. Regardless of the method of the neighborhood viewing, for any solution there should be the value of the objective function determined. In the algorithms viewing the whole neighborhood, for many optimization problems, it is possible to construct the accelerator. This ensures, with the use of partial results, a considerable reduction of the computation time of goal function value for all solutions of the neighborhood. An effective example of the accelerator use was described, among others, by Nowicki and Smutnicki [9]. Below, we present a new two-phase neighborhood search method. In the first phase, for each solution in the neighborhood, there is a lower bound of the value of the objective function determined. At the beginning of the second phase there is created a list of solutions ordered non-decreasingly in reference to lower bound (determined in the first phase). Then, for solutions in the sequence they appear in the ordered list, there is the exact value of the objective function calculated. The computation process is terminated, as soon as the solution whose exact value of the objective function is determined, not greater than the lower bound of the remaining on the solutions list. It is worth noting that the better the lower bound of the objective function value, the less solutions will be verified by calculating the exact value.

For the considered in this paper cyclic flexible job shop problem lower bound can be determined by considering only the first MPS (i.e. graph $G(\pi, 1)$), wherein as the lower bound we assume:

$$LB(\pi) = \max_{v \in \mathcal{A}} \{\lambda_{v,1}\}. \quad (14)$$

The computational complexity of determining the value $LB(\pi)$ is in sequential version $O(om)$, whereas in parallel $O(o)$.

6. Computational experiments

In order to evaluate the acceleration of computations relating to the proposed neighborhood viewing method and the use of vector processing there were computational experiments carried out. The results of golf *AGF* algorithm presented in the work of Bożejko et al. [4] and its two of modifications A_S and A_V were compared. In both algorithms there was a two-phase search of neighborhood applied. In A_S algorithm the search was executed sequentially (procedure **SeqTC**), whereas in A_V algorithm in parallel (procedure **ParTC**). Algorithms were programmed in C++ in Visual Studio 2010. The computations were performed on a PC with an Intel I7-core 2.4GHz on a single core of the processor. Parallel processing was carried out on 128-bit registers using SSE2 commands. Each of them was an 8-element vector consisting of 16-bit representations of the data processed in parallel. Vector processing based on SSE instructions was used, among many others, in the work of Smutnicki et al. [10]. Comparative studies of algorithms were carried out on the instances presented in the works of Barnes and Chambers [1]. For all three algorithms the adopted number of iterations (stop condition) equaled 10 000, whereas the length of tabu list was 15.

Table 1: The operating times and acceleration of algorithms.

Instance	$n \times m$	o	$t(AGF)$	$t(A_S)$	$\frac{t(AGF)}{t(A_S)}$	$t(A_V)$	$\frac{t(A_S)}{t(A_V)}$	$\frac{t(AGF)}{t(A_V)}$
setb4c9	15×11	150	19.59	4.91	4.0	1.49	3.3	13.1
setb4cc	15×12	150	30.22	6.63	4.6	1.97	3.4	15.3
setb4x	15×11	150	18.56	4.92	3.8	1.48	3.3	12.5
setb4xx	15×12	150	23.45	5.78	4.1	1.64	3.5	14.3
setb4xxx	15×13	150	69.33	6.19	11.2	1.73	3.6	40.1
setb4xy	15×12	150	7.47	5.38	1.4	1.63	3.3	4.6
setb4xyz	15×13	150	4.06	0.75	5.4	0.23	3.3	17.7
seti5c12	15×16	225	67.6	12.67	5.3	3.52	3.6	19.2
seti5cc	15×17	225	98.2	15.86	6.2	4.41	3.6	22.3
seti5x	15×16	225	62.86	12.09	5.2	3.38	3.6	18.6
seti5xx	15×17	225	77.13	14.09	5.5	3.81	3.7	20.2
seti5xxx	15×18	225	107.27	17.03	6.3	4.52	3.8	23.7
seti5xy	15×17	225	98.1	15.75	6.2	4.36	3.6	22.5
seti5xyz	15×18	225	121.14	17.97	6.7	4.81	3.7	25.2

In Table 1 there were the results on the time of computations of algorithms $t(A)$, $A \in \{AGF, A_S, A_V\}$. presented. The first column shows the name of example, in the subsequent ones: the number of tasks (n), the number of machines (m) and the number of operations (o). The next three columns include the running times of AGF and A_S . algorithms. In turn, in the last three columns there were the operating times of sequential algorithms AGF and A_S . compared with the running time of the parallel algorithm A_V .

The results of experimental studies clearly indicate that the use of the two-phase method of viewing the neighborhood significantly reduces computation times. For the sequential version of the algorithm the running time is shorter - from 4.0 to 11.2 times (the quotient of $\frac{t(AGF)}{t(A_S)}$). On the other hand, the use of parallel processing realized by vector computing enables additional reduction of time from 3.3 to 3.8 times. Ultimately, the simultaneous use of both methods of computation acceleration allows its users for additional time reduction from 4.6 to 40.1 times.

The quality of solutions generated by AGF algorithm was presented in Table 2 (the other two algorithms A_S and A_V determined the same solutions but in a much shorter time). The cycle time of solutions generated by AGF algorithm were compared with the best known values of the minimum execution time of all tasks C_{max} . It should be noted that C_{max} is the lower bound on the length of cycle time for one of the cyclic models considered in the work of Brucker and Kampmeyer [5]. For each example, based on a minimum time of tasks execution of the first MPS C_{max} and the length of the cycle time

Table 2: The values set by the algorithms of solutions.

Instance	$n \times m$	o	C_{\max}	T^*	PRD
setb4c9	15×11	150	914	903	1.20
setb4cc	15×12	150	907	887.67	2.13
setb4x	15×11	150	925	878	5.08
setb4xx	15×12	150	925	879	4.97
setb4xxx	15×13	150	925	1002	-8.32
setb4xy	15×12	150	910	845	7.14
setb4xyz	15×13	150	903	838	7.20
seti5c12	15×16	225	1174	1130	3.75
seti5cc	15×17	225	1136	1064.5	6.29
seti5x	15×16	225	1198	1141	4.76
seti5xx	15×17	225	1197	1100	8.10
seti5xxx	15×18	225	1197	1136.5	5.05
seti5xy	15×17	225	1136	1064.5	6.29
seti5xyz	15×18	225	1125	1052	6.49

T there was a relative percentage improvement determined

$$PRD = \frac{C_{\max} - T}{C_{\max}} 100\%. \quad (15)$$

For 13 instances the cycle times determined by *AGF* algorithm were significantly lower than the value C_{\max} . This improvement ranged from 1.2 to 8.1 %. In one case (example setb4xxx) determined by *AGF* algorithm, cycle time length was about 8.32% worse than the value C_{\max} .

Given the fact that the best solutions were determined after a small number of iterations, it is possible to state that this algorithm can be successfully used to solve practical examples of large sizes.

7. Summary

In the work there was a cyclical flexible job shop problem considered. A graph model, for a fixed order of operations execution on individual machines was presented. The theorems enabling efficient determination of the minimum cycle time and a lower bound were proven. In order to speed up the calculations there was not only a two-phase method of neighborhood searching proposed but also a parallel method of determining the cycle time for the established order of operations that used vector processing

proposed. Ultimately, the results of computational experiments, which confirmed a significant acceleration of calculations, while maintaining the designated solutions were presented. To sum up, on the basis of the obtained results it can be concluded that the modified *AMF* algorithm designated in a short time fully accepted in practice solutions.

References

- [1] J.W. BARNES and J.B. CHAMBERS: Flexible job shop scheduling by tabu search, Graduate program in operations research and industrial engineering. The University of Texas at Austin, Technical Report Series: ORP96-09, 1996.
- [2] W. BOŻEJKO, M. UCHROŃSKI and M. WODECKI: Parallel hybrid metaheuristics for the flexible job shop problem. *Computers & Industrial Engineering*, **59**(2), (2010), 323-333.
- [3] W. BOŻEJKO, M. UCHROŃSKI and M. WODECKI: Block approach to the cyclic flow shop scheduling. *Computers & Industrial Engineering*, **81** (2015), 158-166.
- [4] W. BOŻEJKO, J. PEMPERA and M. WODECKI: The golf algorithm for the cyclic flexible job shop problem, (to appear).
- [5] P. BRUCKER and T. KAMPMEYER: Cyclic job shop scheduling problems with blocking. *Annals of Operations Research*, **159** (2008), 161-181.
- [6] M.A. GONZÁLEZ, C.R. VELA and R. VARELA: Scatter search with path relinking for the flexible job shop scheduling problem. *European J. of Operational Research*, **245**(1), (2015), 35-45.
- [7] T. KAMPMEYER: Cyclic Scheduling Problems. Ph.D. Thesis, University Osnabrick, 2006.
- [8] F. PEZZELLA, G. MORGANTI and G. CIASCETTI: A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research*, **35** (2008), 3202-3212.
- [9] E. NOWICKI and C. SMUTNICKI: A fast tabu search algorithm for the job shop problem. *Management Science*, **42**, (1996), 797-813.
- [10] C. SMUTNICKI, J. PEMPERA, J. RUDY and D. ZELAZNY: A new approach for multi-criteria scheduling. *Computers & Industrial Engineering*, **90** (2015), 212-220.
- [11] Y. YUAN and H. XU: Flexible job shop scheduling using hybrid differential evolution algorithms. *Computers & Industrial Engineering*, **65**(2), (2013), 246-260.

Parallel patterns determination in solving cyclic flow shop problem with setups

WOJCIECH BOŻEJKO, ZENON CHACZKO, MARIUSZ UCHROŃSKI and MIECZYŚLAW WODECKI

The subject of this work is the new idea of blocks for the cyclic flow shop problem with setup times, using multiple patterns with different sizes determined for each machine constituting optimal schedule of cities for the traveling salesman problem (TSP). We propose to take advantage of the Intel Xeon Phi parallel computing environment during so-called 'blocks' determination basing on patterns, in effect significantly improving the quality of obtained results.

Key words: cyclic scheduling, parallel algorithm, metaheuristics

1. Introduction

In recent years there has been an observed growth in interest in cyclic problems of tasks scheduling, both in the circle of theorists dealing with discrete optimization problems and in the population of industry practitioners. Cyclic production is, in fact, a very effective mode of production in modern flexible manufacturing systems. In the available literature there are many studies on different aspects of cyclic control in companies that produce goods on a mass scale. There are examples of the application of cyclic scheduling in various spheres of industry, transport and logistics (e.g. Pinto et al. [12], Pinedo [11], Mendez et al. [9], Gertsbakh and Serafini [6], Kats and Levner [8]). Unfortunately, the existing models and calculation tools allow one to determine the optimal (minimizing cycle time) control of production systems executing only a small number of tasks.

In this work we considered a cyclic flow shop problem with setup times. Strong NP-hardness of many simplest versions of the cyclic scheduling problem (Smutnicki [14]), in particular of the considered problem, limits the scope of application of exact algorithms

W. Bożejko (corresponding author), and M. Uchroński are with Department of Automatics, Mechatronics and Control Systems, Faculty of Electronics, Wrocław University of Technology, Janiszewskiego 11-17, 50-372 Wrocław, Poland. E-mails: {wojciech.bozejko, mariusz.uchronski}@pwr.edu.pl. Z. Chaczko is with Faculty of Engineering and Information Technology (FEIT), University of Technology, Sydney(UTS), Australia, NSW, Sydney, e-mail: zenon.chaczko@uts.edu.au and M. Wodecki is with Institute of Computer Science, University of Wrocław, Joliot-Curie 15, 50-383 Wrocław, Poland, e-mail: mieczyslaw.wodecki@uw.edu.pl

Received 14.12.2016. Revised 30.04.2017.

in instances with low number of tasks, but in the context of minimizing the design cycle time and the use of exact algorithms, it seems to be fully justified (Brucker et al. [5]). However, due to the NP-hardness, to determine satisfactory solutions there are common fast approximate algorithms used, based on local search techniques, such as: simulated annealing (in parallel version Bożejko et al. [2]) or tabu search (Bożejko et al. [3]). Methods of this type are usually based on a two-level decomposition of the problem: determination of the optimum order of tasks (upper level) and multiple determination of the minimum criteria for a given sequence of tasks (lower level).

While for classic, non-cyclic scheduling problems, a solution to the lower level problem can be obtained in a time-efficient manner by analyzing a specific graph, in case of such a posed problem the solution to lower level is relatively time-consuming because it generally requires solving offlinear programming problem. Therefore, any special properties, including these allowing for more efficient calculation of the cycle time, the search of schedule and limiting the cardinality of locally searched neighborhood or acceleration of the speed of the viewing are very desirable.

In the presented paper, we are proposing the use of new eliminating properties, including the so-called patterns to reduce the number of solutions viewed when generating neighborhood with local search algorithms, such as the tabu search with prohibitions or simulated annealing. Determination of the patterns may be performed either sequentially, or in parallel, using a multiprocessing computations environment. Relevant properties were formulated using the PRAM model machine which is standard for the theoretical verification of the computational complexity of parallel algorithms.

The reminder of hte paper is organized as follows. Section 2 presents a problem description. Its mathematical model is proposed in the section 3. Section 4 contains proposed solution methid in which we define adn use blocks of tasks. Results of computational experiments are discussed in section 5. Conclusions and comments are presented in the Section 6.

2. Problem description

The problem of cyclic production considered in this work may be formulated as follows: there is a set of n tasks given $\mathcal{J} = \{1, 2, \dots, n\}$, to be performed cyclically (repeatedly) on machines from the set $\mathcal{M} = \{1, 2, \dots, m\}$. Any task should be performed in sequence on each of m machines $1, 2, \dots, m$ (technological line). Task $j \in \mathcal{J}$ is a sequence m operations $O_{1,j}, O_{2,j}, \dots, O_{m,j}$. Operation $O_{k,j}$ corresponds to the activity of execution of task j on machine k , in time $p_{k,j}$ ($k = 1, 2, \dots, m, j = 1, 2, \dots, n$). After the completion, and before the start of the next operation a machine setup is performed. Let $s_{i,j}^k$ ($k \in \mathcal{M}, i \neq j, i, j \in \mathcal{J}$) be the setup time between operation $O_{k,i}$ and $O_{k,j}$.

A set of tasks in a single cycle is called a MPS (*minimal part set*). MPSs are processed cyclically, one by one.

One should determine the order of tasks (the same for each machine), which minimizes cycle time, i.e. the moment of commencement of tasks execution from the set \mathcal{J} in the next cycle. In applying this, the following constraints must be met:

- (a) each operation can only be performed by one machine,
- (b) no machine can perform more than one operation at the same time,
- (c) the order of the technological line of operations execution must be preserved,
- (d) execution of any operation cannot be interrupted before its completion,
- (e) each machine, between sequentially performed operations, requires setups,
- (f) each operation is executed sequentially (in successive MPSs) after the cycle time is completed.

The considered problem boils down to determining the starting moments of the tasks execution on machines that meet the constraints (a) - (f), so that the cycle time (the time after which the task is performed in a subsequent MPS) is minimised.

We assume that in each of the MPS, on each machine, the tasks are executed in the same order. Therefore, in a cyclic schedule, the order of tasks execution on machines can be represented by a permutation of the tasks in the first MPS. On this basis we can determine the beginning moments of tasks execution on machines in the first MPS. Increasing them by multiples of the cycle time, we obtain the beginning moments of tasks execution in any of MPSs (commencement of execution of any of the operation in the consecutive MPS should be increased by the cycle time). Let Φ be the set of all permutations of the elements from a set of tasks \mathcal{J} . Therefore, the considered in the work problem comes down to the determination of permutation of tasks (element from the set Φ) minimizing the length of cycle time. In short the problem will be denoted by CFS (*Cyclic Flow Shop*).

3. Mathematical model

Let $[S^k]_{m \times n}$ be a matrix of beginning moments of tasks execution of k -th MPS (for the established order $\pi \in \Phi$), where $S_{i,j}^k$ denotes the beginning time moment of execution of task j on i machine. We assume that tasks in the following MPSs are performed cyclically. This means that there is a fixed $T(\pi)$ (period) such that

$$S_{i,\pi(j)}^{k+1} = S_{i,\pi(j)}^k + T(\pi), \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad k = 1, 2, \dots \quad (1)$$

Period $T(\pi)$ depends of course on permutation π and is called *cycle time* of the system. The minimum value $T(\pi)$, for a fixed π , will be called *minimum cycle time* and denoted by $T^*(\pi)$. Since the order of tasks execution in any MPS is the same, it is enough

to designate a sequence of tasks π for one (*the first*) MPS and make its shift by $k \cdot T(\pi)$, $k=1,2,\dots$ on timeline. For a fixed order of execution of tasks $\pi \in \Phi$, optimum value of cycle time $T^*(\pi)$ can be determined by solving the corresponding linear programming task (see Bożejko et al. [1]). For any order of tasks execution in the first MPS, solving the above linear programming task, there can be the minimum time cycle in polynomial time determined. In case of an exact algorithm's (exhaustive search) solution to CFS problem should be therefore done for each of $n!$ permutations – elements of the set Φ . For real-life instances sizes it will be impossible due to computations time, so in the next chapter we propose an approximate method for solving the CFS problem .

4. Solution method

In many heuristic algorithms solving NP-hard problems, there are neighborhoods viewed, i.e. subsets of solution space. In case where solutions to the problem are permutations, usually the neighborhoods are generated by insert- or swap-type moves and their compositions [4]. They consist of changing positions of elements in the permutation. The number of elements of such neighborhood is at least $n(n-1)/2$, where n is the number of tasks. In practical applications (with large n), neighborhood viewing is the most time consuming part of the algorithm. It follows from the descriptions in the literature concerning computational experiments that the number of iterations of the algorithm has a direct impact on the quality of designated solutions. Hence, there is the search of methods accelerating action of a single iteration of the algorithm. One of them relies in the reduction of the number of elements of the neighborhood, their parallel generation and viewing. In the case of tasks scheduling problems on multiple machines with the minimization of tasks execution time (C_{max}) 'block eliminating properties' are successfully used [7]. Similar properties are to be applied in the algorithm solving the problem of determining minimum cycle time, more specifically - a minimum time of running of a single machine. They allow its users to eliminate elements from the neighborhood that do not directly provide an improvement on the best solution found so far.

The work [3] describes a method solving the CFS problem and a sequential algorithm of search with prohibitions. In a further part of the chapter there is a summary of the main elements of the method presented.

For a set permutation $\pi \in \Phi$ and machine $k \in \mathcal{M}$

$$T_k(\pi) = \sum_{i=1}^{n-1} (p_{k,\pi(i)} + s_{\pi(i),\pi(i+1)}^k) + p_{k,\pi(n)} + s_{\pi(n),\pi(1)}^k \quad (2)$$

is the time of execution of tasks in the order π , with setups performed between task $\pi(n)$, and $\pi(1)$ (i.e. the last task in the given MPS, and the first in the next). It is easy to prove that the minimum cycle time

$$T^*(\pi) = \max\{T_i(\pi) : i = 1, 2, \dots, m\}. \quad (3)$$

Property 1 ([3]). *The necessary condition reducing the value of the minimum cycle time $T^*(\beta)$ is shortening of the running time of k -th machine, i.e. reducing of $T_k(\beta)$, where $T^*(\beta) = T_k(\beta)$.*

Designation of the minimum running time of k -th machine, i.e. the value $\min\{T_k(\delta) : \delta \in \Phi\}$ can be reduced to following traveling salesman problem.

Let $H_k = (\mathcal{V}, \mathcal{E}; p, s)$ be a complete graph, where

- a set of vertices: $\mathcal{V} = \mathcal{J}$,
- a set of edges: $\mathcal{E} = \{(v, u) : v \neq u, v, u \in \mathcal{V}\}$,
- weights of vertices: $p(v) = p_{k,v}$, $v \in \mathcal{V}$,
- weight of edges: $s(e) = s_{i,j}^k$, $e = (i, j) \in \mathcal{E}$.

Property 2 ([3]) *The running time of k -th machine $T_k(\pi)$ is equal to the length (i.e. sum of the weights of vertices and edges) of Hamiltonian cycle $(\pi(1), \pi(2), \dots, \pi(n), \pi(1))$ in graph H_k .*

Property 3 ([3]) *Minimum running time of k -th machine is equal to the length of the salesman path in the graph H_k , i.e. the minimum (due to the length) Hamiltonian cycle.*

Let π_k^* be the optimal salesman path way in a graph H_k ($k = 1, 2, \dots, m$), which is also an optimal (i.e. minimal due to the execution time) order of the tasks from the set \mathcal{J} on k -th machine. This permutation will be called a *pattern* for k -th machine.

In order to reduce the running time of k -th machine $T_k(\pi)$ from π there will be permutations generated, taking into account the existence of the individual elements in the pattern. Patterns also allow to eliminate the elements of the neighborhood which do not give an improvement of the current value of cycle time in local search algorithms.

4.1. Tasks blocks

Let

$$B = (\pi(a), \pi(a+1), \dots, \pi(b)), \quad (4)$$

be a sequence of occurring immediately after another tasks in permutation $\pi \in \Phi$, π_k^* pattern for k -th machine and u, v ($u \neq v$, $1 \leq u, v \leq n$) a pair of numbers such that:

W1: $\pi(a) = \pi^*(u), \pi(a+1) = \pi^*(u+1), \dots, \pi(b-1) = \pi^*(v-1),$
 $\pi(b) = \pi^*(v)$, or

W2: $\pi(b) = \pi^*(u), \pi(b-1) = \pi^*(u+1), \dots, \pi(a+1) = \pi^*(v-1),$
 $\pi(a) = \pi^*(v)$

W3: B is the maximum subsequence due to the inclusion, i.e. it can be enlarged neither by an element $\pi(a-1)$, nor by $\pi(b+1)$, satisfying the constraints **W1** or **W2**).

If the sequence of tasks (4) satisfies the conditions **W1** and **W3** or **W2** and **W3**, then it is called a *block* on k -th machine ($k \in \mathcal{M}$).

Below there is presented a sequential algorithm for determining all of the blocks in the permutation.

Algorithm 1. *Alg_SeqBlock*

$\pi = (\pi(1), \pi(1), \dots, \pi(n))$ - permutation;
 $\pi^* = (\pi^*(1), \pi^*(1), \dots, \pi^*(n))$ - pattern of permutation π ;
 t - number of blocks;
 (b_1, b_2, \dots, b_t) - vector of blocks starting positions in π ;
 $t \leftarrow 1; i \leftarrow 1$;
while ($i \leq n$) **do**
 $b_t \leftarrow i; q \leftarrow (\pi^*)^{-1}(\pi(i))$;
 while ($\pi(i) = \pi^*(q)$) **do**
 $q \leftarrow q + 1; i \leftarrow i + 1$;
 $i \leftarrow i + 1$;

The computational complexity of the algorithm is $O(n)$.

Determination of the pattern (the optimal salesman path in the graph H_k) is an NP-hard problem. Therefore, there will be approximate algorithms used, e.g. 2-opt. For each machine the pattern will be determined before starting the proper algorithm.

4.2. Parallel determination of blocks

To speed up the running of the algorithm for determining the minimum cycle time, we present a method of parallelization of the most time-consuming procedures of determining blocks performed in each iteration.

Property 4 *Determination of blocks for cyclic flow shop problem with setups can be done in time $O(\log n)$ on mn -processor CREW PRAM machine.*

The method of determining the blocks is presented in the Algorithm 2.

Algorithm 2. *Alg_pblocks*

Input: permutations: $\pi = (\pi(1), \pi(1), \dots, \pi(n))$

and $\pi^* = (\pi^*(1), \pi^*(1), \dots, \pi^*(n))$

Output: vector of blocks starting positions (b_1, b_2, \dots, b_k) , k – number of blocks and vector of blocks ending positions (e_1, e_2, \dots, e_k)

Sample input:

$$\pi = (8, 10, 7, 4, 5, 6, 3, 1, 9, 2)$$

$$\pi^* = (9, 3, 1, 6, 10, 7, 4, 5, 2, 8)$$

$$k = 2, b = (2, 7), e = (5, 8)$$

$$B_1 = (10, 7, 4, 5), B_2 = (3, 1)$$

Step 1. parfor $r \in \{1, 2, \dots, p\}$ **do**

$$\pi(0) := \pi(n+1) := \pi^*(0) := \pi^*(n+1) := -1;$$

$$(\pi^*)^{-1}(0) := (\pi^*)^{-1}(n+1) := -1;$$

if (preceding position in π is identical as in π^* , i.e.

$$\pi(r-1) = \pi^*((\pi^*)^{-1}(\pi(r)) - 1)) \text{ **then**}$$

$$B_b[r] := 1;$$

else

$$B_b[r] := 0;$$

if (next position in π is identical as in π^* , i.e.

$$\pi(r+1) = \pi^*((\pi^*)^{-1}(\pi(r)) + 1)) \text{ **then**}$$

$$B_e[r] := 1;$$

else

$$B_e[r] := 0;$$

Step 2. parfor $r \in \{1, 2, \dots, p\}$ **do**

if $((B_b[r] = 0) \text{ and } (B_b[r+1] = 1))$ **then**

$$B_b[r] := 1;$$

else

$$B_b[r] := 0;$$

for sample input:

$$B_b = (0, 1, 0, 0, 0, 0, 1, 0, 0, 0)$$

Step 3. parfor $r \in \{1, 2, \dots, p\}$ **do**

if $((B_e[r] = 0) \text{ and } (B_e[r-1] = 1))$ **then**

$$B_e[r] := 1;$$

else

$$B_e[r] := 0;$$

for sample input:

$$B_e = (0, 0, 0, 0, 1, 0, 0, 1, 0, 0)$$

Step 4. Determine the prefix sum of P elements from table B_b , i.e.

$$\forall_{r \in \{1, 2, \dots, p\}} P[r] = \sum_{i=1}^r B_b[i].$$

for sample input:

$$B_b = (0, 1, 0, 0, 0, 0, 1, 0, 0, 0)$$

$$P = (0, 1, 1, 1, 1, 1, 2, 2, 2, 2)$$

Step 5. parfor $r \in \{1, 2, \dots, p\}$ **do**

if $(B_b[r] = 1)$ **then**

$$b[P[r]] := r;$$

if $(B_e[r] = 1)$ **then**

$$e[P[r]] := r;$$

for sample input:

$$B_e = (0, 0, 0, 0, 1, 0, 0, 1, 0, 0)$$

$$B_b = (0, 1, 0, 0, 0, 0, 1, 0, 0, 0)$$

$$P = (0, 1, 1, 1, 1, 1, 2, 2, 2, 2)$$

$$b = (2, 7), e = (5, 8)$$

Fig. 1 shows the implementation of Algorithm 2 in the form of pblocks procedure of block determination with n processors using OpenMP parallelization library. This algorithm was then run on the Xeon Phi coprocessor. The input data are: the size of permutations in which blocks n , blocks ttn , a permutation pi and permutation – a pattern pi_ptr are determined. Tables b_b and b_e , after the completion of the procedure, contain the beginning and ending positions of the following blocks.

```

int pblocks(int n, int *pi, int *pi_ptr,
            int *b_b, int *b_e) {
    int *pi_ptr_ = new int[n+2];
    int *b = new int[n+1];
    int *b_ = new int[n+1];
    int *bk = new int[n+1];
    int *p = new int[n+2];
    int *p_ = new int[n+2];
    pi_ptr[0] = pi_ptr[n+1] = -1;
    zeros_pi(n, pi_ptr_);
    pi_ptr_[0] = pi_ptr_[n+1] = -1;
    #pragma omp parallel for
    for(int i=1; i<=n; ++i)
        pi_ptr_[pi_ptr[i]] = i;
    #pragma omp parallel for
    for(int i=1; i<=n; ++i)
    {
        if(pi[i-1]==pi_ptr[pi_ptr_[pi[i]] - 1])
            b[i] = 1; else b[i] = 0;
        if(pi[i+1]==pi_ptr[pi_ptr_[pi[i]] + 1])
            bk[i] = 1; else bk[i] = 0;
        p[i] = p_[i] = 0;
    }
    p[1] = b[1]; p[0] = 0;
    #pragma omp parallel for
    for(int i=1; i<=n; ++i)
        if(b[i] == 0 and b[i+1] == 1)
            b[i] = 1; else b[i] = 0;
    #pragma omp parallel for
    for(int i=n; i>=1; --i)
        if(bk[i] == 0 and bk[i-1] == 1)
            bk[i] = 1; else bk[i] = 0;
    for(int i=2; i<=n; ++i)
        p[i] = p[i-1] + b[i];
    int nblocks = 0;
    #pragma omp parallel for
    for(int i=1; i<=n; ++i)
        if(b[i] == 1)
            b_b[p[i]] = i;
            if(bk[i] == 1)
                b_e[p[i]] = i;
    #pragma omp parallel for
    for(int i=1; i<=n; ++i)
    {
        #pragma omp critical
        if(b_b[i] != 0 or b_e[i] != 0)
            nblocks++;
    }
    #pragma omp parallel for
    for(int i=1; i<=nblocks; ++i)
        if(b_b[i] == 0 and b_e[i] != 0)
            b_b[i] = 1;
        if(b_e[i] == 0 and b_b[i] != 0)
            b_e[i] = n;
    delete[] pi_ptr_; delete[] b; delete[] b_;
    delete[] p; delete[] p_;
    return nblocks;
}

```

Figure 1: Function of the parallel block determination.

5. Computational experiments

A parallel procedure for the blocks determination has been implemented in C++ using OpenMP library. The data for computational experiments (permutations of tasks)

were generated randomly. The number of elements of permutations changed in range from 10^3 , to 10^7 . Computational experiments were carried out in a computing environment with shared memory - the coprocessor Intel Xeon Phi 3120 (space 6GB, 1.1 GHz) enabling the use of 228 cores.

The aim of the first phase of the experiments was to demonstrate the effectiveness of the (sequential) mechanism of blocks based on patterns. For this purpose, a procedure for blocks determining was placed in the classic *tabu search* (TS), algorithm with prohibitions with tabu list determined of length 7. There were two versions of the algorithm run - with and without blocks, for a predefined number of 1000 iterations. There was *Percentage Relative Deviation* (PRD), studied to solve the reference solution obtained with the use of NEH algorithm (Nawaz i in. [10]) for test data from work [13]. The results reported in Table 1, indicate that with nearly a twofold shorter time of the algorithm running, the obtained results were much better (32.8% improvement over the NEH) as compared to the version of the algorithm running without the block mechanism (29.0% improvement over the NEH).

Table 1: Comparison of PRD to NEH for TS algorithm with and without the blocks - 1,000 iterations.

$n \times m$	$t[s]$	$t_B[s]$	PRD	PRD_B
20×5	2.2	0.8	-30.0	-32.7
20×10	3.0	0.9	-29.5	-31.6
20×20	4.7	1.2	-29.8	-30.7
50×5	35.7	17.7	-32.5	-34.1
50×10	48.4	22.0	-29.6	-34.5
50×20	73.9	28.3	-28.3	-31.8
100×5	292.1	181.8	-30.7	-36.7
100×10	391.9	203.8	-28.1	-33.6
100×20	604.7	266.4	-27.9	-31.6
200×10	3212.2	1791.1	-26.7	-32.9
200×20	5255.5	2497.7	-26.2	-31.1
Average	902.2	455.6	-29.0	-32.8

The second phase of the experiment consisted of measuring the acceleration of the procedure for the parallel determining of the blocks. The results of computational experiments for the co-processor Intel Xeon Phi 3120 were shown in Tables 2 and 3 and presented in Fig. 2 and 3. For a different number of tasks in permutation the acceleration initially increases quite rapidly, reaches a maximum, and then decreases slowly. The number of processors for which the maximum acceleration is reached depends on the size of the problem. Using the notion of *scalability* of parallel algorithms one can say that for $5000 \cdot 10^3$ tasks in a permutation parallel method for block determination is

characterized for $p = 1, 2, \dots, 32$ with a strong scalability, since with an increase in the number of processors the speedup increases (Fig. 3).

Table 2: Speedup of pblocks procedure.

$\lambda^{(1)}$	$p = 2$	$p = 4$	$p = 6$	$p = 8$	$p = 10$
1	0.003	0.001	0.001	0.001	0.001
2	0.008	0.004	0.003	0.004	0.003
5	0.018	0.008	0.008	0.007	0.007
10	0.037	0.016	0.017	0.017	0.015
20	0.078	0.033	0.036	0.043	0.033
50	0.308	0.154	0.165	0.143	0.147
100	0.632	0.441	0.458	0.443	0.400
200	0.961	0.915	0.986	1.047	0.980
500	1.229	1.578	1.889	2.061	2.171
1000	1.309	1.943	2.498	2.773	3.024
2000	1.302	2.255	2.994	3.490	3.810
5000	1.366	2.478	3.431	4.160	4.880
10000	1.398	2.584	3.612	4.461	5.266

⁽¹⁾ $\lambda = n \cdot 10^3$, coprocessor Intel Xeon Phi 3120A

Table 3: Speedup of pblocks procedure.

$\lambda^{(1)}$	$p = 16$	$p = 32$	$p = 64$	$p = 128$
1	0.001	0.001	0.001	0.001
2	0.003	0.002	0.002	0.001
5	0.007	0.006	0.006	0.004
10	0.015	0.013	0.012	0.008
20	0.033	0.029	0.023	0.019
50	0.129	0.129	0.100	0.073
100	0.400	0.347	0.261	0.195
200	0.988	0.884	0.697	0.507
500	2.234	2.222	1.799	1.285
1000	3.360	3.552	2.967	2.278
2000	4.679	5.307	4.961	3.853
5000	6.306	8.270	8.154	7.043
10000	7.091	9.766	10.775	9.679

⁽¹⁾ $\lambda = n \cdot 10^3$, coprocessor Intel Xeon Phi 3120A

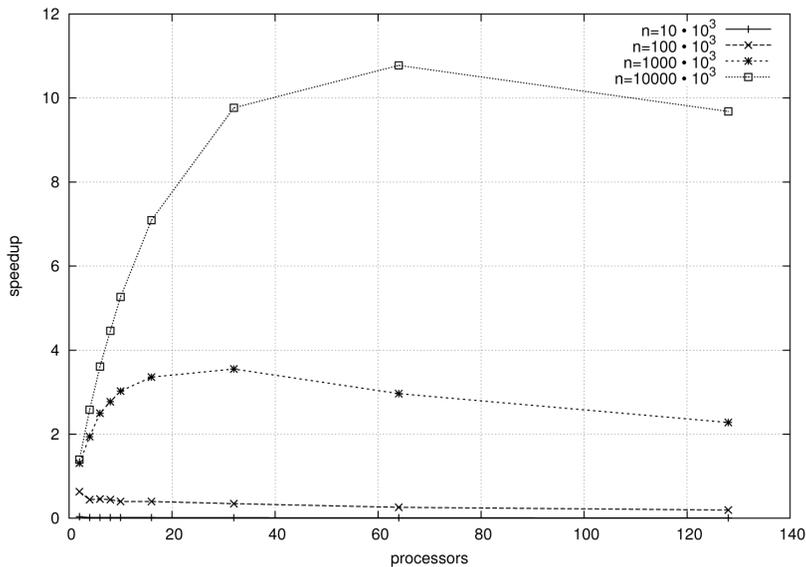


Figure 2: Dependency of speedup on the number of processors – Intel Xeon Phi 3120A.

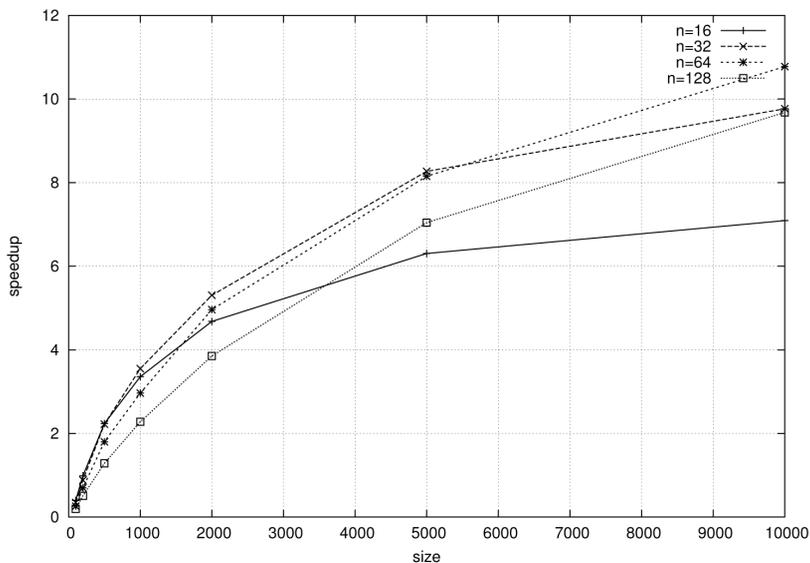


Figure 3: Dependency of speedup on the number of tasks – Intel Xeon Phi 3120A.

6. Conclusions and comments

This paper presents a new concept of blocks for a cyclic flow shop problem with machine setups, using multiple patterns of different sizes. Patterns represent the optimal

order of visiting cities in a traveling salesman problem, ensuring block properties in the problem. The use of the block properties enables for a significant reduction in the number of viewed neighbors in metaheuristic algorithms based on viewing the neighborhoods. Future work in the work's field could be focused on the extension of the pattern approach (which creates blocks in a solution) by researching a distance measure based on the pattern-based neighborhood.

References

- [1] Bożejko W., Uchroński M., Wodecki M. (2016). Parallel metaheuristics for the cyclic flow shop scheduling problem. *Computers & Industrial Engineering*, 95, 156–163.
- [2] Bożejko, W., Pempera, J., Wodecki, M. (2015). Parallel Simulated Annealing Algorithm for Cyclic Flexible Job Shop Scheduling Problem. *Lecture Notes in Artificial Intelligence* No. 9120, Springer, 603–612.
- [3] Bożejko, W., Uchroński, M., Wodecki, M. (2015). Block approach to the cyclic flow shop scheduling. *Computers & Industrial Engineering*, 81, 158–166.
- [4] Bożejko, W., Wodecki, M. (2007). On the theoretical properties of swap multi-moves. *Operations Research Letters*, 35(2), 227–231.
- [5] Brucker, P., Burke, E. K., Groenemeyer, S. (2012). A branch and bound algorithm for the cyclic job-shop problem with transportation. *Computers & Operations Research*, 39, 12, 3200–3214.
- [6] Gertsbakh, I., Serafini, P. (1991). Periodic transportation schedules with flexible departure times. *European Journal of Operational Research*, 50, 298–309.
- [7] Grabowski, J., Wodecki, M. (2004). A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion. *Computers & Operations Research*, 31, 1891–1909.
- [8] Kats, V., Levner, E. (2010). A fast algorithm for a cyclic scheduling problem with interval data. In *Proceedings of the annual operations research society of Israel (ORSIS-2010) conference*, February 2010, Nir Etzion, Israel.
- [9] Mendez, C. A., Cerda, J., Grossmann, I. E., Harjunkski, I., Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and Chemical Engineering*, 30, 913–946.
- [10] Nawaz, M., Enscore, Jr, E.E., Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *OMEGA International Journal of Management Science*, 11, 91–95.

- [11] Pinedo, M. (2005). *Planning and scheduling in manufacturing and services*. New York: Springer.
- [12] Pinnto, T., Barbosa-Povoa, A. P. F. D., Novais, A. Q. (2005). Optimal design and retrofit of batch plants with a periodic mode of operation. *Computers and Chemical Engineering*, 29, 1293–1303.
- [13] Ruiz, R., Stützle, T. (2008). An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 187(3), 1143–1159.
- [14] Smutnicki, C., New features of the cyclic job shop scheduling problem. In *Proceedings of 20th International Conference on Methods and Models in Automation and Robotics MMAR 2015*, IEEE Press, 1000–1005.

Availability analysis of selected mining machinery

JAROSŁAW BRODNY, SARA ALSZER, JOLANTA KRYPEK and MAGDALENA TUTAK

Underground extraction of coal is characterized by high variability of mining and geological conditions in which it is conducted. Despite ever more effective methods and tools, used to identify the factors influencing this process, mining machinery, used in mining underground, work in difficult and not always foreseeable conditions, which means that these machines should be very universal and reliable. Additionally, a big competition, occurring on the coal market, causes that it is necessary to take action in order to reduce the cost of its production, for eg. by increasing the efficiency of utilization machines. To meet this objective it should be proceeded with analysis presented in this paper. The analysis concerns to availability of utilization selected mining machinery, conducted using the model of OEE, which is a tool for quantitative estimate strategy TPM. In this article we considered the machines being part of the mechanized longwall complex and the basis of analysis was the data recording by the industrial automation system. Using this data set we evaluated the availability of studied machines and the structure of registered breaks in their work. The results should be an important source of information for maintenance staff and management of mining plants, needed to improve the economic efficiency of underground mining.

Key words: OEE model, TPM strategy, effectiveness, mining machines.

1. Introduction

The process of the underground exploitation of the coal is very complicated and it is characterized by a huge changeability of mining and geological conditions in which it takes place. The globalization and the growing competition in the energy resources industry forces national mining companies, which want to remain on the market, to take actions one of which goal is higher level of resources use. In particular, it regards to all type of machines and devices.

J. Brodny (corresponding author), is with Institute of Production Engineering, Faculty of Organization and Management Silesian University of Technology, Roosevelta str. 26, 41-800 Zabrze, Poland, e-mail: jaroslaw.brodny@polsl.pl. S. Alszner and J. Krystek are with Institute of Automatic Control, Silesian University of Technology, Akademicka str. 16, 44-100 Gliwice, Poland, e-mail: jolanta.krystek@polsl.pl. M. Tutak is with Institute of Mining, Faculty of Mining and Geology, Silesian University of Technology, Akademicka str. 2, 44-100 Gliwice, Poland, e-mail: magdalena.tutak@polsl.pl

This work is the result of the research project No. PBS3/B6/25/2015 "Application of the Overall Equipment Effectiveness method to improve the effectiveness of the mechanized longwall systems work in the coal exploitation process" realized in 2015-2017, financed by NCBiR.

Received 21.12.2016. Revised 13.05.2017.

Concentration of extraction, realized by reducing the number of longwalls and, at the same time, increasing their performance, makes that the mine's activity is based mostly on one or two working longwalls. Therefore, it is obvious that the effectiveness of the entire mining company depends on efficiency and proper work organization in these longwalls. One of the factors having a significant impact on the effectiveness is optimal capacity utilization of machines, which directly translates into improvement of work efficiency and productivity in these companies. Taking any actions, it is necessary to take into account the specifics of mining companies, which belong to the group of the open plant and differ significantly from stationary enterprises in other industries. In the underground mining exploitation, there is a high risk which is the result of various types of natural hazards and specific work environment. During the exploitation, there are often disturbances of the production process and that generates losses and causes an increase in the cost of coal excavation. The causes of these disturbances are technological as well as technical and organizational factors [10]. Additional costs are generated not only by stopping the extraction process but these are also "lost profits", which are equal to the potential production volume at the time of the stoppage.

The process of coal production involves several stages of which the most important is the obtaining of useful mineral in the exploitation phase (Fig. 1). This exploitation is based on a mechanical cutting out the useful mineral from the rock mass and transporting it from the zone of direct exploitation. Currently, due to the fact that the exploitation process is mostly realized by the longwall system (long, easy for mechanization operational fronts), this zone is called the zone of the longwall face.

Shown in Figure 1 a simplified scheme of coal production process includes also horizontal transport of excavated material to the shaft zone, from which it is transported to the surface by devices using for vertical transport. Subsequently, the output goes to the processing plant and there after taking part with enrichment process the final product, in the form of the right coal with definite parameters, is obtained.

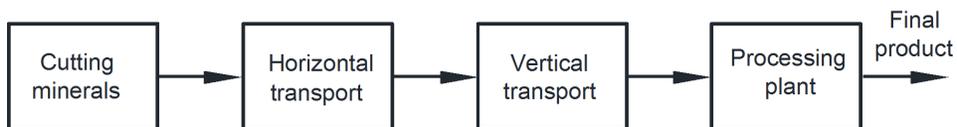


Figure 1: Simplified scheme of coal production process.

The coal cutting process is closely related to the life cycle of the excavation operation mining excavation (called the exploited longwall) and it is shown in Figure 2.

This cycle includes three phases, namely the exploited longwall preparation (so called the disarmament and start-up), the main exploitation and liquidation. All these phases are important for the effectiveness of exploitation process but the final economic result is mainly determined by the second phase, during which the main mineral exploitation is carried out. The proper run of this phase is strongly affected by any type of devices including mining machinery, which take part with the exploitation process and

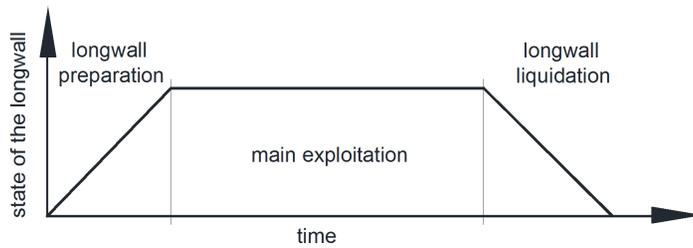


Figure 2: The life cycle of the exploited longwall.

minerals transport, and devices, which protect the whole operational zone (for example mining roof support). Their reliability, availability and performance largely determine the economic efficiency of the whole longwall exploitation. In order to achieve the best results, it is necessary to take actions aimed at optimum use of operated machines by for instance their proper selection to mining and geological conditions, in which they work, providing adequate preventive maintenance, control, etc. A tool that successfully stabilizes machinery's work and at the same time optimizes the cost of its operation is methodology, originating from Japan, for the comprehensive management of machinery and equipment effectiveness, called TPM (Total Productive Maintenance) [1, 2, 5, 7]. To take effective actions in this area it is necessary to conduct research and analysis that are aimed at determining the real utilization of machines on the basis of credible information. For this purpose research, which are described in this paper, has been conducted.

So far, research, which has been carried out in this area, was based only on failure analysis of mining equipment and it was leaned on notes made in the dispatcher's registers. The relatively low reliability of the data made it impossible to determine the actual use of mining equipment and identify the structure and reasons of downtime occurring during working.

In this paper, the data obtained from the industrial automation system was the basis of availability analysis of chosen mining machines. This system, independently from the machine's operators, registers number of machinery parameters in a discreet way. Based on obtained data, an effectiveness analysis of the use of a set of mining equipment belonging to the mechanized longwall system, which is used for direct coal cutting and transporting excavated material from the zone of a longwall face, was conducted. This system also includes mining roof support, which protects mining excavation, and a set of devices that are necessary to carried out the exploitation. Research involved the work of the longwall shearer, the armoured face conveyor, the beam stage loader and the crusher. Machines availabilities and structure of registered breaks in their work were determined on the basis of the obtained data. The study was conducted based on the Total Productive Maintenance strategy (TPM) using the assumptions of the Overall Equipment Effectiveness (OEE) model, which is a tool used for quantitative evaluation of this strategy [1, 3, 4, 6, 8, 9, 11].

2. The TPM strategy and the OEE model

One of the areas where there are opportunities to effectively reduce the cost of mining production is the effectiveness of the use of resources, especially, the mining machinery. Recent years, in Polish underground mining thanks to dynamic technical progress, better and better machines are designed, produced and operated and thus these machines are also more expensive. The increase in technical capabilities of these machines, which are, in many cases, highly modern and powerful, and whose level of reliability is very high, does not always make the benefits linked with their use [10, 12].

The strategy, which is successfully used by stationary production plants to analyze the effectiveness of the equipment use, is Total Productive Maintenance (TPM) [1, 2, 5, 7]. According to the strategy, improvement of the economic efficiency can be achieved through a set of actions and activities aimed at machinery, equipment and other technical measures (renewable resources) maintaining in a failure-free and faultless state, reducing the number of failures, unplanned downtime and lack. This strategy also refers to the human factor, assuming that the effectiveness increasing process of machines and devices has to include changes in the perception of particular technical and organizational activities by employees, to be efficient. Therefore, it is justified to take measures aimed at making employees aware that their knowledge, skills, engagement, responsibility and awareness of their role in the machine maintenance are necessary condition to succeed in this field. Identification of employees with the company, in which they work, is one the assumptions of the TPM strategy. Its application is aimed at reducing failures, unplanned downtime, including so-called micro-downtime, and increasing performance and improving production quality. [2, 5, 7].

In order to effectively make organizational and technical changes, which can result in the improvement of effectiveness of the use of the production resources, it is necessary to analyze the initial state. A tool that allows carrying out such analysis and, at the same time, quantifying the effectiveness of the TPM strategy is the Overall Equipment Effectiveness indicator. [4, 6, 8, 9, 11]. This indicator is the product of three components which include availability and performance of the studied machine and the quality of the obtained product. The values of the partial indicators and the OEE indicator are determined based on the following formula:

$$OEE = A \cdot P \cdot Q \quad (1)$$

where: A – availability, P – performance, Q – quality.

$$D = \frac{D_o - D_t}{D_o} \cdot 100\% \quad (2)$$

where: D_t – downtime, D_o – total time available.

$$W = \frac{W_r}{W_n} \cdot 100\% \quad (3)$$

where: W_r – the real performance during the work shift in tons of excavated material, W_n – the performance founded in the technological plan for the very longwall in tons.

$$J = \frac{J_a}{J_c} \cdot 100\% \quad (4)$$

where: J_a – the real quality of excavated material (that includes the amount of gangue content and sortiment of excavated material), J_c – the quality founded in the technological plan.

Therefore, it is justified to claim that the final indicator (OEE) shows the level of use of the basic time which is needed to achieve a full production.

To determine the value of this indicator for some machine or production system it is very important to identify the causes because of which time losses occur. The most important causes are following [6, 8, 11]:

- failures which are defined as unplanned and often impossible to predict stoppage of the machine for technical reasons, and they make a value of availability indicator decrease as in case of pipeline production process (Fig. 1).
- unplanned downtimes whose the most common causes are logistical problems. Losses resulting from this cause make the value of availability decrease but it is relatively easy to reduce them by, for example, appropriate organizational activities.
- minor failures and stoppages which are most often the result of technical problems and they are removed by operators. They are treated as speed losses and affect a reduction of the performance indicator.
- rearming understood as a change of machine's equipment. In general, it is assumed that rearming is preparing a machine for production process but when the normative time intended to carry out this operation is exceeded, the excess is treated as a loss that make the availability value decrease.
- setting which is treated similarly as rearming.
- loss of performance that are a result from the slowdown of a machine or a system. Errors resulting from improper control, preventive stoppages or exceeded operating parameters (for instance temperature or pressure) can be reasons.
- defective product that causes loss of time which has to be designed to produce a new good product. It reduces the OEE value in the area of quality.

According to the overall equipment effectiveness model for each of studied machines and the entire set of the machines it is necessary to determine the partial indicators in the area of availability, performance and product quality (carbon).

Presented example is focused on the first partial indicator of the OEE model which is the availability indicator of studied equipment. In the next stages of conducted studies it

is necessary to expand this analysis because it should include other indicators as well in order to determine the Overall Equipment Effectiveness indicator for each of examined machines and for the whole set of mining machines.

3. Characteristics of the tested set of machines

The tested set of machines is a part of the mechanized longwall system which is used for underground coal exploitation in the longwall. The longwall system is based on cutting a useful mineral (in this case coal) in the zone of the longwall face whose length can vary from approx. 60 meters to approx. 300 meters. Such long longwall makes great opportunity to make cutting and transporting process mechanized and automated.

Figure 3 shows a scheme of a longwall excavation with the selected main parts [12]. Figure 4 shows its view during exploitation [3].

A set of mining machines, belonging to the mechanized longwall system, which is presented in this analysis concludes the longwall shearer, the armoured face conveyor, the beam stage loader and the crusher designed for crushing large parts of excavated material.

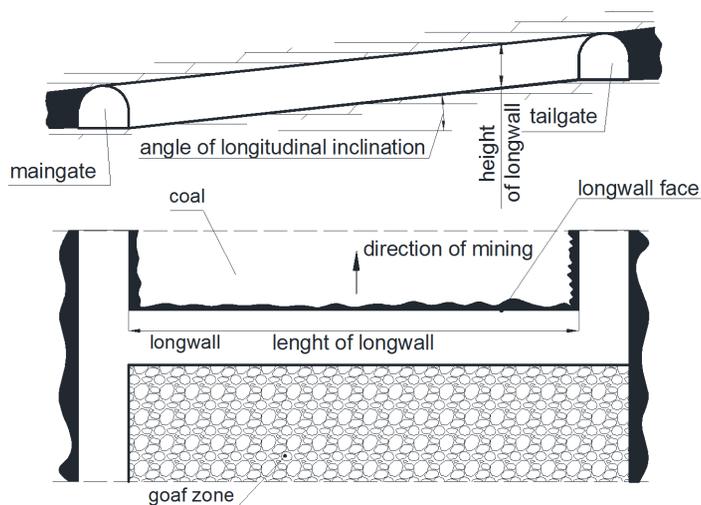


Figure 3: A scheme of a longwall excavation.

The basic machine that is a part of the mechanized longwall system is the longwall shearer. Its task is to cut the coal and load excavated material onto the armoured face conveyor. The shearer is very important for the whole analyzed set of machines because it is the first link in the technological exploitation process. The effectiveness of the analyzed set of machines, as well as of the entire operation process depends on the shearer's reliability. Coal cut by a longwall shearer is transported by an armoured face conveyor

from the longwall to a bottom gate and then it is reloaded onto the beam stage loader and transported out of the zone of the longwall face. The analyzed set of machines includes also crusher whose task is to break large parts of coal to make it possible to transport it further. Inefficiency of a crusher can cause that the exploitation will have to be stopped due to the fact that the coal transport is then impossible.



Figure 4: A view of longwall excavation with equipment during exploitation [3].

Selection of such a set of machines used in this analysis is justified because the quantitative analyses of machines and devices failure that are conducted in mines indicate that most of failures are caused by a cutting machine and conveyors [1, 3, 10]. Therefore, it can be assumed that machines included in the longwall system together with a beam stage loader and a crusher have the greatest impact on the effectiveness of the entire exploitation process.

From the reliability point of view, the studied set of machines is a serial structure system and it means that it properly works when all its components are efficient (Fig. 5).

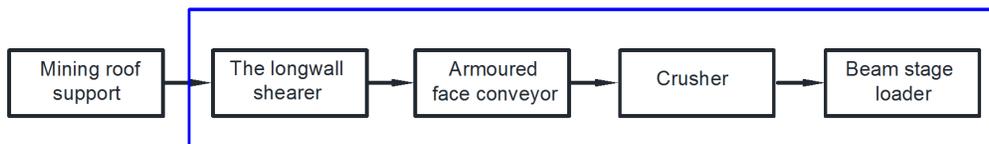


Figure 5: A scheme of reliability structure of the studied system.

Selected parameters of the tested machines are summarized in Table 1. It is important to noticed that some of these parameters are given in two versions: as nominal values and as values that can be achieved by the specific machine in the conditions which are characteristic for the particular longwall.

Table 1: Selected technical parameters of the studied machines

Machine	Parameter	Value of parameter	Value of parameter in longwall conditions
The longwall shearer	Maximum installed power:		
	– cutting heads	2×375 kW	2×375 kW
	– feed	2×60 kW	2×60 kW
	– hydraulic	2×13 kW	2×13 kW
	Work speed	0–12 m/min	0–12 m/min
Armoured face conveyor	Engines power	3 × 400 kW	3 × 400 kW
	Conveyor's performance	1200 t/h	1200 t/h
Beam stage loader	Installed power	1 × 300 kW	1 × 300 kW
	Maximum performance	2000 t/h	1200 t/h
Crusher	Engines power	1 × 200 kW	1 × 200 kW
	Maximum performance	4000 t/h	1200 t/h

4. Determination of availability of the studied set of machines

The specificity of mining industry was taken into consideration during preparing the methodology for determining the OEE for mining machinery. Data collected by an industrial automation system were used to calculate availability of these machines. It was assumed that real work time is difference between total time available included in standard (normative time) and unplanned downtime. This is the time when machines did not work (current consumption of their motors was zero) and there were no problems related to the difficult mining conditions. Thus, the methodology for determining this OEE index included a thorough analysis of registered downtimes and their causes. The specificity of mining exploitation allows for occurrence of such cases that breaks at work of machines are caused by objective factors which are independent of these machines. Performance and quality indicators were also specified using different approach than in case of closed companies. The mass of disintegrated rock related to the value provided in standard was taken into account when calculating performance index. However, in the case of determining the quality index, amount of waste rock in production and sortiment of coal were included.

The whole methodology for determining OEE is therefore based on data not used for other production companies. Its originality results from its adaptation to the mining specificity.

The basis for determining the indicators of availability of each of the studied machine was data which was recorded in the zero-one system. Figure 6 shows an example of temporal waveform of longwall shearer's work in during work shift (lasting 360 minutes).

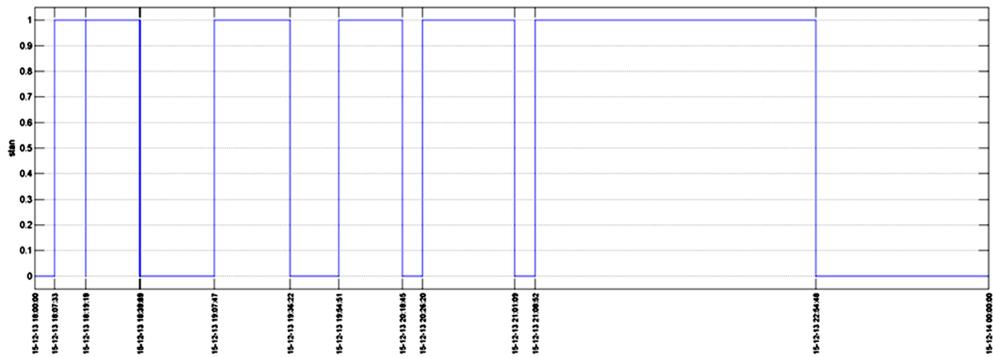


Figure 6: Temporal waveform of longwall shearer's work during one work shift.

Analyzed work shift's real (actual) time in case of the longwall shearer was 13590 seconds and downtime amounted to 8010 seconds. For this change availability rate was therefore 58.9

Figure 7 shows the calculated values of availability indicator of the studied longwall shearer for 64 work shifts and the average value of this indicator during analyzed period (work shift availability and average availability).

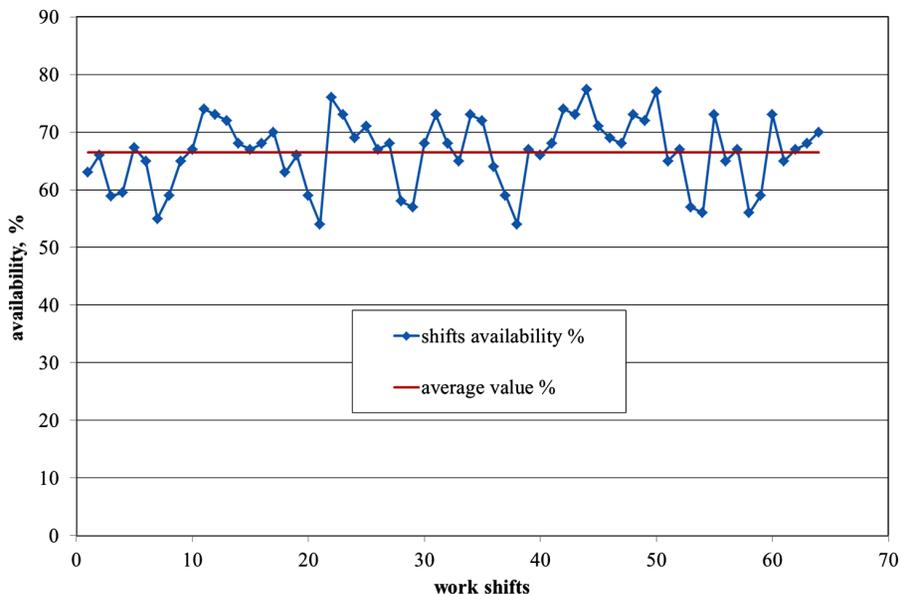


Figure 7: The values of availability indicator of the longwall shearer.

In the similar way, the values of availability indicator of the other studied machines, which are the parts of the longwall system, were determined. (Fig. 8).

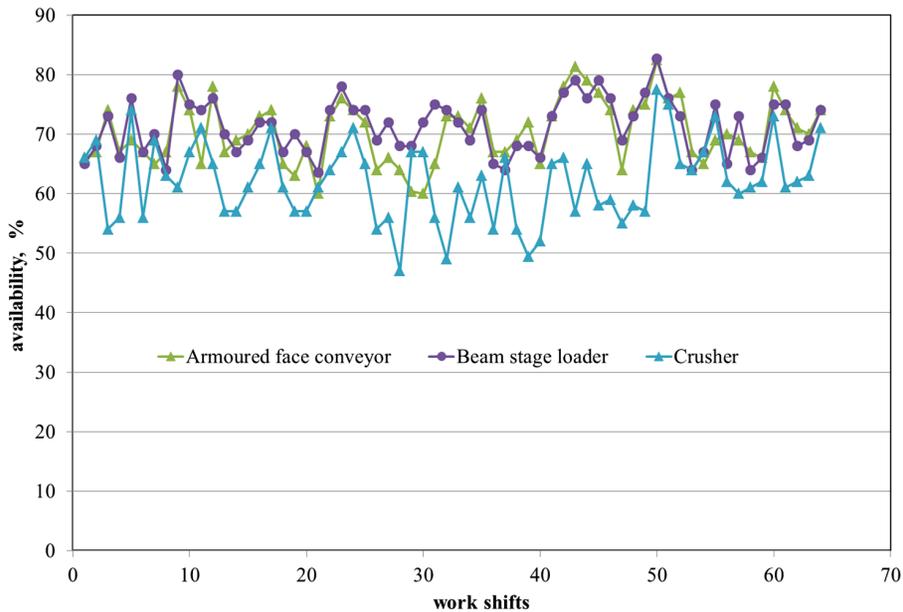


Figure 8: The values of availability indicator of particular machines of the examined set.

Table 2: Selected technical parameters of the studied machines

Machine	Average daily values of availability indicator, %	Maximum daily values of availability indicator, %	Maximum daily values of availability indicator, %
The longwall shearer	$66,63 \pm 5,97$	77,36	54,21
Armoured face conveyor	$70,36 \pm 5,27$	82,45	60,34
Beam stage loader	$71,25 \pm 4,61$	82,65	63,53
Crusher	$62,06 \pm 6,65$	77,43	49,45
The entire set of machines	$67,55 \pm 6,72$	82,65	49,45

On the basis of calculations the average, maximum and minimum daily values of availability indicator of these machines and of the entire set of machines were determined (Table 2).

According to the analysis of the results, it can be claimed that the availability values of the machines are on the low level. Their average value did not exceed 75% for any of the tested machines. It is worth noticing that 75% is widely assumed to be the acceptable lower threshold.

Determined temporal waveforms of machines' work also enabled to determine the structure of the breaks during work. Figure 9 shows a summary of these breaks in case of the studied longwall shearer for all 64 work shifts. Percentage share of the particular breaks (divided into ten categories) was related to the total break time during all studied time.

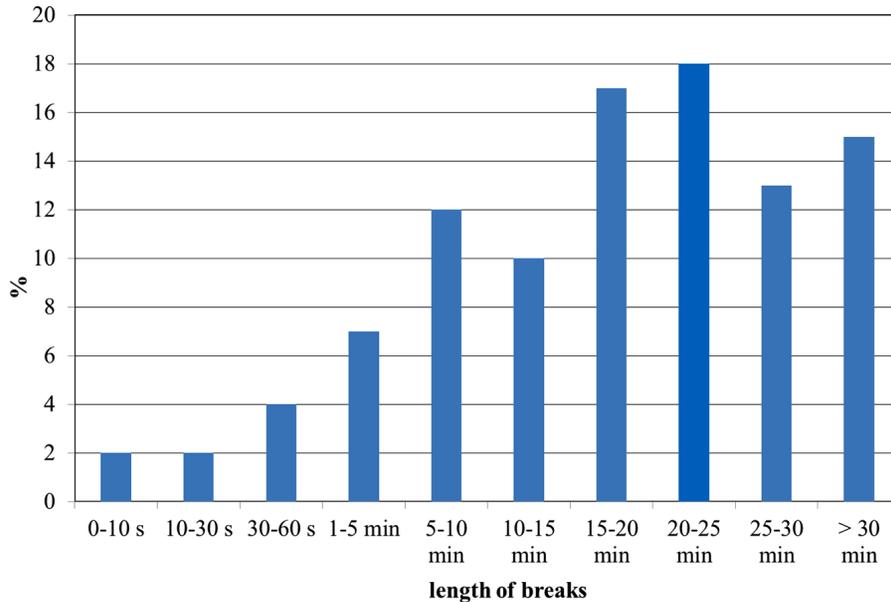


Figure 9: The structure of the breaks in longwall shearer's work.

It can be concluded, base on the results, that the breaks, the duration of which is in the range of 20 to 25 minutes (represent 18% of all time of the breaks), have the highest percentage share of the total breaks time during longwall shearer's work. The diagram clearly shows that the most important for work of this machine are the breaks whose duration is above 5 minutes and which represent approximately 85% of all registered breaks.

5. Conclusions

One of the areas, where there are opportunities to effectively reduce the cost in mining companies, is that one which includes a selection and a use of all types of technical equipment, in particular mining machinery. It is necessary to acquire adequate knowledge about the level of a technical resources use in these companies to make taken actions efficient.

Presented methodology for determining the indicator of effectiveness of the mining machines' use based on data obtained from a industrial automation system should be an important source of knowledge that is necessary to assess the state of machines' work. The conclusions coming from the analysis of the indicators and, in particular, from their variation referred to the taken actions should be the basis for decisions made by appropriate services in order to optimize the use of these machines.

In the present case, the data obtained from the industrial automation system was the primary source of information and based on it, availability indicators of the studied machines were determined. It guaranteed their credibility (and it eliminated the inaccuracy of the registration carried out by dispatchers) and it made the opportunity to register all types of downtime in real time. To identify the causes of the recorded breaks, especially unplanned ones, it is necessary to use the information gained from dispatchers of these machines, but in this area special actions (like trainings), which will raise awareness of the employees, should be taken. Implementation of both of these sources of information should guarantee to obtain reliable and credible information about the state of the studied machines which are essential for the services supervising their use.

The results of the study clearly indicate that there are a lot of reserves in the availability area of examined mining machines and the indicators values are unsatisfactory. In order to improve this situation, it is justified to continue a study and to more completely diagnose the causes resulting in such low values of these indicators.

References

- [1] J. BRODNY, K. STECUŁA and M. TUTAK: Application of the TPM strategy to analyze the effectiveness of using a set of mining machines. *16th Int. Multidisciplinary Scientific GeoConference*, DOI:10.5593/SGEM 2016B12/S03.009p. (2016), 65-72.
- [2] S. BORRIS: Total Productive Maintenance. The McGraw-Hill Companies, New York, 2006.
- [3] G.A. EINICKE, J.C. RALSTON, C.O. HARGRAVE, D.C. REID and D.W. HAINSWORTH: Longwall Mininig Automation. An Application of Minimum-Variance Smoothing. *IEEE Control Systems Magazine*, **6** (2008).
- [4] S. ELEVLI and B. ELEVLI: Performance measurement of mining equipments by utilizing OEE. *Acta Montanistica Slovaca*, **15**(2), (2010), 95-101.
- [5] O. LJUNBERG: Measurement of overall equipment effectiveness as a basis for TPM activities. *Int. J. of Operations & Production Management*, **18**(5), (1998), 495-507.
- [6] W. MAZUREK: Index OEE – Theory and Practice. 2 Ed. Neuron. www.neuron.com.pl, 2014, (in Polish).

-
- [7] S. NAKAJIMA: Introduction to TPM: Total Productive Maintenance. (Translation). Productivity Press, Inc., 129, 1988.
- [8] THE PRODUCTIVITY PRESS DEVELOPMENT TEAM: OEE for Operators. Overall Equipment Effectiveness, ProdPress, Wrocław, 2009, (in Polish).
- [9] M.M. PONCE-HERNÁNDEZ, A. GONZÁLEZ-ANGELES, C.R. NAVARRO-GONZÁLEZ and E. CABRERA-CÓRDOVA: Overall equipment effectiveness (OEE) diagnosis and improving in a small business as an essential tool for business competitiveness. *Research J. of Recent Sciences*. **2**(6), (2013), 58-65.
- [10] H. PRZYBYŁA: The risk of disruption of the extraction process in wall conditions of high concentration of production. *Przegląd Górniczy*, **9** (2009), 103-106, (in Polish).
- [11] D.H. STAMATIS The OEE Primer. Understanding Overall Equipment Effectiveness, Reliability, and Maintainability. Taylor & Francis, 2010.
- [12] W. WARACHIM and J. MACIEJCZYK: Wall coal harvesters. Śląskie Wydawnictwo Techniczne, Katowice, 1992, (in Polish).

The effect of cigarette smoking on endothelial damage and atherosclerosis development – modeled and analyzed using Petri nets

KAJA CHMIELEWSKA, DOROTA FORMANOWICZ and PIOTR FORMANOWICZ

Atherosclerosis as one of the crucial causes of cardiovascular diseases (CVD) is the leading reason of death worldwide. One of the contributing factors to this phenomenon is endothelial dysfunction, which is associated with the impact of various agents and their interactions. Tobacco smoke is one of the well known factors here. For better understanding of its significance a model of its impact on atherosclerotic plaque formation has been proposed. The model contains selected aspects of the influence of tobacco smoke, dual function of nitric oxide (NO) (influence of various mechanisms on NO bioavailability), oxidative stress which promotes low density lipoproteins oxidation, macrophages significance and other mechanisms leading to an aggravation of the endothelial disturbances. The model has been built using Petri nets theory and the analysis has been based on t-invariants. This approach allowed to confirm the important role of inflammation and oxidative stress in atherosclerosis development and moreover it has shown the considerable influence of the cigarette smoke.

Key words: atherosclerosis, endothelial dysfunction, cigarette smoking, modeling, Petri nets, t-invariants

1. Introduction

Despite the progress achieved in recent years, knowledge of the mechanisms associated with atherosclerosis is still incomplete, what impede effective treatment of its complications. Atherosclerosis is a complex and dynamic chronic inflammatory process which occurs in arterial vessels. Modifications and interactions between various factors

K. Chmielewska is with Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland, e-mail: kaja.chmielewska@cs.put.poznan.pl. D. Formanowicz is with Department of Clinical Biochemistry and Laboratory Medicine, Poznan University of Medical Sciences, Grunwaldzka 6, 60-780 Poznan, Poland, e-mail: doforman@ump.edu.pl. P. Formanowicz, the corresponding author, is with Institute of Computing Science, Poznan University of Technology, Piotrowo str. 2, 60-965 Poznan and Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego str. 12/14, 61-704 Poznan, Poland; e-mail: piotr.formanowicz@cs.put.poznan.pl

This research has been partially supported by the National Science Centre (Poland) grant No. 2012/07/B/ST6/01537.

Received 2.01.2017. Revised 6.05.2017.

have influence on this disorder. One of the crucial phenomenon of this process is endothelial dysfunction accompanied by inflammatory process and oxidative stress. The latter is associated with excessive production of reactive oxygen species (ROS). It reflects an imbalance between the amount of ROS and the ability to easily detoxification. On the other hand, formation of free radicals, defined as any molecular species capable of independent existence that contains an unpaired electron in an atomic orbital, can act positively through transmitting signals which stimulate repairing mechanisms. Moreover, inflammation, oxidative stress (respiratory burst) and other atherosclerosis-related processes are additionally stimulated by cigarette smoking.

This network of dependencies is very interesting via its complexity and dynamics. To better understand the nature of the entire disease process the systems approach has been used, because analysis of individual/particular processes or events has been found to be not sufficient [1, 9, 16, 29]. An analysis of the presented model allowed to systematize knowledge about atherosclerosis and its key processes. Furthermore, significant components of the modeled system has been distinguished. The analysis of all mechanisms and interactions between them requires mathematical methods and computer tools. In this case healthy human organism is treated as a system in dynamic equilibrium and a disease state is a consequence of inhibition or accumulation of some molecules. To show the complexity of the presented process the mathematical model, based on Petri nets theory [5, 20, 24], has been created. The analysis of the model has been based on t-invariants, which correspond to subprocesses occurring in the modeled system [12, 18]. Thus, similarities in t-invariants allow to identify subprocesses which interact with each other and for this reason clustering of t-invariants has been done. Additionally, an analysis of MCT sets, which contain transitions corresponding to exactly the same t-invariants, may lead to identification of significant functional blocks of the proposed model.

The model which has been described in this paper is an extended version of the model presented in [3]. The first version of the model focused on endothelial damage and mechanisms which occur in human organism, but it contains only few factors related to cigarette smoking. The extended model presented in this paper includes additional processes and molecules associated with smoking. Mechanisms which have been included in the model presented in [3] promote inflammation and endothelial dysfunction. Selected additional mechanisms which have been included in the extended model are modifications of lipids profile, promotion of prothrombotic state and formation of thrombus.

2. Methods

A structure of a Petri net is a weighted directed bipartite graph which consist of two disjoint subsets of vertices, i.e., places and transitions. Arcs in such a net connect vertices which belong to different subsets, i.e., transitions with places and places with transitions [5, 20, 24]. In the case of modeling of biological processes places (represented by circles) correspond to biological or chemical components. Transitions (represented by rectangles) correspond to elementary subprocesses (e.g., reactions between molecules)

[7, 18]. Other important elements are tokens which represent quantities of particular components included in a model. The distribution of tokens over the places corresponds to the state of the modeled system. Tokens flow through the net, and more precise, they flow between places via transitions, what is related to activation and firing of transitions. A transition is active, when the number of tokens in each place which directly precedes this transition is equal to at least the weight of the arc connecting the place with the transition. Furthermore, an active transition can be fired, what means that tokens flow from each place directly preceding the transition to each place directly succeeding it and the number of flowing tokens is equal to a weight of an appropriate arc. The flow of tokens corresponds to a flow of information, substances etc. through the modeled system [7, 28].

A graphical representation is one of the possibilities of representing Petri nets. It is very intuitive and helps to understand the structure of a modeled system but is not well suited for a formal analysis of its properties. For this purpose another representation, called incidence matrix, is used. Entry a_{ij} of such a matrix $A = [a_{ij}]_{n \times m}$, where n is the number of places, and m is the number of transitions, is an integer number equal to the difference between the numbers of tokens present in place p_i before and after firing transition t_j [20]. Figure 1 shows a graphical representation of an exemplary simple Petri net and its incidence matrix.

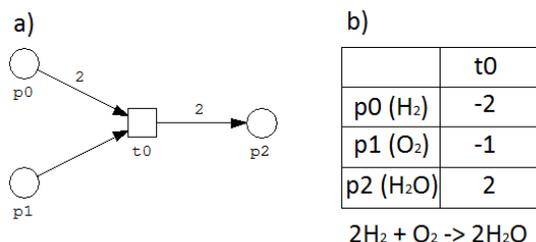


Figure 1: a) A graphical representation of a simple Petri net modeling synthesis of water; b) an incidence matrix of the net presented in part a).

On the basis of incidence matrix A transition invariants (t-invariants) can be calculated [12, 18]. Such invariants play significant roles in an analysis of models of biological systems. A t-invariant is n -element vector x , for which equation $A \cdot x = 0$ is fulfilled. To every t-invariant x there corresponds a set of transitions, called its support, which contains those transitions which correspond to non-negative coordinates of vector x , i.e., $\text{supp}(x) = \{t_j : x_j > 0\}$. t-invariants correspond to some subprocesses which do not change the state of the system [12, 18]. An analysis of t-invariants is based on similarities among them and may lead to discoveries of some interactions between subprocesses which occur in the modeled system. The net should be covered by t-invariants what means that every transition is an element of a support of at least one t-invariant. When the number of t-invariants is high the searching for similarities among them is usually done using clustering methods. According to them similar t-invariants are grouped into set called t-clusters. Further analysis is then performed within the obtained clusters.

Moreover, every t-cluster usually corresponds to some functional module of the biological system, so its biological meaning should be determined. In addition, also transitions can be grouped into structures called Maximal Common Transitions sets (MCT sets), which correspond to functional blocks. Such a set contains transitions being elements of supports of exactly the same t-invariants [7, 8, 12, 28].

The general working scheme is shown in Figure 2.

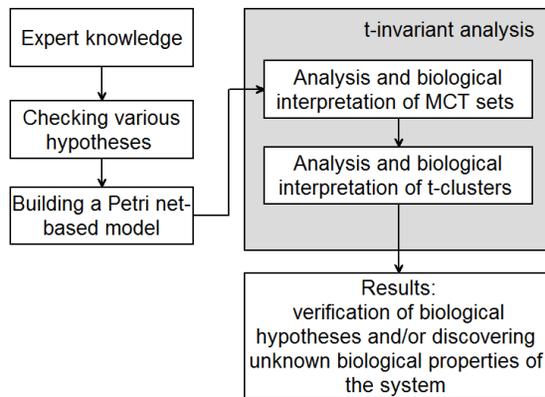


Figure 2: A scheme of work: through expert knowledge and checking various hypotheses to modeling and analysis of biological processes using Petri nets.

3. Informal description of the modeled process

3.1. Endothelial dysfunction

Initiation of endothelial dysfunction is related to presence of various factors, which can have direct and indirect influence on the studied phenomenon. Some of them are high blood pressure, increased glucose level in serum, high concentration of low-density lipoprotein (LDL), cytokines, toxins, etc. The changes in their level/concentration may be influenced by many agents, and cigarette smoking is one of them. The proposed model includes selected aspects of the impact of tobacco smoke [15, 19]. Indirect endothelial damage caused by cigarette smoking is understood as, inter alia: decreased quantity of tetrahydrobiopterin (BH₄ - which is a naturally occurring cofactor for the nitric oxide (NO) synthesis by the nitric oxide synthases (NOS)), decreased quantity of LDL, ROS, free radical and metals.

Damaged endothelium (caused by both direct and indirect mechanisms) expresses vascular cell adhesion protein 1 (VCAM-1) and chemokines. These chemotactic cytokines allow for monocytes adhesion and in consequence they lead to diapedesis. Monocytes movement out of the vessel wall and transformation into macrophages is typical inflammatory response [21]. One of the main function of macrophages is getting rid of damaged cells. Transformation of macrophages into foam cells is a consequence of up-

take of oxLDL through scavenger receptors. In normal condition macrophages after this activity return to blood circulation, but in case when LDL serum concentration is high, macrophages are still present in inflammatory area [27]. The modeled process includes also much more sophisticated role of these white blood cells. Macrophages secrete cytokines, what leads to VCAM-1 synthesis on the endothelial cell surface and vascular smooth muscle cells (VSMC) proliferation [21]. The presence of cytokines induces respiratory burst what results in creation of free radicals, accurately formation of superoxide anion ($O_2^{\bullet-}$) [2, 21], which participates in oxLDL formation [23]. Harmful oxLDL are absorbed by macrophages via scavenger receptors and form foam cells [10, 2], being a crucial compound of atherosclerotic plaque.

3.2. Nitric oxide

Nitric oxide is produced by a group of enzymes called nitric oxide synthases and is involved in many cellular processes, some of them are: regulation of blood pressure, angiogenesis, apoptosis, platelets aggregation, LDL oxidation etc. NO plays important role in CVD, where biochemical and molecular mechanisms which regulate NO bioavailability have significant function. These mechanisms can change NO concentration, causing opposite functions of NO: positive and negative effect. Impairment of NO availability favor CVDs. In this case it promotes atherosclerosis [19, 21, 25]. Cigarette smoke has been found to be another source that leads to decreasing of NO bioavailability [15, 19, 21].

3.3. The effect of cigarette smoking

Cigarette smoking is one of the major risk factors playing an important role in many harmful processes. This part of the model includes selected aspects of its involvement. Substances which are released, as cadmium, metals, free radicals, nicotine, polycyclic aromatic hydrocarbons, aldehydes etc. are engaged in endothelial dysfunction, inflammation, respiratory burst, LDL oxidation and have an important influence on atherosclerosis development. Increased quantity of ROS, free radicals and metals stimulate LDL oxidation, which results in reduction of NO bioavailability [15, 19, 21]. Nicotine, catecholamine, CO_2 lead to increase of blood viscosity, which promotes formation of thrombus. Other elements such as increase and activation of platelets and also increased quantity of fibrinogen, tissue factor, plasminogen activator inhibitor-1, decreased quantity of plasminogen activator and other remaining elements are additional contributors that may stimulate thrombus. Thromboxane A2 and prostacyclin I2 (which also are secreted by cigarette smoke) block the coronary blood vessels. These described processes have more direct influence on atherosclerosis. On the other hand, there are mechanisms which have indirect influence, like stimulating inflammation and endothelium dysfunction. Cigarette smoke secretes macrophages and polycyclic aromatic hydrocarbons, which stimulates chemokines - these molecules belong to inflammatory environment. Moreover, endothelial dysfunction stimulated by cigarette smoking is associated with modification of lipids profile (decreased quantity of HDL and increased quantity of LDL and triglycerides). As is widely known, high LDL level is one of the causes of endothelial damage. Therefore,

it can be noticed, that cigarette smoke is an additional source which stimulates inflammation and aggravates atherosclerotic plaque formation.

4. The model

In this section mathematical model of the analyzed biological system is proposed. The model has been expressed in the language of Petri nets theory [20, 22, 24]. It has been built using Snoopy software [13], and analyzed using R scripts.

The proposed model is shown in Fig. 3. For better transparency of the model it has been divided into two subnets. The first, upper frame presents mechanisms, reactions and molecules which are present in human organism. These processes are associated with response on endothelial dysfunction, LDL oxidation, respiratory burst, NO synthesis and in consequence they may lead to development of atherosclerosis. The second, lower frame presents mechanism, reactions and harmful molecules which are associated with cigarette smoke. These molecules are engaged in all inappropriate processes like stimulation of endothelial damage, oxidative stress promotion and LDL oxidation, reduction of nitric oxide synthesis, also increasing and decreasing quantity of various molecules - which leads to creation of thrombus, increasing blood viscous and development of atherosclerosis. Table 1 includes the names of places which correspond to the biological or chemical components and Table 2 contains the names of transitions which correspond to the reactions and interactions between molecules (represented by places).

The main complications in the modeling and analysis are caused by inhibition reactions (mechanisms decreasing quantity of some elements/molecules). This process has been modeled using Snoopy [13], which allows to use inhibitor arcs, but they are not represented in the incidence matrix and in consequence they are not taken into account in the cluster analysis. The problem of analyzing Petri nets with inhibitor arcs is still open, therefore inhibition reaction was modeled using names of transitions "inhibition" in the presented model. It might be a little unintuitive, so a two examples in Figure 4 have been presented:

- Cofactor BH_4 inhibition:
 - In normal conditions: cofactor BH_4 increases affinity eNOS to L-arginine, and it is important for NO synthesis (Figure 4 part a)).
 - In case of BH_4 inhibition: a decreased quantity of BH_4 leads to inhibition of NO synthesis (in the model this process is related with place "less NO" (Figure 4 part b)).
- eNOS inhibition:
 - In normal conditions: eNOS (nitric oxide synthase) is important molecule, which is engaged in nitric oxide synthesis (Figure 4 part c)).

- In case of eNOS inhibition: indirect inhibition of eNOS is caused by decreasing of BH₄ and direct inhibition is caused by asymmetric dimethylarginine (ADMA) (Figure 4 part d).

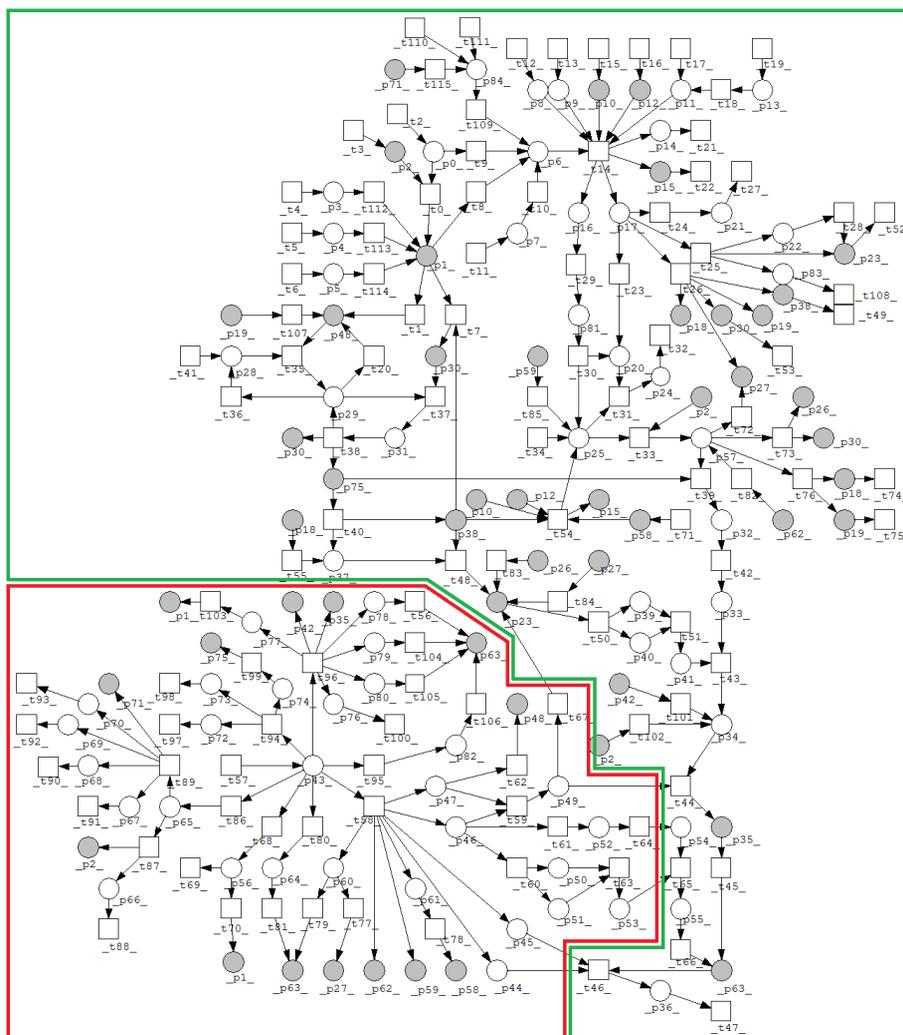


Figure 3: The proposed model is divided into two subnets: the upper frame presents mechanisms and molecules which are present in human organism and the lower frame presents mechanisms and harmful molecules which are associated with cigarette smoke.

Other simplifications included in the model resulted from lack of precise data. Therefore, the model includes transitions with names: “increase”, “decrease”, “less NO”, “NO”. The place “less NO” is related with inhibition of nitric oxide synthesis, and the place “NO” is related with correct NO synthesis. The model also includes additional

arcs which represent reversible reactions, for example: creation of active monocytes. Chemokines and inactive monocytes are involved in creation of active monocytes which can form a complex with VCAM-1 protein. In this process chemokines and VCAM-1 protein can be reused.

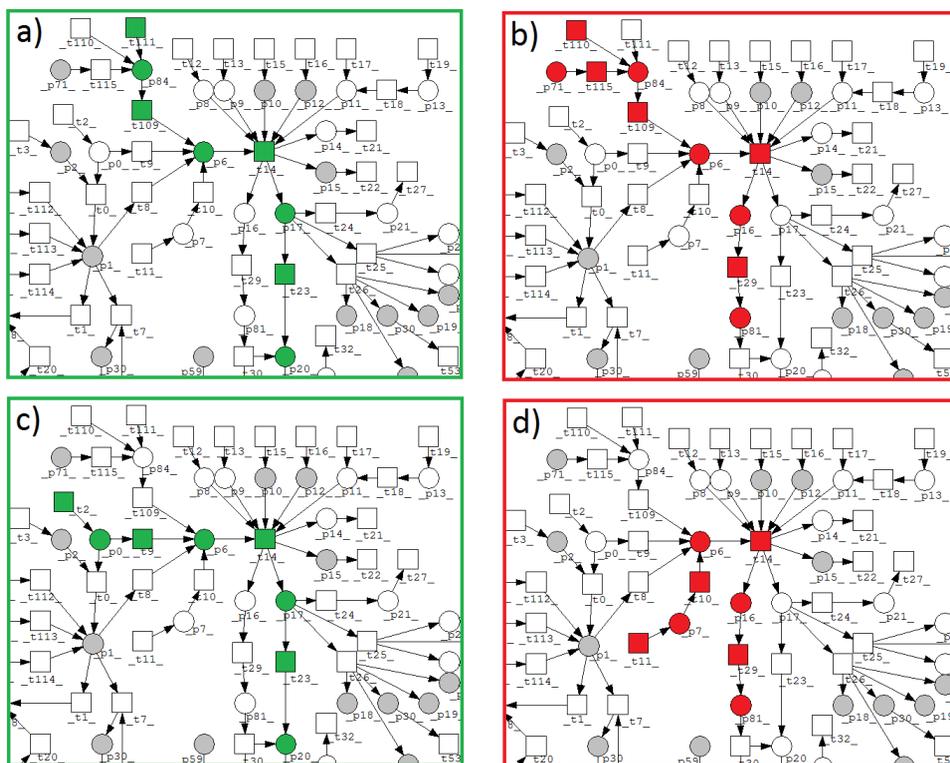


Figure 4: Simplifications caused by inhibition reactions: the left frames a) and c) correspond to situations in normal conditions and the right frames b) and d) correspond to inhibition reactions.

Table 1: List of places. The column “No.” includes the place numbers and the column “Biological meaning” contains the names of biological or chemical components.

No.	Biological meaning	No.	Biological meaning
p_0	healthy endothelium	p_{43}	cigarette smoke
p_1	damaged endothelium	p_{44}	thromboxane A2
p_2	LDL	p_{45}	prostacyclin I2
p_3	high blood pressure	p_{46}	nicotine
p_4	toxins	p_{47}	polycyclic aromatic hydrocarbon
p_5	other factors	p_{48}	chemokines
p_6	eNOS	p_{49}	matrix metalloproteinases (MMPs)
p_7	asymmetric dimethylarginine (ADMA)	p_{50}	adrenaline

No.	Biological meaning	No.	Biological meaning
<i>p</i> ₈	iNOS	<i>p</i> ₅₁	noradrenaline
<i>p</i> ₉	nNOS	<i>p</i> ₅₂	catecholamine
<i>p</i> ₁₀	NADPH	<i>p</i> ₅₃	CO
<i>p</i> ₁₁	L-arginine	<i>p</i> ₅₄	alpha receptors
<i>p</i> ₁₂	O ₂	<i>p</i> ₅₅	high blood viscosity
<i>p</i> ₁₃	L-NMMA	<i>p</i> ₅₆	impairment of fibromuscular dysplasia (FMD)
<i>p</i> ₁₄	citrulline	<i>p</i> ₅₇	oxLDL
<i>p</i> ₁₅	NADP	<i>p</i> ₅₈	ROS
<i>p</i> ₁₆	NO if endothelium is damaged	<i>p</i> ₅₉	free radical
<i>p</i> ₁₇	NO if endothelium is healthy	<i>p</i> ₆₀	cadmium
<i>p</i> ₁₈	macrophage colony stimulating factor (MCSF)	<i>p</i> ₆₁	aldehydes
<i>p</i> ₁₉	monocyte chemotactic protein 1 (MCP-1)	<i>p</i> ₆₂	metals
<i>p</i> ₂₀	NO	<i>p</i> ₆₃	thrombus
<i>p</i> ₂₁	blood pressure	<i>p</i> ₆₄	platelets
<i>p</i> ₂₂	platelet aggregation	<i>p</i> ₆₅	lipid profile
<i>p</i> ₂₃	amplified cell caused by proliferation of VSMC	<i>p</i> ₆₆	triglycerides
<i>p</i> ₂₄	peroxynitrite	<i>p</i> ₆₇	HDL
<i>p</i> ₂₅	superoxide anion	<i>p</i> ₆₈	FMD
<i>p</i> ₂₆	intercellular adhesion molecule 1 (ICAM-1)	<i>p</i> ₆₉	apolipoprotein A1
<i>p</i> ₂₇	adhesion molecules	<i>p</i> ₇₀	selenium
<i>p</i> ₂₈	inactive monocyte	<i>p</i> ₇₁	less quantity of BH4
<i>p</i> ₂₉	active monocyte	<i>p</i> ₇₂	lymphocytes
<i>p</i> ₃₀	vascular cell adhesion molecule 1 (VCAM-1)	<i>p</i> ₇₃	neutrophils
<i>p</i> ₃₁	complex VCAM-1 and monocyte	<i>p</i> ₇₄	auxiliary place
<i>p</i> ₃₂	foam cell	<i>p</i> ₇₅	macrophage
<i>p</i> ₃₃	necrosis core	<i>p</i> ₇₆	oxidative stress markers
<i>p</i> ₃₄	plaque	<i>p</i> ₇₇	endothelin 1 (ET-1)
<i>p</i> ₃₅	tissue factor	<i>p</i> ₇₈	plasminogen activator inhibitor 1
<i>p</i> ₃₆	atherosclerosis	<i>p</i> ₇₉	fibrinogen
<i>p</i> ₃₇	growth factor	<i>p</i> ₈₀	von Willebrand factor
<i>p</i> ₃₈	cytokines	<i>p</i> ₈₁	less NO
<i>p</i> ₃₉	matrix glycoproteins	<i>p</i> ₈₂	plasminogen activator
<i>p</i> ₄₀	collagen	<i>p</i> ₈₃	other mechanisms
<i>p</i> ₄₁	fibrous cap	<i>p</i> ₈₄	BH4 cofactor of eNOS
<i>p</i> ₄₂	white blood cells		

Table 2: List of transitions. The column “No.” includes the transition numbers while the column “Biological meaning” contains their biological functions.

No.	Biological meaning	No.	Biological meaning
t_0	damage caused by LDL	t_{58}	secretion caused by cigarette smoke
t_1	secretion caused by damaged endothelium	t_{59}	activation caused by nicotine and polycyclic aromatic hydrocarbon
t_2	auxiliary transition	t_{60}	release caused by nicotine
t_3	auxiliary transition	t_{61}	induction caused by nicotine
t_4	auxiliary transition	t_{62}	induction caused by polycyclic aromatic hydrocarbon
t_5	auxiliary transition	t_{63}	decrease O ₂ increase CO
t_6	auxiliary transition	t_{64}	activation caused by catecholamine
t_7	expression caused by damaged endothelium	t_{65}	increase the viscosity of blood
t_8	inhibition of eNOS caused by damage endothelium	t_{66}	stimulation caused by highly viscous blood
t_9	secretion caused by healthy endothelium	t_{67}	stimulation caused by MMPs
t_{10}	inhibition of eNOS caused by ADMA	t_{68}	direct damage caused by cigarette smoke
t_{11}	auxiliary transition	t_{69}	remodeling tissue caused by impairment of FMD
t_{12}	auxiliary transition	t_{70}	damage caused by impairment of FMD
t_{13}	auxiliary transition	t_{71}	auxiliary transition
t_{14}	synthesis of NO	t_{72}	expression adhesion molecules caused by oxLDL
t_{15}	auxiliary transition	t_{73}	expression caused by oxLDL
t_{16}	auxiliary transition	t_{74}	auxiliary transition
t_{17}	auxiliary transition	t_{75}	auxiliary transition
t_{18}	inhibition caused by LNMMA	t_{76}	stimulation caused by oxLDL
t_{19}	auxiliary transition	t_{77}	expression caused by cadmium
t_{20}	auxiliary transition	t_{78}	increase caused by aldehydes
t_{21}	auxiliary transition	t_{79}	role in endothelial cell death
t_{22}	auxiliary transition	t_{80}	increase and activation caused by cigarette smoke
t_{23}	high quantity of NO	t_{81}	prothrombotic procoagulative states caused by increase of platelets
t_{24}	regulation caused by NO if healthy endothelium	t_{82}	stimulation of oxidation caused by metals
t_{25}	inhibition caused by NO if healthy endothelium	t_{83}	stimulation caused by ICAM-1
t_{26}	inhibition adhesion molecule	t_{84}	stimulation caused by adhesion molecules
t_{27}	auxiliary transition	t_{85}	stimulate respiratory burst caused by free radical

No.	Biological meaning	No.	Biological meaning
t_{28}	auxiliary transition	t_{86}	modification caused by cigarette smoke
t_{29}	low quantity of NO	t_{87}	increase caused by modification of lipid profile
t_{30}	high quantity of superoxide anion radical	t_{88}	auxiliary transition
t_{31}	reduction	t_{89}	decrease caused by modification of lipid profile
t_{32}	inhibition of oxLDL	t_{90}	tissue remodeling caused by FMD
t_{33}	oxidation	t_{91}	auxiliary transition
t_{34}	auxiliary transition	t_{92}	auxiliary transition
t_{35}	auxiliary transition	t_{93}	auxiliary transition
t_{36}	auxiliary transition	t_{94}	increase caused by development environment inflammatory
t_{37}	adhesion of monocyte	t_{95}	decrease caused by cigarette smoke
t_{38}	transformation	t_{96}	increase caused by cigarette smoke
t_{39}	uptake	t_{97}	auxiliary transition
t_{40}	secretion caused by macrophage	t_{98}	auxiliary transition
t_{41}	auxiliary transition	t_{99}	stimulation caused by inflammatory environment
t_{42}	destruction foam cell	t_{100}	auxiliary transition
t_{43}	formation of plaque composed of necrosis core and fibrous cap	t_{101}	formation caused by white blood cells
t_{44}	plaque rupture	t_{102}	creation caused by LDL
t_{45}	activation of blood platelets	t_{103}	damage caused by endothelin 1
t_{46}	block of the coronary blood vessels	t_{104}	prothrombotic states caused by fibrinogen
t_{47}	auxiliary transition	t_{105}	prothrombotic states caused by von Willebrand factor
t_{48}	proliferation caused by growth factor and cytokines	t_{106}	prothrombotic states caused by plasminogen activator
t_{49}	auxiliary transition	t_{107}	stimulate chemokines
t_{50}	secretion caused by amplified cells	t_{108}	auxiliary transition
t_{51}	formation of fibrous cap	t_{109}	increases the affinity of eNOS to L-arginine
t_{52}	auxiliary transition	t_{110}	inhibition of BH4 cofactor
t_{53}	auxiliary transition	t_{111}	auxiliary transition 120
t_{54}	respiratory burst	t_{112}	damage caused by high blood pressure
t_{55}	simulation growth factor	t_{113}	damage caused by toxins
t_{56}	prothrombotic states caused by plasminogen activator inhibitor 1	t_{114}	damage caused by other factor
t_{57}	auxiliary transition	t_{115}	inhibition of BH4 caused by smoking

5. Analysis

5.1. MCT sets

The proposed Petri net includes 84 places, 115 transitions, 21 non-trivial (i.e., containing more than one transition) MCT sets and is covered by 5344 t-invariants. For all of the MCT sets its biological meaning has been determined and described in Table 3.

Table 3: List of non-trivial MCT sets. The column “MCT set” contains names of these sets, while the column “Transitions” includes names of transitions contained in a given MCT set. The column “Biological meaning” includes biological interpretation (functions and mechanisms) of MCT sets.

MCT set	Transitions	Biological meaning
m_1	$t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{21}, t_{22}, t_{29}, t_{30}, t_{31}, t_{32}$	Nitric oxide synthesis without L-arginine leads to reduction of NO (L-arginine is necessary to proper NO synthesis). This MCT set includes two opposite processes: promotion of LDL oxidation and inhibition of LDL oxidation by superoxide anion reduction to peroxynitrite. This second process acts despite inhibition of NO synthesis.
m_2	$t_{56}, t_{96}, t_{100}, t_{101}, t_{103}, t_{104}, t_{105}$	Increased of various elements caused by cigarette smoke. These elements have influence on endothelial damage and development of prothrombotic states (which stimulate thrombus formation and promote atherosclerosis).
m_3	$t_{60}, t_{61}, t_{62}, t_{63}, t_{64}, t_{65}, t_{66}$	Nicotine induces catecholamine and releases adrenaline and norepinephrine, which lead to increased of the blood viscosity. This MCT set includes stimulation of inflammatory response which is caused by polycyclic aromatic hydrocarbon through stimulation of chemokines.
m_4	$t_{46}, t_{47}, t_{58}, t_{78}, t_{82}, t_{85}$	Secretion of aldehydes (it leads to increased quantity of ROS, which stimulate respiratory burst), metals and free radicals (they have influence on LDL oxidation and respiratory burst), thromboxane A2 and prostacyclin I2 (they promote atherosclerosis by narrowing of the coronary blood vessels).
m_5	$t_{89}, t_{90}, t_{91}, t_{92}, t_{93}, t_{115}$	Modification of lipid profile (caused by cigarette smoke) stimulate decreased quantity of HDL, FMD, apolipoprotein A1, selenium and BH4 (decreased quantity of BH4 leads to inhibition of NO synthesis).
m_6	$t_{39}, t_{42}, t_{43}, t_{50}, t_{51}$	Response of damaged endothelium in which secretion of macrophages is engaged. Macrophages uptake of oxLDL causes formation of foam cells and also destruction (if oxLDL level is high). When foam cell dies necrotic core is released and stimulates plaque creation (development of atherosclerosis).
m_7	$t_{94}, t_{97}, t_{98}, t_{99}$	Development of inflammation caused by cigarette smoke. It stimulates macrophages to increase of cytokines secretion.
m_8	t_{25}, t_{28}, t_{108}	Inhibition of harmful mechanisms (platelet aggregation, proliferation of VSMC, and other), when NO synthesis is correct.
m_9	t_4, t_{112}	Endothelial dysfunction caused by high blood pressure.
m_{10}	t_5, t_{113}	Endothelial dysfunction caused by toxins.
m_{11}	t_6, t_{114}	Endothelial dysfunction caused by other factors.

MCT set	Transitions	Biological meaning
m_{12}	t_{10}, t_{11}	Inhibition of NO synthesis caused by ADMA, which is eNOS inhibitor.
m_{13}	t_{18}, t_{19}	Inhibition of NO synthesis caused by L-NMMA, which is L-arginine inhibitor.
m_{14}	t_{20}, t_{41}	Response of damaged endothelium and transformation to macrophages. Precisely, damaged endothelium secretes chemokines, what induces monocytes.
m_{15}	t_{24}, t_{27}	Regulation of blood pressure in case of proper NO synthesis.
m_{16}	t_{37}, t_{38}	The formation of VCAM-1 and monocyte complex, which leads to macrophages transformation.
m_{17}	t_{44}, t_{45}	Development of atherosclerosis is promoted by plaque rupture and formation of thrombus.
m_{18}	t_{73}, t_{83}	Proliferation of VSMC is stimulated by LDL oxidation.
m_{19}	t_{80}, t_{81}	Increase and activation of platelets (caused by cigarette smoke) promote atherosclerosis.
m_{20}	t_{87}, t_{88}	Increased quantity of LDL and triglycerides caused by modification of lipid profile.
m_{21}	t_{95}, t_{106}	Decreased quantity of plasminogen activator caused by cigarette smoke stimulates prothrombotic state and thrombus formation.

5.2. Clusters analysis

The analysis of the presented model is based mainly on t-invariants, which correspond to some subprocesses occurring in the modeled system. The number of t-invariants is dependent on the nature of the analyzed process. t-invariants can be calculated using many freely available tools, e.g., Charlie [14] or MonaLisa [6]. In our analysis t-invariants have been grouped into t-clusters and the biological meaning has been assigned to each such a cluster. Subprocesses corresponding to t-invariants being elements of the same t-cluster can be related in some way and influence each other. For this reason the analysis of clusters can be crucial for confirmation of some hypotheses and discovering of unknown properties of the modeled system. In Table 4 biological interpretation of 18 t-clusters has been shown. The clusters have been determined using average linkage method and Pearson similarity measure. The Mean Split Silhouette (MSS) is an index which has been used to identify the best clustering in the set of clusterings obtained using various methods (cf. [7, 8]). MSS evaluates a fit of each t-invariant to its cluster and an average quality of a given clustering [17, 26]. Calinski-Harabasz coefficient has been used to find the best number of clusters [4]. This coefficient has been calculated for clusterings whose number of clusters was in the range from 2 to 20 and the optimal number of clusters has been indicated by its highest value.

The obtained results confirm that oxidative stress (respiratory burst) and inflammatory process are the main paths that are related with endothelial dysfunction. These processes lead to increased secretion of ROS and free radicals, which are involved in excessive LDL oxidation. On the other hand, LDL oxidation influence on harmful mechanisms which are associated with decreasing NO bioavailability. These processes promote narrowing of the coronary blood vessels, stimulate increasing blood viscous and creation of

thrombus which leads to development of atherosclerosis. Cluster analysis showed also how big influence on the development of atherosclerosis has cigarette smoking. Both direct and indirect endothelial damage have influence on inflammation through increased quantity of macrophages and cytokines. Disorders in prothrombotic states have influence on creating thrombus and increasing blood viscosity which more directly stimulate atherosclerotic plaque formation.

Clusters analysis revealed how complex and wide is the interaction net of inflammation process and oxidative stress. Both of them are engaged in almost all mechanisms which are described in Table 4. These results reveal which of the processes are crucial for the modeled system.

Table 4: List of t-clusters. The column “t-cluster” contains names of t-clusters, while the column “Biological meaning” includes their biological interpretations.

t-cluster	Biological meaning	t-cluster	Biological meaning
<i>c</i> ₁	Inflammatory response of damaged endothelium: monocytes adhesion, which are engaged in the diapedesis (monocytes movement out of the vessel wall and transformation to macrophages).	<i>c</i> ₁₀	Endothelial dysfunction caused by high LDL level. This LDL can be oxidized, then oxLDL stimulates harmful proliferation mechanism and adhesion of molecules. This cluster includes also inhibition of molecules adhesion and proliferation, when NO synthesis is correct.
<i>c</i> ₂	Tissue remodeling caused by impairment of fibromuscular dysplasia (FMD) - it has direct influence on endothelial dysfunction.	<i>t</i> ₁₁	Modification of lipid profile caused by cigarette smoke has an influence on increased quantity of LDL. This LDL can be oxidized, then oxLDL stimulate harmful proliferation mechanism. This cluster includes also inhibition of molecules adhesion and proliferation, when NO synthesis is correct.
<i>c</i> ₃	The formation of VCAM-1 and monocyte complex, which leads to macrophage transformation. Macrophages secrete cytokines and growth factor, which lead to proliferation of VSMC.	<i>c</i> ₁₂	Endothelial dysfunction caused by high LDL level. This LDL can be oxidized, then oxLDL stimulate harmful proliferation mechanism. This cluster includes also inhibition of proliferation and molecules adhesion, when NO synthesis is correct.
<i>c</i> ₄	Cigarette smoke influence on development inflammation by stimulating macrophages. This cluster includes also harmful activity of macrophages, which leads to proliferation.	<i>c</i> ₁₃	Endothelial dysfunction (caused by high LDL level) stimulates inflammatory response, which is related with LDL oxidation (oxLDL stimulate harmful mechanisms).
<i>c</i> ₅	Endothelial dysfunction caused by high blood pressure induces inflammatory response: secretion of chemokines and adhesion of monocytes.	<i>c</i> ₁₄	Endothelial dysfunction (caused by modification of lipid profile in smokers) stimulates inflammatory response, which is related with LDL oxidation (oxLDL stimulates harmful mechanisms).

t-cluster	Biological meaning	t-cluster	Biological meaning
<i>c</i> ₆	Endothelial dysfunction caused by toxins induce inflammatory response: secretion of chemokines and adhesion of monocytes.	<i>c</i> ₁₅	Endothelial dysfunction caused by various factors stimulates inflammatory response, this process is additionally stimulated by modification of lipid profile in smokers. These processes have influence on LDL oxidation and reduction of NO bioavailability (inhibition of NO synthesis caused by BH4 inhibition, eNOS inhibition by ADMA, L-arginine inhibition by L-NMMA). In other side this cluster includes NO that have vasoregulatory functions and can inhibit LDL oxidation.
<i>c</i> ₇	Endothelial dysfunction caused by other factor (for example high glucose level in diabetes) induce inflammatory response: secretion of chemokines and adhesion of monocytes.	<i>c</i> ₁₆	This cluster includes almost all mechanisms. Distinctive processes: respiratory burst and LDL oxidation. Both are additionally stimulated by cigarette smoke and aggravate atherosclerosis.
<i>c</i> ₈	Endothelial dysfunction caused by impairment of fibromuscular dysplasia (FMD) - it has direct influence on endothelial dysfunction and induce inflammatory response: secretion of chemokines and adhesion of monocytes.	<i>c</i> ₁₇	This cluster includes almost all mechanisms. Distinctive processes: NO synthesis and inflammation, which are additionally stimulated by cigarette smoke.
<i>c</i> ₉	Endothelial dysfunction caused by modification of lipid profile induce inflammatory response: secretion of chemokines and adhesion of monocytes.	<i>c</i> ₁₈	This cluster includes almost all mechanisms. Distinctive processes: NO synthesis and inflammation, which are additionally stimulated by cigarette smoke.

5.3. Conclusions

Systems approach is used for analysis of biological processes as complex systems. This approach forces the detailed knowledge of the modeling process but it can be difficult due to a lack of data and contradictory information. On the other hand, only in this way it is possible to know the nature of these processes, organize knowledge and even discover new dependencies.

Cluster analysis allows the discovery of new and interesting biological meanings, and also confirms the existing dependencies. Clusters are sets of t-invariants, therefore the number of t-invariants is important. When this number is small, all the interactions and dependencies can be considered and the biological interpretations can be determined. In the case of the presented model the number of t-invariants is very high (5344), and some clusters contain over 1000 and 2000 t-invariants, which can be problematic during the analysis. For this reason the descriptions of clusters 15-18 in Table 4 are more general, but they consist distinctive processes. On the other hand, by focusing on the smaller clusters dependencies can be determined in greater detail.

Clusters which contain t-invariants associated with endothelial dysfunctions (caused by high blood pressure, toxins, high glucose level in diabetes), always include a response of endothelial damage (stimulating chemokines and monocytes adhesion). These endothelial dysfunctions can be caused regardless of cigarette smoking. However, subsequent clusters reveal a direct influence of smoking on endothelial damage which are caused by impairment of fibromuscular dysplasia or modification of lipid profile. Clusters which include t-invariants associated with endothelial dysfunction caused by high LDL level (which can be additionally stimulated by cigarette smoke) beside inflammatory response also include oxidation process. If LDL is oxidized, it results in reduction of NO bioavailability. Clusters which contain t-invariants associated with NO synthesis show interesting interactions. NO can act positively and negatively, which is dependent on its concentration. Despite of mechanisms affecting the decrease of NO and promoting of LDL oxidation, NO can act positively and can inhibit harmful processes. The analysis of the clusters that include t-invariants associated with dual functions of NO showed that it can inhibit proliferation. In two of the three clusters containing NO it was revealed that it can inhibit adhesion of molecules and furthermore it can stimulate LDL oxidation.

Due to a lack of accurate data about the concentration of NO and a lack of information about the time of all of the studied reactions it is not possible to determine which mechanisms occur more frequently or faster. Despite this fact, this study confirmed that NO plays an important role in development of atherosclerosis.

The obtained results confirm that oxidative stress and inflammation are closely associated with endothelial dysfunction and development of atherosclerosis and both influence each other. These processes result in excessive production of ROS and free radicals, which are engaged in LDL oxidation and reduction of NO bioavailability. Additional source which stimulates these processes is cigarette smoking. The systems approach that has been used in the study allowed for a better understanding of the analyzed process and for distinguishing important signaling pathways. Participation of cigarette smoking in the atherosclerosis appears to be indisputable. These results are consistent with the literature.

References

- [1] P. BALDAN, N. COCCO, A. MARIN and M. SIMEONI: Petri nets for modelling metabolic pathways: a survey. *Natural Computing*, **9**(4), (2010), 955-989.
- [2] E.BIRBEN, U.M.SAHINER, C.SACKESSEN, S.ERZURUM and O.KALAYCI: Oxidative stress and antioxidant defense. *World Allergy Organ J.*, **5**(1), (2012), 9-19.
- [3] K. CHMIELEWSKA, D. FORMANOWICZ and P. FORMANOWICZ: Modelowanie i analiza uszkodzeń śródbłonna za pomocą sieci Petriego. In: A. Świerniak, J. Krystek (Eds.): *Automatyzacja Procesów Dyskretnych. Teoria i Zastosowania*, Vol. II, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, (2016), 41-56, (in Polish).

- [4] T. CALINSKI and J. HARABASZ: A dendrite method for cluster analysis. *Communications in Statistics*, **3**(1), (1974), 1-27.
- [5] R. DAVID and H. ALLA: Discrete, Continuous, and Hybrid Petri Nets. Springer Science & Business Media, 2005.
- [6] J. EINLOFT, J. ACKERMANN, J. NÖTHEN and I. KOCH: MonaLisa–visualization and analysis of functional modules in biochemical networks. *Bioinformatics*, **29**(11), (2013), 1469-70.
- [7] D. FORMANOWICZ, A. KOZAK, T. GŁOWACKI, M. RADOM and P. FORMANOWICZ: Hemojuvelin - hepcidin axis modeled and analyzed using Petri nets. *J. of Biomedical Informatics*, **46**(6), (2013), 1030-1043.
- [8] D. FORMANOWICZ, M. RADOM, P. ZAWIERUCHA and P. FORMANOWICZ: Petri net-based approach to modeling and analysis of selected aspects of the molecular regulation of angiogenesis. *PLoS ONE*, **12**(3), (2017), e0173020.
- [9] P. FORMANOWICZ: On the border between biology, mathematics and computer science. *BioTechnologia*, **92**(3), (2011), 217-220.
- [10] J. FROSTEGÅRD: Immunity, atherosclerosis and cardiovascular disease. *BMC Medicine*, **11**(1), (2013), 117.
- [11] D. GRASSI, G. DESIDERI, L. FERRI, A. AGGIO, S. TIBERTI and C. FERRI: Oxidative stress and endothelial dysfunction: say NO to cigarette smoking. *Current Pharmaceutical Design*, **16**(23), (2010), 2539-2550.
- [12] E. GRAFAHREND-BELAU, F. SCHREIBER, M. HEINER, A. SACKMANN, B.H. JUNKER, S. GRUNWALD, A. SPEER, K. WINDER and I. KOCH: Modularization of biochemical networks based on classification of Petri net t-invariants. *BMC Bioinformatics*, **9**(90), (2008).
- [13] M. HEINER, M. HERAJY, F. LIU, C. ROHR and M. SCHWARICK: Snoopy - a unifying Petri net tool. *Lecture Notes in Computer Science*, **7347**, (2012), 398-407.
- [14] M. HEINER, M. SCHWARICK and J. WEGENER: Charlie – an extensible Petri net analysis tool. In *Proc. PETRI NETS 2015*, Brussels, Springer, LNCS. **9115**, (2015), 200-211.
- [15] M. KITAMI and M.K. ALI: Tobacco, metabolic and inflammatory pathways, and CVD risk. *Global Heart*, **7**(2), (2012), 121-128.
- [16] E. KLIPP, W. LIEBERMEISTER, C. WIERLING, A. KOWALD, H. LEHRACH and R. HERWIG: Systems biology. A textbook. Wiley-VCH, Weinheim, 2009.

- [17] L. KAUFMAN and P.J. ROUSSEEUW: Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley and Sons, 1990.
- [18] I. KOCH, W. REISIG and F. SCHREIBER: (Ed.) Modeling in systems biology. The Petri net approach. Springer, London, 2011.
- [19] B. MESSNER and D. BERNHARD: Smoking and cardiovascular disease: mechanisms of endothelial dysfunction and early atherogenesis. *Arteriosclerosis, Thrombosis, and Vascular Biology*, **34**(3), (2014), 509-515.
- [20] T. MURATA: Petri nets: Properties, analysis and applications. *Proc. of the IEEE*, **77**(4), (1989), 541-580.
- [21] K.M. NASEEM: The role of nitric oxide in cardiovascular diseases. *Molecular Aspects of Medicine*, **26**(1-2), (2005), 33-65.
- [22] C.A. PETRI: Communication with Automata (in German). Schriften des Instituts für Instrumentelle Mathematik, Bonn, 1962.
- [23] S. PARTHASARATHY, A. RAGHAVAMENON, M.O. GARELNAB and N. SANTANAM: Oxidized low-density lipoprotein. *Methods in Molecular Biology*, **610** (2010), 403-417.
- [24] W. REISING: Understanding Petri Nets. Modeling Techniques, Analysis Methods, Case Studies. Springer-Verlag, Berlin, Heidelberg, 2013.
- [25] M. ROSSELLI, P.J. KELLER and R.K. DUBEY: Role of nitric oxide in the biology, physiology and pathophysiology of reproduction. *Human Reproduction Update*, **4**(1), (1998), 3-24
- [26] P.J. ROUSSEEUW: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. of Computational and Applied Mathematics*, **20** (1987), 53-65.
- [27] B. SUN, B.B. BOYANOVSKY, M.A. CONNELLY, P. SHRIDAS, D.R. VAN DER WESTHUYZEN and N.R. WEBB: Distinct mechanisms for OxLDL uptake and cellular trafficking by class B scavenger receptors CD36 and SR-BI. *J. of Lipid Research*, **48**(12), (2007), 2560-2570.
- [28] A. SACKMANN, M. HEINER and I. KOCH: Application of Petri net based analysis techniques to signal transduction pathways. *BMC Bioinformatics*, **7**(482), (2006).
- [29] Z. SZALLASI, J. STELLING and V. PERIVAL: System modeling in cellular biology. From concepts to nuts and bolts. The MIT Press Cambridge, Massachusetts, 2006.
- [30] V.M. VICTOR, M. ROCHA, E. SOLÁ, C. BANULS, K. GARCIA-MALPARTIDA and A. HERNÁNDEZ-MIJARES: Oxidative stress, endothelial dysfunction and atherosclerosis. *Current Pharmaceutical Design*, **15**(26), (2009), 2988-3002.

Information management in passenger traffic supporting system design as a multi-criteria discrete optimization task

ADAM GALUSZKA, JOLANTA KRYPEK, ANDRZEJ SWIERNIAK, CARMEN LUNGOCI
and TOMASZ GRZEJSZCZAK

This paper presents a concept of an Integrated System of Supporting Information Management in Passenger Traffic (ISSIMPT). The novelty of the system is an integration of six modules: video monitoring, counting passenger flows, dynamic information for passengers, the central processing unit, surveillance center and vehicle diagnostics into one coherent solution. Basing on expert evaluations, we propose to present configuration design problem of the system as a multi-objectives discrete static optimization problem. Then, hybrid method joining properties of weighted sum and ϵ -constraint methods is applied to solve the problem. Solution selections based on hybrid method, using set of exemplary cases, are shown.

Key words: multi objective optimization, discrete static optimization, Pareto solutions, integrated systems.

1. Introduction

In recent years one can observe intensive development of monitoring and diagnostics systems enhanced by the development of modern technologies. It also applies to rail vehicles, which devotes more and more attention in scientific publications, standards and regulations due to the need to monitor individual modules of the vehicle as well as some elements of railway infrastructure [1, 9, 10]. We found similarities in this process in two countries: Poland and Romania.

Polish railway is after the liberalization process of the rail passenger market, where there has been a significant increase in the shares of companies outside the PKP Group, mainly local carriers from 1% to 48% [2]. An important factor in this process is the creation of market competition in the segment of interregional transport (dynamic develop-

A Galuszka (corresponding author), J. Krypek, A. Swierniak and T. Grzejszczak are with Silesian University of Technology, 44-100 Gliwice, Akademicka str. 16, Poland. E-mails: {adam.galuszka, jolanta.krypek, andrzej.swierniak, tomasz.grzejszczak}@polsl.pl. C. Lungoci is with Transilvania University, Politehnicii 1, Brasov 500019, Romania, e-mail: lungoci@unitbv.ro

Work partly financed from BK funds in 2017.

Received 22.12.2016. Revised 29.04.2017.

ment of the companies in the local passenger transport). This factor results in favorable conditions for the development of new features and facilities offered by the railway.

Romania, after the 1989 revolution, has one of the most used railway networks in Europe, but at the same time, lagging behind in maintaining its infrastructure. This, combined with the economic decline of the 1990s, lead to a period of relative decline of CFR group. Some less traveled routes, especially in rural areas, have been cancelled and the existing rolling stock has entered in a lack of repairs period. The situation continued until 1998 when the National Society of Romanian Railways was reorganized into four independently-funded institutions to increase the efficiency. Romanian railways situation has improved also due to a better economic situation of the country after 2000, which allowed the start of major investment projects. A liberalization process allowed the access to railway infrastructure to all licensed railway operators, which led to the establishment of private rail operators. Some railway lines, so-called "non-interoperable sections" were leased to private operators, passenger transport is done by them, using their own or leased rolling stock [11].

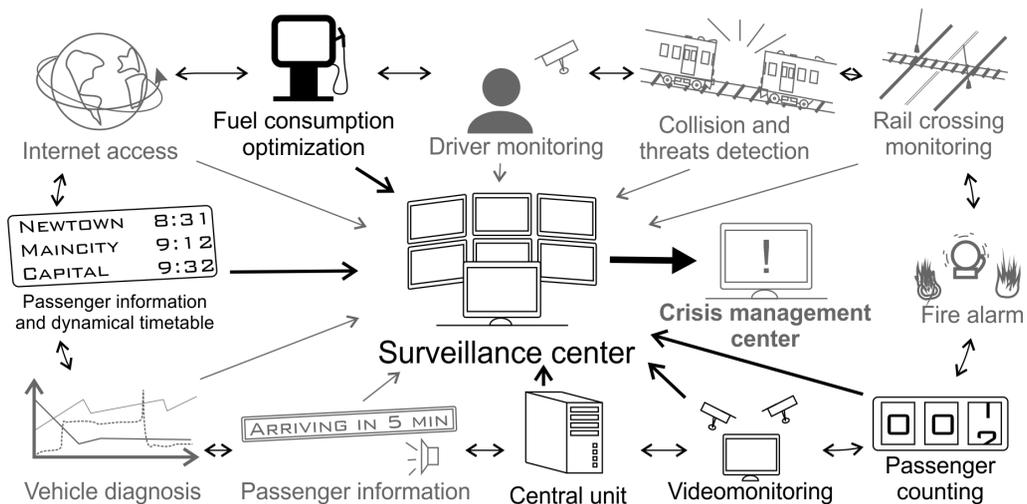


Figure 1: The structure of the telemetry system of Supporting Information Management in Passenger Traffic (ISSIMPT). The highlighted items are analyzed in the article

The designed system in its current state concerns the passengers safety aspect. The system is realizable thanks to installation of following modules: video monitoring, counting passenger flows, dynamic information for passengers, the central processing unit, surveillance center and vehicle diagnostics and integration with the Control Center according to diagram from Figure 1 [3]. Other modules configuration problem has been considered in our earlier works [6, 7]. To some extent, the described project of ISSIMPT affects the aspect of reducing the effect of the digital divide by providing access to Internet services implemented by the module wireless access to the Internet and Intranet in public transport.

As part of the work associated with the development of conceptual assumptions, for each module, a team of expert reviewed three different methods of its implementation. Each of the methods has been evaluated by the design team, with regard to three evaluation criteria:

1. Functionality and expandability;
2. Compliance with standards;
3. Costs.

The grading scale is following: 1 - very low, 2 - low, 3 - medium, 4 - high, 5 - very high. It is worth noticing that grading in third criterium: Costs, high rating (high grade in terms of costs) means low price.

Comparative analysis of possible construction of video monitoring module are presented in Table 1 and of passengers counting module - in Table 2. Other modules has been analysed in similar way and finally all results are summarized in Table 3.

Considerations are based on Table 3 containing the assessment of the suitability of all six designed modules that are mentioned above. The stated problem is the task of maximizing the objective functions vector consisting of 3 components:

$$\max \leftarrow F(x) = (f_1(x), f_2(x), f_3(x)), \quad (1)$$

where:

- $f_1(x)$ is a sum of grades in criterium 1, depending on method,
- $f_2(x)$ is a sum of grades in criterium 2, depending on method,
- $f_3(x)$ is a sum of grades in criterium 3, depending on method,
- x - space of feasible solutions.

The grades are in $\langle 1, 5 \rangle$ range, thus for 6 modules the range of sum values is $\langle 6, 30 \rangle$.

1.1. Example

Let variant number 1 of feasible solution will be the choice of method 1 concerning implementation of each of the build modules. Therefore on the basis of Table 4, the new table describing this solution can be build (Table 3). It can be observed that variant 1 has low functionality (low sum of grades in criterion 1), but is cheap (high sum of grades in criterion 3). The example shows that the criteria are contradictory, thus there exist no ideal solution of the problem, that is the indication of the best module build method.

Table 1: Comparative analysis of monitoring methods

Parameter	Method 1	Method 2	Method 3
	Machinist optical observation	Standard video monitoring within the vehicle and the railway station	Video monitoring module of ISSIMPT
Functionality and expansion possibility	Vehicle operator observation. The method is limited by the field of view of the door mirrors. Inability of entrance monitoring.	Video monitoring allowing camera observation from cameras installed on and within the vehicle. Ensuring the safety of structural and (partially) the safety of travelers	Image observation, integration of module with fire alarm. Observation of images from mirror, track, pantograph and entrance cameras. Ensuring the security of passengers and protection against attacks of aggression. Scalable system.
Score:	1	3	5
Compliance with the standards and guidelines	No compliance with the standards and the applicable guidelines for the monitoring.	Partial compliance with standards concerning the structural parts and security of passengers resulting from the technical conditions.	Compliance with the standards and current guidelines
Score:	1	4	5
General costs (installation, observation, ease of use, costs)	Low cost, low efficiency	Interference in the vehicle hull, involving the need to install additional devices. Expandability and scalability of the system. Average costs, high efficiency.	Interference in the vehicle hull, involving the need to install additional devices. Upgradeability and scalability of the system and integration by the newly developed modules. Open architecture. High implementation costs, very high efficiency of video monitoring system adapted to the dynamic situation within the vehicle.
Score:	4	3	2

Scores scale: 1- very low, 2-low, 3-medium, 4-high, 5-very high.

1.2. Contribution

In the paper a method that optimally solves the problem of Information Management in Passenger Traffic Supporting System design is proposed. This method joins classical multi-criteria optimization methods: weighted sum and ϵ -constraint methods and was proposed in earlier works to solve simpler problems [6, 7]. In section 2 the problem of modeling system configuration as a optimization problem is presented, in section 3 optimization results are presented and, finally results are concluded in section 4.

Table 2: Comparative analysis of passengers counting methods

Parameter	Method 1	Method 2	Method 3
	IR/thermal	Laser	3D stereosc. cameras
Functionality and expansion possibility	Necessary deduction of people movement direction, limited system, low sensitivity, historical solution.	The system is not intelligent, and there is no possibility of distinguishing people. Two people passing through the laser beam at the same time are counted as one.	Observation and analysis of the dynamic image from 3D cameras, which can be added to the monitoring system. Easy to verify of the quantities entering/outgoing passengers.
Score:	1	2	5
Compliance with standards and guidelines	Compliance with the standards.	Compliance with the standards.	Compliance with the standards.
Score:	5	5	5
General costs (installation, observation, ease of use, costs)	Low cost, low efficiency	Low cost, low efficiency.	Precise measurement with use of advanced stereoscopic cameras. Relatively high costs.
Score:	4	4	2

Scores scale: 1- very low, 2-low, 3-medium, 4-high, 5-very high.

Table 3: Designed modules grades

Method		Crit. 1	Crit. 2	Crit. 3
Videomonitoring	1:	1	1	4
	2:	3	4	3
	3:	5	5	2
Passenger counting	1:	1	5	4
	2:	2	5	4
	3:	5	5	2
Passenger information and dynamical timetable	1:	1	2	4
	2:	3	3	3
	3:	5	4	1
Central unit	1:	1	3	5
	2:	3	3	3
	3:	5	4	3
Surveillance center	1:	3	5	5
	2:	4	4	4
	3:	5	4	5
Fuel consumption optimization	1:	2	5	4
	2:	4	5	3
	3:	5	5	3

Table 4: Exemplary feasible solution

Variant	Module 1			Module 2			Module 3			Module 4			Module 5			Module 6								
	Method	Crit. 1	Crit. 2	Crit. 3	Method	Crit. 1	Crit. 2	Crit. 3	Method	Crit. 1	Crit. 2	Crit. 3	Method	Crit. 1	Crit. 2	Crit. 3	Method	Crit. 1	Crit. 2	Crit. 3				
1	1	1	1	4	1	1	5	4	1	1	2	4	1	1	3	5	1	3	5	5	1	2	5	4

2. The problem of system configuration choice as a multi-criteria optimization task

Real optimization problems are often formulated as a multi objective ones, e.g. [4, 13]. It is assumed that the optimal solution is the indication of methods from individual modules resulting from maximizing rating. Basing on Table 4 it is assumed that:

- $i = 1, 2, \dots, 6$ is a module index,
- $j = 1, 2, 3$ is a criterium index,
- $k = 1, 2, 3$ is a method index,

then $x_{i,j,k}$ describes the values from Table 3 (i.e. $x_{4,2,2} = 3$).

The set of all $x_{i,j,k}$ constitute the space of feasible solutions $\Omega = \{x_{i,j,k} \in \mathbb{Z}^n : 1, 2, 3, 4, 5\}$ for all i, j, k . The aim of the problem is to select the configuration resulting from the maximization of individual ratings criteria, that is:

- $f_1(x)$ - criterium 1 defined as sum of grades for all modules.
 $f_1(x) = \sum_{i=1}^6 x_{i,1,k}; i = 1, 2, \dots, 6; k = 1, 2, 3;$
- $f_2(x)$ - criterium 2 defined as sum of grades for all modules.
 $f_2(x) = \sum_{i=1}^6 x_{i,2,k}; i = 1, 2, \dots, 6; k = 1, 2, 3;$
- $f_3(x)$ - criterium 3 defined as sum of grades for all modules.
 $f_3(x) = \sum_{i=1}^6 x_{i,3,k}; i = 1, 2, \dots, 6; k = 1, 2, 3;$

For the exemplary configuration from Table 3, the values substituted to consecutive criteria are:

$$\begin{aligned}
 f_1(x) &= \sum_{i=1}^6 x_{i,1,1} = 1 + 1 + 1 + 1 + 3 + 2 = 9; \\
 f_2(x) &= \sum_{i=1}^6 x_{i,2,1} = 1 + 5 + 2 + 3 + 5 + 5 = 21; \\
 f_3(x) &= \sum_{i=1}^6 x_{i,3,1} = 4 + 4 + 4 + 5 + 5 + 4 = 25.
 \end{aligned} \tag{2}$$

This states that the variant 1 has low functionality ($f_1(x) = 9$) and low cost ($f_3(x) = 25$). Number of all possible configurations of the system is: $L = 3^6 = 729$.

On order to chose the besteonfiguration, a hydride method has been applied. This method consist of elements of weighted sum methods and ε -constraints [5, 6, 12]. The objective vector has been divided into two sets: primary objectives F_p and secondary objectives F_s :

$$\begin{aligned} F_p(x) &= [f_1(x), f_2(x), \dots, f_{kp}(x)]^T, \\ F_s(x) &= [f_{kp+1}(x), f_{kp+2}(x), \dots, f_k(x)]^T, kp < k, \end{aligned} \quad (3)$$

which leads to the following problem statement:

$$\min_{x \in \Omega} F'(x) = \sum_{i=1}^{kp} w_i f_i(x), \quad (4)$$

with constraints:

$$f_i(x) \leq \varepsilon_i, i = kp + 1, kp + 2, \dots, k. \quad (5)$$

The advantage of the above problem formulation is that the result fulfills the secondary objective at least on ε level. Other objectives are meet according to the set weights.

3. Optimal system configurations

For the described problem from equation 1, the divided objective functions are in form:

$$F_p(x) = [f_1(x), f_3(x)]^T, F_s(x) = f_2(x). \quad (6)$$

Thus the problem can be described as:

$$\min_{x \in \Omega} F'''(x) = (-w_1 f_1(x) - w_3 f_3(x)), -f_2(x) \leq -\varepsilon. \quad (7)$$

Despite the objective function $f_2(x)$ being called secondary, in the process of configuration selection it is the most important. Due to standards compliance assurance the solution has to fulfill the given weights and constraints. Regardless of the set weights w_1 and w_3 , the solution ensures the standards compliance at desired level. Figure 2 presents the objective subspace $(-f_1(x), -f_3(x))$ with Pareto solutions for $\varepsilon = 27$.

Thus, one can generate a set of optimization problems assuming different weights, e.g.: case 1: $w_1 = 1, w_3 = 1$ (or $w_1 = 0.5, w_3 = 0.5$ if one normalize the sum of wiegths to value 1) in this example the main objectives are equivalent; case 2: $w_1 = 0, w_3 = 1$, it is a border case, where optimization problem is reduced to costs; case 3: $w_1 = 1, w_3 = 0$, it is a border case, where optimization problem is reduced to functionalities.

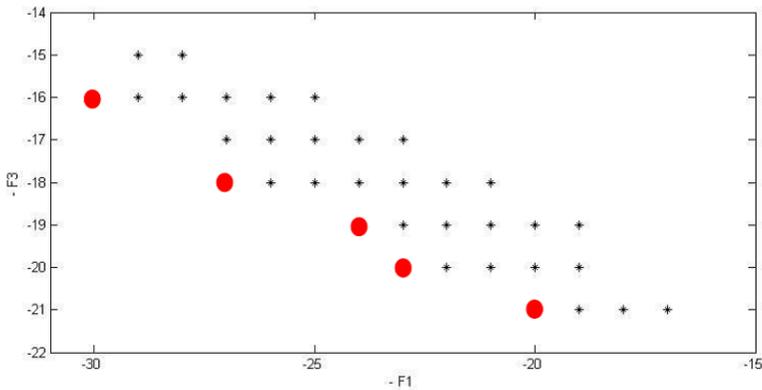


Figure 2: Objective subspace $(-f_1(x), -f_3(x))$ with Pareto solutions for $\epsilon = 27$

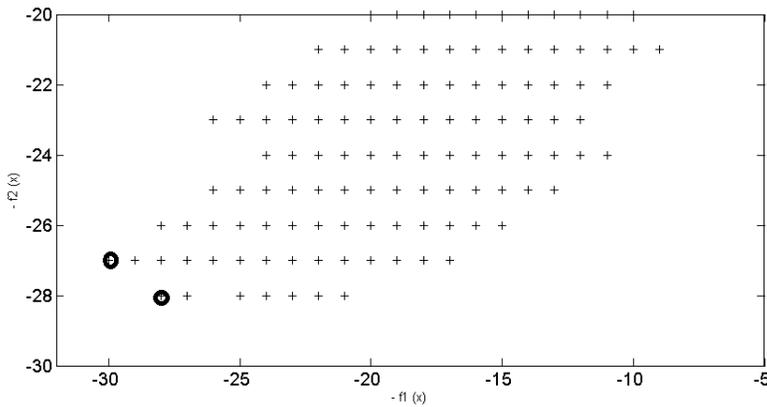


Figure 3: objective space for F_t problem

Table 5: Exemplary optimal solution

Variant	Module 1			Module 2			Module 3			Module 4			Module 5			Module 6								
	Method	Crit. 1	Crit. 2	Crit. 3																				
723	3	5	5	2	3	5	5	2	3	5	4	1	3	5	4	3	1	3	5	5	3	5	5	3
729	3	5	5	2	3	5	5	2	3	5	4	1	3	5	4	3	3	5	4	5	3	5	5	3

The optimal solution e.g. for $\epsilon = 28$ is variant number 723 with objective function value $F'''(723) = 44$, while for $\epsilon = 27$ is variant number 729 with objective function value $F'''(729) = 46$. The final system configurations resulting from the solution are presented in Table 5.

In order to compare the solution to possible solutions that are not limited by costs we present solution space for limited two objective problem F_I :

$$F_I(x) = [f_1(x), f_2(x)]^T \quad (8)$$

in Figure 3. Pareto front is now limited to two solutions leading to values $F_I(728) = [30, 27]^T$ and $F_I(726) = [28, 28]^T$, respectively.

4. Conclusion

The article presents the problem of designing the system configuration comprising of six modules as multi-criteria optimization task. Considered in the design of each of the modules in three different ways, with each of the methods has been assessed by experts in terms of the three criteria. Using the basic methods of multi-criteria optimization, indicated sets of Pareto solutions of possible system configuration and sample solutions optimized using assumed values of additional parameters. It should be mentioned that pilot installation of integrated system has been implemented and successfully tested on electrical and diesel type passenger trains in Poznan and Warsaw, Poland, as result of research and development project No UOD-DEM-1-243/001.

References

- [1] J. MLYNCZAK: Analysis of Intelligent Transport Systems (ITS) in Public Transport of Upper Silesia. Springer Berlin Heidelberg. Modern Transport Telematics, 2011, 164-171.
- [2] Summary of the state of rail safety. Report for the third quarter of 2013. Railway Transport Office, Warsaw, November 2013.
- [3] T. HEJCZYK, B. WSZOLEK, A. GALUSZKA, J. MLYNCZAK and R. BURDZIK: Application of safety and communication modules as an integrated intelligent system in rail vehicles. *Vibroengineering Procedia*, **4** (2014), 105-110.
- [4] A. GALUSZKA and A. SWIERNIAK: Non-cooperative game approach to multi-robot planning. *Int. J. of Applied Mathematics and Computer Science*, **15**(3), (2005), 359-367.
- [5] M. EMMERICH: Matlab Help Optimization Toolbox, Multiobjective Optimization. 2005.
- [6] A. GALUSZKA, A. SWIERNIAK, T. HEJCZYK, B. WSZOLEK and J. MLYNCZAK: Design of rail passengers safety and comfort system as three-objective discrete

- static optimization problem. Carlos Enrique Palau Salvador (Ed.): *ISC'2015, 13th Annual Industrial Simulation Conf.*, Valencia, Spain, (2015), 89-94.
- [7] A. GALUSZKA, A. SWIERNIAK, T. HEJCZYK and J. MLYNCZAK: Hybrid optimization method for design of rail passengers safety and comfort system. *Procedia Technology*, **22** (2016), 905-912.
- [8] HU ZHENGQUN, JUNXIA CUI, J. ZHANG, L. ZHANG and L. CHANG: Railway safety monitoring system based on CAPS. *Fifth Int. Conf. on Measuring Technology and Mechatronics Automation*, (2013), CD-ROM.
- [9] J. KRZYSTEK and T. TRZNADEL: Application of the algorithm DBR of Constraints Theory to production planning. *Przegląd Elektrotechniczny*, **88**(10b), (2012), 163-169, (in Polish).
- [10] J.R. WILSON, A. MILLS, T. CLARKE, J. RAJAN and N. DADASHI (Eds): *Rail Human Factors Around the World: Impacts on and of People for Successful Rail Operations*. CRC Press, Croydon 2012.
- [11] T. POPESCU: *Proiectul Feroviar Romanesc*. Editura Simetria, 2014, (in Romanian).
- [12] M. BELAZZOUG, M. BOUDOUR and K. SEBAA: FACTS location and size for reactive power system compensation through the multi-objective optimization. *Archives of Control Sciences*, **20** (2010), 473-489.
- [13] K. SKRZYPCZYK: Time optimal tracking a moving target by a mobile vehicle – game theoretical approach. *Przegląd Elektrotechniczny*, **86**(3), (2010), 211-215, (in Polish).

Sensitivity analysis of signaling pathway models based on discrete-time measurements

MALGORZATA KARDYNSKA and JAROSLAW SMIEJA

The paper is focused on sensitivity analysis of large-scale models of biological systems that describe dynamics of the so called signaling pathways. These systems are continuous in time but their models are based on discrete-time measurements. Therefore, if sensitivity analysis is used as a tool supporting model development and evaluation of its quality, it should take this fact into account. Such models are usually very complex and include many parameters difficult to estimate in an experimental way. Changes of many of those parameters have little effect on model dynamics, and therefore they are called sloppy. In contrast, other parameters, when changed, lead to substantial changes in model responses and these are called stiff parameters. While this is a well-known fact, and there are methods to discern sloppy parameters from the stiff ones, they have not been utilized, so far, to create parameter rankings and quantify the influence of single parameter changes on system time responses. These single parameter changes are particularly important in analysis of signalling pathways, because they may pinpoint parameters, associated with the processes to be targeted at the molecular level in laboratory experiments. In the paper we present a new, original method of creating parameter rankings, based on an Hessian of a cost function which describes the fit of the model to a discrete experimental data. Its application is explained with simple dynamical systems, representing two typical dynamics exhibited by the signaling pathways.

Key words: sensitivity analysis, signaling pathways, measurement uncertainty, discrete-time measurements.

1. Introduction

Signaling pathways (or regulatory pathways) are cascades of biochemical processes involving creation, degradation and modification of various molecules, specific for a given pathway, as well as their transport between cellular compartments such as cytoplasm, nucleus, mitochondrion, etc. These processes are activated by events taking place

The Authors are with Institute of Automatic Control, Silesian University of Technology, Akademicka str. 16, 44-100 Gliwice, Poland. E-mails: {malgorzata.kardynska, jaroslaw.smieja}@polsl.pl. M. Kardynska is the corresponding author.

The work has been supported by the NCN grant DEC-2013/11/B/ST7/01713 (JS, MK). Calculations were performed on the Ziemowit computational cluster (<http://www.ziemowit.hpc.polsl.pl>) created in the POIG.02.01.00-00-166/08 project (BIO-FARMA) and expanded in the POIG.02.03.01-00-040/13 project (Syscancer).

Received 11.12.2016. Revised 30.04.2017.

inside a cell (such as, e.g., DNA damage), changes in extracellular environment (e.g. in its chemical content or temperature), direct interactions with other cells (following their binding) or physical stresses (radiation, mechanical stress). They are regulated by positive and negative feedback loops that in many cases are not fully understood. Therefore, use of methods that have their origins in automatic control may help in learning these mechanisms, through formulating hypotheses about the structure of regulatory networks governing intracellular processes. Ultimately, such knowledge will support development of the protocols of external regulation of cell behavior.

There are many methods that can be used to describe the dynamics of signaling pathways. In this paper we are focused on deterministic models described by ordinary differential equations, in which variables denote concentrations of molecules involved in a given pathway. Each parameter (or, in case of Michaelis-Menten kinetics, a pair of parameters) correspond to a single biochemical process.

Due to rapid advances in experimental techniques, our knowledge about biochemical processes occurring in living cells is continuously expanding. In the literature there is a growing number of highly dimensional models, which describe the dynamics of signaling pathways components [1]. The more processes are taken into account, the more complex models arise with a large number of parameters. However, methods of measuring biochemical parameters are limited and often inaccurate [2]. Therefore, any mathematical model should be checked with respect to its sensitivity to parameter changes. In general, such model should be robust with respect to small parameter changes. Nevertheless, some parameters are always more important than others. The sensitivity analysis is the tool to be used to determine how a change of parameters influences the system behavior. It provides information about the most important parameters that have the greatest impact on the system output [3]. In the particular application that is considered in this paper, each parameter is associated with a particular biochemical process. Hence, sensitivity analysis may provide insights into how biological experiments should be planned to gain maximum information. Moreover, parameters with the highest ranking indicate prospective molecular drug targets affecting a pathway that is involved in a given disease.

Sensitivity analysis have been developed for over half a century, initially for applications in engineering [4, 5, 6]. While sensitivity methods proved to be helpful in analysis of various pathways [7, 8, 9, 10, 11], they were focused on simulation results whose units were clearly determined (as concentration units). However, in most cases biological experiments provide data about the fold increase of the number of molecules or of the concentration, while their absolute values are not known. In that case the same methods of sensitivity analysis may lead to false conclusions [12]. Furthermore, measurements are discrete in time with irregular and sparse sampling periods. Even if live microscopy is used, data is collected every couple of minutes. For these reasons it is necessary to develop methods which take into account specific properties of biological systems and experimental data.

In this paper we present a new method for creating parameter rankings. It is based on the Hessian of a cost function describing the fit of the model to discrete measurement data [13]. The rankings allow not only to identify the most important parameters of the

model, which is facilitated by the methods developed in earlier works [14], but allows to quantify and rank the importance of single parameters. These parameters should be determined with the highest accuracy when developing a biologically relevant model and may provide hints to indicate prospective molecular targets for new drugs [15]. In addition, the proposed method takes into account the measurement uncertainty at specific sampling times, which is also an improvement over standard sensitivity analysis methods. Most of these methods assume that the model perfectly describes the biological process, which is not true - mathematical models are built on the basis of experimental measurements that are subject to uncertainty and in the case of biological experiments may be very high.

In the following section the concept of stiff and sloppy parameters and the method to find them are introduced, followed by the definition of the proposed ranking. Then, the rankings obtained with the proposed method are shown and compared with those given by standard sensitivity analysis for two models, each representing particular dynamics exhibited by signaling pathways. Both examples show that the presented method allows to create reliable parameter rankings that helps to identify parameters substantially affecting the fit of the model to experimental data.

2. Stiff and sloppy parameters

Sensitivity analysis is usually divided into two major categories: local and global. Local sensitivity analysis describes how the system output changes when parameters deviate in a close neighborhood of their nominal values. Global sensitivity, in turn, describes how the system output changes when multiple parameters are allowed to change in a relatively wide range [16, 17]. The method presented in this work may be classified as a local one.

Let the model be described by the state equation:

$$\frac{d\mathbf{X}}{dt} = F(\mathbf{X}, \mathbf{U}, \theta), \quad \mathbf{X}(t_0) = \mathbf{X}_0, \quad (1)$$

where \mathbf{X} is a vector of state variables, representing concentrations or the average number of molecules of proteins, enzymes or transcripts involved in the signaling pathway, \mathbf{U} denotes control vector, which is usually a scalar in biological systems, θ is a vector of parameters. A solution of the described model is defined by:

$$\mathbf{X}(\theta, t). \quad (2)$$

Model quality can be evaluated using the least-squares cost function [13] that describes the difference between the variable values obtained from simulation and quantified experimental data:

$$C_s(\theta) = \sum_n \frac{1}{2} \frac{(x_s(\theta, t) - x_{s,n})^2}{\sigma_{s,n}^2} = \sum_n \frac{1}{2} r_{s,n}^2, \quad (3)$$

where $x_{s,n}$ is the value of s -th state variable in the n -th sample measured with the uncertainty $\sigma_{s,n}$, $x_s(\theta, t)$ is a solution of the model at corresponding time t , while $r_{s,i}$ is the residual describing the deviation of a dynamical variable $x_s(\theta, t)$ from its measured values. If the model perfectly fits to measurement data the cost function C_s is equal to 0, and the vector of parameters θ , giving a perfect fit, is denoted as θ^* .

To analyze model sensitivity to parameter variation, let us consider the Hessian matrix corresponding to the cost function C_s calculated at θ^* . Since the value of one biochemical parameter may vary in the range of several orders or more from another, to eliminate the impact of relative changes in parameter values the derivatives with respect to $\log \theta$ are taken [14]:

$$H_{j,k}^{C_s} = \frac{\partial^2 C_s}{\partial \log \theta_j \partial \log \theta_k}, \quad (4)$$

where j and k denotes j -th and k -th parameter, respectively.

Instead of (4), the Hessian approximation H^{C_s} can be used - the so called Fisher information matrix $J^T J$ [18]:

$$J = \frac{\partial r_{s,n}}{\partial \log \theta}. \quad (5)$$

The Hessian matrix H^{C_s} is positive definite and symmetric, so it has real eigenvalues λ and eigenvectors v [19]. It describes the surface of deviations of the model from measured data. For a model with N_p parameters the surface is an N_p -dimensional ellipsoid in parameter space. The principal axes of the ellipsoids are the eigenvectors of H^{C_s} , while the width (denoted by d_i) of the ellipsoids along each principal axis is given by:

$$d_i = \frac{1}{\sqrt{\lambda_i}}. \quad (6)$$

The narrowest axes are called *stiff* and they define directions in the parameter space, leading to large changes in the model response. The broadest axes, called *sloppy*, represent the directions along which parameters changes, even in a wide range, do not result in a worse fitting of the model to experimental data [20]. The meaning of eigenvalues and eigenvectors of Hessian H^{C_s} is illustrated with a simple example of a hypothetical model with two parameters: θ_1 and θ_2 (Fig. 1), where Hessian describes an ellipse in the θ_1/θ_2 parameter space, d_1 and d_2 denote the width of the ellipse along each principal axis, corresponding to eigenvalues λ_1 and λ_2 , respectively, while v_1 and v_2 denote the eigenvectors of H^{C_s} and define the position of the ellipse. In this example, the eigenvectors v_1 and v_2 are associated with a large value d_1 and small value d_2 , respectively. Since v_2 eigenvector depends mostly on θ_1 , this parameter is *stiff*, i.e. the change in its value leads to much greater change in system response than in the case θ_2 was changed.

It should be stressed, however, that models of signaling pathways are much more complex and often include dozens of parameters. Therefore graphical presentation of results illustrating deviations of the model from measured data is not possible. In this

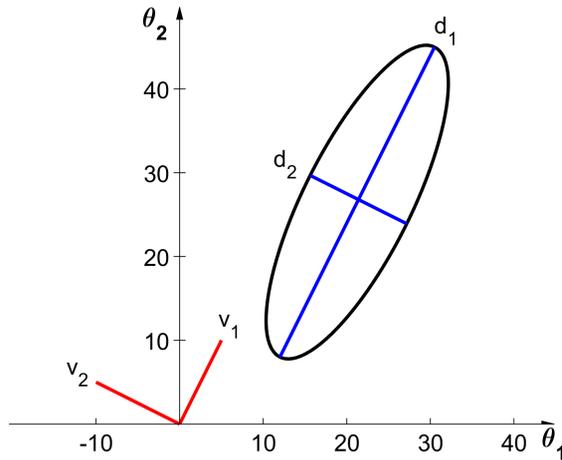


Figure 1: An ellipse illustrating deviations of the model from measured data in a 2-dimensional parameter space.

case, plots showing parameter rankings are the preferred way of presenting the results. The method of creating parameter rankings, proposed in this paper, is based on eigenvectors and eigenvalues of the Hessian H^{C_s} . R_j , denoting the ranking value for the j -th parameter, can be defined as:

$$R_j = \sum_i \left| \frac{v_{j,i}}{d_i} \right|, \quad (7)$$

where d_i is the width of the ellipsoid along i -th principal axis, and $v_{j,i}$ is the element of the i -th eigenvector corresponding to the j -th parameter.

3. Ranking examples

To show applicability of the proposed method, two examples are presented in this section. Both represent typical dynamics exhibited by signaling pathways - with and without oscillations. For each model its step response have been simulated and the measurements have been sampled for arbitrarily chosen time points, shown in the figures, with 10% uncertainty.

The results obtained are compared to standard rankings based on the area under the curve (AUC) of sensitivity functions with L^1 norm [21] as a metric. While other methods can be found in the literature [22], sensitivity functions constitute the most often used

base for the rankings in local analysis of signaling pathways [7, 8, 10, 11, 23], unless it is qualitative behavior of the pathway that is under consideration [24].

3.1. Transcription-translation pathway

As the first example, let us consider a simple pathway, in which gene transcription is activated, leading to production of mRNA and, subsequently, protein, whose concentrations are denoted by x_m and k_p , respectively. These molecules are degraded in a first-order process. In the simplest case such system is described by linear state equations:

$$\frac{dx_m}{dt} = k_m u - k_{dm} x_m, \quad (8)$$

$$\frac{dx_p}{dt} = k_p x_m - k_{dp} x_p, \quad (9)$$

where u represents the system input (induction of transcription), k_m , k_p , k_{dm} and k_{dp} are mRNA and protein production and degradation rates, respectively.

Let us also assume that only the protein levels are observed in the experiment, i.e. x_p is the output variable.

Then, the system may be alternatively represented by the transfer function

$$K(s) = \frac{X(s)}{U(s)} = \frac{k}{(1 + sT_1)(1 + sT_2)}, \quad (10)$$

whose parameters have been arbitrarily chosen as $T_1 = 1/k_{dm} = 0.1$, $T_2 = 1/k_{dp} = 1$ and $k = k_p k_m / (k_{dm} k_{dp}) = 1$ (in this system only three parameters are identifiable).

The model step response with sampled measurements is shown in Fig. 2.

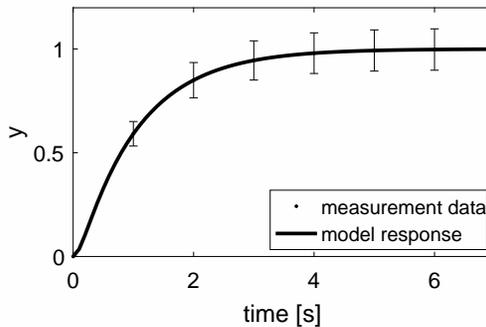


Figure 2: The step response of a second-order inertial system with parameters $T_1 = 0.1$, $T_2 = 1$ and $k = 1$ and the standard deviation of measurement data.

Parameter rankings, obtained with the procedure described in the previous section are shown in Fig. 3a.

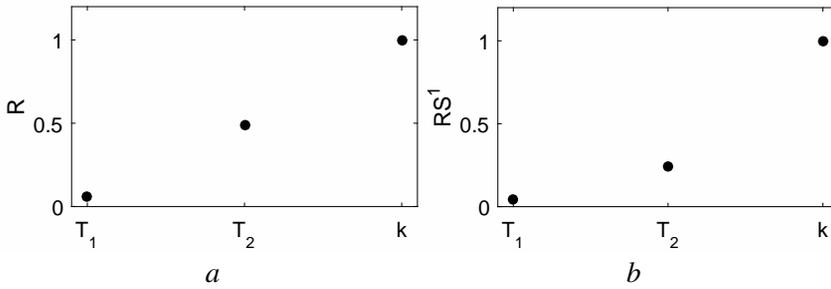


Figure 3: Parameter rankings for the second-order system based on: (a) the proposed method and (b) sensitivity functions.

The order of parameters importance is the same for both rankings. However, in the ranking based on sensitivity functions, the change in system response caused by change of T_2 value is similar to the one caused by T_1 change. The ranking created with the method proposed in this paper assigns much greater importance to T_2 , which is a greater time constant and determines transient system response. These differences result from taking into account the measurement uncertainty in the cost function C_s (3). As mentioned before, uncertainty of measurement, defined as the standard deviation σ , has been assumed to be 10% of measurement value. Fig. 2 shows that the absolute standard deviation of measurements made at time-points: 1s and 2s (before reaching the steady state) is smaller than the standard deviation of measurements in steady state. As a result, deviation of the model response at points with a lower σ will be treated as more important than deviation at points with a higher σ . This explains the higher position of parameter T_2 in our ranking, compared to the parameter k , which is responsible for the steady state system response and according to Fig. 2 cannot be precisely determined due to measurement uncertainty.

3.2. A closed-loop regulatory module

As a second example a model of p53/Mdm2 regulatory module has been considered. It is one of the simplest oscillatory systems that can be fitted to experimental data, described by the following equations [25]:

$$\frac{d(p53)}{dt} = ms_1 - k_{d1} \cdot (p53) \cdot (Mdm2_{nuc})^2, \quad (11)$$

$$\frac{d(Mdm2_{cyt})}{dt} = n \cdot \left(s_2 + \frac{s_3 \cdot (p53)^3}{s_4^3 + (p53)^3} \right) - k_1 k_2 \cdot \frac{Mdm2_{cyt}}{k_2 + (p53)}, \quad (12)$$

$$\frac{d(Mdm2_{nuc})}{dt} = \frac{k_1 k_2 \cdot (Mdm2_{cyt})}{k_2 + (p53)} - k_{d2} \cdot (Mdm2_{nuc}), \quad (13)$$

where the variables $p53$, $Mdm2_{cyt}$ and $Mdm2_{nuc}$ denote concentrations of total p53 protein, cytoplasmic Mdm2 and nuclear Mdm2, respectively. It is a minimal model reflect-

ing oscillatory response of p53 protein to system excitation, e.g., DNA damage. The parameters m and n are the numbers of p53 and Mdm2 gene copies, respectively, s_1 , s_2 , s_3 and s_4 are the production rates per gene copy, k_{d1} and k_{d2} are p53 and Mdm2 degradation rates and k_1 , k_2 are Mdm2-mediated nuclear import rates. Nominal parameter values are given in Table 1.

Table 1: p53/Mdm2 model parameters.

Parameter	s_1	s_2	s_3	s_4	k_{d1}
Value	16	8	80	$1 \cdot 10^5$	$1 \cdot 10^{-13}$
Unit	s^{-1}	s^{-1}	s^{-1}	—	s^{-1}
Parameter	k_{d2}	k_1	k_2	m	n
Value	$2.2 \cdot 10^{-4}$	$3.5 \cdot 10^{-3}$	$2.3 \cdot 10^3$	2	2
Unit	s^{-1}	s^{-1}	—	—	—

Let the system output be the p53 concentration, measured, e.g., with an experimental technique called *Western Blotting*. A characteristic feature of biological measurements, including the one mentioned in the preceding sentence, is their large uncertainty, affected by many different factors. As a result, the standard deviation of measurements may vary significantly, as shown in Figure 4 (measurement points are marked for illustration only - their values have been calculated in simulation, with the random error superimposed on the results).

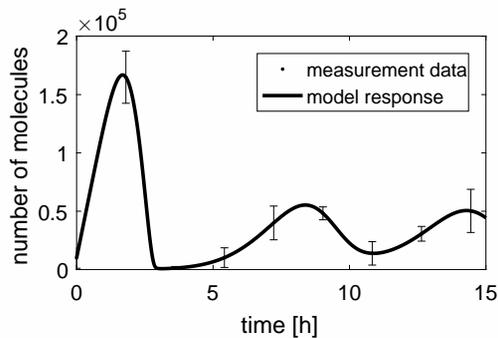


Figure 4: p53 model responses against the measurement data.

As in the previous example, two different sensitivity rankings have been calculated. The results are shown in Fig. 5.

Methods used for rankings calculation produced different results. In the case of signaling pathways, it may be more difficult to determine the impact of individual param-

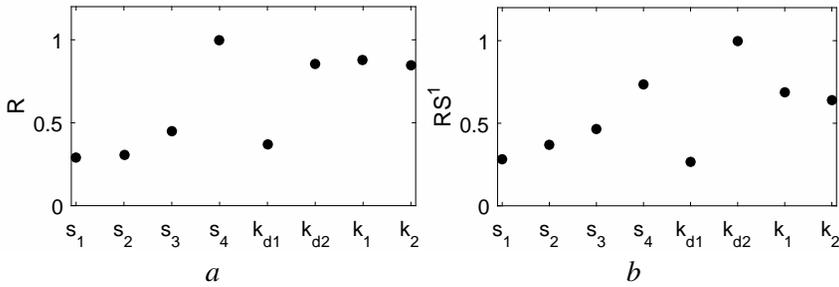


Figure 5: Parameter rankings for p53/Mdm2 model based on: our method (a) and sensitivity functions (b).

eters since they may affect many characteristics of the system response. For example, both parameter rankings indicate the high position of parameter k_{d2} , which affects the amplitude of p53 oscillations and introduces a small phase shift. In the parameter ranking based on sensitivity functions the parameter k_{d2} is indicated as the most important, while in the ranking taking into account the measurement uncertainty its position is slightly lower. On the other hand, the parameter s_4 has been found to be more relevant according to our method. In order to determine which of these two parameters has more significance on the fit of the model to experimental data, we performed simulations with parameters k_{d2} and s_4 changed by 15%. On Fig. 6 we compared received time courses with the model response for the nominal parameter set and measurement data. When we analyze Fig. 6 we find that despite the significant change in the model response caused by change of the parameter k_{d2} , the model may still quite good fit to the experimental data due to measurements uncertainty. Therefore, in the ranking taking into account the measurement uncertainty the position of parameter k_{d2} is lower. Similarly, we can explain the differences for parameter s_4 , which also significantly changes the model response, however, the model response after changing the parameter s_4 by 15% slightly worse fits to the experimental data.

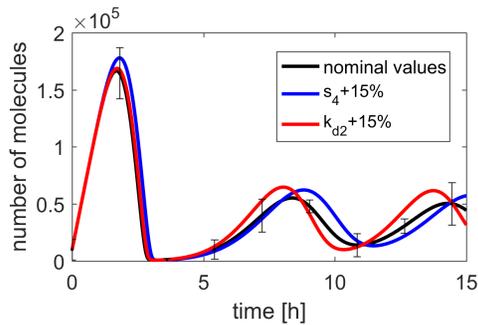


Figure 6: A comparison of p53 model responses after the change of selected parameters.

4. Conclusion

The new method for creating parameter rankings based on the known method called *sloppy / stiff* sensitivity analysis has been proposed. It facilitates taking into account the impact of measurement uncertainty, which is a major problem in analysis of any kind of biological experimental data.

Two simple examples have been used to show that the presented method allows to create reliable parameter rankings that helps to identify parameters substantially affecting the fit of the model to experimental data. This allows to choose parameters that should be determined with the highest accuracy in the experimental research and, as a consequence, the method can be used to plan biological experiments and helps to use the funds for experimental research in the most efficient way.

References

- [1] R. WILLIAMS, J. TIMMIS and E.E. QWARNSTROM: Computational models of the NF- κ B signalling pathway. *Computation*, **2**(4), (2014), 131-158.
- [2] S.J. MAERKL and S.R. QUAKE: A systems approach to measuring the binding energy landscapes of transcription factors. *Science*, **315** (2007), 233-237.
- [3] D.A. RAND: Mapping the global sensitivity of cellular network dynamics. *J. Royal Society Interface*, **5** (2008), 59-69.
- [4] R.W. NEWCOMB: Linear Multiport Synthesis. McGraw-Hill, NY, 1966.
- [5] J.J. CRUZ: Feedback Systems. McGraw-Hill, New York, 1972.
- [6] R.K. BRAYTON and R. SPENCE: Sensitivity and Optimization. Elsevier, Amsterdam, 1980.
- [7] M. BENTELE, I. LAVRIK, M. ULRICH, S. STOSSER, D.W. HEERMANN, H. KALTHO, P.H. KRAMMER and R. EILS: Mathematical modeling reveals threshold mechanism in cd95-induced apoptosis. *The Journal of Cell Biology*, **166**(6), (2004), 839-851.
- [8] H. YUE, M. BROWN, J. KNOWLES, H. WANG, D.S. BROOMHEAD and D.B. KELL: Insights into the behaviour of systems biology models from dynamic sensitivity and identifiability analysis: a case study of an NF- κ B signalling pathway. *Molecular BioSystems*, **2**(12), (2006), 640-649.
- [9] E. BALSACANTO, A.A. ALONSO and J.R. BANGA: An iterative identification procedure for dynamic modeling of biochemical networks. *BMC Systems Biology*, **4**(11), (2010).

- [10] V.RAIA, M. SCHILLING and M. BOEHM: Dynamic mathematical modeling of IL13-induced signaling in Hodgkin and Primary Mediastinal B-Cell Lymphoma Allows Prediction of Therapeutic Targets. *Cancer Research*, **71** (2011), 693-704.
- [11] K. RATEITSCHAK, F. WINTER, F. LANGE, R. JASTER and O. WOLKENHAUER: Parameter identifiability and sensitivity analysis predict targets for enhancement of STAT1 activity in pancreatic cancer and stellate cells. *PLoS Computational Biology*, **8**(12), (2012), e1002815.
- [12] J. SMIEJA, M. KARDYNSKA and A. JAMROZ: The meaning of sensitivity functions in signaling pathways analysis. *Discrete and Continuous Dynamical Systems – series B*, **10** (2014), 2697-2707.
- [13] B.C. DANIELS, Y.J. CHEN, J.P. SETHNA, R.N. GUTENKUNST and C.R. MYERS: Sloppiness, robustness, and evolvability in systems biology. *Current Opinion in Biotechnology*, **19** (2008), 389-395.
- [14] R.N. GUTENKUNST, J.J. WATERFALL, F.P. CASEY, K.S. BROWN, C.R. MYERS and J.P. SETHNA: Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, **3**(10), (2007), 1871-1878.
- [15] A. MARIN-SANGUINO, S.K. GUPTA, E.O. VOIT and J. VERA: Biochemical pathway modeling tools for drug target detection in cancer and other complex diseases. *Methods in Enzymology*, **487** (2011), 319-369.
- [16] J. LEIS and M. KRAMER: Sensitivity analysis of systems of differential and algebraic equations. *Computers & Chemical Engineering*, **9** (1985), 93-96.
- [17] P. IGLESIAS and B. INGALLS (Ed): Control theory and systems biology. MIT Press, 2010.
- [18] J.J. WATERFALL, F.P. CASEY, R.N. GUTENKUNST, K.S. BROWN, C.R. MYERS, P.W. BROUWER, V. ELSER and J.P. SETHNA: Sloppy-model universality class and the Vandermonde matrix. *Physical Review Letters*, **97**(15), (2006), 150601.
- [19] R.A. HORN and C.R. JOHNSON: Matrix Analysis. Cambridge University Press, 1990.
- [20] K.S. BROWN and J.P. SETHNA: Statistical mechanical approaches to models with many poorly known parameters. *Physical Review E*, **68** (2003), 021904.
- [21] M. KARDYNSKA and J. SMIEJA: L-1 and L-2 norms in sensitivity analysis of signaling pathway models. *21st Int. Conf. on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, (2016), 589-594.

- [22] K. PUSZYNSKI, P. LACHOR, M. KARDYNSKA and J. SMIEJA: Sensitivity analysis of deterministic signaling pathways models. *Bulletin of the Polish Academy of Sciences Technical Sciences*, **60**(3), (2012), 471-479.
- [23] K. A. KIM, S. L. SPENCER, J. G. ALBECK, J. M. BURKE, P. K. SORGER, S. GAUDET and H. KIM DO: Systematic calibration of a cell signaling network model. *BMC Bioinformatics*, **11** (2010), 202.
- [24] M. KARDYNSKA and J. SMIEJA: Sensitivity analysis of signaling pathways in the frequency domain. *Information Technologies in Medicine*, **2** (2016), 275-285.
- [25] B. HAT, K. PUSZYNSKI and T. LIPNIACKI: Exploring mechanisms of oscillations in p53 and nuclear factor- κ B systems. *IET Systems Biology*, **3** (2009), 342-355.

Vibration control in semi-active suspension of the experimental off-road vehicle using information about suspension deflection

JERZY KASPRZYK, PIOTR KRAUZE, SEBASTIAN BUDZAN and JAROSLAW RZEPECKI

The efficiency of vibration control in an automotive semi-active suspension system depends on the quality of information from sensors installed in the vehicle, including information about deflection of the suspension system. The control algorithm for vibration attenuation of the body takes into account its velocity as well as the relative velocity of the suspension. In this paper it is proposed to use the Linear Variable Differential Transformer (LVDT) unit to measure the suspension deflection and then to estimate its relative velocity. This approach is compared with a typical solution implemented in such applications, where the relative velocity is calculated by processing signals acquired from accelerometers placed on the body and on the chassis. The experiments performed for an experimental All-Terrain Vehicle (ATV) confirm that using LVDT units allows for improving ride comfort by better vibration attenuation of the body.

Key words: vibration control, magnetorheological damper, linear variable differential transformer, skyhook.

1. Introduction

Semi-active automotive suspension systems [9] have numerous applications, especially for vehicles used in significantly varying road conditions. This type of suspension is characterized by low energy consumption, inherent stability and ability to adapt to different road conditions. Generally, semi-active devices used in vehicles are magnetorheological (MR) dampers [11], in which the relationship between damping force and the piston velocity depends on the instantaneous viscosity of the MR fluid filling the damper. This viscosity can be adjusted by the magnetic field induced inside the piston. State of the MR fluid can be changed from liquid to semi-solid within milliseconds. Controlling current flowing through the coil allows for modifying the MR damper characteristics according to the ride conditions. In the recent literature numerous algorithms

The Authors are with Institute of Automatic Control, Silesian University of Technology, Akademicka str. 16, 44-100 Gliwice, Poland. The corresponding author is J. Kasprzyk, e-mail: jerzy.kasprzyk@polsl.pl

The partial financial support of this research by Polish Ministry of Science and Higher Education is gratefully acknowledged.

Received 2.01.2017. Revised 25.04.2017.

for the MR damper control can be found, such as, e.g., Skyhook or Groundhook [8]. In [6] linear and nonlinear feedback control algorithms for MR dampers are compared. Besides, information about the road profile which is obtained in advance can be applied in feed-forward control using, e.g., FxLMS approach [7].

Proper control of the damping parameters can be adjusted regarding to the information from different types of sensors, like LVDT [2], vision cameras [10], laser range scanners [1], or using other measurement methods like, e.g., structured light, RGB-D, infrared, ultrasonic sensors or multi-sensors approach [12]. Application of a laser range scanner in the experimental ATV [5] was tested during a special student program called PBL (Project Based Learning), but some disadvantages have been identified, e.g., changes of the vehicle position in the z-axis may result in replacing some of the scan lines, and produce incorrect values of the distance between the scanner and the road object. Besides, solutions based on image processing are computationally demanding, whereas the real-time processing imposes limits on the architecture of the system which should be simple, reliable and accurate. Thus, it is assumed that information about vehicle motion and, indirectly, about road roughness should be acquired from accelerometers and LVDT sensors, which measure the suspension stroke based on change of the position of the movable magnetic rod.

The paper is organised as follows. Section 2 refers to an experimental set-up together with the architecture of the measurement and control system applied in this research. In Section 3 Skyhook approach for vibration attenuation is described. Next, methods of vertical velocity estimation are analyzed in Section 4, and experimental results of vibration attenuation are presented in Section 5. Final conclusions are drawn in Section 6.

2. Description of the system

The experimental set-up is based on the off-road vehicle (see Fig. 1) modified by replacing the classical dampers by the MR ones produced by the Lord Corporation. Basic version of the test platform is equipped with the following devices: accelerometers from Freescale Semiconductor, peripheral measurement and control units, the main controller based on the Beaglebone-White single-board computer. In this research the measurement system has been extended by additional devices, i.e.: LVDT sensors and dedicated signal conditioners produced by Peltron, the National Instruments (NI) sbRIO platform which serves as a controller dedicated to LVDT units, and an IMU (Inertial Measurement Unit) assembled using the STMicroelectronics integrated circuit.

Transfer of measurement data from accelerometers to the main controller through the CAN bus is realized by a controller program written in C language. Software for LVDT sensors has been created using LabVIEW which supports NI platforms. A NI sbRIO platform is additionally connected to the main suspension controller via the RS-232 communication protocol. Communication with IMU has been established using the

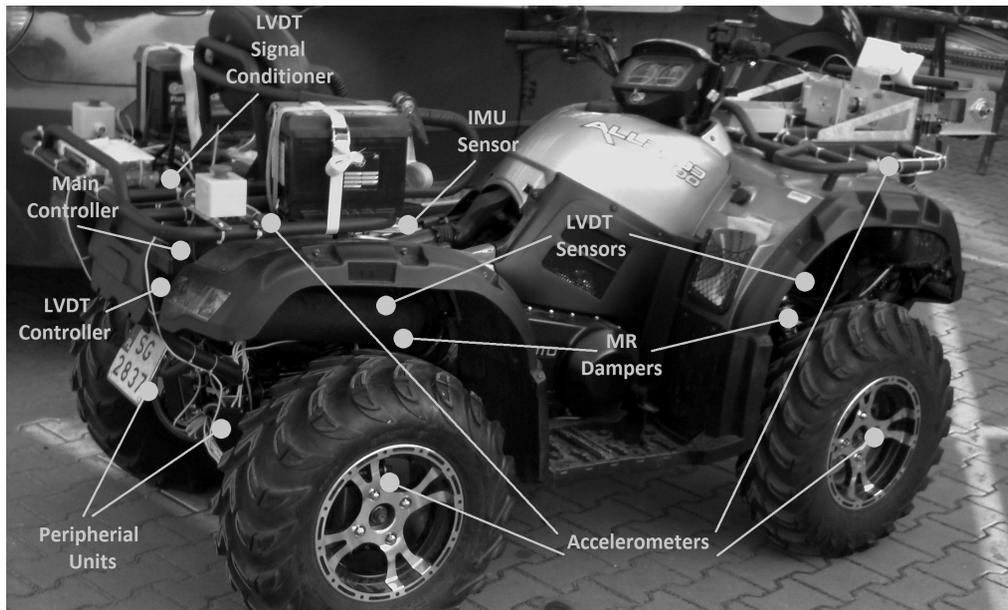


Figure 1: ATV with vibration control and vehicle motion measurement system

same NI controller. The developed measurement and control system is presented in the block diagram in Fig. 2.

Four three-axis accelerometers are installed on the vehicle body as well as four next to each wheel. They are used for measuring the absolute vertical acceleration of the ATV body (the sprung mass) and the chassis (the unsprung mass). However, the vibration control algorithm requires estimates of the absolute vertical velocity of the body and the relative velocity of the suspension system. Commonly, in order to obtain velocity estimates, the acceleration signals are integrated. Here, LVDT sensors have been also placed in the vicinity of the suspension shock-absorbers, on the same screws as dampers. They measure relative suspension displacement of the suspension system. Therefore, the LVDT signals can be differentiated and used instead of accelerometers for estimation of the relative velocity of the suspension elements.

3. Vibration control

The goal of vibration control is the minimisation of vibration of the selected vehicle part while driving through an obstacle, in this case a beam lying on the track. Shortly, it means that the suspension should be soft when wheels reach the beam, but when the ATV comes down, dampers should become harder to reduce oscillations of the vehicle body.

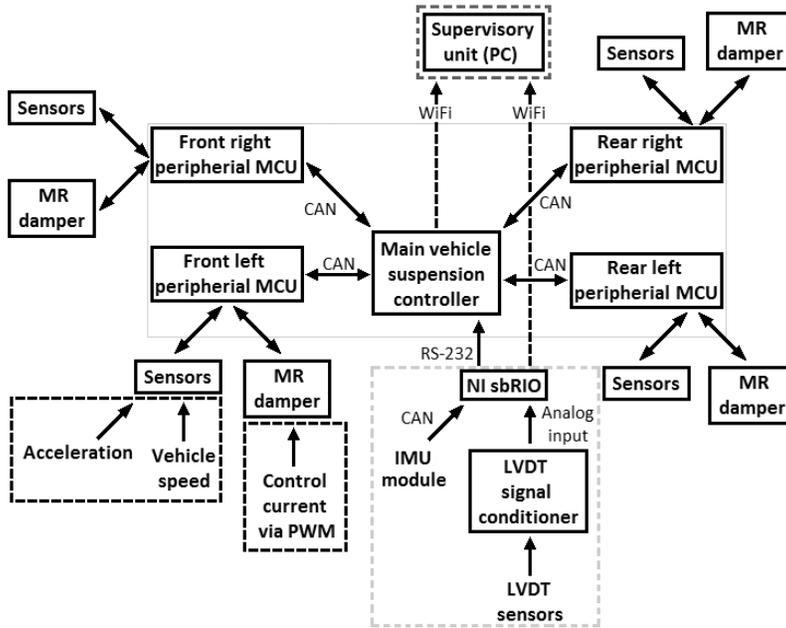


Figure 2: Architecture of the proposed measurement and control system

A classical approach to vibration control in the semi-active suspension system is based on the Skyhook algorithm [3], where the optimal control signal is able to isolate the sprung mass from the base excitation. It means that a force generated by the damper should be proportional to the absolute vertical velocity of the sprung mass v_s :

$$F_{alg} = -\delta \cdot v_s, \quad (1)$$

where the gain factor δ should be properly tuned to obtain good vibration attenuation.

Thus, the damper control signal (i.e. current controlling viscosity of the MR fluid in the damper) can be calculated as follows:

$$i_{ctrl} = \begin{cases} i(F_{alg}, v_{mr}) & \text{for } v_s(v_s - v_u) > 0, \\ 0 & \text{for } v_s(v_s - v_u) \leq 0, \end{cases} \quad (2)$$

where: v_u is the velocity of the unsprung mass (the base excitation) and $v_{mr} = v_s - v_u$ denotes the relative velocity of the damper piston. The inverse model of the MR damper

$$i(F_{alg}, v_{mr}) = \left[\frac{-F_{alg} - c_0 \cdot v_{mr} - \alpha_0}{\alpha_1 \cdot \tanh(\beta_0 \cdot v_{mr}) + c_1 \cdot v_{mr}} \right]^2 \quad (3)$$

was identified in special experiments [4], where $\alpha_0 = 62.42$, $\alpha_1 = 1340$, $\beta_0 = 39.95$, $c_0 = 802.8$ and $c_1 = 488.5$.

It should be noticed that the following velocities are used in the above equations: the absolute velocity of the vehicle body v_s and under-body parts v_u as well as the relative velocity of the damper piston, v_{mr} . They should be estimated based on the appropriate measurement signals. Since each quarter of the vehicle suspension is controlled independently using the proposed Skyhook algorithm, the set of variables v_u , v_s , v_{mr} , F_{alg} reflects any part of the suspension.

4. Velocity estimation

Usually, in this type of applications, a vertical velocity of the vehicle body is estimated by integration of a signal acquired from an accelerometer mounted on the body, whereas the damper piston velocity is calculated as a difference between the estimated body velocity and the velocity of the unsprung mass estimated using a signal from an accelerometer mounted on the chassis, near a wheel. In order to improve the accuracy of the current calculation according to (3), and consequently the quality of vibration control, we propose to estimate the relative velocity v_{mr} based on differentiation of the signal from the LVDT sensor measuring the displacement of the damper piston.

4.1. Data preprocessing

Since, 3-axis accelerometers were used in this application, signals measured by the sensor was transformed into the vertical acceleration with respect to the vehicle's reference coordinate system. Typical measurements taken from the accelerometer and the LVDT deflection sensor located in the front right side of the vehicle are presented in Fig. 3. These signals were acquired with the sampling interval 2 ms while crossing the obstacle at zero control current of the MR damper. It can be stated that signal from the accelerometer exhibits an offset caused by the gravitational acceleration as well as a big noise induced by the vehicle engine. Also, the offset can be observed in the measured deflection of the suspension caused by the non-zero point of the LVDT operation, but influence of the engine noise is small.

Generally, two types of measurement disturbances can be distinguished in this case: sensor-induced and engine-induced. The sensor-induced noise is an inherent parameter related to the operating range of frequencies. Comparison of measurement noise and measured signals in frequency domain is presented in Fig. 4. The sensor-induced noise was acquired for a stationary vehicle with the engine off, whereas measurements were taken for the vehicle driving the test route. Based on presented power spectral densities it can be evaluated that signal-to-noise ratio for the whole 250 Hz frequency range is equal to 51 dB for the acceleration measurements and 57 dB for the suspension deflection measurements.

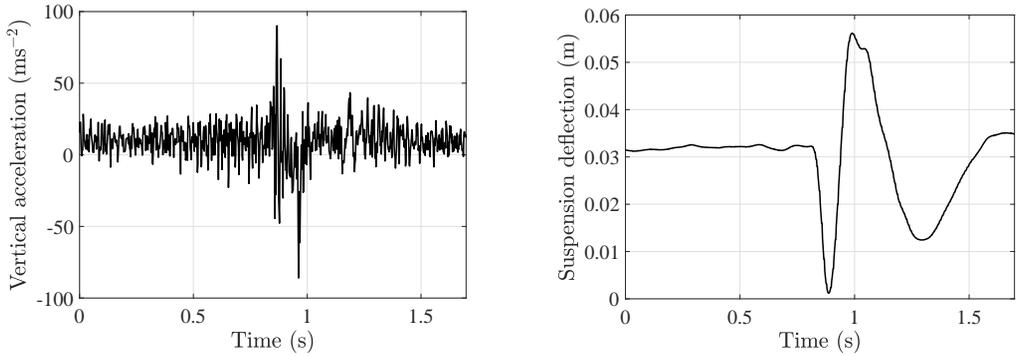


Figure 3: Sample measurement signals: body acceleration (left), suspension deflection (right).

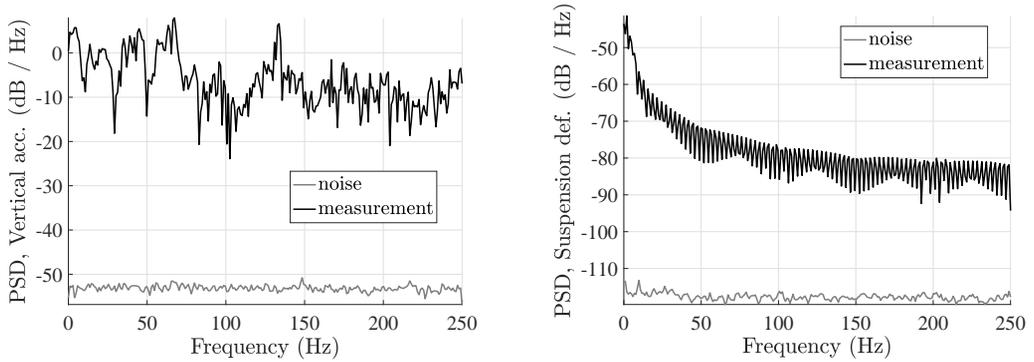


Figure 4: Power spectral density of measured signals and measurement noise for accelerometer (left), LVDT (right).

Because integration of signals may lead to problems with an offset and a signal drift, so invariant influence of the gravitational acceleration was excluded from the resultant acceleration signal a_j by subtracting an estimated averaged acceleration $a_{avg,j}$ from the raw measurement $a_{sensor,j}$:

$$a_j(n) = a_{sensor,j}(n) - a_{avg,j}(n), \quad (4)$$

where n denotes the time instant, and the averaged acceleration is the result of low-pass filtering in the discrete-time domain:

$$a_{avg,j}(n) = [H_{lp}(z^{-1})]^2 \cdot a_{sensor,j}(n) = \left[\frac{0.001}{1 - 0.999z^{-1}} \right]^2 \cdot a_{sensor,j}(n), \quad (5)$$

where z^{-1} is the delay operator. Here index j refers to the relevant part of the vehicle body: front or rear, and left or right.

In the case of vehicle vibration control the frequency-range-in-interest of the acceleration signals varies from about 1 Hz to 25 Hz [9], so time constant of the low-pass digital filter denoted here as $H_{lp}(z^{-1})$ was set to 2 seconds resulting in a cut-off frequency equal to 0.5 Hz.

Finally, the velocity estimation can be implemented as integration with inertia according to the following formula:

$$v_j(n) = H_{int}(z^{-1}) \cdot a_j(n) = \frac{T_s}{1 - 0.99z^{-1}} \cdot a_j(n), \quad (6)$$

where T_s denotes the sampling interval and dynamics of the inertial part of the filter $H_{int}(z^{-1})$ can be characterized by a time constant set to 0.2 seconds. The estimated velocity of the vehicle parts are used in the Skyhook algorithm directly according to (2), or they are used for estimation of the relative velocity applied in the inverse model (3).

In the second approach the relative velocity can be estimated by differentiation of the signal from LVDT sensor measuring the displacement of the damper piston:

$$v_j(n) = \frac{(1 - z^{-1}) \cdot x_j(n)}{T_s}, \quad (7)$$

where x_j denotes a deflection of a suspension.

4.2. Experimental results

Both methods of velocity estimation were compared in experiments performed as individual trips by a beam with a height of 0.08 m and for a vehicle speed of about 20 km/h. Vehicle vertical movement was recorded by 8 accelerometers and 4 LVDT sensors positioned as shown in Fig. 1. Operation of the suspension system was tested for the different levels of current controlling the MR dampers: 0, 0.07, 0.13, 0.27, and 0.53 A. Here only a few examples of experimental results can be presented.

Sample plots of the relative velocity estimated by both methods for 5 consecutive runs for the same control current are shown in Fig. 5. This confirms the greater impact of errors caused by subtraction of two estimates calculated by integration. It may suggest that using LVDT-based estimates gives the opportunity to get better results of control in comparison with the acceleration-based approach.

Repeatability of experiments was validated performing 5 runs for the same configuration of the suspension system. In order to exclude disturbances generated by the vehicle engine signals were additionally filtered by the 20th-order Chebyshev low-pass filter of first type with the cut-off frequency equal to 35 Hz. Next, the velocity was estimated and time diagrams of velocity signals were averaged as follows:

$$v(n) = \frac{1}{5} \sum_{k=1}^5 v_k(n), \quad (8)$$

where k refers to a consecutive experiment.

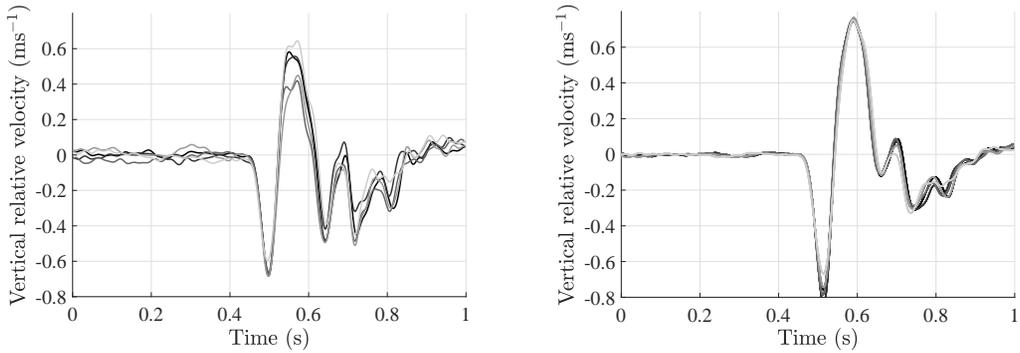


Figure 5: Vertical relative velocity of the left front part of the suspension for 5 consecutive rides for current 0.07 A, estimated using: accelerometers (left), LVDT (right).

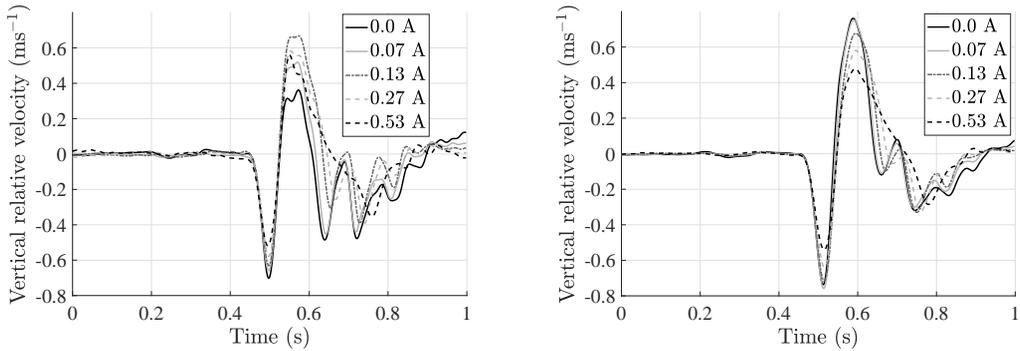


Figure 6: Vertical relative velocity of the right front part of the suspension estimated for different control currents using: accelerometers (left), LVDT (right).

The averaged values of velocity estimates are shown in Fig. 6 and 7. They depict the relative movement of the right and left front part of the suspension, respectively. Reduction of the maximum amplitude of the relative velocity according to increase of the control current can be easily noticed for results obtained by using LVDT measurements. This is in line with expectations, that the higher current should result in better damping. Plots of the velocity estimated using acceleration measurements indicate some irregularities in waveforms, not existing in reality, as well as occurrence of the offset. Furthermore, in the case of acceleration-based estimates, decreasing of the amplitude for increasing control current is not met which confirms greater deterioration of these estimates in comparison to the LVDT-based results.

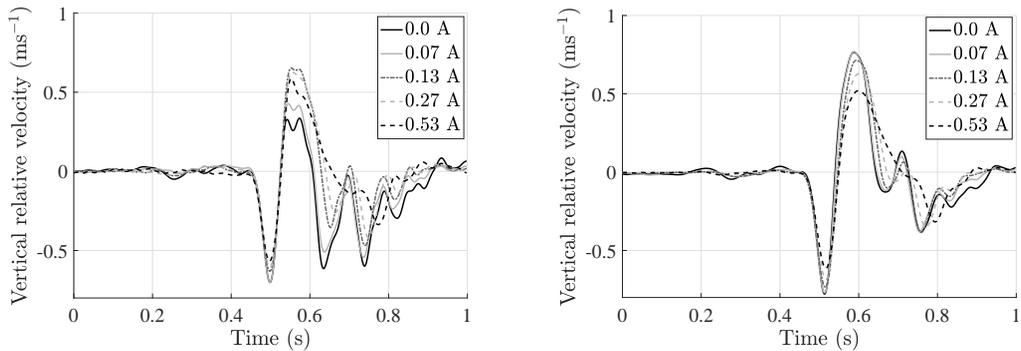


Figure 7: Vertical relative velocity of the left front part of the suspension estimated for different control currents using: accelerometers (left), LVDT (right).

5. Results of vibration attenuation

Effectiveness of vibration attenuation was tested for both methods of the relative velocity estimation. Control current was calculated according to equation (2) independently for each MR damper. Generally, the ride comfort is assessed on the basis of the acceleration acting on the driver or the passenger, so we propose to evaluate the effectiveness of control algorithms using a signal from IMU situated near the driver seat. The main problem in implementation of the Skyhook algorithm is the proper choice of the gain δ . The mathematical model of the system is non-linear and the optimisation procedure for δ selection is very difficult to perform, so the gain factor in (2) was determined experimentally by the trial and error method. For each part of the suspension δ was changed within the range defined on the basis of previous research. It was found that the best results for control algorithm using estimation of the relative velocity based on acceleration sensors can be obtained for δ equal approximately 3000, whereas for estimation based on the LVDT sensors the gain factor δ was a little smaller and was 2500. However, it was also stated that effectiveness of vibration attenuation is not very sensitive to δ , which can vary over a fairly wide range.

Exemplary results of the acceleration measured by IMU obtained for both methods of the relative velocity estimation are shown in Fig. 8. It can be easily observed that the maximum value of the acceleration, having the greatest impact on the feeling of comfort ride, reaches about 18 ms^{-2} for control using accelerometers, whereas it reaches about 12 ms^{-2} for control using LVDT. Thus, this confirms the supposition that using LVDT sensors for estimation of the relative velocity may improve effectiveness of vibration attenuation in this case.

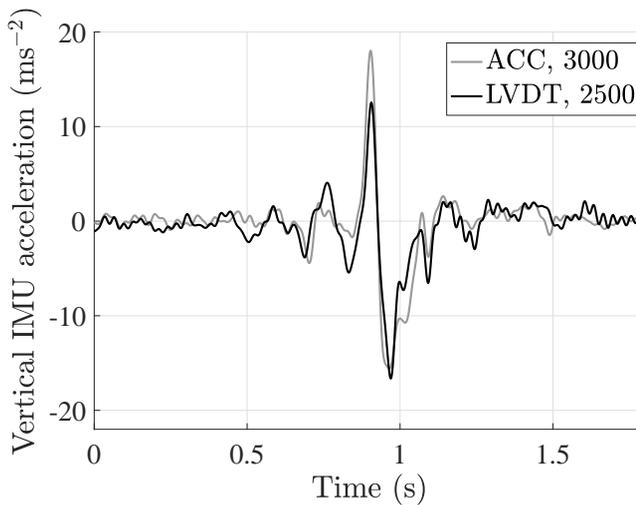


Figure 8: Acceleration measured by IMU for both methods of relative velocity estimation, averaged over different vehicle rides

6. Conclusions

In this paper vibration control in the semi-active automotive suspension using two ways of the relative velocity estimation has been considered. One method represents classical approach based on subtracting the velocity of sprung and unsprung masses, estimated on the basis of signals acquired from accelerometers. The other method uses signals from LVDT units. It was shown that LVDT-based estimates are less susceptible to measurement noise and using them in the inverse model of the MR damper to calculate the control current may lead to better vibration attenuation of the vehicle body. It also seems that estimation of the body velocity required in equation (1) based on some kind of combination of signals from accelerometers and LVDT sensors may improve the results of vibration control. This will constitute the next stage of the research.

References

- [1] S. BUDZAN and J. KASPRZYK: Fusion of 3D laser scanner and depth images for obstacle recognition in mobile applications. *Optics and Lasers in Engineering*, **77** (2016), 230-240.
- [2] A. CHAMSEDDINE and H. NOURA: Control and Sensor Fault Tolerance of Vehicle Active Suspension. *IEEE Trans. on Control Systems Technology*, **16** (2008), 416-433.

- [3] D. KARNOPP, M.J. CROSBY and R.A. HARWOOD: Vibration control using semi-active force generators. *J. of Engineering for Industry*, **96** (1974), 619-626.
- [4] J. KASPRZYK, J. WYRWAŁ and P. KRAUZE: Automotive MR damper modeling for semi-active vibration control. *Proc. of the IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Besancon, France, (2014), 500-505.
- [5] J. KASPRZYK, S. BUDZAN, J. RZEPECKI, R. MAREK and P. KRAUZE: Vibration control in semi-active suspension system of the experimental off-road vehicle using information about road roughness. *Proc. of the 20th KKAPD*, Zakopane, Poland, (2016).
- [6] P. KRAUZE and J. KASPRZYK: Comparison of linear and nonlinear feedback control for a half-car model with MR dampers. *Proc. of the 21 IEEE Int. Conf. on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, (2016), 965-970.
- [7] P. KRAUZE and J. KASPRZYK: Mixed skyhook and FxLMS control of a half-car model with magnetorheological dampers. *Advances in Acoustics and Vibration*, (2016), Article ID 7428616, <http://dx.doi.org/10.1155/2016/7428616>.
- [8] B. SAPINSKI: Magnetorheological Dampers in Vibration Control. AGH University of Science and Technology Press, 2006.
- [9] S.M. SAVARESI, C. POUSSOT-VASSAL, C. SPELTA, O. SENAME and L. DUGARD: Semi-Active Suspension Control Design for Vehicles. Elsevier. 2010.
- [10] Z. YANKUN, C. HONG and N. WEYRICH: A single camera based rear obstacle detection system. *IEEE Intelligent Vehicles Symp.*, (2011), 485-490.
- [11] G.Z. YAO, F.F. YAP, G. CHEN, W.H. LI and S.H. YEO: MR damper and its application for semi-active control of vehicle suspension system. *Mechatronics*, **12** (2002), 963-973.
- [12] S-J. YU, S.R. SUKUMAR, A.F. KOSCHAN, D.L. PAGE and M.A. ABIDI: 3D reconstruction of road surfaces using an integrated multi-sensory approach. *Optics and Lasers in Engineering*, **45** (2007), 808-818.

Towards emergence phenomenon in business process management

MACIEJ KORYL and DAMIAN MAZUR

A standard solution regarding business process management automation in enterprises is the use of workflow management systems working by the Rule-Based Reasoning approach. In such systems, the process model which is designed entirely before the implementation has to meet all needs deriving from business activity of the organization. In practice, it means that great limitations arise in process control abilities, especially in the dynamic business environment. Therefore, new kinds of workflow systems may help which typically work in more agile way e.g. following the Case-Based Reasoning approach. The paper shows another possible solution – the use of emergence theory which indicates among other conditions required to fulfill stimulation of the system (for example the business environment) to run grass-roots processes that lead to arising of new more sophisticated organizing forms. The paper also points the using opportunity of such techniques as the processing of complex events to fulfill key conditions pointed by the emergence theory.

Key words: business process management, adaptive case management, emergence.

1. Introduction

Standardization work in the area of business process management began in the early 90s of the last century. A key role in the initial phase of the field development played the organization called *Workflow Management Coalition (WfMC)*, which defined basic terms and first rules. Initially, the issue was viewed in a relatively narrow meaning as ‘task flow management’ and ‘document flow management’. In [8] a definition appears, which shows the then understanding of the issue: it is understood as automation of the business process as a whole or a part, in the time of which documents, information or rules are directed from one participant to another to perform an action resulting from the defined rules. In the consecutive years, the meaning of the terms was extended to encompass global perspective on business processes in an organization, seen mainly from the angle of achieved goals. Today, organizations such as *Object Management Group* or the abovementioned *WfMC* agree that the notion of business processes modelling should be analysed more broadly which is reflected by the definition included among others,

The Authors are with Rzeszow University of Technology, W. Pola str. 2, 35-959 Rzeszow, Poland. E-mails: maciej.koryl@gmail.com, mazur@prz.edu.pl. D. Mazur is the corresponding author.

Received 19.12.2016. Revised 10.05.2017.

in the [16]: business process management is a discipline that deals with the topics of modelling, automation, manufacturing, control, monitoring and optimization of the flow of business tasks in order to reinforce the fulfilment of organization goals, with participation of systems, staff, customers, and co-workers within and outside the boundaries of an organization. The new perspective except for clear orientation towards process aim considers the factors that were not present before:

- clients of organization are also active constituent of the considered system and take part in shaping the processes,
- the notion of action flow should be interpreted in a free way – the order of actions may be defined, but it can also be undefined.

Using the nomenclature applied in software engineering, management of processes in the enterprise should be ‘agile’ – more strongly directed to the goals of an organization and aims realized by the clients, rather than performing exactly defined procedures. Following this direction also influences the software systems designed to aid automation processes realized in a company. These systems change from the so-called workflow systems in the direction to more advanced applications, which could be called ‘business objectives realization systems’. Development of such solutions is a complex task that requires knowledge and experience from different areas. The following paper shows a proposition of using the theory of self-organising (or self-growing) systems, the emergence theory, in connection with modern techniques of information management such as *Complex Event Processing* to develop a system that supports functioning of companies that realize business processes in a dynamically changing environment.

2. Workflow processes as orchestration of the use cases

One of the methods of describing a process performed in an organization is a graphical notation that is usually specially formalized and extended version of the UML activity diagram. One of them is a notation called *Business Process Model and Notation* (BPMN), which allows defining actions which occur in the business process and all possible flows between them as well as events occurring during process realization (standard published in [2]). Formally, the model developed by using BPMN is a graph, the nodes of which are actions and arcs are the flows between them.

On the other hand, one of the ways to describe the required information system functions are *use cases* (UC). A use case describes the interaction between an actor (usually a system user) and the system in the form of a scenario that consists of well-defined actions attributed to one of the sides. Realization of a properly defined use case leads to creating business value within a wider process – it contributes to meet a fragment of requirements on the way to realize a process objective. Use cases may be defined by using different abstraction levels, from the most general ones that represent a business process, to the most specific ones that represent single interaction (action) in a process. The ability to

differentiate the abstraction level that is further described in [3] allows, in describing system reactions, assuming a level in which a singular use case will correspond to an action in the BPMN model. Such a model will then comprise a description of time relations between use cases. The BPMN model, apart from documentation and regulation functions relating to realized processes in a company, can also serve the function of an executable algorithm started using special software that aids workflow control. The action of defining the process model for the purpose of further automation using a control tool is called process orchestration and the person who is responsible for its realization is a 'process engineer'.

3. Decision based on defined rules

The theory of use cases (e.g. [3]) enumerates about ten elements that may be used to describe the use case. However, it may be assumed that the minimum set of information necessary to comprehend and implement an UC consists of the following elements:

- name of the use case – short name that describes well the aim of use case realization,
- initial conditions – system conditions required to start the UC,
- final conditions – system state after realization of the UC,
- basic scenario – a typical scenario that leads to fulfillment of the UC goal,
- alternative scenarios (extensions) – possible alternative flows of interaction that lead or lead not to the fulfillment of the UC goal.

In result of the realization of business process aims (i.e. realization of a subsequent UC that comprise the process) the system condition changes. The aim of the process realization is to lead the system to a particular, desired final state. For example, in the hereafter analyzed process of credit sale, such a state may be a signed credit agreement (a legal act) as well as the balance on the customer's account. The aim of the process engineer is to create a model that will include such range of use cases that after the realization of the last of them the process goal will be fulfilled (in other words – the final conditions of the last analyzed UC will correspond to target system conditions). A key issue that accompanies orchestration of use cases is the order of their realization. The basic limitation that affects the UC order is the necessity to guarantee fulfillment of initial conditions defined for each of them. This rule, in most cases, does not lead to one solution (unambiguous model), therefore the process engineer, while defining the process flow, has to use additional indicators and external knowledge and quite often has only intuition at his disposal. The consequence of the UC ordering is finding a proper business rule that is a function of the process state the execution of which serves as a

trigger of the flow to the right direction to the next action (i.e. running of the next UC). The defined rules will be implemented in a decision node that follows directly after the realized action thanks to which it is possible to define clearly the flow of further processes. Designing the right order of UC as well as finding rules that enable automation of flows between them is a complicated task that requires extensive experience. Most often a system started and been working according to a specified model requires further observation and model correction by empirical research. The method of workflow control based on the rules mentioned above is called *Rule-Based Reasoning* (RBR).

4. Cloud of use cases

A business process of a cash credit sale will serve as an example for further discussion. Servicing of cash credit sale is characterized by relatively high requirements concerning minimization of servicing time, in comparison to the sale of other financial products (e.g. mortgages). This requirement implies the necessity to implement software that ensures a high level of automation in the process. Fig. 1 shows set of use cases that occur in the discussed process as well as a set of roles that are taken by actors involved in the process of selling credits. Relations between actors and use cases indicate potential doers of particular activities, but such a relation does not have to unambiguously indicate which of the roles are responsible for the realization of the UC: the model that was shown assumes, for example, that the credit scoring may be done both by a consultant and a credit analyst. This model also does not indicate the order of the UC tasks realization and does not even introduce assumptions to the UC set the realization of which is necessary to bring the system to the final condition.

5. Ordering

When considering a typical approach to business process modeling, it is necessary to order use cases in one coherent process spanning from the initial moment (event) to the final event. In the case of a more complex process, there usually exist many possible solutions that differ from each other regarding the order of the realized actions. If initial conditions of each use case were empty (use case may be applied in any state of the system), $17!$ (i.e. about $3,5 * 10^{14}$) solutions would exist in the discussed process. Even if relatively few items may be ordered in a different way, a rich set of possible solutions exists (e.g. 5040 options for seven free items). Such a set cannot be practically applied and evaluated in a real business environment. The criteria defined by using *Key Performance Indicators* (KPI) are useful while choosing a proper variant. Examples of such indicators are: total time of process realization, staff workload (total work time), number of people involved, client load (amount of information gathered, consultation time), share of positive conclusions (number of the clients acquired), share of negative decisions in the

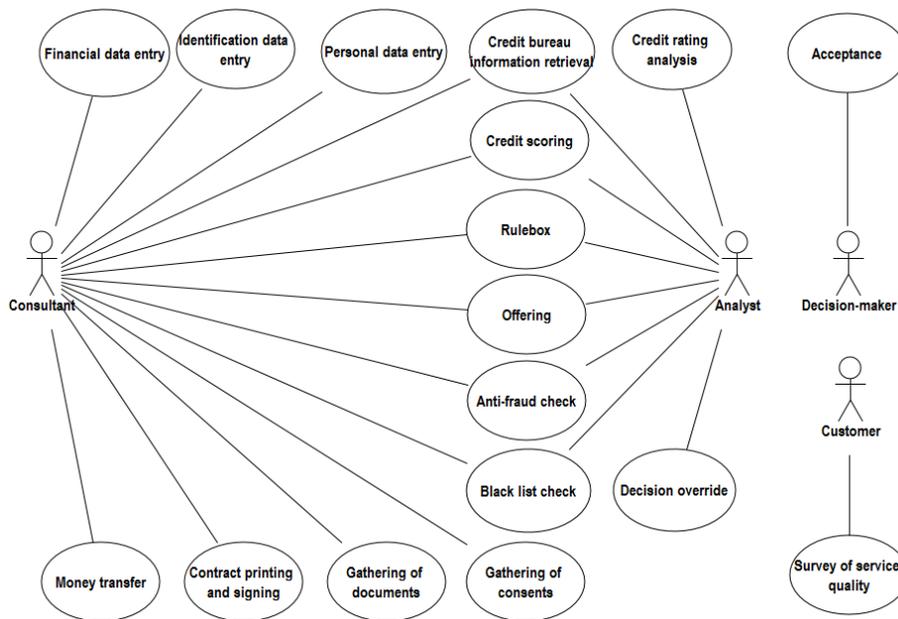


Figure 1: Typical use cases in the process of credit sale

confirmation stage, staff workload in relation to the number of the acquired contracts, business value of the acquired contracts, quality of contracts (measured, for example, by the share of credits in worse situation after the defined time), client's satisfaction (expressed by means of answers to questions in a questionnaire) etc. A task of deciding on the process model is usually realized by bank workers in a proper business role focused on doing analyses and implementation of optimal solutions (process engineer, business analyst, etc.). Figs. 2-4 show different variants of ordering several initial actions in the discussed process.

One of the KPI which will have a significantly different value in each of the exemplary variants is an indicator that can be called 'offer preparation time' and defined as the time between the start of the process and display of the list of offers prepared for the client. It will be the longest (the least beneficial) in the first variant and the shortest in the second one. On the other hand, the indicator that was called 'share of acceptances in approval center' will probably be the most beneficial in the first variant. Maybe the last variant would allow reaching the optimal solution, but it only is a hypothesis.

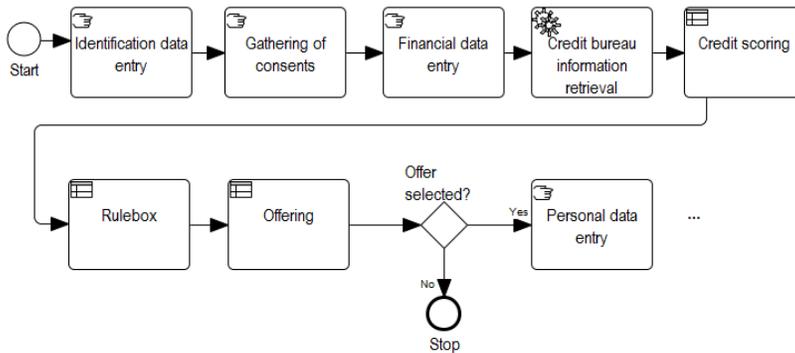


Figure 2: Variant 2 - 'defensive'

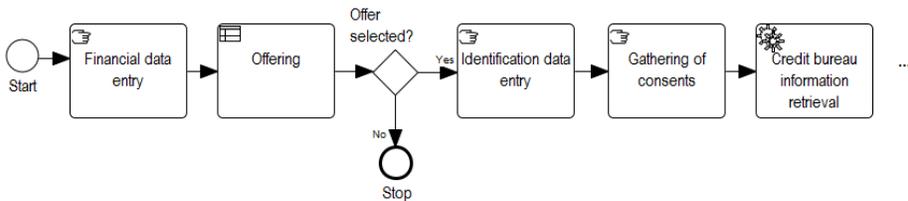


Figure 3: Variant 1 - 'offensive'

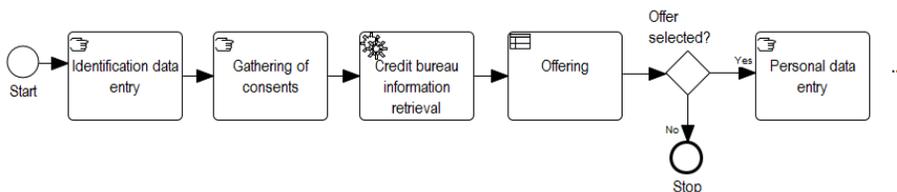


Figure 4: Variant 3 - 'maybe balanced'

6. Current trends compared to proposed solution

Nowadays, in the area of workflow management yet another type of alternative solution in comparison with the RBR approach is distinguished – the approach based on business cases called *Case-Based Reasoning* (CBR), in which problems are solved by re-using previous solutions to similar problems, after their necessary adaptation to the new situation. The authors of [17] propose combining the two techniques: solutions found within CBR serve to build rules within RBR. Early freezing of definitions of processes should be rejected, and observation, learning, and continuous adjustment should

be practiced when knowledge and experience of the organization grows. Recent years have brought very interesting practical solutions in the area of non-formalized approach to process management. The *Adaptive Case Management* (ACM) approach [11] and *Case Management Model and Notation* (CMMN) standard have been elaborated, and first management tools which implement such idea have also appeared. It should be recognized as an important and a valuable step in the direction of ‘paradigm switch’ in process management domain. The ACM approach ends with a process-centric and automatic way of problems solution towards knowledge- and communication-based one. The approach proposed in the current paper is close to CBR as both methods can be classified as ‘agile’ methods of process management in which a key role is played by actors involved in actions as well as the function of feedback and knowledge extension by experience. The ACM is used as a part of proposed practical solution, but one more step is suggested on the way towards the new kind of automation of process management. Another modern direction in BPM related to proposed approach is process mining technique [1] developed as a part of data mining area. One aspect of process mining is a control-flow discovery, i.e., automatic construction of the process model (e.g. a Petri net, BPMN graph) describing the causal dependencies between activities. Some parts of the solution proposed have the same functions as the process mining technique, so it seems to be useful for the future development to include that technique as a part of the solution. The insights provided by approaches similar to proposed are very valuable for the development of the next generation of Process-Aware Information Systems (PAIS) [5]. The PAIS are defined as a software system that manages and executes operational processes involving people, applications, and information sources based on process models. The system proposed may be partially treated as a kind of PAIS, but it seems to exceed PAIS definition since it has constituents which have non-process nature. After research of similar areas, it may be said that proposed solution consists of elements that are under current development, but they are not tied together to challenge such holistic approach as proposed.

7. The next step – emergence

According to the [6] emergence refers to what happens when a system of mutually related and relatively simple elements organizes itself, showing more intelligent better-adapted responses of a higher level. This definition points the increase of the system ‘intelligence’ thanks to which there appear more complex ‘behaviors’. Referring to the science of knowledge theory, the increase of intelligence can be treated as a simultaneous increase of both declarative knowledge (‘knowledge that’) and the procedural knowledge (‘knowledge how’). It is an important advice from the perspective of further discussion related to the implementation of the theory of emergence in a software system.

Systems organize themselves when relevant conditions are met. In [6] five of them are distinguished:

- 1) the system consists of a large number of actors (“All we need is thousands of individuals and a few simple rules of interaction”),
- 2) actors receive feedback from the background,
- 3) between actors there occur constant and free communication (in this case communication that is incidental and not planned),
- 4) actors have the ability and skill of recognizing recurring patterns,
- 5) no authoritarian control exists, instead of it, indirect control regulates the system.

Can those conditions be met in the case of an environment in which business processes are realized? A problem occurs already in the first condition – the size of the system measured as the number of actors performing actions. Who is an actor in case of a business process? If we assume that according to [16] actors (and a part of the analyzed system) are both representatives of organizations that offer products, as well as clients that use the offer, then the condition of involved participants in typical business environments will be fulfilled in most cases. The necessity of access to information feedback (the second condition) imposes requirements connected with monitoring of the information system. The system must compute in a continuous way, indicators that allow for conclusions about the effectiveness of the decisions being made (the previously mentioned KPI indicators) and the results must be known to the process participants. They also have to know what is the connection between actions that are realized by them as well as the decisions made and values of indicators and what influence on their unit benefit or public benefit have particular values of indicators. The condition connected with the existence of feedback (negative or positive) may be met by providing appropriate functions in the software. The third condition concerns communication between actors. This condition is fulfilled in two manners: through direct contact between actors and using relevant communication functions of the information system. In contemporary companies, collaboration is often remote and therefore the condition concerning free communication encounters obstacles. Thus, there appears a particular requirement for the software: it has to effectively provide all relevant communication functions for the analyzed issue on the level not worse than direct communication between involved parties. The fourth condition is relatively the most difficult to fulfil because it requires not only to ‘design’ the system appropriately (to decide on proper structure and communication mechanisms) but it requires from the system certain ‘computing power’, that will allow for effective recognition of recurring sequences of actions and decisions in relation to the result of the whole process or even results of the whole population of processes. Fulfillment of this requirement requires, in a special way, the introduction of support by using conceptual and software tools such as hereafter described *Complex Event Processing* approach. The last condition means both lack of direct control (there is no specified actor who manages the process realization) as well as the necessity of the occurrence of certain ‘bottom-up’ forms of control: in the system there must exist

mechanisms of mutual control that will eliminate unwanted reactions, namely those that obstruct its growth. An example of such mechanisms in software systems are systems of opinions and comments that enable to eliminate individuals that use rules not accepted by the community.

8. Emergence of processes

Let us assume that the environment that realizes the business process of cash credit sale has been organized in such a way that conditions which stimulate emergence are fulfilled:

- 1) there is properly large number of actors (bank workers, customers),
- 2) the system for sales support calculates and presents effectiveness indicators in a continuous manner,
- 3) there functions a system of communication between actors that is easily accessible and convenient in use as it 'encourages' making decisions,
- 4) the system for sales support searches recurring behaviors of actors in a continuous manner, it can relate actions taken in more complex sequences, and it attributes the required effectiveness indicators to them,
- 5) there exists no 'top-down' management of task completion; neither a person nor software system allocates tasks to do. Actors themselves choose tasks to do within the goal they strive to achieve. There are also no mechanisms for 'top-down' evaluation of task completion. At the same time, there are systems implemented to allow for 'bottom-up' control (e.g. system of comments).

What may emergence effects be expected in such a system? What 'higher organization forms' will appear here? The first element which we expect is self-organisation of the process flow. The system must acquire the final state that is known and well described using relevant rules. It is achieved as result of the realization of specific (formalized) tasks that belong to a certain set, but the order of their realization is not fixed. There is also no requirement concerning completion of all the tasks from the set because also the realization of a sub-set can lead the system to the target state. Regarding process self-organization, we may expect the following recurring elements to emerge:

- sequences of actions within one role,
- decisions in connection with a particular system state,
- sequences of actions and decisions encompassing many roles,
- processes as wholes.

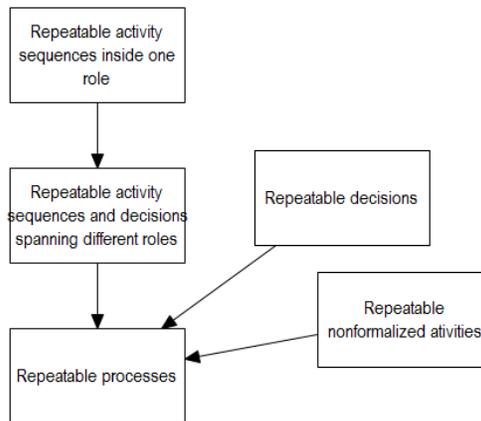


Figure 5: Emerging elements

The second effect that we should expect is the emergence of new activities that do not appear in the set of defined sentences that will be attached to sequences on the same basis as formalized tasks (Fig. 5). These can encompass recurring communication acts or other activities with a clear structure that are not formalized yet. Such a situation takes place mainly in case of growing business needs the meeting of which is not possible by using the system component elements designed until that point.

An effective conceptual and technical tool that serve to detect recurring patterns is *Complex Event Processing* (CEP). It is an approach that encompasses methods of tracking and analyzing sequences of events and finding relationships that appear as well as using the recognized relationships for further deductions and decision making. [9] is most often recognized as the first complete description of the approach. Integration of the business processes realization software with CEP software enables on-going analysis of the stream of events that take place in a process to find regularities according to the defined rules. In the case of the discussed issue, there are several categories of patterns that may be subjected to detection by using CEP, such as:

- detection of frequently recurring simple sequences of actions (without branches and decision nodes),
- detection of recurring decisions connected with the process state and information included in registered data,
- detection of action sequences tied the KPI indicators (e.g. sequences that lead to the most beneficial value for a particular indicator or combination of indicators).

The work of the rule-based CEP engines often leads to the development of declarative knowledge base of the system which can subsequently be used as feedback for the

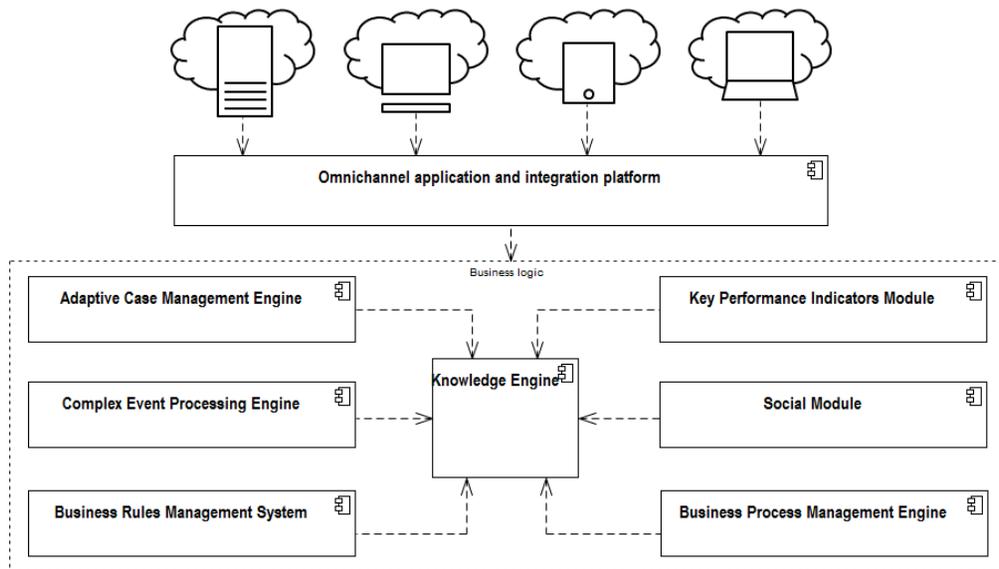


Figure 6: Proposed components of the system

support and automation of actions realized by emerging procedural knowledge. The simplest example of applying such knowledge is a system of hints for the most beneficial ways of process continuation or even automation of the realization of next actions in the case when (according to the defined rules) the system detects domination of one of the paths.

9. System stimulating emergence

The open question is what set of functions should support information technology system which would help the community to develop emergent behavior. The first attempt to describe such software might be made by taking the emergence conditions mentioned above and by carrying out analysis how to satisfy each of them, keeping at the same time additional goal that coherent and useful software must come into existence. Thanks to the use of such analytical approach, the first sketch of the required system appears. It is shown in the form of the component model in Fig. 6. To confirm the correctness of the analysis and that the software meets expectations, an empirical experiment in the proper business environment would serve. In the next subsections, the responsibility of each component is shortly described.

9.1. Omnichannel application and integration platform

The *omnichannel* business model, described in-depth in [15], is at the moment promoted as the most appropriate way of cooperation between customers and service providers in nowadays business conditions. The omnichannel approach may be seen as the ability to be in constant contact between parts through multiple communication channels at the same time where the same data and process information are accessible and coherent. Such capability should also be present in discussed system as its primary function which would enable access to the other more specialized functions. This part of the system provides a bi-directional asynchronous connection between front-end multi-technological applications handled by actors engaged in communication and core modules described below. Such a function supports the first condition of emergence phenomenon i.e. involving large enough number of process participants.

9.2. Adaptive Case Management Engine

This module acts as the main process engine allowing actors to do their business tasks, giving them the opportunity to optimize the way of doing that and to adapt their behavior when conditions are changing. The ACM engine enables to shape own actor's path to the final goal and at the same time tracks his steps and builds a database for further analysis and reasoning which is provided by the *Knowledge Engine* component. Thanks to the ACM module, 'bottom-up' model of management is accessible and mentioned above the fifth condition of emergence may be satisfied.

9.3. Complex Event Processing Engine

The CEP approach enables to find recurring patterns in the stream of business events. Several types of pattern matching approaches are described in [9] and implemented in a real software and many of them have found their practical implementation in manifold software applications (interesting examples are described in [10]). In discussed system, the CEP engine recognizes patterns in actors' behavior tracking sequences of chosen tasks, makes abstractions of those events and sends it to the *Knowledge Engine* component. That function is required to fulfill the fourth condition of emergence mentioned above, i.e. recurring patterns recognizing skill.

9.4. Business Rules Management System

BRMS software typically stores, executes and monitors some business logic which may be externalized from software code-base and described using special executable notation. In process management systems such kind of software usually supports the description of the logic of decision nodes where the outcome depends on the state of the business case under processing and sometimes depends on the state of the external environment. Described system uses BRMS engine to store and execute decision logic made by actors. It is an open and non-trivial issue how such automation may be provided and leads to a more fundamental question about action logic recording.

9.5. Key Performance Indicators Module

The second condition of emergence listed previously concerns continuous feedback about the efficiency of decisions made and activities taken. A typical solution for efficiency measurement in BPM systems is the calculation of *Key Performance Indicators* and further analysis based on them. Special parts of business applications usually visualize such indexes and let to reason and adapt procedures of action by them. Adjustment involves qualified ‘process engineers’ and in the most cases is not automated. Discussed solution, besides visualization of KPIs, have to automate their application and therefore values of KPIs are sent to *Knowledge Engine* module where they are combined and confronted with decisions made and activity sequence patterns found.

9.6. Social Module

The *Social Module* provides a convenient way of performing communication acts between parties involved in cooperation. Both service providers and customers can easily contact each other, ask the question, give advice, make a proposal or just share experience and knowledge. This module serves not only as popular internet communicator but it works in the context of current business activities carried out by parties and registers and classifies such communication acts enabling further reasoning provided by the *Knowledge Engine*. Tactics governing this extended social module may be theoretically aided by *Speech Acts Theory* introduced in [14] and then broadly developed by many authors and practically adopted in agent-like software.

9.7. Business Process Management Engine

The BPM part of discussed system runs re-usable sequences of tasks, which are products of emergence phenomena. This module is also used to define embedded micro-sequences having a rigid structure which represent algorithms of single activities and as a whole are called from ACM engine.

9.8. Knowledge Engine

A module called *Knowledge Engine* is the central part of the system which main responsibility is to find symptoms of emergence phenomena and to utilize their power. It derives knowledge from other modules and makes it usable. For example, it finds regularity in task sequences generated by choices of actors by confrontation with combined values of measured KPIs. Regular and most valuable patterns are then used as a proposition of future choices with a constant evaluation of gathered effects if such hint was utilized. An intrinsic part of this module is knowledge database built by increasing experience of the system. Very interesting issue, which requires further study, is how to represent procedural knowledge making it useful for process automation.

10. Final remarks

An interesting direction that can be used in further development of the proposed approach is knowledge extension techniques which besides feedback make use also of possible simulations of process continuation as well as evaluation of the results of simulated actions before taking real actions. This idea was described in [2], where the technique was called ‘projective simulation’. An additional advantage of this approach is the introduced randomness and probabilistic evaluation which gives additional potential in extending knowledge by using solutions that were not selected and that can be selected due to lack of formal obstacles defined as initial conditions of use cases.

The proposed direction is not free from risk. First and foremost it assumes much deeper immersion of actors in the business environment than it takes place in typical commercial activity. Relatively high awareness of process participants and very high motivation is required to meet this assumption. It is necessary to meet complex technical requirements that enable free and satisfactory participation of actors in communication processes that are necessary for the system to develop. Thus, it is necessary to use modern, often mobile equipment and develop sophisticated software that will provide high-quality information and its usage would be as natural as in the case of bottom-up processes occurring in nature.

References

- [1] W.M.P. VAN DER AALST: *Process Mining. Data Science in Action*, Springer, 2016.
- [2] H.J. BRIEGEL and G. DE LAS CUEVAS: *Projective Simulation for Artificial Intelligence*. Scientific Reports, 2012.
- [3] A. COCKBURN: *Writing Effective Use Cases*. Addison-Wesley Professional, 2000.
- [4] G. DI MARZO SERUGENDO, M-P. GLEIZES and A. KARAGEORGOS (Ed.): *Self-organising Software. From Natural to Artificial Adaptation*. Springer-Verlag Berlin Heidelberg, 2011.
- [5] M. DUMAS, W.M.P VAN DER AALST and A.H. TER HOFSTEDÉ: *Process Aware Information Systems: Bridging People and Software Through Process Technology*. Wiley-Interscience, 2005.
- [6] S. JOHNSON: *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*. Touchstone, New York, 2001.
- [7] H.D. JORGENSEN and S. CARLSEN: *Emergent Workflow: Planning and Performance of Process Instances*. *Proc. of the 1999 Workflow Management Conf. – Workflow-based Applications*, (1999).

-
- [8] P. LAWRENCE (Ed.): *Workflow Handbook 1997*. Workflow Management Coalition. John Wiley & Sons Ltd, 1997.
- [9] D. LUCKHAM: *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, 2002.
- [10] D. LUCKHAM: *Event Processing for Business. Organizing the Real-Time Enterprise*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2012.
- [11] H.R. MOTAHARI-NEZHAD and K.D. SWENSON: Adaptive Case Management: Overview and Research Challenges. *IEEE Int. Conf. on Business Informatics*, (2013).
- [12] OBJECT MANAGEMENT GROUP (OMG): *Business Process Model and Notation (BPMN)*. <http://www.bpmn.org>.
- [13] A. RANTRUA, M. GLEIZES and C. HANACHI: Flexible and emergent workflows using adaptive agents. *ICCCI 5th Int. Conf. on Computational Collective Intelligence, Technologies and Applications*, (2013).
- [14] J.R. SEARLE: *Speech Acts*. Cambridge University Press, 1969.
- [15] P.C. VERHOEF, P.K. KANNAN and J.J. INMAN: From multi-channel retailing to omni-channel retailing. Introduction to the special issue on multi-channel retailing. *J. of Retailing*, **91**(2), 2015.
- [16] M. VON ROSING, H. VON SCHEEL and A. SCHEER: *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM*. Morgan Kaufmann, 2014.
- [17] B. WEBER and W. WILD: *An Agile Approach to Workflow Management*. Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn, 2012.

Active resources concept of computation for enterprise software

MACIEJ KORYL

Traditional computational models for enterprise software are still to a great extent centralized. However, rapid growing of modern computation techniques and frameworks causes that contemporary software becomes more and more distributed. Towards development of new complete and coherent solution for distributed enterprise software construction, synthesis of three well-grounded concepts is proposed: Domain-Driven Design technique of software engineering, REST architectural style and actor model of computation. As a result new resources-based framework arises, which after first cases of use seems to be useful and worthy of further research.

Key words: domain-driven design, REST, actor model.

1. Introduction

Enterprise software systems working in deployment environment of huge corporations such as banks or industrial plants consist of many separate products, typically from several to several dozen parts. One software product serves from hundreds to above thousand use cases and at the same time interacts with use cases of other products due to automation of complex business processes. As regards the computation character, two types of processes may be listed:

- processes of interactive character, often automated by usage of workflow tools, responsible for entire process composition from atom elements representing well defined and cohesive activities. Process composition in such manner is called orchestration;
- processes of batch character, consisted of processing fragments one following another or one running parallel with another, typically iterated on collections of business objects characteristic for particular area, such as contracts, transactions, orders, etc. Nowadays implementations of such batch processes are supported by modern software frameworks dedicated to that purpose.

The Author is with Rzeszow University of Technology, W. Pola str. 2, 35-959 Rzeszow, Poland, e-mail: maciej.koryl@gmail.com

Received 18.12.2016.

In both cases, constructed processes consist of many functions working in close cooperation, often exposed as APIs of systems or APIs of their components. As long as cooperation is carried in synchronous way, complexity of such arrangement may be controlled, even if number of involved systems is substantial and number of interactions is high. In synchronous systems number of possible states in which system may occur is countable and predictable at the stage of software designing or detectable during testing. After applying asynchronous model of communication, complexity of system violently grows with increase of possible different states, which number is non-linear function of possible states of constituents of the system and number of interactions between them ([4]). During many years of computation theories development and many years of software engineering implementation practices, meaningful conceptual and technical tools were established, but that does not mean that problem of complexity has passed away or even has been minimized to notable degree. The matter is broadly recognized in specialized computation areas dealing with well-established algorithms, but still is not enough captured in commercial products development, govern by its own specificity connected with high number of software users, many different business objects and huge number of unpredictable interactions (sample characteristic of such systems is shown in [8]). In these days, problem is more and more complicated because of limitations of monolithic systems and the need for introduction of distributed software, for example in the form of microservices ([12]), which are adapted for horizontal scaling and well suited in actual hardware capabilities. Several conceptual and technical tools currently used in software engineering discipline, dedicated to distributed processing are described in [3], where one of them is an actor model of computation, which after many years of academic development, currently gains great popularity in commercial area. Proposition which is shown in this paper constitutes coherent and complete framework based on well-trying techniques and design patterns with actor model between them and has working implementation in Java programming language with use of modern tools for enterprise applications such as Spring Framework and noSQL databases. Proposed solution has been used to build some parts of banking transactional system, which supports batch processes such as massive transactions processing or interaction with trade platform. As a foundation of the idea, three engineering concepts act: Domain-Driven Design technique proposed in [5] and broadly accepted in software community, Representational State Transfer architectural style introduced in [6] and currently becoming the most popular way of interaction between web components, and the actor model of computation described in [10] nowadays gaining mature and useful implementations. In addition, the solution was enriched by use of standard language of agent communication in multi-agent systems – Agent Communication Language, which semantics was found as very suitable for required interactions.

2. Fundamental concepts

2.1. Resource as a central point of the REST model

Representational State Transfer (REST) is an architectural style implementing fundamental rules of the web and HTTP standard. REST was introduced by dissertation [6] and at present has obtained great popularity as ‘the web used correctly’. Central idea of the style is to treat all things which have identity as resources and give them globally unique Uniform Resource Identifier (URI). Resources named in this way may communicate together using hyperlinks. Communication is provided by usage of standard HTTP commands with their established semantics. Important rule of REST is that communication ought to be stateless, i.e. parties cannot keep state of communication assuming that next message will be continuation of previous one. In proposed approach the resource term plays key role as external representation of computation units and set of REST rules and good practices in interactions modeling are applied.

2.2. Aggregate in Domain-Driven Design concept

Domain-Driven Design (DDD) concept introduced in [5] is an approach to software development which pays attention to key meaning of domain model in software design. Domain model plays central role in whole process of development acting as universal medium of communication between all participants and providing stable base for software structure. DDD technique is divided into two stacks of patterns: strategic and tactical ones. First of them serves as toolset for taking control over complexity of extensive software and second consists of a set of building blocks, which is sufficient for complete design of each kind of enterprise software on some level of abstraction. The most important pattern from tactical stack is the aggregate building block and there is plenty of rules explained in literature, how to build useful aggregates (for example [18]). Aggregate is a graph of objects tied together into one coherent object offering common set of services for external world. The only way to access aggregate constituents capabilities is aggregate root, the central entry point to that software unit. Thanks to such construction, aggregate guarantees the consistency of changes of whole structure, controls its internal state and gives convenient way for its access. In proposed framework, aggregate plays important role as representation of stable state of a resource and also as a part of resource in dynamical state by offering its behavioral capabilities.

2.3. Actor model of computation

The actor model of computation developed many years ago and firstly published in [10] was thought as conceptual tool for understanding of concurrency. Many software frameworks based on actor model have been built to this day, but broad utilization in enterprise software area is still scarce. Currently, attention in that idea is growing, stimulated by development of multi-core processors and development of cloud computing solutions with necessity of computation distribution. Theory of actor model treats actors as universal primitives with capability to carry out each kind of needed computation

([9]). Actors are independent units of computation loosely coupled together, only by asynchronous messages passing and the only reliable knowledge which actor has about other actor is its mailbox address. When an actor receives a message it may do some computation, send messages to participants, create additional actors as its children or may change its own behavior preparing itself for future course of situation. In proposed framework, actors will support implementation of dynamical state of resource.

2.4. Agent Communication Language

Agent Communication Language (ACL) is a definition of standard language used in multi-agent systems to model conversations between involved parties. Its origins are in philosophical theory of speech acts ([15]), which state that each utterance has not only informative, but also performative function, i.e. causes consequences in receiver's activity. ACL has been drawn up by Foundation for Intelligent Physical Agents (FIPA) as FIPA-ACL set of standards [7]. In proposed solution ACL syntactics and semantics are used to model communication between resources, especially by use of standard vocabulary of performatives denoting the type of communicative acts.

3. Active resources model of computation

3.1. The active resources term

As 'active resources' are considered resources, which at moment of interaction may be under change originated from other interaction, computation processes or any other factor. As active resource is in unstable state and its properties may change in time, another object cannot assume that something is true about that resource, even if resource still exists or not. An observer cannot have knowledge about its state, but can have only some beliefs. For example, if some procedure in banking system completes payment and has information about sufficient balance of debited account, it cannot assume that payment will be successful. It ought to be ready for receiving information from target resource that operation has succeeded or not. As regards software design, active resource is represented by synthesis of three concepts:

- REST resource, which brings unambiguous global identification of resource and convenient language of communication for presentation and change of its state. It also provides availability of many technical frameworks ready for use for implementation of interactions;
- DDD aggregate, which gives a comprehensive way of modeling software external and internal structure, its behavior and rules forming objects identity. DDD technique also brings possibility of effective and cheap implementation thanks to help of modern frameworks for enterprise software such as Spring Framework [16] which was broadly used to implement proposed solution;

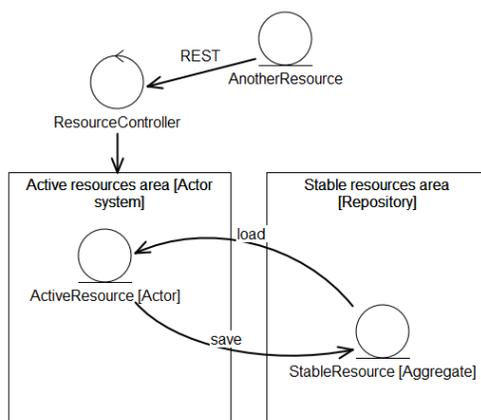


Figure 1: Two areas of resources

- actor, which offers its capability of long-term existence and sophisticated communication abilities. Utilization of actor model is possible and reliable due to existence of mature implementations such as Akka Framework [1], which was used with support of patterns based on it ([2, 18, 20]).

3.2. Two areas of resources

Any resource in the solution may stay in one of two states: stable state, when no change of its properties is possible and active state, when its properties may dynamically change. Therefore, symbolically two areas are distinguished: stable resources area and active resource area as was presented in Fig. 1.

When message to resource in stable area was directed, resource is moved to active area, where it acquires ability to act. If system detects that active resource is idle (does not perform any activity and has empty mailbox), resource may be removed from active area, but it depends on used strategy of supervising. In the system implementation these areas as represented by DDD repository pattern and by actor system respectively. For resource migration into active area, dedicated to such kind of resources, area supervisor is responsible. Sample body of supervisor's function of message handling is shown on Listing 1. Some essential comments have been placed in the code.

Listing 1: Supervisor's message handling function

```

public void onReceive(Object message) {
    if (message instanceof CreateResource) {
        // request to create new resource
        CreateResource msg = (CreateResource) message;
        // create active resource
    }
}
  
```

```
ActorRef activeResource =
    context().actorOf(componentNameProps(),
        msg.resourceName());
// and send message to the newborn in active state.
// It will be responsible for immediate
// creation of it's stable representation
activeResource.tell(msg, self());

} else if (message instanceof PerformAct) {
    // message to existing resource
    PerformAct msg = (PerformAct) message;
    // is resource in active state?
    ActorRef activeResource =
        this.getContext().getChild(msg.resourceName());
    if (activeResource == null) { // no
        // enter existing resource into active state
        activeResource =
            context().actorOf(componentNameProps(),
                msg.resourceName());
    }
    // send message to resource in active state
    activeResource.tell(msg, self());
}
}
```

4. Example of distributed batch processing supported by the new concept

Each processing routine in transactional system may be treated as active resource. Examples of such resources are: standing orders processing, interest calculation, interest capitalization, incoming or outgoing payments processing etc. Initialization of processing is therefore implemented as request for creation of resource of some kind. System ordering computation sends request to microservice which is responsible for handling such processing. Typically it will be microservice which owns resources being processed, for example customer contracts or registered payments. It also may be separate microservice as in example below, when processing involves different resources. Ordering system sends POST command with REQUEST performative and in return receives URI of active resource which probably will be created in the target system. Ordering system have to be ready to accept confirmation of resource creation sent by target system – the CONFIRM performative meaning that processing has been started. Thanks to received URI, ordering system is able to contact with active resource in target system and ask it about its state (QUERY_REF performative sent by GET command) or to order further requests, for example hold computation or abandon it (CANCEL performative sent by POST command). Ordering system should be ready to accept information from resource

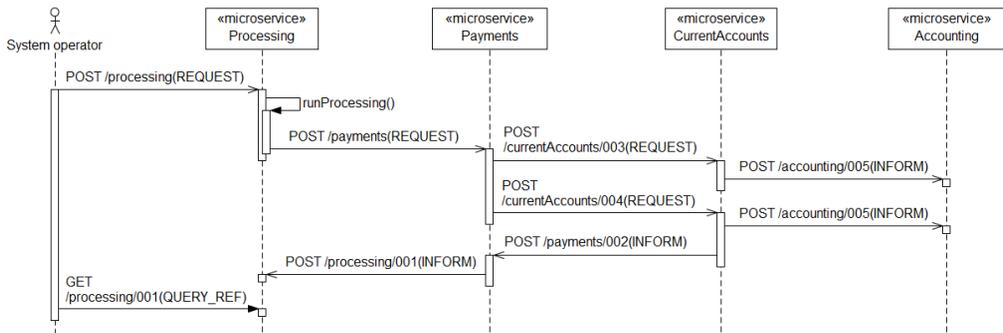


Figure 2: Sample processing supported by the new model

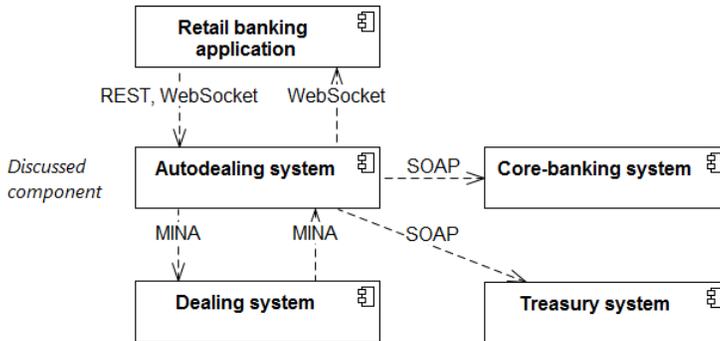


Figure 3: Component model of 'autodealing' example

(which may be sent as INFORM performative by POST command) or explicitly order such information (e.g. results of computation). Sequence diagram presented in Fig. 2 illustrates example of interactions between microservices working on some kind of processing in banking software. For clarity only single set of interactions for one transaction was shown.

5. Example of real-time cooperation and problems solved

Second example concerns real-time computation provided in enterprise system offering auto-dealing functionality, i.e. ability to make unassisted currency exchange transactions on trading platform (typically such transactions are supported by brokers). The whole solution consists of several components. The most important of them are shown in Fig. 3.

The first attempt to implement *Autodealing system* was made using typical service-oriented approach of stateless nature. Implementation and functional tests of the solution

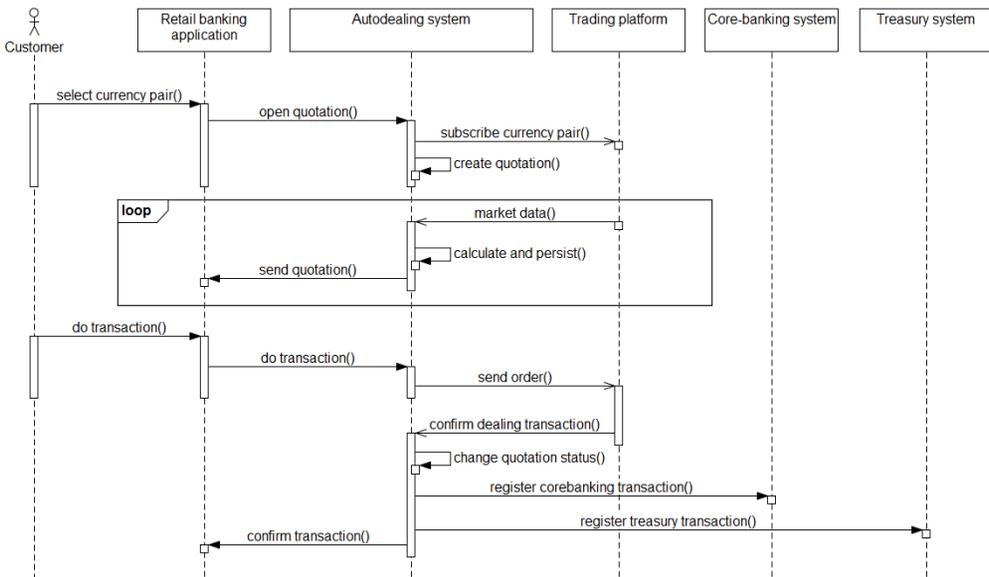


Figure 4: Real-time cooperation supported by service-oriented approach

revealed some problems of diverse origin. In the majority of cases functionally satisfying solution was found, but applied mechanisms were of different type and architecture of the system began to drift into worrying direction. Coherent mechanism was demanded and finally the active resources framework has filled the gap. The most interesting problems and solutions provided by the framework are described below. Description concerns key interactions in system which are shown in Fig. 4.

1. The first problem is connected with events ordering and occurs when first *market data* message arrives when quotation is still being created or persisted and doesn't exist yet. System cannot properly address *market data*. The simplest protection is to ignore *market data* if quotation doesn't exist, but user would wait unacceptably long time if frequency of messages is low (e.g. one per 10 second).
2. The other problem concerns resource access synchronization and appears when frequency of *market data* messages is high and new message arrives when previous is still being handled. System cannot synchronize processing by use of database transactions because remote invocations exist. The simplest way is to ignore incoming messages when the old one is being processed, but it may lead to unwanted business effect of losing the most beneficial offers. Some advanced queuing mechanism should be used to ensure proper order of servicing.
3. Another problem appears after user decides to finalize transaction and system sends *order* message to *Trading platform*. Until platform doesn't confirm transaction, still *market data* messages arrives to the system. In that state, such messages

have to be ignored. Typical solution is to use special *status* property of object and make behavior conditional on its value.

4. One of functional requirements is that user has to be informed about the result of dealing transaction immediately when confirmation from *Trading platform* is got, and the next parts of confirmation handling procedure (interaction with *Core-banking* and *Treasury* domain systems) should be done later. To fulfil such a requirement special tools for asynchronous processing have to be applied.
5. Since communication with domain systems may be unreliable, it should be repeated in controlled way if failure occurs, without blocking main use case scenario. Such a requirement causes that some equipment for background processing have to be used.
6. Until user doesn't select the most interesting currency pair and register private subscription in *Trading platform*, public rates are broadcasted to each observer. To do that, *Autodealing system* have to maintain global list of active observers of each currency pair or list of pairs for each active observer, and therefore special concurrent structure must be applied.
7. Mentioned above list of observers is a static structure, and it cannot be directly used to actively inform observers about communication problems happening, such as temporal unavailability of dealing or domain systems. Low amount of maintenance information causes discomfort and growing impatience when any problem appears.

The active resource framework naturally helps to solve the problems mentioned above.

1. Messages arriving at active resource are enqueued in its mailbox and each of them are processed in exclusive mode (i.e. at the moment resource services only one task). So, hazard cannot occur: creation of resource always would finish before next message is taken to service.
2. All messages in resource mailbox are ordered and are computed exclusively - this property blocks synchronization problem. In both cases (1. and 2.) prioritization function of resource mailbox plays important role: the functional requirement states that when newer message of the same type is already present in queue, it has higher priority and should be served first. Older message should be skipped in such case. Discussed solution supports that requirement.
3. Third problem may be solved by actor capability to dynamically change its behavior. After ordering of transaction, actor switches its handling function to the new one which accepts only confirmation and rejection messages.

4. Asynchronous processing is the normal way which active resource uses to perform all its tasks. In this example, resource can inform user about *Trading platform* answer and continue cooperation with other systems.
5. Repeated actions may be initiated and controlled by active resource in a few ways. Common pattern is to address message to itself: when error is detected, actor sends message to its own mailbox causing future effect of message processing. Since active resource carries its state, repeated executions may be controlled and, for example, ended after some number of failures.
6. Instead of keeping some kind of global structures storing addresses of observers, built-in function of discussed framework can be used: each type of resource has its own supervisor which keeps mailbox addresses of all its child resources in active state. This address book (which is an actor system function) lets properly broadcast required information to resources, and each resource can inform its front-end client.
7. Each active resource can actively react to problems appearing and can notify connected user in the same way as above. Thereafter it can send information update when conditions are changing.

Generally, one might say, that the new concept more naturally fits requirements of real-time transactional systems than typical service-oriented solution. Examples above point out that it solves many problems of different nature, simplifies software development and introduces coherence into its architecture.

6. DDD tactical tool-set extension

The set of design patterns contained in tactical part of DDD includes the aggregate as a central element and a few additional items. They belong to two sub-layers of business logic layer: *application* and *domain* one. The concept of active resources introduces new sub-layer including additional patterns responsible for active behaviour: the *actors* layer. Enriched model of DDD tactical patterns is shown on Fig. 5.

7. Final remarks

New concept of computation in enterprise software and examples of its application in some processing and real-time routines were presented. Similar solutions work in real banking software and currently is under further research and development, but first results are very promising. After a few first cases of use of the new solution, there might be told that characteristic features of development process based upon new framework are low cost of use cases implementation and low level of defects detected during quality

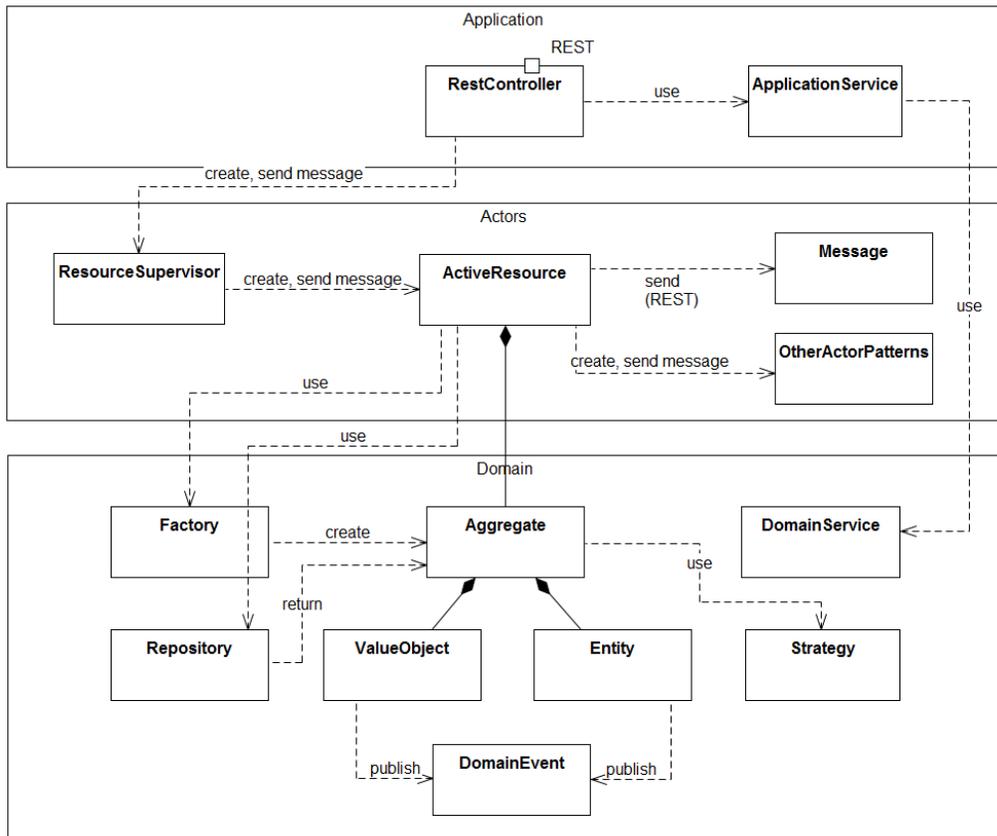


Figure 5: DDD tactical patterns enriched with *actors* layer

assurance phase. The concept of active resource is thought as a broader idea and may serve as fundamental framework able to maintain whole transactional solution. It may be applied to serve both kind of enterprise computation: batch processing, where active resource controls process, and computation of real-time or interactive character, which is more unpredictable and more demanded than batch processing, and active resource helps to control its complexity. Broader research is currently going on. Next planned stage is attempt to build complete new module of transactional system with dominance of resources of the new kind. Additional direction of theoretical and practical development would be step towards utilization of some more sophisticated ideas, such as emergent properties theory, speech acts theory and indeterminism, with hope, that they may help in better understanding of complex software processes.

References

- [1] Akka Framework, <http://akka.io>. Access 11.2016.
- [2] J. ALLEN: *Effective Akka*. O'Reilly Media, Inc., 2013.
- [3] P. BUTCHER: *Seven Concurrency Models in Seven Weeks*. Pragmatic Bookshelf, 2014.
- [4] K.M. CHANDY and L. LAMPORT: Distributed snapshots: Determining global states of Ddistributed systems. *ACM Trans. on Computer Systems*, **3**(1), (1985), 63-75.
- [5] E. EVANS: *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional, 2003.
- [6] R.T. FIELDING: *Architectural Styles and the Design of Network-based Software Architectures*. PhD dissertation, University Of California, Irvine, 2000.
- [7] FIPA Standards, <http://www.fipa.org/repository>. Access 11.2016.
- [8] M. FOWLER: *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2002.
- [9] C. HEWITT: *Actor Model of Computation: Scalable Robust Information Systems*. Cornell University Library, arXiv:1008.1459, 2015.
- [10] C. HEWITT, P. BISHOP and R. STEIGER: A universal modular actor formalism for artificial intelligence. *IJCAI'73 Proc. of the 3rd Int. Joint Conf. on Artificial Intelligence*, (1973), 235-245.
- [11] S. MILLETT and N. TUNE: *Patterns, Principles, and Practices of Domain-Driven Design*. John Wiley & Sons, 2015.
- [12] S. NEWMAN: *Building Microservices. Designing Fine-Grained Systems*. O'Reilly Media, 2015.
- [13] T. O'CONNOR: Emergent properties. *American Philosophical Quarterly*, **31** (1994), 91-104.
- [14] M. NASH and W. WALDRON: *Applied Akka Patterns*. O'Reilly Media, Inc., 2016.
- [15] J.R. SEARLE: *Speech Acts*. Cambridge University Press, 1969.
- [16] Spring Framework, <https://spring.io>. Access 11.2016.
- [17] G. SUKUMAR: *Distributed Systems: An Algorithmic Approach*. Chapman and Hall/CRC, 2014.

-
- [18] V. VERNON: *Implementing Domain-Driven Design*. Addison-Wesley Professional, 2013.
- [19] V. VERNON: *Reactive Messaging Patterns with the Actor Model: Applications and Integration in Scala and Akka*. Addison-Wesley Professional, 2015.
- [20] D. WYATT: *Akka Concurrency*. Artima Press, 2013.

Mobile devices and computing cloud resources allocation for interactive applications

HENRYK KRAWCZYK and MICHAŁ NYKIEL

Using mobile devices such as smartphones or iPads for various interactive applications is currently very common. In the case of complex applications, e.g. chess games, the capabilities of these devices are insufficient to run the application in real time. One of the solutions is to use cloud computing. However, there is an optimization problem of mobile device and cloud resources allocation. An iterative heuristic algorithm for application distribution is proposed. The algorithm minimizes the energy cost of application execution with constrained execution time.

Key words: mobile devices, computing cloud, task allocation, optimization.

1. Mobile cloud computing

The capabilities of mobile devices are increasing at a very fast pace. Average computing power of smartphone CPU increased almost 4 times between 2011 and 2014 [1]. Access to the Internet is a fundamental feature of mobile devices today, and the connection speed of GSM networks almost tripled in 2013 - average download rate was around 520 Kbps in 2012, and almost 1.4 Mbps in 2013 [4]. It is estimated that by the year 2018 there will be more than 7.4 billion mobile devices with constant access to 3G or 4G networks, and global traffic in these networks will exceed 15 exabytes per month. It is a result of the continuous growth of user demand, expecting direct access to Internet services, multimedia, social networks and others.

While the computing power, memory capacity, size and resolution of the screens, and number of various sensors in mobile devices grew rapidly, there is no similar development of battery capacity. For example, the first iPhone, presented in 2007, had battery capacity of 5180 mWh, and the iPhone 5s that debuted in 2013 had a 5960 mWh battery. The huge increase in computing power in mobile devices corresponded with only a 15% gain in battery capacity [20].

One of solutions for this problem is integration between the computing cloud and a mobile device [19]. Thanks to the new generation of mobile networks, applications are

The Authors are with Gdańsk University of Technology, Poland. E-mails: {hkrawk, micnykie}@pg.gda.pl. H. Krawczyk is the corresponding author.

Received 20.12.2016. Revised 29.04.2017.

able to transmit significant amounts of data to services in the cloud. The idea of using resource-rich remote machines to extend the capabilities of smartphones and tablets, allows one to run computationally expensive tasks, while using minimal amounts of device energy.

Many models and architectures for integrating mobile devices with the cloud were proposed in the literature [18]. They differ by scope of integration and method of application optimization – some were designed to maximize application performance [3], others focus on minimizing energy consumption. Some solutions require manual optimization, by annotating which components should be run on the cloud, and which on the mobile device [15]. There are also a few frameworks that are using multi-objective optimization [24]. Unfortunately, most of the proposed solutions require significant changes in the mobile operating system [21].

Analysis of existing solutions in the field of mobile cloud computing leads to the conclusion that there are a few common problems and challenges:

1. Implementing the mobile-cloud integration requires using specific architecture, patterns, compiler or language; so an existing application must be rewritten for the most part.
2. Application partition must be often done manually, and automatic methods usually take into account only one objective.
3. Software frameworks require significant modifications in the mobile device's operating system, which is not practical in real applications.

The solution proposed in this paper is an attempt to solve these problems and at the same time accommodate experience from existing research in the field. The main difference from frameworks like Clonecloud [3] or ThinkAir [11] is that the proposed solution focuses on energy optimization while keeping the execution time under a specified value. It also does not require any changes in the mobile operating systems, unlike the abovementioned Clonecloud [3] or Cloudlets [21]. In contrast to eXCloud [14] or μ Cloud [15], the optimization and allocation is fully automatic.

2. Interactive application model

Almost all mobile applications have some form of graphical user interface (*GUI*), and can be described as interactive applications; like games, social media, or different kinds of content editing applications. The user is performing actions during the whole life cycle of the application, inputs data, modifies the state in a nondeterministic way, and the application reacts to these actions. Optimization of this type of application is a bigger challenge than optimizing some batch programs, where the input is provided at the start of execution, and data flow is predictable.

Interactive applications are most commonly based on asynchronous events, coming directly from the user, like mouse clicks, button presses, touches of the screen, but also indirect interactions like timer events or network messages. In the case of the mobile applications events from various sensors should also be considered, e.g. changes of the device orientation, or the GPS location.

In response to these events, the application is presenting the output, typically in a visual way. Event handling and rendering of the GUI are implemented with the event queue and the event loop, which periodically consumes and processes events. Modern mobile applications are targeting to render 60 frames per second, which means that every event must be processed in under 16ms.

Obviously, there are events that cannot be processed in that time because, for example, the event requires some interaction with remote services. Hence, the event handling is usually done asynchronously, i.e. the actual processing is performed by threads different than GUI rendering. Only after the processing is finished, the event with results is sent to the event queue and it causes re-rendering of the interface. The concept of the event loop is presented in figure 1.

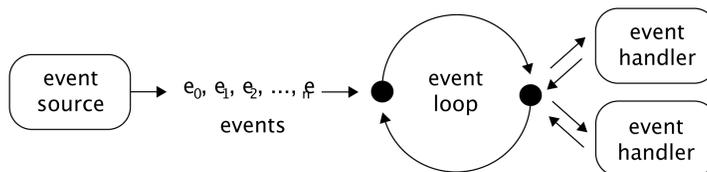


Figure 1: The event handling loop

The proposed application model is based on the functional-reactive programming (*FRP*) paradigm. The core concept of this paradigm is the reactive, instead of proactive, approach to data flow, i.e. reacting to events instead of passing messages to functions. The FRP is similar to the classic observer pattern, but the event flow is modeled with *event streams*, also called *observables* or *signals* in some programming languages. Event processing is done strictly by so called *operator functions*, which are adopted from the functional programming paradigm, e.g. mapping, filtering or reducing.

One of the first programming languages that introduced the FRP concept was Haskell [17]. However, it was never broadly used in programming interactive applications, most likely because of its purely functional nature. One of the new languages that is supposed to target the FRP paradigm is Elm [6]. There are also many frameworks and libraries for popular programming languages that make it easy to use FRP concepts, like Reactive Extensions [13] for C# or RxJS [16] and Cycle.js [23] for JavaScript.

Asynchronous events could be modeled as streams, i.e. sequences of events happening over time [7]. Event stream S_e could be interpreted as a sequence of events v_i . The values of v_i could be arbitrary, and the sequence could be finite, infinite or empty:

$$S_e = \langle v_i, v_{i+1}, v_{i+2}, \dots \rangle \quad (1)$$

In interactive applications we could distinguish three types of streams:

- input streams - events are generated outside the application and processed inside it,
- internal streams - events are generated and processed inside the application,
- output streams - events are generated inside the application but processed outside.

An example of an input stream is the stream of user interactions, e.g. the stream of touch events. In this case the values are coordinates (x, y) of the finger touch, and x_{max} i y_{max} is the width and height of the screen in pixels:

$$S_{touch} = \langle (307, 204), (521, 149), (122, 501), \dots \rangle \quad (2)$$

$$x \in [0, x_{max}], y \in [0, y_{max}]$$

Other examples of input streams include network messages, timers or various types of system interruptions.

Internal event streams are modeling the data flow inside the application. They are generated by transforming one input stream, or a combination of multiple inputs. For example, a stream of user interface element interactions can be generated from the screen touch events. In this case the event values are identifiers of the GUI element:

$$S_{action} = \langle (home_button), (back_button), (text_input), \dots \rangle \quad (3)$$

The application is usually generating multiple output streams. The most common one is the stream of the graphical user interface state, which can be represented as the following sequence of matrices presenting pixel colors $p_{j,k}$:

$$S_{frames} = \left\langle \begin{bmatrix} p_{1,1} & \cdots & p_{1,w} \\ \vdots & \ddots & \vdots \\ p_{h,1} & \cdots & p_{h,w} \end{bmatrix}_i, \begin{bmatrix} p_{1,1} & \cdots & p_{1,w} \\ \vdots & \ddots & \vdots \\ p_{h,1} & \cdots & p_{h,w} \end{bmatrix}_{i+1}, \dots \right\rangle \quad (4)$$

The important characteristic of all event streams is that they are immutable, i.e. the event value v_i is constant during the whole application lifecycle.

Input streams S_{in} are processed and eventually converted to output streams S_{out} with functions f called operators. Theoretically, the whole application can be modeled as a single operator such as:

$$S_{out} = f(S_{in}) \quad (5)$$

In practice, the application is divided into multiple operators with smaller scope, however all of them have a common signature, i.e. they convert input streams to output streams. One exception to that rule is the merge operator f_{merge} , that just combines multiple streams into one:

$$S_{out} = f_{merge}(S_0, S_1, \dots, S_n) \quad (6)$$

Operators can be divided into two categories, *pure* and *impure*, analogically to functions. Pure operators do not have any side-effects, and are stateless. It is a very convenient property of an operator, because it guarantees that for any input event value v , the operator will always produce the same result w :

$$f(\langle v_i, v_{i+1}, \dots \rangle) = \langle w_i, w_{i+1}, \dots \rangle : v_i = v_j \implies w_i = w_j \quad (7)$$

$$i \neq j \wedge i, j = 0, 1, 2, \dots$$

Thanks to this property, pure operators are deterministic, therefore it is easy to cache the results, and test the correctness of the operator. Regarding the application optimization, and partitioning the application to mobile devices and the cloud, the most important feature of a pure operator is their independence from the rest of the application. In other words, they can be isolated and moved to other environments without migrating any extra state.

Two primary examples of pure operators are mapping and filtering. Mapping operator f_{map} is essentially a function in a mathematical sense, that for every input value v_i assigns an output value $g(v_i)$:

$$f_{map}(\langle v_i, v_{i+1}, \dots \rangle) = \langle g(v_i), g(v_{i+1}), \dots \rangle \quad (8)$$

The filtering operator f_{filter} generates an event stream with unchanged values, however some values can be omitted:

$$f_{filter}(\langle v_i, v_{i+1}, \dots \rangle) = \langle v_i : g(v_i) > 0 \rangle \quad (9)$$

For practical reasons, not all operators could be pure and stateless. The most common operator that requires storing an internal state is the accumulation operator f_{scan} , sometimes called scan operator:

$$f_{scan}(\langle v_0, v_i, v_{i+1}, \dots \rangle) = \langle w_i : w_0 = h(v_0, 0) \wedge w_i = h(v_i, h(v_{i-1})) \rangle \quad (10)$$

The interaction between the computer and the user is a two-way process in which both sides produce and consume data, described in literature as *human-computer interaction (HCI)*. The process is cyclical, however there is no strict order of the interactions, i.e. the user can execute multiple actions without response from the computer, and the computer can produce output without analyzing any input from the user.

In practice all interactions between the user and the computer are done through various I/O (*input/output*) devices. The computer may be modeled as a system, which includes I/O drivers and an application working in the operating system environment. The application is not interacting directly with the user, instead it is communicating with an abstraction layer provided by the operating system. Hence, from the application's point of view, we can simplify this model to the interaction between an application and an operating system. The communication is usually done in an asynchronous way, in the form of events, signals or interruptions. In consequence the interaction between the application and the system can be modeled as a cycle of stream processing.

Although the whole interaction is a cyclic and nondeterministic process, the event processing performed inside the application can be modeled as a clearly defined directed acyclic graph (DAG), in which nodes are operators, and edges are internal event streams. The graph represents all possible execution flows for a single iteration – a time between a single user interaction and the application outputting a new state (result). It is important that although two operators are connected with an edge (stream) they not necessarily have to produce and consume an event in every iteration. Hence, in the lifecycle of the application some operators may be executed more frequently than others. An example model of an application, namely a mobile chess game, is presented in figure 2.

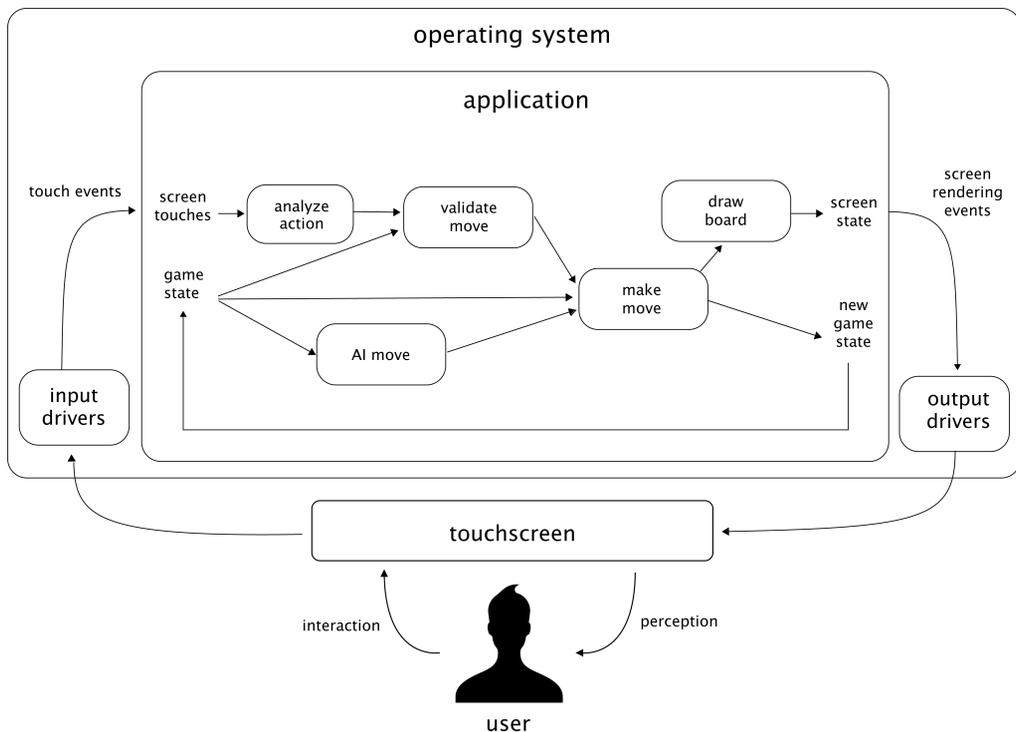


Figure 2: Model of a chess game application

The application contains one input stream – screen touches. It is important to note that the application does not have any global state, instead the game state is passed inside the application in the form of a cyclic stream. There are $N = 5$ operators identified in the application. The screen touches are mapped to actions (*analyze action*), i.e. certain moves of the figures on the chessboard. From the stream of potential figure moves only the valid moves are selected (*validate move*), and then merged with the current state of the game, which in turn generates the new state of the game (*make move*). After every other generated state the computer move is produced (*AI move*). In order to do this, the state stream is mapped to an optimal move – computed by the AI algorithm – and merged

with the aforementioned potential figure moves stream. Lastly, the current state of the game is drawn as a chessboard on the screen (*draw board*).

It is possible to split operators further, and obtain a graph with higher granularity. Especially the *AI move* operator, seems like a quite complex mapping function, and thus a good candidate to divide into a group of simpler operators. However, due to the implementation of the AI algorithm (a variant of the Minimax algorithm [10]), the resulting set of operators would be tightly connected, with significant amounts of communication between them, so moving some of those operators to the cloud would most likely result in lower performance of the application.

3. Problem description and solution space

Optimization of a mobile application, modeled as described in section 2, relies on partitioning the operators into two sets: those executed on the mobile device and those executed in the computing cloud. The objective of the optimization is to minimize the cost c of application execution and execution in the assumed time-frame t_{max} . The cost is interpreted as the sum of all operators' costs:

$$C = \sum_{n=0}^{N-1} c(f_n) \quad (11)$$

In general, the cost is proportional to the CPU time used for executing the operator, and may be also interpreted as the energy required to run the operator. Because some of the operators may be executed in parallel on multi-core processors, the execution time is equal to the critical path in the application model.

Theoretically, the cost and time of execution of every operator may be infinite, because the number of cycles in an interactive application may be infinite. Furthermore, the number of cycles cannot be estimated, because it depends on nondeterministic behavior of the user. In order to find the exact solution of the optimization, every possible combination of events should be considered. For instance, in the chess game application there are 20 possibilities for the first move, and the number of possible game states is increasing exponentially. The game finally ends in one of three states: white (user) win, black (computer) win, or draw. The illustration of the game iterations is presented in figure 3.

The number of all possible games P for I iterations can be calculated based on the number of possible states m on every turn:

$$P = m_0 \cdot m_1 \cdot \dots \cdot m_{I-1} \quad (12)$$

The average number of turns in a chess game is estimated at around 40 for advanced players [2]. In every turn there are two iterations – white player move and black player move – so we can estimate that there are around 80 iterations in a typical chess game.

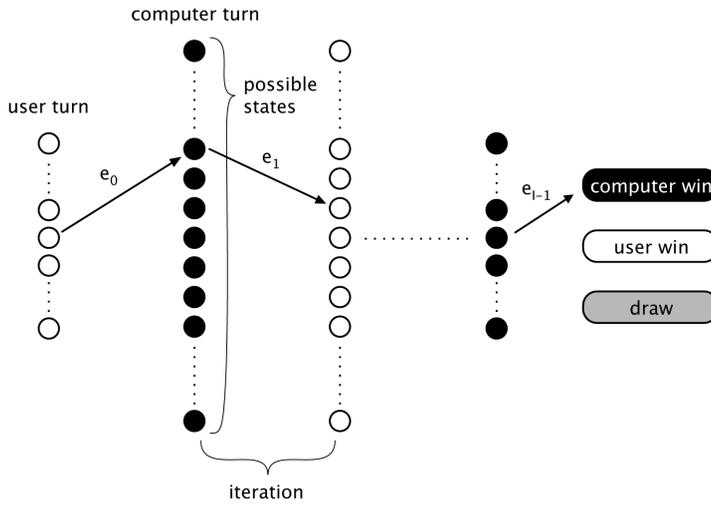


Figure 3: Illustration of chess game iterations

The number of possible states can be different for every turn because of the varying number and positions of the figures. However, the average number of states is estimated at around 30 throughout the whole game [22]. Therefore, the number of possible chess games could be estimated as:

$$P \approx \bar{m}^I = 30^{80} \approx 10^{118} \quad (13)$$

Considering that number of possible games, finding the exact optimization of the whole game is impossible in a reasonable time. Hence, we will analyze the cost and the time of a single iteration, i.e. execution of all operators between subsequent input events:

$$C = \sum_{i=0}^{I-1} \sum_{n=0}^{N-1} c(f_n(e_i)) \quad (14)$$

Despite the fact that, for every event, the cost and the time of executing the operator may be different, the average cost and time of a single iteration is equal to the sum of average costs and execution times of all operators:

$$\bar{C}_{it} = \frac{1}{I} \sum_{i=0}^{I-1} C_i = \frac{1}{I} \sum_{i=0}^{I-1} \sum_{n=0}^{N-1} c(f_n(e_i)) \quad (15)$$

Therefore, in order to minimize the cost of application execution, we must minimize the average cost of executing every operator. Taking into consideration the fact that operators may be executed in two environments – a mobile device or the cloud – the cost and time of the data transfer must be included in the calculations. The easiest way to achieve that is to model the communication as an additional *transfer operator*, that is added on every internal stream which is transferred between the environments.

4. Iterative optimization algorithm

To perform the exact optimization of an interactive application: first, we would have to construct a graph that covers all the iterations, i.e. an I number of connected operator graphs. Therefore, the optimization must be performed on a graph that has $N * I$ nodes, which would be very expensive for a high number of iterations, even when using a heuristic algorithm [12].

However, the iterative nature of interactive applications allows you to effectively use a greedy optimization strategy. In every iteration the algorithm finds the single most profitable transition, i.e. the operator that, when moved to the other environment, would have the maximum impact on lowering the summary execution cost, and still keep the execution time in the assumed time-frame. We expect that after some number of iterations, that is significantly lower than the total number of iterations for the given application, the heuristic algorithm should achieve near-optimal application partition.

```

function OPTIMIZE(operators)
   $dc_{max} \leftarrow 0$ 
   $operator_{max} \leftarrow null$ 
  for all operators do
    if operator in cloud then
       $dt \leftarrow mobileTime(operator) - cloudTime(operator)$ 
      if  $t + dt \leq t_{max}$  then
         $dc \leftarrow cloudCost(operator) - mobileCost(operator)$ 
        if  $dc > dc_{max}$  then
           $dc_{max} \leftarrow dc$ 
           $operator_{max} \leftarrow operator$ 
    else
       $dt \leftarrow cloudTime(operator) - mobileTime(operator)$ 
      if  $t + dt \leq t_{max}$  then
         $dc \leftarrow mobileCost(operator) - cloudCost(operator)$ 
        if  $dc > dc_{max}$  then
           $dc_{max} \leftarrow dc$ 
           $operator_{max} \leftarrow operator$ 
  return  $max_{operator}$ 

```

The algorithm pseudocode is presented in the listing. It returns an identifier of an operator that should be moved from the mobile device to the cloud or vice versa. The $mobileTime()$, $mobileCost()$, $cloudTime()$ and $cloudCost()$ functions return costs and execution times for a given operator computed from the previous operator calls. We assume that initial costs and times are equal to 0. The proposed algorithm's time complexity for a single iteration is $O(N)$ depending on the number of operators.

A software framework and management module is required to enable dynamic allocation of an operator to the cloud or a mobile device in every iteration. Such an offloading framework was developed, and a testing environment was set up, in order to test the algorithm. The framework was built on top of RxJS [16] and CycleJS [23] libraries that support developing interactive applications with architecture based on event streams.

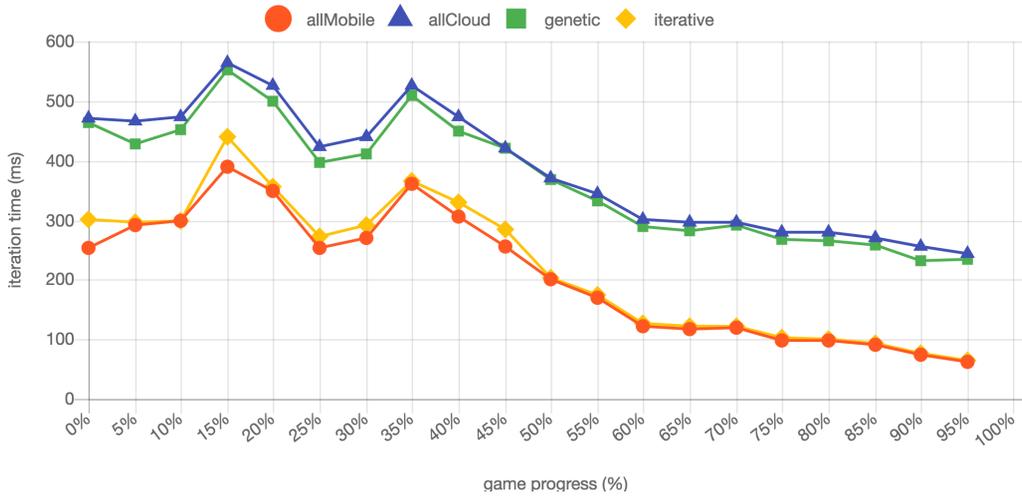
The proposed iterative heuristic was also compared to the genetic optimization algorithm [12] that calculated operator allocation a priori, before the application execution. The genetic algorithm was proposed for static applications, where execution times are constant throughout the lifecycle of the application, so the allocation for given operator is also constant. As the algorithm requires historical data to perform the optimization, operator execution and transfer times for 20 games were measured and averaged.

The tests were performed for the implemented chess application corresponding to presented one in figure 2. Four methods of application allocation were evaluated:

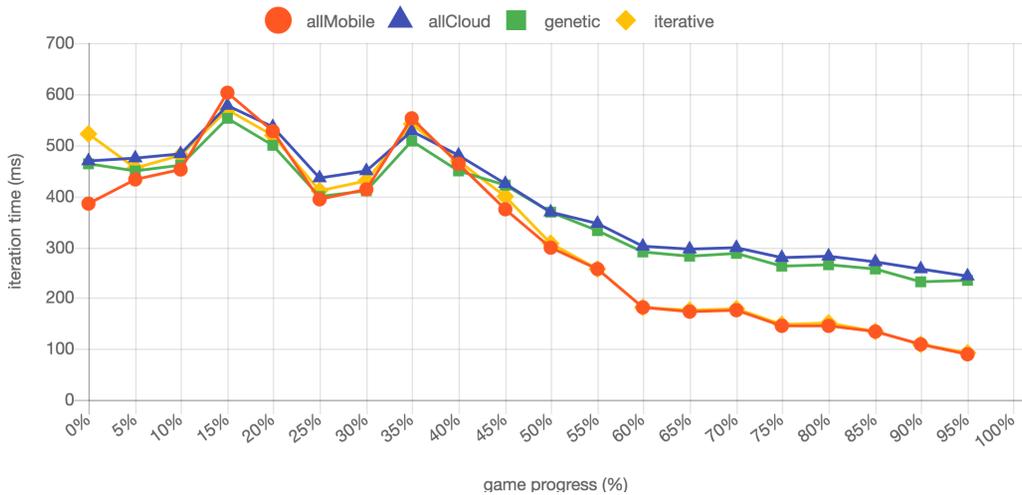
- *allMobile* - the application running only on the mobile device,
- *allCloud* - all operators running in the cloud, with the mobile device acting as a thin client,
- *genetic* - operators allocated to the mobile device or to the cloud before the application execution by the heuristic genetic algorithm, optimization based on the average costs for all iterations
- *iterative* - operators allocated to the mobile device or to the cloud on every iteration by the heuristic optimization algorithm proposed above.

The tests were run on a mobile device emulator. As the performance of the actual mobile devices vary greatly between different models, several ratios of the cloud computing performance to the mobile device performance were simulated, ranging from the 1:1 ratio (the same performance on both environments) to 3:1 ratio (3 times faster computation on the cloud). The network latency, for communication between the device and the cloud, was set to 200ms. The charts presented in figure 4 show the varying iteration duration (in milliseconds) throughout the whole game, from the beginning (0%) until the end (100%). The data was averaged from 20 different chess games.

The average duration of the whole game for different cloud to mobile performance ratios and allocation configurations are presented in table 1. Although the iterative optimization algorithm does not give the optimal solution in every case, the difference from the best allocation is always relatively small. Hence, with the iterative optimization algorithm enabled, and with dynamic operator allocation, the average game duration across all the cases is 13.4% lower than the *allMobile* allocation, and 12.5% lower than the *allCloud* one. The *genetic* optimization algorithm favored moving the single operator with the highest cost to the cloud (*AI move*), but because this algorithm does not consider changing execution times, the results are comparable to the *allCloud* scenario.



(a) 1:1 performance ratio

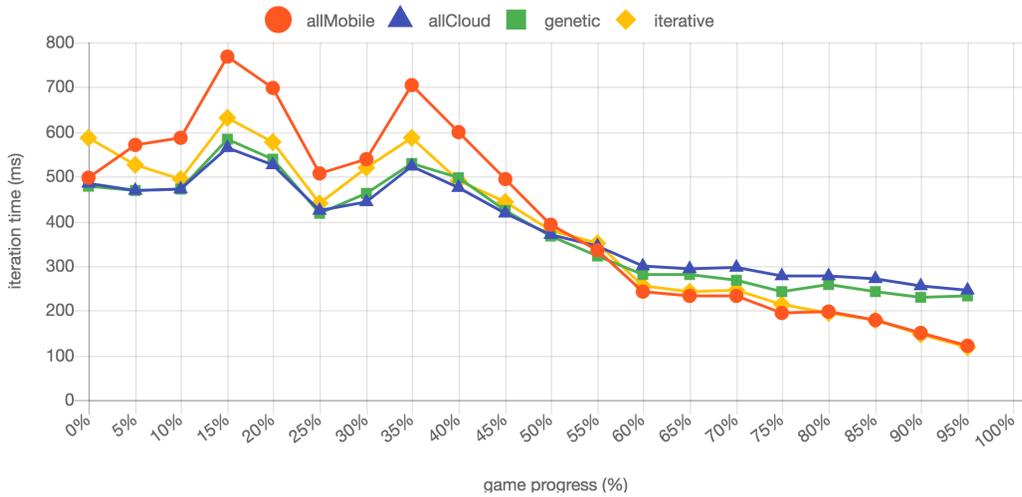


(b) 1.5:1 performance ratio

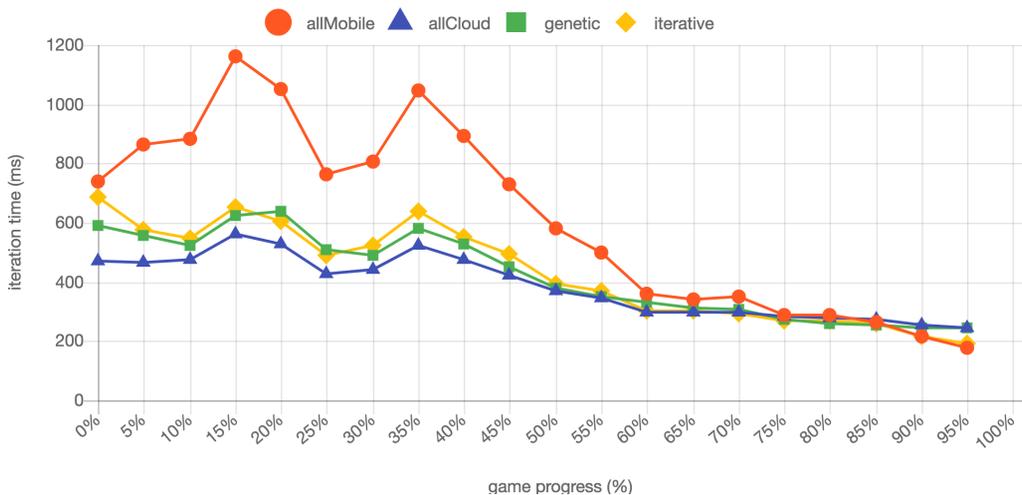
Figure 4: Iteration duration (in milliseconds) throughout the game

5. Conclusions

Optimization of an interactive mobile application using cloud computing has three possible results: all operators are executed on the device, all operators are offloaded to the cloud, or the operator graph is partitioned between those two environments. The example



(c) 2:1 performance ratio



(d) 3:1 performance ratio

Figure 4: Iteration duration (in milliseconds) throughout the game

chess game application showed that using only the mobile device is not optimal, because of the complex processing involved in computing the AI moves. Offloading the whole application to the cloud is also not ideal, because of the significant output data size for the *draw board* operator. In fact, the optimal solution is to move only one operator to

Table 1: Average game durations

performance ratio	allMobile	allCloud	genetic	iterative
1:1	18.60	34.77	33.31	19.70
1.5:1	27.99	35.08	33.40	28.94
2:1	36.63	34.78	34.19	33.71
3:1	54.72	34.73	37.97	38.34
average	34.49	34.84	35,88	30.17

the cloud, for the middle part of the game, hence the iterative heuristic algorithm and dynamic operator allocation are suitable in this case.

References

- [1] H. BAUER, Y. GOH, S. SCHLINK and C. THOMAS: The supercomputer in your pocket. *McKinsey on Semiconductors*, (Autumn), (2012), 14-27,
- [2] Chessgames Services LLC: Chess Statistics. <http://www.chessgames.com/chessstats.html>, 2016.
- [3] B. CHUN, S. IHM, P. MANIATIS, M. NAIK and A. PATTI: Clonecloud: elastic execution between mobile device and cloud. *Proc. of the 6th Conf. on Computer Systems*, (2001), 301-314.
- [4] Cisco, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018. 2014.
- [5] E. CUERVO, A. BALASUBRAMANIAN, D. CHO, A. WOLMAN, S. SAROIU, R. CHANDRA and P. BAHL: MAUI: making smartphones last longer with code offload. *Proc. of the 8th Int. Conf. on Mobile Systems, Applications, and Services*, (2010), 49-62.
- [6] E. CZAPLICKI: Elm: Concurrent FRP for Functional GUIs. Senior thesis, Harvard University, 2012.
- [7] O. ETZION and P. NIBLETT: Event Processing in Action. Manning Publications Co., 2010.
- [8] J.X. HAO and J.B. ORLIN: A faster algorithm for finding the minimum cut in a directed graph. *J. of Algorithms*, **17**(3), (1994), 424-446.

- [9] R. KEMP, N. PALMER, T. KIELMANN and H. BAL: Cuckoo: a computation offloading framework for smartphones. *Mobile Computing, Applications, and Services*. Springer, 2010, 59-79.
- [10] R.E. KORF and D.M. CHICKERING: Best-first minimax search. *Artificial Intelligence*, **84**(1), (1996), 299-337.
- [11] S. KOSTA, A. AUCINAS, P. HUI, R. MORTIER and X. ZHANG: Unleashing the power of mobile cloud computing using thinkair. arXiv:1105.3232 [cs.DC], (2011).
- [12] H. KRAWCZYK, M. NYKIEL and J. PROFICZ: Mobile offloading framework: Solution for optimizing mobile applications using cloud computing. *Computer Networks*, Springer International Publishing, (2015), 293-305.
- [13] J. LIBERTY, P. BETTS and S. TURALSKI: *Programming Reactive Extensions and LINQ*. Springer, 2011.
- [14] R.K. MA, K.T. LAM and C. WANG: eXCloud: Transparent runtime support for scaling mobile applications in cloud. *Int. Conf. on Cloud and Service Computing*, (2011), 103-110.
- [15] V.MARCH, Y. GU, E. LEONARDI, G. GOH, M. KIRCHBERG and B.S. LEE: μ cloud: towards a new paradigm of rich mobile applications. *Procedia Computer Science*, **5** (2011), 618-624.
- [16] Microsoft Open Technologies: The Reactive Extensions for JavaScript. <https://github.com/Reactive-Extensions/RxJS>, 2016.
- [17] H. NILSSON, A. COURTNEY and J. PETERSON: Functional reactive programming, continued. *Proc. of the 2002 ACM SIGPLAN Workshop on Haskell*, (2002), 51-64.
- [18] M. OTHMAN, S.A. MADANI and S.U. KHAN: A survey of mobile cloud computing application models. *IEEE Communications Surveys & Tutorials*, **16**(1), (2014), 393-413.
- [19] A. PATHAK, C. HU, M. ZHANG, P. BAHL and Y. WANG: Enabling automatic offloading of resource-intensive smartphone applications. Purdue University, Electrical and Computer Engineering Technical Report, 2011.
- [20] M. SATYANARAYANAN: Mobile computing: the next decade. *Proc. of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, (2010).
- [21] M. SATYANARAYANAN, P. BAHL, R. CACERES and N. DAVIES: The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, **8**(4), (2009), 14-23.

-
- [22] C. SHANNON: Programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and J. of Science*, **41**(314), (1950), 256-275.
- [23] A. STALTZ: Cycle.js – a Functional and Reactive JavaScript Framework for Cleaner Code. <http://cycle.js.org/>, 2016.
- [24] X. ZHANG, A. KUNJITHAPATHAM, S. JEONG and S. GIBBS: Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing. *Mobile Networks and Applications*, **16**(3), (2011), 270-284.

Discrete-time feedback stabilization

WOJCIECH MITKOWSKI, WALDEMAR BAUER and MARTA ZAGÓROWSKA

This paper presents an algorithm for designing dynamic compensator for infinite-dimensional systems with bounded input and bounded output operators using finite dimensional approximation. The proposed method was then implemented in order to find the control function for thin rod heating process. The optimal sampling time was found depending on discrete output measurements.

Key words: stabilization, feedback stabilization, finite dimensional stabilization, infinite dimensional systems, finite dimensional approximations, continuous-discrete system.

1. Introduction

One of the main areas of automatic control is related to stabilization problems. Usually, in real time application, an algorithm consisting of two stages is used: 1. Bring the system to the valid region of linearization. 2. Stabilize the system using linear approximation. This approach is justified by topological similarity of a nonlinear system and its linearization (valid only for hyperbolic systems without purely imaginary eigenvalues).

Feedback design (design of the stabilizing controller) depends on the system form (usually we have either differential equations or transfer function for time invariant systems).

The design of finite dimensional feedback is useful due to multiple reasons: 1. It is possible to use simple, finite-dimensional methods, e.g., Lyapunov functions and in consequence, Lyapunov equations strictly linked with algebraic Riccati equations. 2. Some of the systems have predefined structure, e.g., the hoisting machine (long line is a distributed system, and the drive may be modeled with finite-dimensional system).

The design of finite-dimensional controllers for infinite systems with finite set of unstable modes (or at least weakly damped ones) is widely analyzed in literature. This class of the systems was described by Triggiani (1975) [34], or even earlier by Fattorini (1967) [11]. Using small disturbance methods and building appropriate invariant sets, Schumacher (1981, 1983) [30, 31] proposed finite dimensional stabilizing controllers

The Authors are with AHG University of Science and Technology, Krakow, Poland. W. Mitkowski is the corresponding author, e-mail: wojciech.mitkowski@agh.edu.pl. W. Mitkowski is the corresponding author.

This work was supported by the AGH (Poland) - the project No 11.11.120.817.

Received 14.12.2016. Revised 30.04.2017.

for distributed and delayed systems (see also Kamen (1985) [14]). Similar results were obtained by Curtain (1984) [4] for parabolic systems with infinite input-output operators. Also the works of Curtain and Salomon (1986) [5], and Sakawa (1983, 1984, 1985) [27-29] are worth noticing. Balas (1983) [2] proposed a finite dimensional dynamic compensator for finite dimensional approximations of infinite systems. Similar methods were proposed by Kobayashi (1983) [16]. Gibson (1981) [13] used finite dimensional approximation of algebraic Riccati equation. The detailed description of those works was done, e.g., by Mitkowski (1991) [20] with 229 books and papers analyzed.

The design of stabilizing controllers is still an interesting problem (see, e.g. Przyuski (2014) [26]), especially as there are more efficient numerical tools. Thanks to computers, nowadays, we can analyze complex mathematical models of distributed parameter systems, e.g. models of non-integer order Obrzeczka (2014) [22], Sierociuk (2015) [32], Oprzêdkiewicz (2016) [24] which sometimes better describe real systems.

In this work, we focused on an algorithm of stabilization of linear infinite dimensional system with bounded input and bounded output operators and with finite set of unstable modes (weakly damped) using finite discrete stabilization. As an example, we used diffusion equation which models the heating process of a thin rod.

2. Problem description

Consider a closed-loop system (with continuous time) shown in Fig. 1.

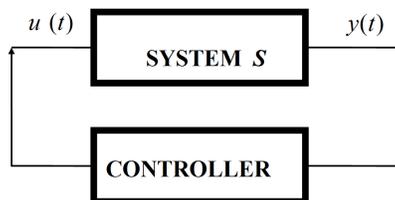


Figure 1: Closed-loop system.

Finite dimensional stabilization problem: for a given infinite system S find a stabilizing controller (finite dimensional) such that the closed-loop system is exponentially stable with predefined damping coefficient.

In digital control, it is necessary to use a discrete system (computer or other device with discrete time). In order to use a discrete stabilizing controller in continuous time system, we need to use the system (see Fig. 2) in the form of a series of pulser, continuous linear system S , and ZOH (Zero Order Hold) with input $u(k)$ and output $y(k)$, $k = 1, 2, 3, \dots$.

If the pulser and ZOH work synchronously with time step $h > 0$, then the parameters of discrete linear system S^d denoted for simplicity with A, B, C are given by the formulas

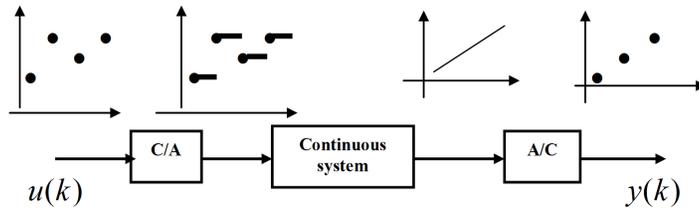


Figure 2: Continuous-discrete system.

calculated on the basis of continuous system:

$$A := e^{Ah}, \quad B := \int_0^h e^{At} B dt, \quad C := C \quad (1)$$

For a valid controller (both continuous and discrete), we need the controllability and observability of continuous system S . The conditions for time step $h > 0$ which guarantee that the discrete system is also controllable and observable are known and may be found, e.g., in Mitkowski (1991, p. 141) [20].

3. The decomposition of the system

There is a group of infinite dimensional systems which can be stabilized using finite dimensional methods. Let us now consider a system (see for example Pazy (1983) [25], Slemrod (1974) [33], Wang (1972) [35], Curtain and Pritchard (1978) [7], Curtain and Zwart (1995) [8])

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), & y(t) &= Cx(t) \\ x(t) &\in X, & u(t) &\in U, & y(t) &\in Y \end{aligned} \quad (2)$$

For further use we will denote it as $S(A, B, C)$. Let us now assume that (2) fulfills the following conditions:

- X, Y, U – Hilbert spaces, $\dim U < +\infty$.
- A is an infinitesimal generator C_0 of semi-group $T_A(t)$, for $t \geq 0$ in X .
- $B \in L(U, X), C \in L(X, Y)$ are bounded.
- A is a discrete operator with finite number of eigenvalues with $\operatorname{Re} s > \beta, \beta < +\infty$.

Taking into account the conditions above, we can decompose (2) into (Triggiani (1975) [34]):

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} u(t), \quad (3)$$

$$y(t) = C_1 x_1(t) + C_2 x_2(t) + C_3 x_3(t),$$

$$\begin{aligned} x_i(t) &\in X_i, i = 1, 2, 3, X = X_1 + X_2 + X_3, \\ \dim X_1 &< +\infty, \quad \dim X_2 = p < +\infty. \end{aligned} \quad (4)$$

The spectrum of A (see (2)) is depicted in the Fig. 3. The operator A_1 is responsible for unstable (or weakly damped) part of the system (3). The operators A_2 and A_3 are exponentially stable.

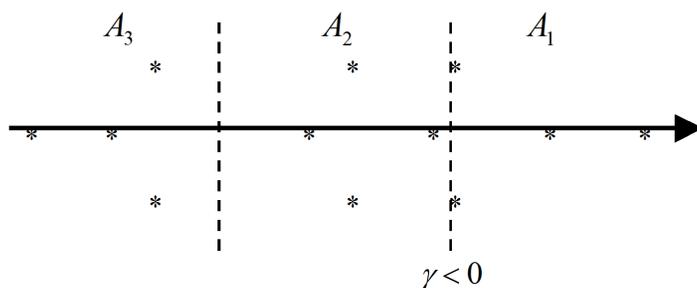


Figure 3: Discrete spectrum of A .

Let us now add the following assumptions:

- $\sup\{\operatorname{Re} s : s \in \lambda(A_3)\} < 0, \sup\{\operatorname{Re} s : s \in \lambda(A_2)\} = \gamma < 0$.
- The pair (A_1, B_1) is controllable, the pair (C_1, A_1) is observable.
- $\dim X_2 = p \rightarrow +\infty \Rightarrow \|B_3\| \rightarrow 0$ and $\|C_3\| \rightarrow 0$.

The last assumption is fulfilled if, e.g., self-adjoint generator A has compact resolvent (the eigenvectors form a basis of the given space).

4. Finite-dimensional stabilizing controller

Let us now consider dynamic feedback Mitkowski (1988 [19, p. 519], 1991, [20, p. 233]) of form:

$$\begin{bmatrix} \dot{w}_1(t) \\ \dot{w}_2(t) \end{bmatrix} = \begin{bmatrix} A_1 - G_1 C_1 + B_1 K_1 & -G_1 C_2 \\ B_2 K_1 & A_2 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} + \begin{bmatrix} G_1 \\ 0 \end{bmatrix} y(t), \quad (5)$$

$$u(t) = K_1 w_1(t), \quad w_i(t) \in X_i, \quad i = 1, 2.$$

Let us assume that the conditions mentioned in previous section are fulfilled. There exists a finite dimensional stabilizing controller (5), such that the closed-loop system (2) with (5) is exponentially stable with predefined damping coefficient $\alpha \in (\gamma, 0)$, see Sakawa (1983) [27]. See also Mitkowski (1982, 1986, 1988) [17, 19], Mitkowski (1991, [20, p. 230]) for further details.

The design of feedback (5) may be reduced to finding the matrices K_1 and G_1 which can be done using methods known from finite dimensional system's analysis, e.g., LQ design. The desired damping coefficient $\alpha \in (\gamma, 0)$ can be found by increasing $p = \dim X_2$.

A discrete version of the controller (Mitkowski (1991, [20, p. 236]) can be obtained using formulas (1) and remembering that the system is asymptotically stable if the eigenvalues lie inside the unit circle. The matrices K_1 i G_1 should be found in a way that guarantees that the eigenvalues of matrices $A_1 + B_1 K_1$ and $A_1 - G_1 C_1$ lie inside the unit circle (for example, we can set them as zeros).

5. Example

Let us now consider the process of heating a thin rod (Oprzedkiewicz (2003, 2016) [23, 24]) depicted in the Fig. 4.

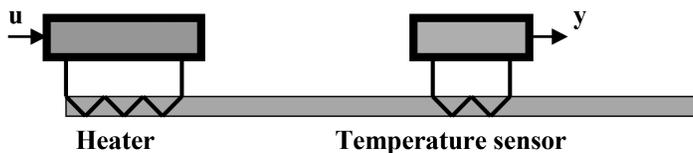


Figure 4: Heating of a thin rod.

A simplified mathematical model of the analyzed process has the form

$$\begin{aligned}
 \frac{\partial x(z,t)}{\partial t} &= a \frac{\partial^2 x(z,t)}{\partial z^2} - R_a x(z,t) + b(z)u(t), \quad t \geq 0, \quad z \in [0, 1], \\
 \frac{\partial x(z,t)}{\partial z} \Big|_{z=0} &= \frac{\partial x(z,t)}{\partial z} \Big|_{z=1} = 0, \quad t \geq 0, \\
 x(z,0) &= 0, \quad z \in (0, 1), \\
 y(t) &= \int_0^1 c(z)x(z,t)dz.
 \end{aligned} \tag{6}$$

where

$$\begin{aligned}
 b(z) &= \begin{cases} 1 & \text{for } 0 \leq z \leq z_0 \\ 0 & \text{for } z_0 < z \leq 1 \end{cases} \\
 c(z) &= \begin{cases} \bar{c} & \text{for } z_1 \leq z \leq z_2 \\ 0 & \text{for } 0 \leq z < z_1 \quad \text{and} \quad z_2 < z \leq 1 \end{cases} \\
 x(z,t) &= \sum_{i=0}^{\infty} x_i(t)h_i(z)
 \end{aligned}$$

After the decomposition, we have system $S(A, B, C, D)$, where

$$\begin{aligned}
 A &= \text{diag}(\lambda_0, \lambda_1, \lambda_2, \dots), \quad B = [b_0 \ b_1 \ b_2 \ \dots]^T, \\
 C &= [c_0 \ c_1 \ c_2 \ \dots], \quad D = 0,
 \end{aligned}$$

and

$$\begin{aligned}
 X &= L^2(0, 1; R), \quad \lambda_i = -i^2 \pi^2 a - R_a, \quad i = 0, 1, 2, \dots \\
 h_i(z) &= \begin{cases} 1 & \text{for } i = 0 \\ \sqrt{2} \cos(i\pi z) & \text{for } i = 1, 2, 3, \dots \end{cases} \\
 b_i &= \int_0^1 b(z)h_i(z)dz, \quad c_i = \int_0^1 c(z)h_i(z)dz,
 \end{aligned} \tag{7}$$

we have the following parameters for model (6) (verified in a laboratory, Oprzedkiewicz (2003, 2004) [26, 21]):

$$a = 0.000945, \quad R_a = 0.0271, \quad \bar{c} = 25.7922 \quad z_0 = 1/13, \quad z_1 = 25/52, \quad z_2 = 27/52$$

From (7), we have

$$\begin{aligned}
 A &= \text{diag}(-0.0269 - 0.0358 - 0.0624 - 0.1068 - 0.1690 - 0.2490 - 0.3467 \\
 &\quad - 0.4621 - 0.5954 - 0.7464 - 0.9152 - 1.1017 - 1.3060 - 1.5281 - 1.7679 \\
 &\quad - 2.0255 - 2.3009 - 2.5940 - 2.9049 - 3.2335 - 3.5800 - 3.9441 - 4.3261 \\
 &\quad - 4.7258 - 5.1433)
 \end{aligned}$$

$$B = [0.0769 \ 0.1077 \ 0.1046 \ 0.0995 \ 0.0926 \ 0.0842 \ 0.0745 \ 0.0638 \\ 0.0526 \ 0.0412 \ 0.0299 \ 0.0190 \ 0.0090 \ -0.0000 \ -0.0077 \ -0.0139 \\ -0.0187 \ -0.0218 \ -0.0234 \ -0.0235 \ -0.0223 \ -0.0200 \ -0.0168 \ -0.0130 \\ -0.0087]^T$$

$$C = [1.0171 \ 0 \ -1.4348 \ -0.0000 \ 1.4244 \ -0.0000 \ -1.4070 \ 0.0000 \\ 1.3830 \ -0.0000 \ -1.3524 \ -0.0000 \ 1.3156 \ -0.0000 \ -1.2729 \ -0.0000 \\ 1.2246 \ -0.0000 \ -1.1711 \ -0.0000 \ 1.1130 \ -0.0000 \ -1.0507 \ 0.0000 \ 0.9848]$$

In order to perform the simulation, the heating process was implemented with use of Matlab/Simulink environment (see Fig. 5).

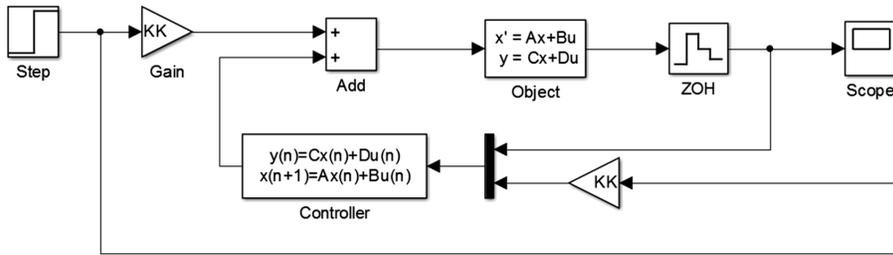


Figure 5: Simulink system.

The zero-order-hold is necessary to simulate a measurement device (e.g. thermometer) with various sampling times. We used the Tustin method (see e.g., Astrom 1990, [1, p. 212]) to discretize the compensator and then find the appropriate sampling frequency. It transforms the continuous system $S(A, B, C, D)$ into a discrete one for a given sampling time h using the formulas

$$\begin{aligned} A^+ &= \left(I + \frac{h}{2}A\right)\left(I - \frac{h}{2}A\right)^{-1} \\ B^+ &= A^{-1}(A^+ - I)B \\ C^+ &= C \\ D^+ &= D \end{aligned} \quad (8)$$

During the simulation we wanted to find optimal sampling time of the compensator for various sampling frequencies for temperature measurement. We used the performance indicator proposed by Bini and Buttazzo (2014) [3]:

$$J(N) = \frac{1}{N} \int_0^T |\dot{u}(t)| dt \quad (9)$$

During the simulations, we set $T = 200$ [s]. For optimization, we used golden search with parabolic interpolation implemented in Matlab Optimization Toolbox. The optimization constraints were chosen as $1 \leq N \leq 10^6$. The results are gathered in Tab. 1.

Table 1: The results of optimization

Temperature sampling frequency [Hz]	Optimal number of samples N_{opt}	Sampling time $h = \frac{T}{N_{opt}}$ [s]
10	23700	0.0084
1	23896	0.0083
0.1	68957	0.0029
0.03	84140	0.0024
0.02	48284	0.0041
0.01	69997	0.0028

It can be seen that sampling time of the controller increases with increasing sampling frequency. This means that we have a buffer in the controller for doing necessary calculations. The accuracy of temperature measurements and controller performance are depicted in the Figs 6 and 7.

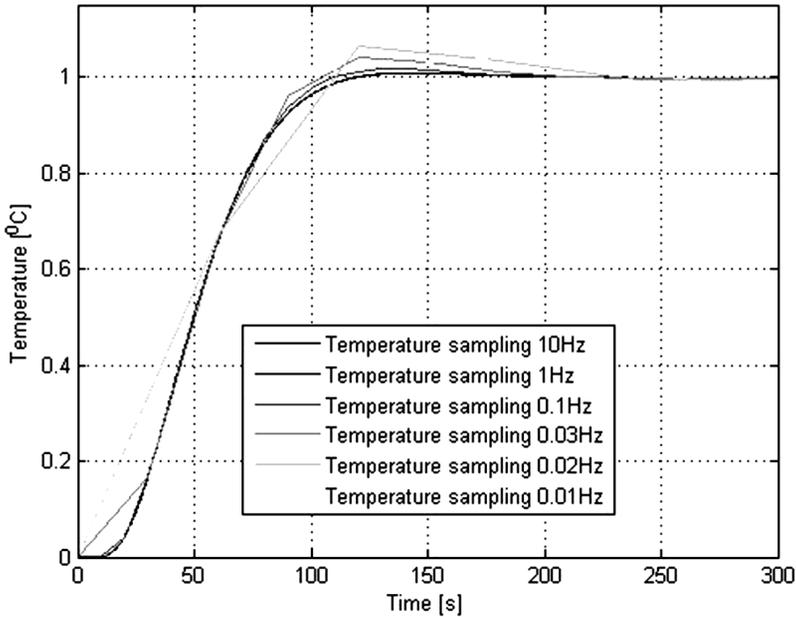


Figure 6: Temperature for various sampling frequencies.

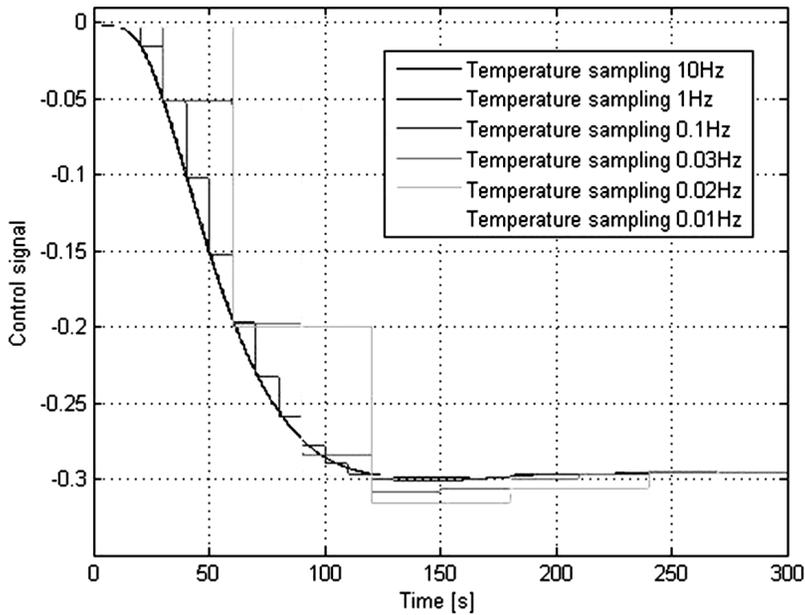


Figure 7: Control signal for various sampling frequencies.

6. Comparison quality index

For all numerical experiments the golden search with parabolic interpolation method has been chosen for tuning N_{opt} parameter. Initial value for all experiments have value 1.

The tests will be conducted for the following quality index:

$$1. \frac{1}{N} \int_0^T |\dot{u}(t)| dt, \quad 2. \frac{1}{N} \int_0^T u^2 dt \quad 3. \frac{1}{N} \int_0^T |u| dt \quad 4. \frac{1}{N} \int_0^T t u dt$$

It can be seen, that all quality index give the same result for the same sampling frequencies (see Fig. 8, Fig. 9 and Fig. 10). But calculating are the faster for the quality index of form 1 (see Tab. 2).

7. Conclusion

The main purpose of this work was to present possible way of approximating infinite dimensional systems with finite dimensional ones. The resulting system can then be discretized and implemented in digital controllers. The results were confirmed with simulation as we analyzed the process of thin rod heating. We found optimal sampling time

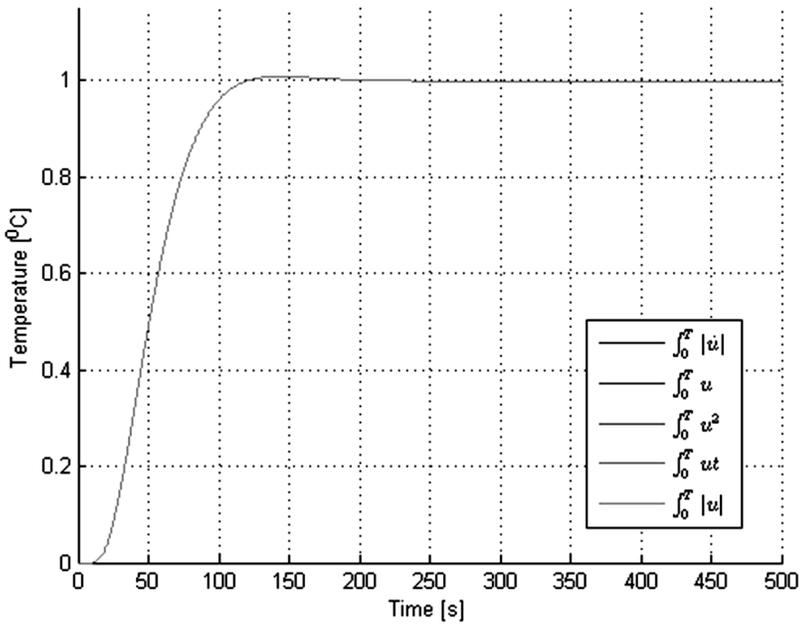


Figure 8: Temperature for various quality index with sampling frequencies 10Hz.

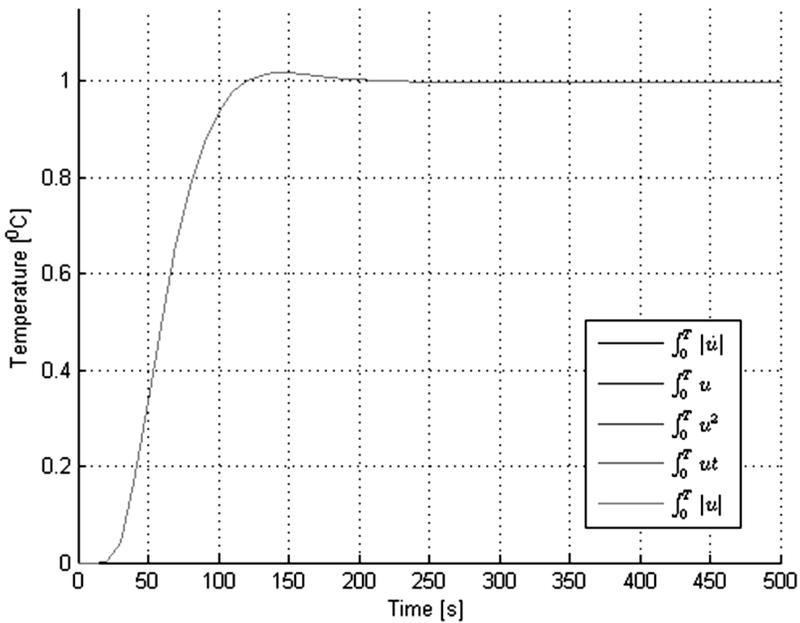


Figure 9: Temperature for various quality index with sampling frequencies 0.01Hz.

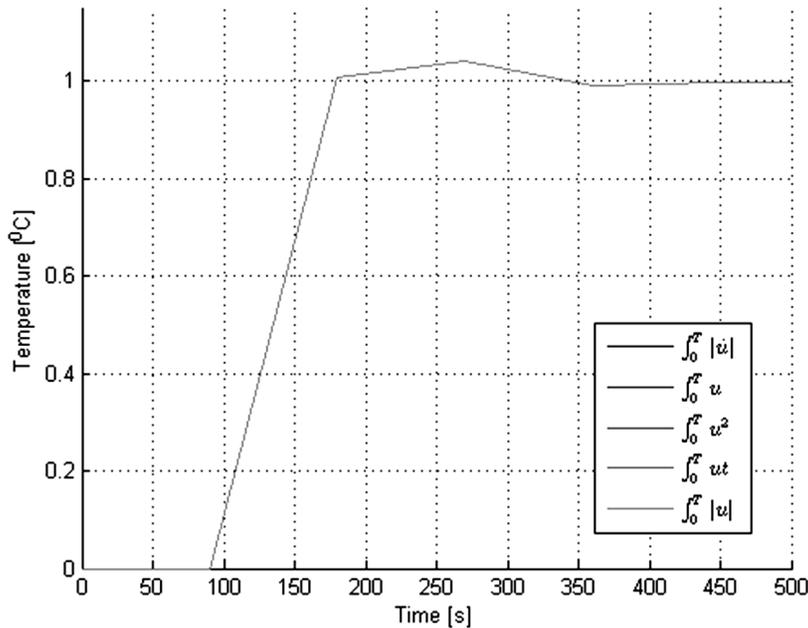


Figure 10: Temperature for various quality index with sampling frequencies 0.1Hz

for the compensator depending on various output sampling frequencies and comparison result for different quality index.

Nevertheless, the proposed algorithm is general and may be used for control of various systems. One of the possible way of applications may be non-integer order diffusion equation Gal and Warma (2016) [12], see also Evans (2007) [10]. However, it will require further analysis and research, as the methods for integer order systems cannot be directly applied to them.

References

- [1] K.J. ASTROM: Computer Controlled Systems. Theory and Design. Second Edition. Prentice-Hall, 1990.
- [2] M.J. BALAS: The Galerkin method and feedback control of linear distributed parameter systems. *J. Mathematical Analysis and Application*, **91**(2), (1983), 527-546.
- [3] E. BINI and G.M. BUTTAZZO: The optimal sampling pattern for linear control systems. *IEEE Trans. On Automatic Control*, **59**(1), (2014), 78-90.

Table 2: The results of optimization

Quality index number	Temperature sampling frequency [Hz]	Error	Time of calculation [s]
1	10	0.0545	200
	1	0.0548	191
	0.1	0.0586	93
	0.03	0.0687	61
	0.02	0.0853	102
	0.01	0.0889	80
	2	10	0.0545
1		0.0548	448
0.1		0.0586	135
0.03		0.0687	113
0.02		0.0853	87
0.01		0.0889	90
3		10	0.0545
	1	0.0548	454
	0.1	0.0586	158
	0.03	0.0687	119
	0.02	0.08537	87
	0.01	0.08897	90
	4	10	0.0545
1		0.0548	432
0.1		0.0586	173
0.03		0.0687	116
0.02		0.0853	102
0.01		0.0889	101

- [4] R.F. CURTAIN: Finite dimensional compensators for parabolic distributed systems with unbounded control and observation. *SIAM J. Control and Optimization*, **22**(2), (1984), 255-276.
- [5] R.F. CURTAIN and D. SALOMON: Finite dimensional compensators for infinite dimensional systems with unbounded input operators. *SIAM J. Control and Optimization*, **24**(4), (1986), 797-816.
- [6] R.F. CURTAIN: Linear-quadratic control problem with fixed endpoints in infinite dimensions, *J. of Optimization Theory and its Applications*, **44**(1), (1984), 55-74.
- [7] R.F. CURTAIN and A.J. PRITCHARD: *Infinite-Dimensional Linear Systems Theory*. Springer, Berlin, 1978.

- [8] R.F. CURTAIN and H. ZWART: An Introduction to Infinite-Dimensional Linear Systems Theory. Springer, New York, NY, 1995.
- [9] Z. EMIRSAJŁOW: Feedback control in LQCP with a terminal inequality constraint. *J. of Optimization Theory and Applications*, **62**(3), (1989), 387-403.
- [10] K.P. EVANS and N. JACOB: Feller semigroups obtained by variable order subordination. *Revista Matemática Complutense*, **20**(2), (2007), 293-307.
- [11] H.O. FATTORINI: On complete controllability of linear systems. *J. Differential Equations*, **3**(3), (1967), 391-402.
- [12] C. GAL and M. WARMA: Elliptic and parabolic equations with fractional diffusion and dynamic boundary conditions. *Evolution Equations and Control Theory*, **5**(1), (2016), 61-103.
- [13] J.S. GIBSON: An analysis of optimal modal regulation: convergence and stability. *SIAM J. Control and Optimization*, **19**(5), (1981), 686-707.
- [14] E.W. KAMEN, P.P. KHARGONEKAR and A. TANNENBAUM: Stabilization of time-delay systems using finite-dimensional compensators. *IEEE Trans. on Automatic Control*, **30**(1), (1985), 75-78.
- [15] T. KATO: Perturbation Theory for Linear Operators. Springer, Berlin, 1966.
- [16] T. KOBAYASHI: Discrete-time observers and parameter determination for distributed parameter systems with discrete-time input-output data. *SIAM J. on Control and Optimization*, **21**(3), (1983), 331-351.
- [17] W. MITKOWSKI: Stabilization of linear distributed systems. *3rd Symp. IFAC, Control of Distributed Parameter Systems*, Toulouse, France, (1982), p. IV.10-IV.13.
- [18] W. MITKOWSKI: Feedback stabilization of second order evolution equations with damping by discrete-time input-output data. *IMACS-IFAC Symp. on Modelling and Simulation for Control of Lumped and Distributed Parameter Systems*, Lille, France, (1986), 355-358.
- [19] W. MITKOWSKI: Stabilizacja liniowych układów nieskończone wymiarowych za pomocą dynamicznego sprzężenia zwrotnego (Stabilization of infinite-dimensional linear systems by dynamic feedback). *Archiwum Automatyki i Telemekhaniki*, **33**(4), (1988), 515-528, (in Polish).
- [20] W. MITKOWSKI: Stabilizacja systemów dynamicznych (Stabilization of Dynamic Systems). WNT, Warszawa, 1991.
- [21] W. MITKOWSKI and K. OPRZĘDKIEWICZ: A sample time assign for a discrete interval parabolic system with the two-dimensional uncertain parameter space. *Systems Science*, **30**(1), (2004), 43-50.

- [22] A. OBRĄCZKA and W. MITKOWSKI: The comparison of parameter identification methods for fractional partial differential equation. *Solid State Phenomena*, **21** (2014), 265-270.
- [23] K. OPRZĘDKIEWICZ: The interval parabolic system. *Archives of Control Sciences*, **13**(4), (2003), 415-430.
- [24] K. OPRZĘDKIEWICZ, E. GAWIN and W. MITKOWSKI: Modeling of heat distribution with the use of non-integer order, state space model. *Int. J. of Applied Mathematics and Computer Science*, **26**(4), (2016), 749-756.
- [25] A. PAZY Semigroups of Linear Operators and Applications to Partial Differential Equations. Springer, New York, 1983.
- [26] K.M. PRZYŁUSKI: On an infinite dimensional linear-quadratic problem with fixed endpoints: The continuity question. *Int. J. of Applied Mathematics and Computer Science*, **24**(4), 723-733.
- [27] Y. SAKAWA: Feedback stabilization of linear diffusion systems. *SIAM J. Control and Optimization*, **21**(5), (1983), 667-676.
- [28] Y. SAKAWA: Feedback control of second order evolution equations with damping. *SIAM J. Control and Optimization*, **22**(3), (1984), 343-361.
- [29] Y. SAKAWA: Feedback control of second-order evolution equations with unbounded observation. *Int. J. Control*, **41**(3), (1985), 717-731.
- [30] J.M. SCHUMACHER: Dynamic Feedback in Finite- and Infinite -dimensional Linear Systems. Ph. D. dissertation, Dep. Math., Vrije Universiteit, Amsterdam, The Netherlands, 1981.
- [31] J.M. SCHUMACHER: A direct approach to compensator design for distributed parameter systems. *SIAM J. Control and Optimization*, **21**(6), (1983), 823-836.
- [32] D. SIEROCIUK, T. SKOVRANEK, M. MACIAS, I. PODLUBNY, I. PETRAS, A. DZIELINSKI and P. ZIUBINSKI: Diffusion process modeling by using fractional-order models. *Applied Mathematics and Computation*, **257**(1), (2015), 2-11.
- [33] M. SLEMROD: A note on complete controllability and stabilizability for linear control systems in Hilbert space. *SIAM J. Control*, **12**(3), (1974).
- [34] R. TRIGGIANI: On the stabilization problem in Banach space. *J. of Mathematical Analysis and Applications*, **52**(3), (1975), 383-403.
- [35] P.K.C. WANG: Model feedback stabilization of linear distributed system. *IEEE Trans. on Automatic Control*, **17**(4), (1972), 552-553.

An exact block algorithm for no-idle RPQ problem

JAROSLAW PEMPERA

In the work a single-machine scheduling problem is being considered, in which all tasks have a fixed availability (release) and delivery time. In the analyzed variant no-idle time is allowed on a machine. The purpose of optimization is to determine such order of tasks that minimizes the makespan, i.e. the time of execution of all the tasks. There is also a number of properties of the problem presented, in particular there are formulated block eliminating properties for no-idle constraint. There was an exact B&B algorithm based on the block properties proposed.

Key words: scheduling, single machine, no-idle, B&B algorithm.

1. Introduction

In a single-machine problem with jobs' release time and delivery time (RPQ) there must be a number of tasks executed on one machine. No idle time means that since the commencement of the first task the machine must perform the next task without interruption until the completion of the last task. Therefore, there must be a schedule of tasks minimizing delivery time of all the tasks determined.

No-idle constraint occurs in production systems, in which there are thermal or chemical reaction processes. In such systems, machine downtimes can not only cause failure of machinery, deterioration in the quality of products but can also generate additional costs related to energy expenditure. No-idle constraint can also be used in planning of outsourcing services, in which costs are dependent on the length of the contracts or in case of renting of expensive equipment for the execution of orders. Undoubtedly, both RPQ and RPQ no-idle problems are NP-hard. In case of RPQ problem there are efficient algorithms based on Branch-and-Bound (B&B) method proposed by Carlier [1] and Grabowski et al. [4]. In the first algorithm the node in the B&B search tree method is created in the time $O(n \ln n)$ using the algorithm Schrage [10], whereas in the second algorithm the nodes are created on the basis of the concept of a block task. Due to short duration of execution of these algorithms (a few seconds for 50–10000 tasks) they can be used to estimate lower bounds for the construction of multi-machine problems with

The Author is with Department of Automatics, Mechatronics and Control Systems, Faculty of Electronics, Wrocław University of Technology, Janiszewskiego str. 11-17, 50-372 Wrocław, Poland, e-mail: jaroslaw.pempera@pwr.edu.pl

Received 9.11.2016. Revised 16.05.2017.

the criterion of minimizing the completion time of all tasks such as: flow shop problems, job shop problems, flexible flow shop and job shop problems.

Construction of algorithms for simple scheduling problems is important in scientific context because the properties of these problems can be used in the construction of exact and heuristic algorithms for multi-machine scheduling problems. Discovered by Grabowski et al. block properties have been successfully used not only in the construction of exact algorithms for flow shop and job shop problems (the exact solution of small size), but also in the construction of one of the most effective heuristic algorithms for the following problems: flow shop [3, 7], job shop [8], no-store (blocking) flow shop [9] etc.

Study on task scheduling with no-idle constraint on machines mainly relate to the flow shop problem. There were proposed exact algorithms [11] and heuristic algorithms based on the construction [2], [5] and local search methods [6]. The paper presents the new properties of the problem with no-idle time on machine. Due to the significant similarity to the block properties of RPQ problem they will also be called block properties. Using these properties there has been an exact algorithm proposed based on the B&B method.

2. Problem description

On machine there must be executed n task from the set $J = \{1, \dots, n\}$. Each task has a known execution time $p_j > 0$, $j \in J$ and cannot start earlier than the release time $r_j \geq 0$. The machine cannot perform more than one task at a time. The tasks on the machine must be carried out without any interruption. Once the task is completed the finished product is delivered to the customer in time $q_j \geq 0$. Let S_j , C_j and D_j , be respectively the starting moment, the completion time and delivery moment of task j , $j \in J$. There must be determined the tasks schedule minimizing the moment of all the tasks delivery.

Let π be a permutation defined on the set J describing the sequence of tasks execution on one machine. Feasible schedule for tasks execution in the order of π must meet the following restrictions:

$$S_j \geq r_j \quad \text{for } j \in J, \quad (1)$$

$$C_j = S_j + p_j \quad \text{for } j \in J, \quad (2)$$

$$D_j = C_j + q_j \quad \text{for } j \in J, \quad (3)$$

$$S_{\pi(j)} = C_{\pi(j-1)} \quad \text{for } j = 2, \dots, n. \quad (4)$$

Constraints (1–3) are obvious, while equation (4) imposes the performance of tasks on the machine in order π with no idle time.

Let $D_{\max} = \max_{j \in J} D_j$ be the moment of all the tasks delivery. We want to find the sequence of tasks on the machine π^* , such that

$$D_{\max}(\pi^*) = \min_{\pi \in \Pi} D_{\max}(\pi), \quad (5)$$

where Π is the set of all permutations defined on the set J .

2.1. Example

There must be $n = 10$ tasks performed on a machine. The release times, execution times and delivery times of tasks are summarized in Table 1. In Fig. 1 there are three schedules for tasks execution presented in the Gantt chart form. Schedules (A) and (B) have been designated for the permutations generated by the use of Schrage algorithm (A-RPQ, B-RPQ no-idle) while schedule (C) is the optimal schedule for the RPQ no-idle problem. The delivery moment of all tasks is: (A)–42, (B)–46, (C)–43.

Table 1: Test data

Parametr	1	2	3	4	5	6	7	8	9	10
release time – r_i	9	16	15	27	6	4	26	12	3	11
execution time – p_i	4	1	1	2	1	1	2	4	1	1
delivery time – q_i	26	13	22	12	28	1	5	17	20	0

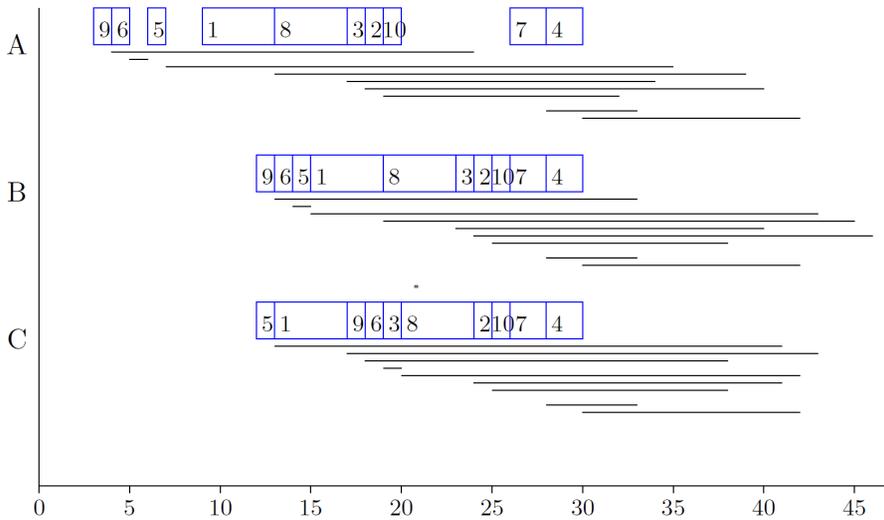


Figure 1: Gantt chart

3. Graph model

Before the presentation of graph model it must be noted that the equation (4) can be replaced by two inequalities:

$$S_{\pi(j)} \geq S_{\pi(j-1)} + p_{\pi(j-1)} \quad \text{dla } j = 2, \dots, n, \quad (6)$$

$$S_{\pi(j-1)} \geq S_{\pi(j)} - p_{\pi(j-1)} \quad \text{dla } j = 2, \dots, n. \quad (7)$$

For fixed processing order π , there is a graph $G(\pi) = (V, E(\pi))$ defined with the set of nodes V and a set of directed arcs $E(\pi)$. A set of nodes $V = J \cup \{s, t\}$ consists of n nodes representing tasks and two fictitious nodes s (initial, source) and t (final, target). The node representing the task j is burdened with weight equal to the duration of the task p_j , $j \in J$, while the fictitious nodes with weight equal to zero.

A set of arcs $E(\pi)$ consists of four subsets of arcs:

$$E^s = \{(s, j) : j \in J\}, \quad (8)$$

arc $(s, j) \in E^s$ corresponds to the constraint (1) and has weight r_j ,

$$E^t = \{(j, t) : j \in J\}, \quad (9)$$

arc $(j, t) \in E^t$ corresponds to the constraint (3) and has weight q_j ,

$$E^1(\pi) = \{(\pi(j-1), \pi(j)) : j = 2, \dots, n\}, \quad (10)$$

arc $(\pi(j-1), \pi(j)) \in E^1(\pi)$ corresponds to the constraint (6) and has weight 0. Arcs from the set $E^1(\pi)$ corresponds to the processing order of tasks π and will be called (in short) machine arcs. The last subset constitute arcs resulting from the no-idle constraint

$$E^2(\pi) = \{(\pi(j), \pi(j-1)) : j = 2, \dots, n\}, \quad (11)$$

arc $(\pi(j), \pi(j-1)) \in E^2(\pi)$ corresponds to the constraint (7) and has weight $-p_{\pi(j)} - p_{\pi(j-1)}$, which results from transformation of (7) to

$$S_{\pi(j-1)} \geq S_{\pi(j)} + p_{\pi(j)} - p_{\pi(j)} - p_{\pi(j-1)} \quad \text{for } j = 2, \dots, n. \quad (12)$$

Propertie 1 For fixed processing order of tasks execution π on a single machine, the earliest starting moment for the task j , $j \in J$ is equal to the length of the longest path to the node representing this task (without weight of node) in the graph $G(\pi)$.

Propertie 2 For fixed processing order π , the earliest delivery moment of all the tasks is equal to the length of the longest path to a node t in the graph $G(\pi)$.

It follows from the construction of a graph $G(\pi)$ that every path begins in the source node s , what is more, the longest path in the graph $G(\pi)$ to node t is also the longest path in this graph. This path will be called *critical path*.

The set of arcs of the graph $G(\pi)$ consists of arcs with negative weights, therefore, determination of the length of the longest paths requires the use of the Bellman-Ford algorithm with complexity of $|V| \cdot |E|$.

Propertie 3 *The lengths of the longest paths in $G(\pi)$ can be determined in time $O(n)$.*

Proof. For the proof, it is enough to note that the lengths of the longest paths have a fixed value after execution of a relaxation of arcs in the following order: (i) arcs from the set E^s , (ii) arcs from the set $E^1(\pi)$ in the order of $(\pi(1), \pi(2)), \dots, (\pi(n-1), \pi(n))$, (iii) arcs from the set $E^2(\pi)$ in the order of $(\pi(n), \pi(n-1)), \dots, (\pi(2), \pi(1))$, (iv) arcs from the set E^t .

Any critical path in the graph $G(\pi)$ begins with an arc belonging to E^s and ending with an arc belonging to E^t . Therefore, it will be described with the pair (a, b) , where $(s, a) \in E^s$ i $(b, t) \in E^t$. For $a = b$ the solution π is the optimal solution. Let us consider two cases: (i) $a < b$ and (ii) $a > b$.

In case of (i) a sequence $B_{a,b} = (\pi(a), \pi(a+1), \dots, \pi(b))$ will be called *block of tasks*. The block theorem [4] can be formulated as follows:

Propertie 4 *If $B_{a,b}$ is a block in π and $\beta \in \Pi$ is any permutation such that $D_{\max}(\beta) < D_{\max}(\pi)$, there is a task $\pi(k) \in \{\pi(a+1), \dots, \pi(b)\}$ (or $\pi(k) \in \{\pi(a), \dots, \pi(b-1)\}$) such that in the permutation β task $\pi(k)$ precedes all other tasks in a block (it is processed after all other tasks in a block).*

In case (ii) the sequence $B_{a,b} = (\pi(a), \dots, \pi(n), \pi(1), \dots, \pi(b))$ will be called *i-block of tasks*. In a way analogous to 4 property, the following property can be proved for the *i-block* $B_{a,b}$:

Propertie 5 *If $B_{a,b}$ is i-block in π and $\beta \in \Pi$ is any permutation such that $D_{\max}(\beta) < D_{\max}(\pi)$, there is a task $\pi(k) \in \{\pi(a+1), \dots, \pi(n)\}$ (or $\pi(k) \in \{\pi(1), \dots, \pi(b-1)\}$) such that in the permutation β task $\pi(k)$ precedes all other tasks in the set $\{\pi(a), \dots, \pi(n)\}$ (it is processed after all other tasks from the set $\{\pi(1), \dots, \pi(b)\}$).*

4. Exact block algorithm

With each node of Branch and Bound method there is associated a permutation determining the order of tasks π , block or *i-block* of tasks $B_{a,b}$ and the relation of partial order of R . The relation R implies a constraint "if $(i, j) \in R$, then task j can begin only after the task i is completed". The space of solutions is divided into subspaces (branching) on the basis of tasks in the block (or *i-block*) $B_{a,b}$.

A new node in a search tree of B&B algorithm is created by generating a new permutation and adding constraints to relation R . For $k = a + 1, \dots, x$ ($x = b$ for the block and $x = n$ for *i-block*) a new permutation is generated by shifting task $\pi(k)$ on position a in

π . To the relation R there are added constraints $\{(\pi(k), \pi(j)) : j \neq k, j = a, \dots, x\}$, which means that the task $\pi(k)$ must be executed before all other tasks from the set $\{a, \dots, x\}$. But for $k = x, \dots, b - 1$ ($x = a$ for the block and $x = 1$ for i -block) a new permutation is generated by shifting task $\pi(k)$ on position b w π . To the relation R there are added constraints $\{(\pi(j), \pi(k)) : j \neq k, j = x, \dots, b\}$ which means that the task $\pi(k)$ must be executed after all other tasks from the set $\{x, \dots, b\}$.

The search node of tree is cut off if the relation R is contradictory. Similarly as in the exact block algorithm for RPQ problem, the node generated by shifting task $\pi(k)$ is cut off, if $r_{\pi(k)} > r_{\pi(a)}$ in case of inserting into the position a and if $q_{\pi(k)} > q_{\pi(b)}$ in case of inserting into the position b .

In Fig. 2 there was the course of the proposed algorithm for data from example presented. Each node is described in a separate rectangle, in which there is a permutation, node number, and D_{\max} (in the lower right corner) presented. Tasks forming block (i -blok) were linked with the use of thick line, in addition, the tasks $\pi(a)$ and $\pi(b)$ were highlighted in bold.

Transfers of tasks resulting from the division strategy were illustrated by arcs, wherein the arcs leading to the nodes were indicated by dashed lines. The initial permutation π (Node 1) was generated by Schrage algorithm. In the permutations π there is i -block $B_{9,6}$. In the case of tasks 4 and 5 there are conditions $r_4 > r_7$ and $q_5 > q_3$ and $q_1 > q_3$ fulfilled, therefore the corresponding nodes will be cut off. Finally, as a result of the division there are three child nodes created by shifting: (i) task 8 to position 6 (node 2), (ii) task 6 to position 6 (node 7), (iii) task 9 to position 6 (node 12). Child nodes are connected to a parent node with an arc with the assigned number of a task shifted as a result of generating of a new permutations. In any child the number of restrictions remembered in relation of partial order increases. It is the main mechanism of cutting off child nodes. This can be seen on the example of node 11, wherein the child nodes generated by shifting tasks 5, 1, 3 and 8 were cut off because of the restrictions added while creating node 10, i.e. shifting tasks 9, which generates restriction: task 9 must be performed after tasks 5,1,8 and 3. The algorithm found the optimal solution in node 4 with a value of 43 and generated a total of 14 nodes (permutations) in reference to 10! all possible. There were tests conducted on the effectiveness of the block algorithm on multiple instances of data generated at random of size from 10 to 80 tasks. In most cases the number of nodes generated by algorithm did not exceed a few dozen and duration of action did not exceed a few hundred milliseconds. In addition, during the operation of the algorithm, i -blocks were generated relatively rarely. This means that for many instances of RPQ problem, in exact solution the removal of idle-time of machine does not increase the time of delivery of all tasks.

5. Summary

In the work, for single-machine scheduling problem with release and delivery dates with no-idle time on machines there was the notion of i -block of tasks formulated which

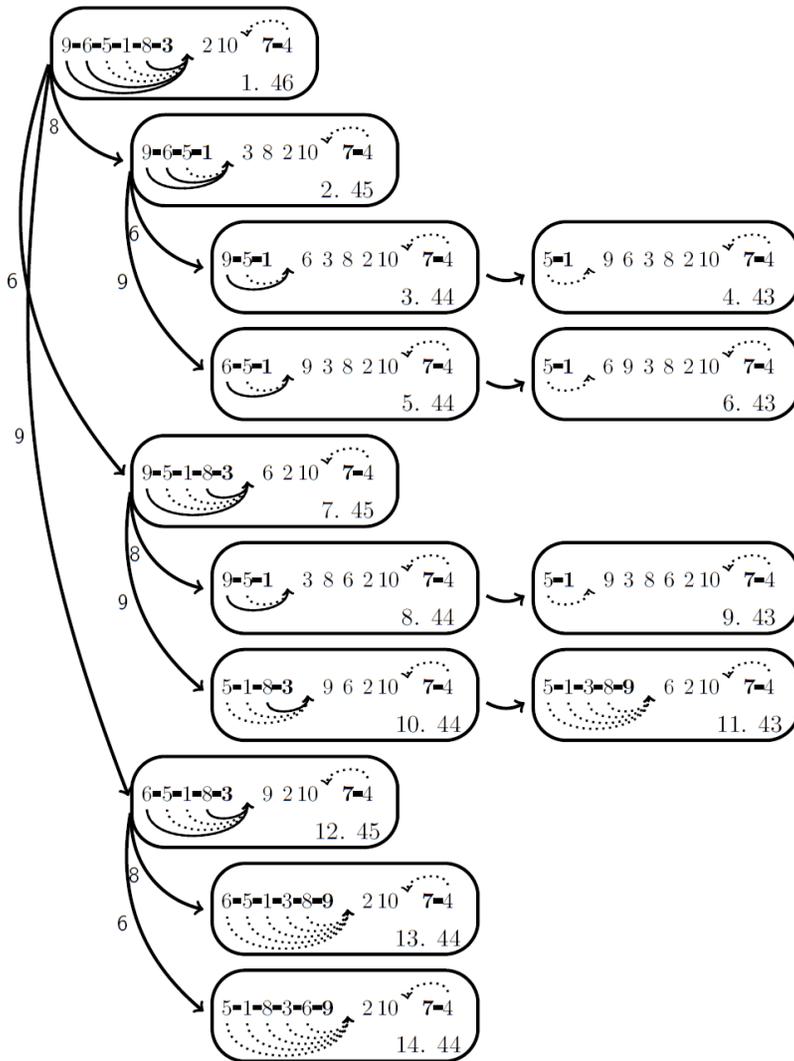


Figure 2: The calculation process for the data from example

is an extension of the well-known block of tasks. There was an exact algorithm proposed based on B&B method in which block properties have been used in the branch and bound strategy. It follows from the experimental studies on algorithms that solution to a significant number of instances of size of dozens of tasks takes no more than 1 second on a PC. The author of the paper believes that, based on *i*-blocks of tasks there can be efficient local search algorithms constricted for multi-machine tasks scheduling problems.

References

- [1] J. CARLIER: The one-machine sequencing problem. *European J. of Operational Research*, **11**(1), (1982), 42-47.
- [2] O. CEPEK, M. OKADA and M. VLACH: Note: On the two-machine no-idle flow-shop problem. *Naval Research Logistics*, **47**(4), (2000), 353-358.
- [3] J. GRABOWSKI and J. PEMPERA: New block properties for the permutation flow shop problem with application in tabu search. *The Journal of the Operational Research Society*, **52**(2), (2001), 210-220.
- [4] J. GRABOWSKI, E. NOWICKI and S. ZDRZAŚKA: A block approach for single-machine scheduling with release dates and due dates. *European J. of Operational Research*, **26** (1986), 278-285.
- [5] P.J. KALCZYNSKI and J. KAMBUROWSKI: A heuristic for minimizing the makespan in no-idle permutation flow shops. *Computers & Industrial Engineering*, **45** (2005), 146-154.
- [6] B. LIU, L. WANG and Y. JIN: An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, **35**(9), (2008), 2791-2806.
- [7] E. NOWICKI and C. SMUTNICKI: A fast taboo search algorithm for the job shop problem. *Management Science*, **24**(5), (1996), 530-534.
- [8] E. NOWICKI and C. SMUTNICKI: A fast tabu search algorithm for the permutation flow-shop problem. *European J. of Operational Research*, **91**(1), (1996), 160-175.
- [9] Q.-K. PAN and L. WANG: No-idle permutation flow shop scheduling based on a hybrid discrete particle swarm optimization algorithm. In press at *Int. J. of Advanced Manufacturing Technology*.
- [10] L. SCHRAGE: Obtaining optimal solution to resource constrained network scheduling problem. Unpublished manuscript, 1971.
- [11] P. VACHAJITPAN: Job sequencing with continuous machine operation. *Computers & Industrial Engineering*, **6**(3), (1982), 255-259.
- [12] R.E. KALMAN: Mathematical description of linear dynamical system. *SIAM J. Control*, **1**(2), (1963), 152-192.
- [13] F.C. SHWEPPE: *Uncertain Dynamic Systems*. Prentice-Hall, Englewood Cliffs, N.J., 1970.

The study of the influence of micro-environmental signals on macrophage differentiation using a quantitative Petri net based model

KATARZYNA RZOSIŃSKA, DOROTA FORMANOWICZ and PIOTR FORMANOWICZ

The complexity of many biological processes, which, thanks to the development of many fields of science, becomes for us more and more obvious, makes these processes extremely interesting for further analysis. In this paper a quantitative model of the process of macrophage differentiation, which is essential for many phenomena occurring in the human body, is proposed and analyzed. The model is expressed in the language of Petri net theory on the basis of one of the three hypotheses concerning macrophage differentiation existing in the literature. The performed analysis allowed to find an importance of individual factors in the studied phenomenon.

Key words: Petri nets, atherosclerosis, macrophages, t-invariants.

1. Introduction

The discovery of high independence between phenomena of macrophages differentiation and the T helper cells and the existence of macrophages subpopulations with different phenotype profiles M1 and M2 [19], opens new roads to better understanding the patomechanisms of many diseases and brings hope for their effective treatment in the future [28, 17].

The motivation for these studies were the results of recent research, which shed new light on immunological processes. Abnormal functioning of the immune system is the basis of many civilization-related diseases, inter alia the atherosclerosis. This disease ceased to be seen as a simple disorder caused by deposition of lipids in the arte-

K. Rżosińska is with Institute of Computing Science, Poznan University of Technology, Piotrowo str. 2, 60-965 Poznan, Poland, e-mail: katarzyna.rzosinska@cs.put.poznan.pl. D. Formanowicz is with Department of Clinical Biochemistry and Laboratory Medicine, Poznan University of Medical Sciences, Grunwaldzka str. 6, 60-780 Poznan, Poland, e-mail: doforman@ump.edu.pl. P. Formanowicz, the corresponding author, is with Institute of Computing Science, Poznan University of Technology, Piotrowo str. 2, 60-965 Poznan and Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego str. 12/14, 61-704 Poznan, Poland; e-mail: piotr.formanowicz@cs.put.poznan.pl

This research has been supported by the National Science Centre (Poland) grant No. 2012/07/B/ST6/01537.

Received 23.01.2017. Revised 19.05.2017.

rial endothelium. Now we know that its etiology is highly complex and associated with multiple processes, such as inflammation, oxidative stress, immune disturbances and hyperlipidemia. It should be emphasized here that macrophage accumulation within the vascular wall is a hallmark of atherosclerosis [3]. Therefore, the role of macrophages to keep organism homeostasis is crucial.

Unfortunately, the mechanisms underlying macrophage differentiation are not fully known. The macrophages are very important cells for the development of many diseases, thus it is significant to determine how they are maturing and how the process of activation of particular phenotypic groups is going. In this paper macrophage differentiation process has been studied. There are three main hypotheses about this phenomenon in the literature. We have considered them and chosen one for analysis. Because of the aforementioned complexity of the studied phenomenon, for its modeling and analysis a systems approach based on the language of Petri nets theory has been used. Hence, an in-depth analysis has been made.

In this article an extended version of the model proposed in [25] is presented and analyzed. The model has been extended by addition of the influence of macrophage-derived secretory products on the maturation of the monocytes.

The structure of the paper is as follows. In section 2 basic notions concerning Petri nets used in the other parts of this work are briefly presented. In section 3 biological aspects of the process of macrophage differentiation are described. In section 4 the Petri net based model of this process is presented, while in section 5 the results of its analysis are described. The paper ends with conclusions given in section 6.

2. Petri nets

Petri nets have been proposed in 1960s by Carl A. Petri as a formalism for modeling and analysis of concurrent computer systems [23]. For years they have been mainly used in the context of technical systems but in the mid 1990s it has been realized that nets of this type can be also used for describing and analysis of biological systems [24, 10].

A Petri net has a structure of a weighted directed bipartite graph. In such a graph a set of vertices can be divided into two disjoint subsets in such a way that there is no arc connecting vertices of the same subset. In Petri nets vertices from one of these subsets are called places while elements of the other subset are transitions. When a Petri net is a model of some biological system, places correspond to its passive components (e.g., substrates or products of reactions) and transitions are counterparts of some elementary subprocesses occurring in the system (e.g., chemical reactions). Hence, in Petri nets arcs connect places with transitions and transitions with places. There is also another, very important type of components of Petri nets, i.e., tokens. They bring dynamics to the net, which is one of its fundamental properties. Tokens flow between places through transitions, what corresponds to a flow of substances, information etc. in the modeled system. The flow of tokens is governed by a simple transition activation and firing rule, according to which a transition is active if in every place directly preceding this transition there

is a number of tokens which is equal to at least the weight of an arc connecting that place with the transition. An active transition can be fired what means that tokens flow from all places directly preceding it to places directly succeeding the transition and the number of flowing tokens is equal to the weight of an appropriate arc. There are two exceptions to this rule. A transition which is not preceded by any place, called an input transition, is continuously active (so, it can fire at any time). Moreover, a transition which is not succeeded by any place, called an output transition, does not generate tokens when fired. Input and output transitions often represent some interfaces between a modeled system and its environment. A distribution of tokens over the set of places, called marking, corresponds to a state of the modeled system [21].

There is a very intuitive graphical representation of Petri nets, where transitions are depicted as rectangles or bars, places as circles, arcs as arrows and tokens as dots or positive integer numbers within places. This representation is very helpful in understanding a structure of the model and in its simulation but it is not very well suited for an analysis of its formal properties. For this purpose another representation, called incidence matrix, is more suitable. In such matrix $A = (a_{ij})_{n \times m}$ rows correspond to places, columns correspond to transitions and entry a_{ij} is equal to the difference between the numbers of tokens residing in place p_i before and after firing transition t_j .

In the analysis of Petri net based models of biological systems especially important are transition invariants (t-invariants). An invariant of this type is vector x being a solution of the equation $A \cdot x = 0$. These vectors are dependent only on the structure of the network and the distribution of tokens does not affect them. For each Petri net minimal t-invariants [16] are looked for, i.e., those that are not linear combinations of other t-invariants. Algorithms for calculating t-invariants are described in several papers, e.g., in [4] (there are also freely available software tools for calculating invariants, e.g., INA [11], Charlie [9] and MonaLisa [5]). With t-invariant x there is associated set $s(x) = \{t_j : x_j > 0, j = 1, 2, \dots, m\}$ containing transitions corresponding to positive entries of vector x and is called a support of this invariant.

Supports correspond to subprocesses which do not change a state of the modeled system. Hence, they are especially important and an analysis of dependencies among such subprocesses may lead to discoveries of some unknown properties of the system. Such an analysis can be done by searching for similarities among t-invariants. These similarities (properly defined) correspond to common parts of the supports. Transitions being elements of support intersections correspond to elementary processes composing subprocesses being counterparts of the supports. These subprocesses can interact with each other through the common elementary processes. In order to find similar t-invariants they are usually grouped into sets called t-clusters using standard clustering algorithms. Each of such clusters contains t-invariants similar to each other according to some similarity measure. Moreover, each of them corresponds to some subprocess of higher order [8, 7]. There are many clustering algorithms and similarity measures which may provide various clusterings (e.g., sets of clusters). Hence, in order to obtain a proper clustering (from the viewpoint of the analysis of the model of the biological system), the algorithm, the

similarity measure and also the number of clusters should be carefully chosen, what is not an easy task.

In addition, in the case where a Petri net contains a great number of transitions, they can be grouped into the so-called Maximum Common Transition sets (MCT sets). Each of these sets contains transitions being elements of supports of exactly the same minimal t-invariants and corresponds to some functional module of the modeled system [8, 7]. The formal definition of MCT sets and some considerations about their importance in the biological context can be found, e.g., in [26].

In figure 1 there is shown an example of a simple Petri net consisting of six transitions and four places. Among them there are two input transitions, representing input signals (named Input1 and Input2) and output transition, representing an output signal (named Output). The net is covered by two t-invariants. The subnets corresponding to these invariants are marked with red and green arrows. As can be seen, the support of the t-invariant marked in green does not contain transitions Input2 and Output. These transitions are in the support of the second t-invariant (the red one) but it does not contain transition Input1 and those ones forming the cycle in the net. It can be observed that in order to appear signal “Output” signal “Input2” is necessary but signal “Input1” is not. In addition, signal “Input2” (together with “Input1”) is also necessary to start processes forming the cycle if in the initial marking there are no tokens in any of the places being elements of the cycle.

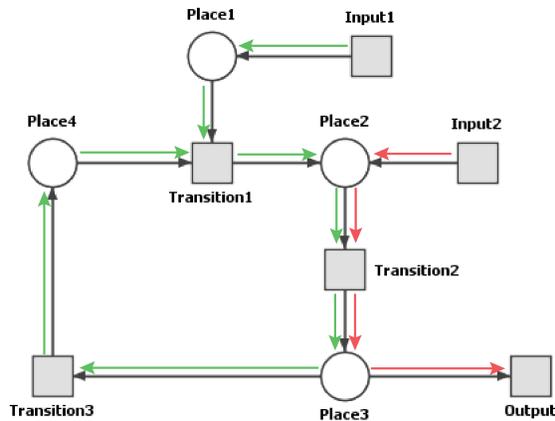


Figure 1: Simple Petri Net showing a cyclic process with two input and one output. It is covering by two t-invariants, marked by red and green arrows.

3. The process of macrophage differentiation

Macrophages, a type of white blood cells, are phylogenetically the oldest group of the immune cells. They are characterized by high heterogeneity and versatility of the immune response. Macrophages initiate and direct virtually all immune responses [20].

The main function of macrophages is phagocytosis, however, they are equipped with a number of other activities, which gives them the possibility to a comprehensive immune response. They produce, inter alia, a number of cytotoxic and effector molecules such as cytokines, reactive oxygen species (ROS), and reactive nitrogen species (RNS). In addition, macrophages belong to antigen presenting cells (APC), so they can activate the T-cells and start the specific immune response. Diversity of macrophages function allows to divide them into resident tissue macrophages, tissue macrophages, and monocyte-derived macrophages [12].

Resident macrophages differentiate in the early stages of embryonic development and have ability to regenerate its population. Other macrophages arise mainly as a result of monocytes maturation. Monocytes are also not homogeneous group. There are two subpopulations of monocytes, classical and non-classical differing in the expression level of certain surface receptors. Both of these populations are formed in the bone marrow, from where they go into the bloodstream. Then they are recruited into the tissues and differentiate into various types of macrophages [12].

M1 macrophages, also called inflammatory macrophages, are activated in the presence of pathogens and inflammatory factors. They are characterized by the generation of cytotoxic compounds, such as ROS, nitric oxide (NO), citrulline, high levels of IL-12, and low levels of IL-10. Moreover, these macrophages produce plurality of pro-inflammatory cytokines, such as IL-23, IL-1, IL-6 and tumor necrosis factor (TNF). This phenotypic profile is capable of an antigen presentation (APCs). However, metabolism adapted to the production of toxic substances is not indifferent for the cells. Generally, the macrophages activated in this subtype, are able to self deactivation, when the cytotoxic functions are no longer required for the status of the human organism. However, if the process in M1 mode lasts too long, cells are not longer able to deactivate, and their destructive metabolism ultimately leads to apoptosis [19].

M2 macrophages, called anti-inflammatory macrophages, create the second phenotype class. This is the basic activation program. Cells of these type are responsible for the maintenance of tissue homeostasis, remodeling, growth and regeneration of cells, damaged by injury or inflammation. Among this class, three subtypes are distinguished, i.e., M2a, M2b and M2c. They differ in the expression and their function in the organism. Attraction M2a induces Th2 response, stimulates the type II inflammatory response (cytotoxic) and combats parasites. It is characterized by secretion of IL-10, IL-1ra, polyamides, and decoy IL-1RII. In turn, the M2b program has immunological response regulatory activity and is distinguished by TNF, IL-1, IL-6 secretion, high levels of IL-10, and low levels of IL-12 secretion. These types of cells are also involved in the activation of Th lymphocytes and inhibition of tumor growth. M2c, a third of these programs, is referred as immunosuppressant and is designed to inhibit inflammatory reactions and is involved in reconstruction and recomposing of the tissues. M2c produce, among others, IL-10 and transforming growth factor beta (TGF- β). They have a unique metabolic machinery (plasticity) that allows them to switch from M1 to M2 [20]. However, these populations are much more heterogeneous, meaning that the M1 (routine heal mode) and M2 (inhibit mode) are extrema of the spectrum of intermediate phenoty-

phes. A multitude of factors influencing the phenotype of these versatile cells makes that macrophages may take mixed activity.

There are three main hypotheses of macrophage differentiation. The first one assumes that each one of subpopulations of monocytes can differentiate into specific macrophages phenotype. According to this, classical monocytes and monocyte-derived tissue macrophages can differentiate into M1 macrophages, while M2 macrophages are formed from non-classical monocytes and resident macrophages [12]. According to the second hypothesis, macrophages phenotype depends on factors affecting differentiation of monocytes in a tissue. Micro-environmental signals and cytokines present in the tissue are different for various inflammatory conditions. The resultant of these factors affect to the expression of the macrophages involved in the immune response. Usually local population of macrophages include both subtypes. However, the percentage of any fraction is different and affected by many factors and circumstances. At the beginning of the inflammation an amount of M1 macrophages is much larger than M2 fraction, the number of which increases with time and reaches the largest share in the post-inflammatory phase [12, 1]. According to the third hypothesis, mature macrophages have ability to change their phenotype from pro-inflammatory (M1/inhibit mode) to anti-inflammatory (M2/heal mode), and vice versa, depending on the different conditions in the tissues [13, 22, 27].

In vitro studies have demonstrated that macrophage activation into M1 phenotype takes place under the influence of infectious agents such as lipopolysaccharide (LPS), granulocyte macrophage colony stimulating factor (GM-CSF), and pro-inflammatory cytokines: tumor necrosis factor alpha (TNF- α) and interferon gamma (IFN- γ). Activation of the M2 program is dependent on the presence of certain cytokines (IL-4 and IL-13), anti-inflammatory agents (IL-10, TGF- β) and activity Fc and TL receptors. Influence of IL-4 and IL-13 gives rise to a subpopulation M2a of macrophages. Activation of phenotype M2b depends on the simultaneous launch of an immune complex receptors Fc γ and TL. M2C - the third of the programs, is stimulated by IL-10 and TGF- β .

The activity of particular classes of macrophage phenotypes stimulates the differentiation and activity of another monocytes. M2b activity leads to a local increase in the level of IL-10, which in turn leads to the formation of increased number of macrophages M2c. In contrast, the IL-12 secretion by M1 macrophages activates T-cells, which produce IFN- γ , which, as stated above, is a major factor inducing the classic macrophage activation M1. The share of particular subpopulation of macrophages in the tissue has an impact on the subsequent differentiation of macrophages [18].

For better understanding the intricacies of the discussed biological process, it has been schematically depicted in the figure 2. According to the accepted hypothesis, the possibility of differentiation all types of macrophages to all phenotypic classes, depending on the microenvironment conditions was considered. Some of macrophages tend to differentiate into a specific phenotype more than others. It has been presented by the lines thickness. Inflammatory factors, and their effect on the macrophage differentiation to type M1 and stimulation of recruitment of classical monocytes to the tissue, are highlighted in red. In green, yellow and blue color are marked anti-inflammatory factors

which promote the formation M2a, M2b and M2c macrophages, respectively. Each phenotypic class has a different expression profile and level of interleukins secretion. All types of macrophages secrete IL-10 but M1 secrete it in very limited amount, what is signified on the figure by dotted lines.

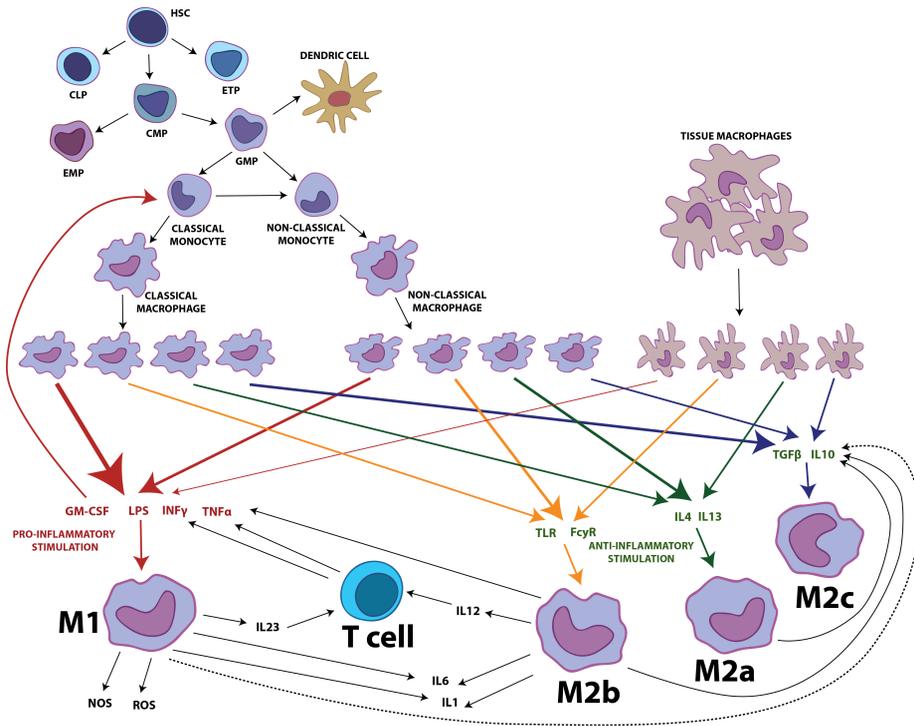


Figure 2: The analyzed process of macrophages polarization.

4. The model

The model created for the purpose of this study has been based on the second hypothesis of macrophage differentiation, according to which the macrophage phenotype depends on the micro-environmental signals. Accordingly, the model is limited to the differentiation and maturation of monocytes into macrophages, without taking into account the possibility of changing phenotype of already mature macrophages. Moreover, the ability to inactivate macrophages and the presence of memory macrophages has not been taken into account. The model contains the activity of selected cytokines, directly related with the macrophages differentiation process. In addition to monocyte derived macrophages, four main types of resident tissue macrophages are included in the model: Langerhans cells, microglial cells, Kupffer cells and Alveolar macrophages. The Petri

net based model consists of 67 transitions and 41 places described in Tables 1 and 2 and it is shown in Fig 3.

With each of the transitions there is associated a literature reference, mentioned in Table 3. The analyzed process is not autonomous, the individual elements are involved in other reactions in the organism. This fact, likewise the formal requirements of the Petri net, made it necessary to include output transitions to the model, that signify the contribution of some factors in other physiological processes, whose specification was not important for this study. For example, only part of IL-23 produced by M1 macrophages is involved in stimulating T-cells, other molecules of IL-23 may participate in other processes not included in the model. All similar phenomena are indicated in column “Biological meaning” of Table 2 by word “output”.

The model described in this paper is available in SBML and SPEED formats on the website <http://www.cs.put.poznan.pl/krzosinska/research.html>

Table 1: List of places

Place	Biological meaning	Place	Biological meaning
p_0	Langerhans cells (LC)	p_{21}	lipopolysaccharides (LPS)
p_1	microglial cells	p_{22}	toll-like receptors (TLR)
p_2	Kupffer cells	p_{23}	receptors for the Fc region of IgG (Fc γ R)
p_3	Alveolar macrophage	p_{24}	IL-4
p_4	hematopoietic stem cells (HSC)	p_{25}	IL-13
p_5	common lymphocyte progenitor (CLP)	p_{26}	IL-10
p_6	common myeloid progenitor (CMP)	p_{27}	transforming growth factor β (TGF- β)
p_7	early T-lineage progenitor (ETP)	p_{28}	M2a
p_8	granulocyte and monocyte progenitor(GMP)	p_{29}	M2b
p_9	megakaryocyte/erythrocyte progenitors(MEP)	p_{30}	M2c
p_{10}	macrophage/dendritic cells progenitors (MDP)	p_{31}	M1
p_{11}	common monocyte progenitor(cMoP)	p_{32}	nonclassical monocyte
p_{12}	common dendritic progenitors (CDP)	p_{33}	classical monocyte
p_{13}	dendritic cells	p_{34}	nitric oxide (NO)
p_{14}	non-classical monocyte in blood	p_{35}	ROS
p_{15}	classical monocyte in blood	p_{36}	IL-6
p_{16}	monocyte derived macrophages	p_{37}	IL-12
p_{17}	non-classical macrophage	p_{38}	IL-23
p_{18}	tumor necrosis factor alpha (TNF- α)	p_{39}	IL-1
p_{19}	interferon γ (IFN- γ)	p_{40}	T cells
p_{20}	granulocyte-macrophage colony-stimulating factor (GM-CSF)		

THE STUDY OF THE INFLUENCE OF MICRO-ENVIRONMENTAL SIGNALS ON MACROPHAGE DIFFERENTIATION USING A QUANTITATIVE PETRI NET BASED MODEL

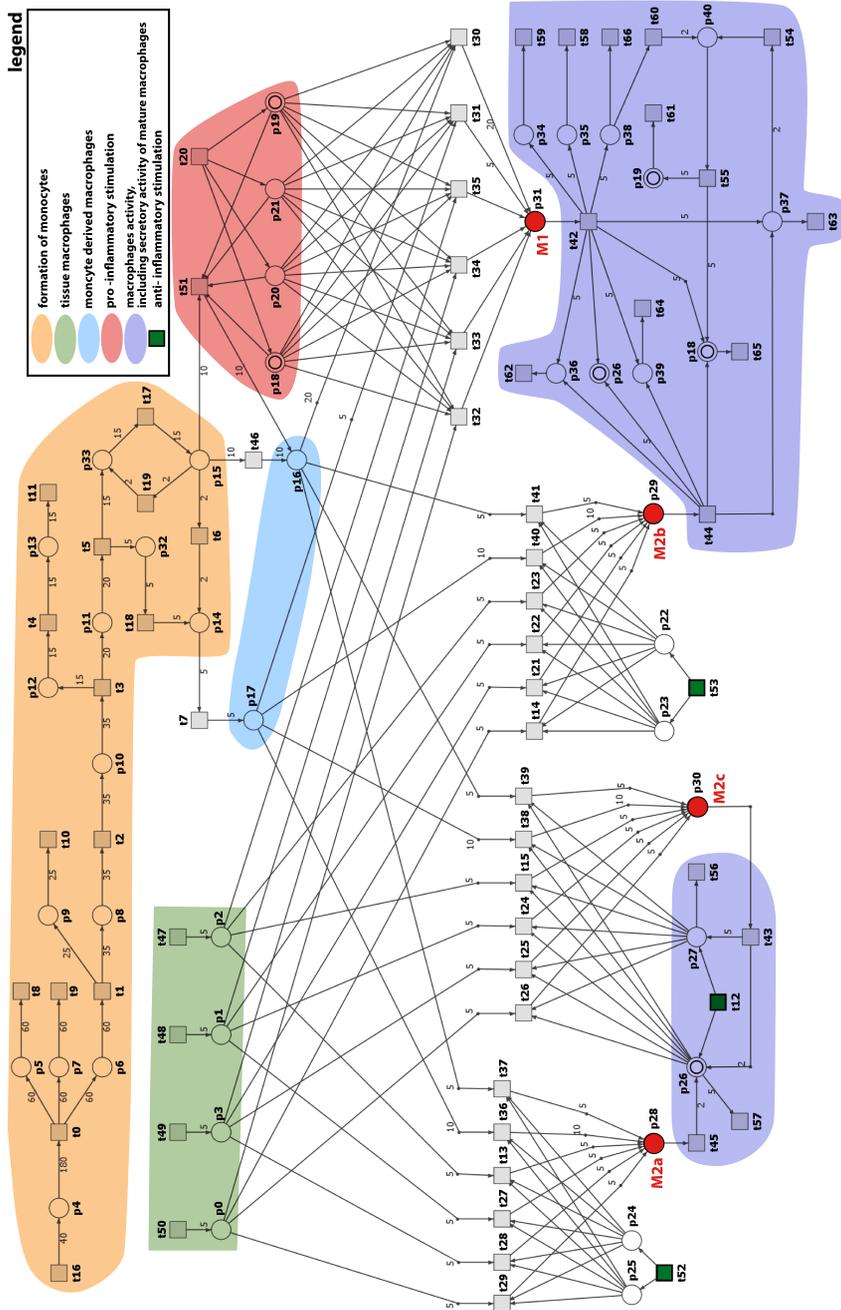


Figure 3: Macrophage differentiation Petri net based model, with biological descriptions of the most important parts of the net.

Table 2: List of transitions

Transition	Biological meaning	Transition	Biological meaning
t_0	HSC differentiation	t_{34}	microglial to M1 activation
t_1	CMP differentiation	t_{35}	Kupffer cells to M1 activation
t_2	GMP differentiation	t_{36}	non-classical monocytes to M2a activation
t_3	MDP differentiation	t_{37}	classical monocytes to M2a activation
t_4	maturation of dendritic cells (DC)	t_{38}	non-classical monocytes to M2c activation
t_5	exit from the bone marrow	t_{39}	classical monocytes to M2c activation
t_6	loss of expression of Ly6C	t_{40}	non-classical monocytes to M2b activation
t_7	non-classical monocytes maturation	t_{41}	classical monocytes to M2b activation
t_8	lymphoid line	t_{42}	M1 activity
t_9	formation of T-cells	t_{43}	M2c activity
t_{10}	erythropoiesis	t_{44}	M2b activity
t_{11}	presenting antigens	t_{45}	M2a activity
t_{12}	anti-inflammatory stimulation	t_{46}	recruitment of classical monocytes into tissue
t_{13}	Kupffer cells M2a activation	t_{47}	Kupffer cells self-renewal
t_{14}	LC to M2b activation	t_{48}	microglial self-renewal
t_{15}	Kupffer cells M2c activation	t_{49}	Alveolar macrophage self-renewal
t_{16}	HSCs self-renewal	t_{50}	LC self-renewal
t_{17}	transport to blood classical monocytes	t_{51}	increase in the recruitment of classical monocytes into tissue
t_{18}	transport to blood non-classical monocytes	t_{52}	IL-13 i IL-4 input
t_{19}	return to the marrow of classical monocytes	t_{53}	Fc γ R and TLR input
t_{20}	proinflammatory stimulation	t_{54}	T-cells activation by IL-12
t_{21}	alveolar macrophage to M2b activation	t_{55}	T-cells activity
t_{22}	microglial to M2b activation	t_{56}	TGF- β activity
t_{23}	Kupffer cells to M2b activation	t_{57}	IL-10 activity
t_{24}	microglial to M2c activation	t_{58}	ROS output
t_{25}	alveolar macrophage to M2c activation	t_{59}	NO output
t_{26}	LC to M2c activation	t_{60}	T-cells activation by IL-23
t_{27}	microglial to M2a activation	t_{61}	IFN output

Transition	Biological meaning	Transition	Biological meaning
t_{28}	alveolar macrophage to M2a activation	t_{62}	IL-6 output
t_{29}	LC to M2a activation	t_{63}	IL-12 output
t_{30}	classical monocytes to M1 activation	t_{64}	IL-1 output
t_{31}	non-classical monocytes to M1 activation	t_{65}	TNF output
t_{32}	LC to M1 activation	t_{66}	IL-23 output
t_{33}	alveolar macrophage to M1 activation		

Table 3: The references to transitions

References	Transition
[6]	$t_0, t_1, t_2, t_3, t_4, t_8, t_{10}, t_{11}, t_{16}, t_{52}, t_{53}, t_{54}, t_{58}, t_{59}, t_{61}, t_{62}, t_{63}, t_{64}, t_{65}, t_{66}$
[12]	$t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}, t_{21}, t_{22}, t_{23}, t_{24}, t_{25}, t_{26}, t_{27}, t_{28}, t_{29}, t_{30}, t_{31}, t_{32}, t_{33}$
[18]	$t_{42}, t_{43}, t_{44}, t_{45}, t_{51}$
[19]	$t_{12}, t_{42}, t_{43}, t_{44}, t_{45}, t_{46}$

5. Analysis of the model

The net is covered by 126 t-invariants. Moreover, there are six multi-element MCT sets and 42 trivial (i.e., single-element) MCT sets.

The biological meaning of t-invariants has been established on the basis of the analysis of the significance of the transitions from the support of particular t-invariants. Covered the entire Petri Net by t-invariants, and determining the biological significance for each of them, is a key step in the analysis of biological Petri Net models.

Invariant x_1 corresponds to flow of classical monocytes between the bone marrow and blood. Invariant x_6 describes outside activity of IL-10 and TGF- β caused by anti-inflammatory stimulation. The Tables 4 and 5 describe the remaining invariants that reflect the tissue macrophages differentiation and bone marrow-derived macrophages differentiation to all phenotypic classes, as well as their activity, including the ability to stimulate T cells. Table 4 shows the bone marrow derived macrophages and the resident macrophages, differentiating only into one of specific phenotypes. In Table 5 there are described those t-invariants which depict the formation of mixed populations. In this group a major phenotype class arises from non-classical monocyte, and the second phenotype from classical monocytes. There do not exist invariants indicating that tissue macrophages are involved in the formation of mixed populations. Also not found any

t-invariant, which suggests formation of a mixed population consisting more than two phenotypes.

In the model there exist six multielement MCT sets shown in the Table 6 and 42 Trivial MCT-transitions shown in the Table 7. MCT m_1 corresponds to monocytes formation in the bone marrow and maturation of non-classical monocytes. The set of m_2 presents M1 macrophages activity for the formation of cytotoxic compounds like ROS and RNS. Set m_3 comprises three output transitions corresponding to outflow of TNF- α , IL-1 and IL-6 from the system. m_4 and m_5 are two-element sets of transitions, corresponding to stimulation of macrophages to differentiation to M2a and M2b and the activity of these macrophages. Set m_6 also includes only two transitions - they correspond to synthesis and outflow of IFN- γ from the system.

To find additional biological links between elements of the model, clusters analysis was performed. The t-invariants have been clustered, using several algorithms: UPGMA, Centroid method, complete linkage, McQuitty's method, median method, single linkage and Ward's method, including the following metrics: Correlation, Pearson, Minkowski distance, Maximum distance, Canberra distance, Manhattan distance, Euclidean distance and binary distance. Calinski-Harabasz index [2] and Mean Split Silhouette (MSS) [14] were used to appoint the best algorithm and the best similarity measure with the optimal numbers of clusters. The MMS allow to determine the best clustering method, by checking how each t-invariants is matched to given class. The high CH value means the number of clusters generated by tested clustering method is better, than clustering with lower CH value.

The high values of both coefficients showed clustering using the Pearson and Correlation measures. Finally, clustering by UPGMA algorithm with Pearson similarity measure has been chosen. It grouped t-invariants into six t-clusters (see in Table 8). In this case, the coefficient values were high on mean for the entire clustering, as well as for each cluster separately. In addition, this clustering contains only one single transition clusters.

Single-element cluster c_1 corresponds to circulation of classic monocytes between bone marrow and bloodstream (invariant x_1). All other clusters contain invariants corresponding to formation of monocytes in the bone marrow and their transport into a blood and transformation of classical monocytes into non-classical ones, by the loss of expression, and increasing a recruitment of classical monocytes into the tissue by proinflammatory factors. Clusters c_2 , c_3 and c_6 contain t-invariants that represent T-cells activation, wherein in c_2 and c_3 clusters T-cells are activated by IL-12, while in cluster c_6 activation is influenced by two interleukins, i.e., IL-12 and IL-23. In clusters c_4 and c_5 , there is no corresponding t-invariants to the T cells activation. In turn, the input transitions corresponding to the proinflammatory and anti-inflammatory factors that stimulate macrophages to activate the relevant phenotypes, i.e., M1, M2a and M2b are present in all the clusters. However, an input transition for the external stimulation for differentiating of macrophages to the M2c type occurs only in the cluster c_3 . The obtained clusters demonstrate differences in the presence of t-invariants corresponding to the differentia-

tion of macrophages (tissue and the bone marrow-derived) to the different phenotypes, and activity of the differentiated macrophages:

- Cluster c_2 (11 t-invariants): contains invariants corresponding to the differentiation of tissue macrophages to M2a, non-classical monocytes to M2a and M2b and classical monocytes to M2a.
- Cluster c_3 (18 t-invariants): contains invariants corresponding to the differentiation of tissue macrophages to M2c, non-classical monocytes to all classes of phenotypes and classical monocytes to M2a and M2c.
- Cluster c_4 (11 t-invariants): contains invariants corresponding to the differentiation of tissue macrophages to M2b, non-classical monocytes to M2a, M2a, M2c and classical monocytes to M2b.
- Cluster c_5 (17 t-invariants): contains invariants corresponding to the differentiation of tissue macrophages to M1, non-classical monocytes to all classes of phenotypes and classical monocytes to M2a, M2b and M1.
- Cluster c_6 (68 t-invariants): contains invariants corresponding the differentiation of tissue macrophages M1 and M2b, and differentiation of both non-classical and classical monocytes to all classes of phenotypes.

Table 4: t-invariants corresponding to various ways of T-cell activation. The homogeneous population of macrophages.

M1	tissue macrophages	x_{77}	x_{78}	x_{79}	x_{80}	x_{81}	x_{82}	x_{83}	x_{84}
	non-classical monocytes	x_{85}	x_{86}	x_{87}	x_{88}	x_{89}	x_{90}	x_{91}	x_{92}
M2a	tissue macrophages	x_2	x_3	x_4	x_5				
	non-classical monocytes	x_{73}	x_{74}	x_{75}	x_{76}				
M2b	tissue macrophages	x_{11}	x_{12}	x_{13}	x_{14}	x_{23}	x_{24}	x_{25}	x_{26}
	non-classical monocytes	x_{69}	x_{72}						
M2c	tissue macrophages	x_7	x_8	x_9	x_{10}				
	non-classical monocytes	x_{52}							

The table shows t-invariants corresponding to differentiation of tissue and non-classical macrophages to only one of phenotypic population. t-invariants corresponding to T-cell activation by IL-23 are marked in blue, those corresponding to activation of T-cells by IL-12 are marked in green while in yellow are marked t-invariants corresponding to activation of T-cells by both of these interleukins. The t-invariants which are not marked correspond to the cases, where the path of T-cell stimulation is not active.

Table 5: t-invariants corresponding to various ways of T-cell activation. The mixed population of macrophages.

M1	M1	x_{65}	$x_{66} \uparrow$	x_{67}	$x_{68} \uparrow$	x_{69}	$x_{70} \uparrow$	x_{71}	$x_{72} \uparrow$	
	M2a	x_{76}	$x_{82} \uparrow$	x_{79}	$x_{85} \uparrow$	x_{77}	$x_{83} \uparrow$	x_{80}	$x_{86} \uparrow$	
	M2b	x_{105}	$x_{107} \uparrow$	$x_{116} \uparrow$	x_{119}	$x_{121} \uparrow$	x_{106}	$x_{108} \uparrow$	x_{120}	$x_{122} \uparrow$
	M2c	x_{88}	$x_{94} \uparrow$	x_{91}	$x_{97} \uparrow$	x_{89}	$x_{95} \uparrow$	x_{92}	$x_{98} \uparrow$	
M2a	M1	x_{42}	$x_{43} \uparrow$	x_{48}	$x_{49} \uparrow$	x_{44}	$x_{45} \uparrow$	x_{50}	$x_{51} \uparrow$	
	M2a	x_{36}	$x_{37} \uparrow$							
	M2b	x_{40}	$x_{41} \uparrow$	x_{46}	$x_{47} \uparrow$					
	M2c	x_{38}	$x_{39} \uparrow$							
M2b	M1	x_{101}	$x_{102} \uparrow$	x_{115}	x_{103}	$x_{104} \uparrow$	x_{117}	$x_{118} \uparrow$		
	M2a	x_{75}	$x_{81} \uparrow$	x_{78}	$x_{84} \uparrow$					
	M2b	x_{99}	$x_{100} \uparrow$	x_{113}	$x_{114} \uparrow$					
	M2c	x_{87}	$x_{93} \uparrow$	x_{90}	$x_{96} \uparrow$					
M2c	M1	x_{59}	$x_{60} \uparrow$	x_{65}	$x_{66} \uparrow$	x_{61}	$x_{62} \uparrow$	x_{67}	$x_{68} \uparrow$	
	M2a	x_{53}	$x_{54} \uparrow$							
	M2b	x_{57}	$x_{58} \uparrow$	x_{63}	$x_{64} \uparrow$					
	M2c	x_{55}	$x_{56} \uparrow$							

The table shows t-invariants corresponding to macrophage differentiation leading to the formation of mixed populations. The first column shows populations of macrophages resulting from the non-classical monocytes. In the second column there are subpopulations arising from classical monocytes. As in Table 4 t-invariants corresponding to T-cell activation by IL-23 are marked in blue, those corresponding to activation of T-cells by IL-12 are marked in green while in yellow are marked t-invariants corresponding to activation of T-cells by both of these interleukins. The t-invariants which are not marked correspond to the cases, where the path of T-cell stimulation is not active. The arrow at an invariant indicates the occurrence of an increase in the recruitment of classical monocytes by inflammatory agents.

Table 6: List of non-trivial MCT sets

MCT set	Contained transitions	Biological meaning
m_1	$t_0, t_1, t_2, t_3, t_4, t_5, t_7, t_8, t_9, t_{10}, t_{11}, t_{16}, t_{18}$	monocytes formation in the bone marrow and maturation of non-classical monocytes
m_2	t_{42}, t_{58}, t_{59}	M1 macrophages activity for the manufacture of cytotoxic compounds like ROS and RNS.

MCT set	Contained transitions	Biological meaning
m_3	t_{62}, t_{64}, t_{65}	outflow of TNF- α , IL-1 and IL-6 from the system
m_4	t_{44}, t_{53}	stimulate macrophages to differentiate to M2b and the M2b activity
m_5	t_{45}, t_{52}	stimulate macrophages to differentiate to M2a and the M2a activity
m_6	t_{55}, t_{61}	production and outflow IFN- γ from the model

Table 7: Trivial MCT sets

Transitions composing trivial MCT sets
$t_6, t_{12}, t_{13}, t_{14}, t_{15}, t_{17}, t_{19}, t_{20}, t_{21}, t_{22}, t_{23}, t_{24}, t_{25}, t_{26}, t_{27}, t_{28}, t_{29}, t_{30}, t_{31}, t_{32}, t_{33}, t_{34}, t_{35}, t_{36}, t_{37}, t_{38}, t_{39}, t_{40}, t_{41}, t_{43}, t_{46}, t_{47}, t_{48}, t_{49}, t_{50}, t_{51}, t_{54}, t_{56}, t_{57}, t_{60}, t_{63}, t_{66}$

Table 8: Clusters composition

t-cluster	t-invariants
c_1	x_1
c_2	$x_2, x_3, x_4, x_5, x_{35}, x_{36}, x_{37}, x_{75}, x_{78}, x_{81}, x_{84}$
c_3	$x_6, x_7, x_8, x_9, x_{10}, x_{38}, x_{39}, x_{52}, x_{53}, x_{54}, x_{55}, x_{56}, x_{87}, x_{88}, x_{90}, x_{93}, x_{94}, x_{96}$
c_4	$x_{11}, x_{12}, x_{13}, x_{14}, x_{40}, x_{41}, x_{57}, x_{58}, x_{69}, x_{99}, x_{100}$
c_5	$x_{15}, x_{16}, x_{17}, x_{18}, x_{42}, x_{43}, x_{59}, x_{60}, x_{70}, x_{76}, x_{82}, x_{101}, x_{102}, x_{105}, x_{107}, x_{109}, x_{110}$
c_6	$x_{19}, x_{20}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}, x_{31}, x_{32}, x_{33}, x_{34}, x_{44}, x_{45}, x_{46}, x_{47}, x_{48}, x_{49}, x_{50}, x_{51}, x_{61}, x_{62}, x_{63}, x_{64}, x_{65}, x_{66}, x_{67}, x_{68}, x_{71}, x_{72}, x_{73}, x_{74}, x_{77}, x_{79}, x_{80}, x_{83}, x_{85}, x_{86}, x_{89}, x_{91}, x_{92}, x_{95}, x_{97}, x_{98}, x_{103}, x_{104}, x_{106}, x_{108}, x_{111}, x_{112}, x_{113}, x_{114}, x_{115}, x_{116}, x_{117}, x_{118}, x_{119}, x_{120}, x_{121}, x_{122}, x_{123}, x_{124}, x_{125}, x_{126}$

6. Conclusions

The previously published [25] our basic model focused mainly on the influence of the micro-environment on the macrophage differentiation process. The main problem that occurred during the building of the model was the discrepancy between knowledge and the availability of accurate, quantitative and qualitative data, which are required by the formalism of Petri nets. The model that has been build and then analyzed in this paper has been extended (as compared to the previous one) by mutual macrophages interactions. It should be underlined that the activity of the already differentiated macrophages to the particular phenotype is very important, because it affects next differentiating

macrophages. The products of the activity of every phenotype are factors that directly affect the subsequent differentiation of macrophages or indirectly affect this differentiation, inter alia by the activation of T-cells. The analysis of the extended model allowed to observe new relationships between the cells.

The assumption of the model was based on the possibility of differentiation of all types of macrophages to both of the phenotypic classes. The origin of macrophages and an influence of micro-environmental factors affect a likelihood of the acquisition of the specific phenotypic activity by them. Proinflammatory factors, contributing to the activation of phenotype M1, stimulate the recruitment of classical monocytes from the bloodstream to the target tissues, in order to enhance macrophages population that is involved in the local inflammation.

From the analysis of the model there have been drawn several conclusions. It has been revealed that macrophages recruited into tissue, by inflammatory stimulation, can differentiate into all phenotypic classes, not only to the proinflammatory one (M1). This means that the proinflammatory factors can initiate the anti-inflammatory response of macrophages M2, by increasing the pool of macrophages in a tissue. In the model there are also t-invariants suggesting that the stimulation of monocytes migration by proinflammatory factors leads to the increased maturation of macrophages (to the anti-inflammatory phenotypes) from the M2 macrophages without leading to the activation of the proinflammatory M1 phenotype. Furthermore, due to high inflows of classical monocytes into a tissues it is possible the rise of a mixed population of macrophages in the immune response by simultaneous activation of a part of the macrophages to the different phenotypes. In the model mixed populations of macrophages contain only two phenotypes in which classical macrophages play only auxiliary roles and independently does not differentiate into a homogeneous population.

Although various types of macrophages differ in their activity considerably and also in their secreted substances, IL-10 was always produced in the analyzed network. For this reason an external stimulation of macrophages for differentiation to the phenotype M2c has no real effect on the formation of such macrophages. Macrophages M2c are formed only as a result of macrophage activity of other classes of phenotypes, which produce IL-10.

M1 and M2b macrophages can activate T-cells, but it does not always happen. Macrophages M1, regardless of the origin, produce IL-12 and IL-23, however, it does not always lead to the activation of T-cells. Such activation may only take place with the participation of one of the cytokines or both of them. Similarly, macrophages M2b always produce IL-12, but this does not always lead to activation of the T-cells.

The above findings, formulated on the basis of t-invariants, MCT sets and clustering analyzes, could be interesting starting points for further study of macrophage differentiation mechanisms. A further work on modeling and analysis of macrophage differentiation will focus on the impact of other immune system cells on the studied phenomenon and mutual impact of interleukins and cytokines on macrophages.

References

- [1] L. ARNOLD, A. HENRY, F. PORON, Y. BABA-AMER, N. VAN ROOIJEN, A. PLONQUET, R.K. GHERARDI and B. CHAZAUD: Inflammatory monocytes recruited after skeletal muscle injury switch into antiinflammatory macrophages to support myogenesis. *J. of Experimental Medicine*, **204** (2007), 1057-1069.
- [2] T. CALINSKI and J. HARABASZ: A dendrite method for cluster analysis. *Communications in Statistics*, **3** (1974), 1-27.
- [3] G. CHINETTI-GBAGUIDI, S. COLIN and B. STAEL: Macrophage subsets in atherosclerosis. *Nature Reviews Cardiology*, **12** (2015), 10-17.
- [4] J.M. COLOM and M. SILVA: Convex geometry and semiflows in P/T nets: a comparative study of algorithms for computation of minimal P-semiflows. *Lecture Notes in Computer Science*, **483** (1991), 79-112.
- [5] J. EINLOFT, J. ACKERMANN, J. NÖTHEN and I. KOCH: MonaLisa-visualization and analysis of functional modules in biochemical networks. *Bioinformatics*, **29** (2013), 1469-1470.
- [6] P.J. DELVES, S.J. MARTIN, D.R. BURTON and I.M. ROITT: Roitt's Essential Immunology. 13th Edition. Wiley-Blackwell, 2017.
- [7] D. FORMANOWICZ, A. KOZAK, T. GŁOWACKI, M. RADOM and P. FORMANOWICZ: Hemojuvelin-hepcidin axis modeled and analyzed using Petri nets. *J. of Biomedical Informatics*, **46** (2013), 1030-1043.
- [8] D. FORMANOWICZ, A. SACKMANN, A. KOZAK, J. BŁAŻEWICZ and P. FORMANOWICZ: Some aspects of the anemia of chronic disorders modeled and analyzed by petri net based approach. *Bioprocess and Biosystems Engineering*, **34** (2011), 581-595.
- [9] M. HEINER, M. SCHWARICK and J. WEGENER: Charlie – an extensible Petri net analysis tool. In *Proc. PETRI NETS*, Brussels, Springer, LNCS, **9115** (2015), 200-211.
- [10] R. HOFESTÄDT: A Petri net application of metabolic processes. *J. of System Analysis, Modelling and Simulation*, **16** (1994), 113-122.
- [11] INA Home page. Available at <http://www2.informatik.hu-berlin.de/starke/ina.html>.
- [12] P. ITALIANI and D. BORASCHI: From monocytes to M1/M2 macrophages: phenotypical vs. functional differentiation. *Frontiers in Immunology*, **5** (2014), 514.

- [13] P. ITALIANI, E.M. MAZZA, D. LUCCHESI, I. CIFOLA, C. GEMELLI, A. GRANDE, C. BATTAGLIA, S. BICCIATO and D. BORASCHI: Transcriptomic profiling of the development of the inflammatory response in human monocytes in vitro. *PLoS One*, **9** (2014), e87680.
- [14] L. KAUFMAN and P.J. ROUSSEEUW: Finding groups in data: an introduction to cluster analysis. John Wiley and Sons, New York, 1990.
- [15] I. KOCH, W. REISIG and F. SCHREIBER (Ed.): Modeling in Systems Biology. The Petri Net Approach. Springer, London, 2011.
- [16] K. LAUTENBACH: Exakte Bedingungen der Lebendigkeit für eine Klasse von Petri-Netzen. Number 82 in Berichte der GMD. Gesellschaft für Mathematik und Datenverarbeitung, Sankt Augustin, 1973.
- [17] N. LEITINGER and I.G. SCHULMAN: Phenotypic polarization of macrophages in atherosclerosis. *Atherosclerosis, Thrombosis and Vascular Biology*, **13** (2013), 1120-1126.
- [18] A. MANTOVANI, A. SICA, S. SOZZANI, P. ALLAVENA, A. VECCHI and M. LOCATI: The chemokine system in diverse forms of macrophage activation and polarization. *Trends in Immunology*, **25** (2004), 677-686.
- [19] F.O. MARTINEZ and S. GORDON: The M1 and M2 paradigm of macrophage activation: time for reassessment. *F1000 Prime Reports*, **6** (2014), 13.
- [20] C.D. MILLS and K. LEY: M1 and M2 Macrophages: The Chicken and the Egg of Immunity. *J. of Innate Immunology*, **6** (2014), 716-726.
- [21] T. MURATA: Petri nets: Properties, analysis and applications. *Proc. of the IEEE*, **77** (1989), 541-580.
- [22] K.J. MYLONAS, M.G. NAIR, L. PRIETO-LAFUENTE, D. PAAPE and J.E. ALLEN: Alternatively activated macrophages elicited by helminth infection can be reprogrammed to enable microbial killing. *J. of Immunology*, **182** (2009), 3084-3094.
- [23] C.A. PETRI: Communication with automata. *Schriften des Instituts für Instrumentelle Mathematik*, Bonn, (1962), (in German).
- [24] V.A. REDDY, M.L. MAVROVOUNIOTIS and M.N. LIEBMAN: Petri net representation in metabolic pathways. *Proc. of the 1st Int. Conf. on Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, (1993), 328-336.
- [25] K. RZOSIŃSKA, D. FORMANOWICZ and P. FORMANOWICZ: Ilościowy model procesu różnicowania makroforagów oparty na sieciach Petriego. In: A. Świerniak, J. Krystek (Eds.): Automatyzacja procesów dyskretnych. Teoria i zastosowania,

Vol. II, 205-216, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2016, (in Polish).

- [26] A. SACKMANN, M. HEINER and I. KOCH: Application of Petri net based analysis techniques to signal transduction pathways. *BMC Bioinformatics*, **7** (2006), 482.
- [27] R.D. STOUT, C. JIANG, B. MATTA, I. TIETZEL, S.K. WATKINS and J. SUTTLES: Macrophages sequentially change their functional phenotypes in response to changes in microenvironmental influences. *J. of Immunology*, **175** (2005), 342-349.
- [28] C. VAROL, A. MILDNER and S. JUNG: Macrophages: Development and Tissue Specialization. *Annual Review of Immunology*, **33** (2015), 643-675.
- [29] J. YANG, L. ZHANG, C. YU, X-F YANG and H. WANG: Monocyte and macrophage differentiation: circulation inflammatory monocyte as biomarker for inflammatory disease. *Biomarkers Research* **2** (2014), 1.

Vehicle navigation in populated areas using predictive control with environmental uncertainty handling

KRZYSZTOF SKRZYPCZYK and MARTIN MELLADO

This paper addresses the problem of navigating an autonomous vehicle using environmental dynamics prediction. The usefulness of the Game Against Nature formalism adapted to modeling environmental prediction uncertainty is discussed. The possibility of the control law synthesis on the basis of strategies against Nature is presented. The properties and effectiveness of the approach presented are verified by simulations carried out in MATLAB.

Key words: motion planning, prediction, uncertainty handling, game theory.

1. Introduction

Nowadays more and more mobile robotic solutions are used outdoor. That means robotic vehicles often have to work in populated, dynamic environments under very little human supervision. However coexisting with humans and operating efficiently in such environment, requires a robot must be able to navigate in harmony with traffic participants - humans. Thus the problem of automated navigation in dynamic environments has become an important challenge of contemporary Robotics [1] [3] [5] [8] [9]. In contrast to static and supervised environments, navigating robot in dynamic and uncertain conditions requires many issues to be solved e.g. moving objects detection and tracking, environmental changes prediction, motion planning, and many others. The most common approach to a synthesis of navigation systems which are intended to operate in dynamic conditions is predictive control. Its effectiveness depends on quality of the environmental dynamics model which is created using environment's state observations. Inherent features of each prediction are its inaccuracy and uncertainty. The first one is usually related to a quality of measurement devices or procedures. The second one concerns predictability of the process dynamics. The navigation strategy is calculated using the future, estimated environment's state. In the case the prediction model is ac-

K. Skrzypczyk (corresponding author), is with Institute of Automatic Control, Silesian University of Technology, Akademicka str. 16, 44-100 Gliwice, Poland, e-mail: krzysztof.skrzypczyk@polsl.pl. M. Mellado is with Instituto de Automática e Informática Industrial Universidad Politécnica de Valencia, Spain, e-mail: martin@ai2.upv.es

This work has been partially supported by BK grant.

Received 9.01.2017. Revised 27.04.2017.

curate enough and the process is predictable, the navigation plan computed using the prediction usually results in successful, collision free motion even for relatively long time horizon. But in the case when it differs significantly from the real one, the vehicle movement strategy may lead to a collision. Therefore it is very important to provide methods for handling prediction uncertainty in the control process. There are some approaches to uncertainty handling and analysis. The most popular approaches reported in the literature are: probabilistic approach, fuzzy set based one as well as methods based on Bayesian network formalism [6]. Unfortunately they are quite technically challenging and thus hard to apply in real-world robotic solutions. The key issue in uncertainty robust control is to take action which minimize a risk resulting from a difference between the modeled and the real state of the control process. The technique which allows for relatively easy modelling decision processes uncertainty is Game Theory, in particular Games Against Nature (G.A.N.) [7]. Nature in the G.A.N. formalism symbolizes fictitious player whose behavior cannot be considered as rational. Generally G.A.N. are commonly used in economics, telecommunication or informational sciences. However, this technique can be used also in engineering and technical problems, like control or robots motion planning [10] [11] [4]. Using the game theoretical framework prediction based motion planning can be formulated as a sequence of two-person games in the strategical form. In such games one player is identified with the motion planning system. This system is able to influence the navigation process using decisions (controls) taken from its admissible decision space. The second player is identified with possible to happen, environmental states influencing the process. In this case, these states can be defined as alternative scenarios corresponding to prediction deviations coming from model inaccuracies. By that means uncertainty regarding the prediction can be modeled and involved in the motion control problem. Playing various strategies against such opponent we can obtain the motion plan which takes into account prediction uncertainty. In the remaining part of this paper the motion problem is stated and modeled in the game theoretical framework. Exemplary simulations are presented to illustrate the approach discussed.

2. Problem overview

A discussion on uncertainty handling in the predictive navigation of a vehicle is concerned with a system framework presented in Fig.1. An operation idea of this system can be described as follows: The perception subsystem provides information about environment state. The information is assumed to be valuable enough to provide objects' detection and tracking.

In this work none of specific sensory system is considered, but it is assumed it is able to provide information about objects' location. These information generally can be noised and affected by many inaccuracies and they are stored in a data buffer of the size defined. Using these data the objects' motion model is created and a prediction of objects' location is computed in the given time horizon. The aforementioned inaccuracies

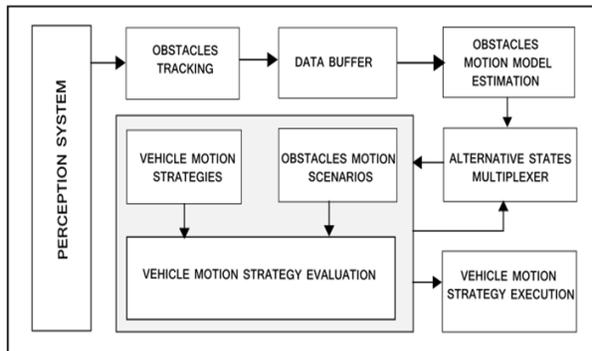


Figure 1: A diagram of the predictive navigation system with uncertainty handling

influence the prediction. These predictions are used for finding the navigation strategy - speed and direction of movement which minimize the risk of collision. It is assumed the environment changes prediction is not precise and can differ (even much) from the real state. In order to deal with this fact, alternative scenarios of predicted environmental state are taken into account. These scenarios are input data for the decision making module. In this work the scenarios represent assumed different than identified motion model's parameters. The role of this module is to evaluate particular environmental state - navigation strategies combinations and find the best navigation strategy in the sense of criteria applied. A reasonable navigation strategy should minimize the collision risk in the prediction horizon, on the one hand, and minimize the deviation from the desired movement direction, on the other one.

2.1. Navigation task formulation

Usually, navigation problems are decomposed, and analysed as a sequence of simple, short-term, point-to-point navigation ones. The solution of such problem is a sequence of controls that allows the vehicle reaching the desired position defined in the base coordinate frame. In this work the problem is reformulated. From the wheelchair driver perspective the more convenient way for defining the short navigation goal is to follow selected straight-line path instead of reaching an intermediate way point. Let us define the desired path selected in the time n as a tuple:

$$\Upsilon_0(n) = (l(n), \Delta w) \quad (1)$$

where $l(n)$ is the ray starting at the point $(x_{R,n}, y_{R,n})$, angled at $\theta_{R,n}^*$. The threshold value Δw determines maximal, feasible vehicle distance to the line $l(n)$. The path (1) defines a lane of a width Δw , the vehicle is supposed to move inside. Thus a primary navigation task can be perceived as a sequence of paths (1) which allows for reaching the destination point. Such an action is more intuitive in the context of a social navigation process.

2.2. Problem statement

$$R(n) = [x_{R,n}, y_{R,n}, \theta_{R,n}]^T \quad (2)$$

Let us assume (2) the vehicle is equipped with a sensory system which is able to detect N objects moving in its sensing range and to calculate their positions. Moreover the system is provided with a buffer in which information on objects locations collected in the past can be stored. Assuming the system is a discrete time one with a sampling period Δt , let us define the set containing locations of objects:

$$P_i = \{p_{i,k}\}, i = 1, 2, \dots, N, k = \langle n - M, n \rangle \quad (3)$$

where $p_{i,k}$ is a vector containing coordinates defining location of the i th object in the k th time point. The current time is indexed by n and the number of past observations stored in the buffer is equal to M . Using data defined in (3) an estimation of future locations of objects can be found:

$$\hat{p}_{i,h}^m = \bar{p}_i^m(h), h = n + 1, \dots, n + H \quad (4)$$

where H denotes the prediction horizon and m is the model used for calculating the prediction.

Now the short-term navigation task which is the vehicle movement in the desired direction can be stated as follows: In the current time n , using the information on estimated, future objects' location (4), find the control (the direction and velocity of the vehicle):

$$u_R^*(n) = [\theta_R(n), v_R(n)]^T \quad (5)$$

which applied for H subsequent sampling periods minimize the risk of collision with obstacles and provides the smallest possible deviation from the desired travel direction.

3. Prediction uncertainty handling

The vehicle motion strategy is determined using the environmental state prediction (4) and thus highly depends on the quality of this anticipation. Each prediction bears uncertainty and therefore it must be taken into account while planning the motion. In this study we propose to handle prediction uncertainty using game theory framework. Let us consider the process as a two-person game in a strategical form:

$$G = (D, C) \quad (6)$$

where D is a decision space of the game while C denotes costs incurred in the aftermath of decisions taken. The decision space is defined as a Cartesian product of the player decision set D_G and the decision set D_N of a fictitious opponent named Nature [7]. The

uncertainty considered in this study is represented by various states of Nature specified in the set (7). In turn, each state of Nature represents hypothetical navigation scenario in the presence of circumstances different than predicted in (4). Therefore the uncertainty is handled by considering a number of predictions obtained using various models. Thus the decision set of Nature is defined by:

$$D_N = \{m\}, \quad m \in [1, K] \quad (7)$$

where m is the model used for calculating the prediction (4). The player's decision set contains admissible controls that can be taken by the vehicle. Let us define this set as:

$$D_G = \left\{ u_{R,k} = [v_{R,k}, \theta_{R,k}]^T \right\} = V_R \times \Theta_R \quad (8)$$

where:

$$V_R = \{v_{R,i}\}, i = 1, \dots, \bar{V}_R, \quad \Theta_R = \{\theta_{R,j}\}, j = 1, \dots, \bar{\Theta}_R \quad (9)$$

The set V_R contains admissible values of the vehicle velocity. Similarly the set Θ_R contains possible values of the vehicle orientation that can be taken. The motion parameters values defined in (8) correspond to actions usually taken by pedestrians, e.g. marching faster, slower, giving way to an approaching object, stopping etc. Thus, the size of the decision space is not large. A result of applying the control chosen from (8) in the presence of hypothetical scenario distinguished in (7) is evaluated using the function which is to be minimized:

$$C : D \rightarrow \Re \quad (10)$$

The values of this function are represented by the matrix $C = [c_{ij}]$. Rows of this matrix represent admissible controls $i \in D_G$, while columns the scenario $j \in D_N$ considered. The function (10) modelling process is presented in the next section.

4. Process modelling

Let us assume a two component form of the cost evaluation index:

$$C(u_{R,k}, m) = \beta_r C_{risk}^m + \beta_p C_{pdev} \quad (11)$$

where C_{risk}^m denotes the navigation risk assessment according to the strategy $u_{R,k}$ assuming the m th prediction variant (7). The second component expresses the cost of going off the primary path (1).

4.1. Navigation risk evaluation

In this work the following risk evaluation index is proposed:

$$C_{risk}(u_{R,k}, m) = \frac{1}{\sum_{h=n+1}^H \min_{i=1, \dots, N} |\hat{p}_{R,h}(u_{R,k}) - \hat{p}_{i,h}^m|_{L=2}} \quad (12)$$

This index evaluates the risk of applying the action $u_{R,k} \in D_G$ for the time H and is calculated by summarizing distances to the nearest objects. Its value is calculated for the m th prediction variant and is minimized while finding the best control strategy with respect to the collision avoidance aspect. An influence of this component on the index value is tuned experimentally using the weighting factor β_r .

4.2. Getting off the path cost

The next aspect of the control $u_{R,k}$ suitability evaluation is calculating the cost related to the path (1) tracking error. Since the vehicle is intended to move inside the lane defined by (1) the suitability of the action is evaluated twofold:

$$C_{pdev}(u_{R,k}) = C_{dist} + C_{ang} \quad (13)$$

The first component expresses the cost related to getting off the primary path. The second one maps vehicle movement direction change into the cost space. These costs are calculated for the vehicle movement prediction in the time horizon H . The first component is determined using the following formula:

$$C_{dist}(u_{R,k}) = w_d \sum_{h=n+1}^{n+H} \hat{d}(\hat{p}_{R,h}, l(n)) \quad (14)$$

where $\hat{d}(\hat{p}_{R,h}, l(n))$ is the predicted vehicle-path distance calculated in the time point h according to:

$$\hat{d}(\hat{p}_{R,h}, l(n)) = \begin{cases} \hat{d}(\hat{p}_{R,h}, l(n)) & \text{for } \hat{d}(\hat{p}_{R,h}, l(n)) \geq \Delta w \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

The weighting factor w_d is tuned experimentally and is used for balancing the influence of control goals defined by (13). The second component of the cost function is related to the vehicle orientation change resulting from taking the action. This factor is modeled as:

$$C_{ang}(u_{R,k}) = w_a |\theta_{R,k} - \theta_{R,n}^*| \quad (16)$$

The weighting factor w_a similarly as in (14) is tuned experimentally.

5. Control strategy

The next stage, after designing and calculating the cost function (11) is to find the best control strategy. This strategy has to take into account various possible environmental scenarios representing states of Nature. Therefore the control for the time horizon H is determined as the solution of G.A.N. In this study Hurwicz criterion is applied. This criterion characterises an optimistic decision maker, who expects favorable for him circumstances to happen. According to his policy, with some level of optimism α , selects a strategy which is the result of the following optimization task:

$$i_0 = \min_i [\alpha \min_j c_{ij} + (1 - \alpha) \max_j c_{ij}] \quad (17)$$

Please note that for α equal to 0 this criterion is equivalent to the pessimistic (Wald) one.

6. Simulation results

In order to evaluate the approach considered in this paper, simulation study is carried out. The system functioning is evaluated using a number of scenarios typical for crowded space navigation. In this section an exemplary, very common scenario, is selected as an illustration. The simulated navigation task consists in moving the vehicle along the path set with the given velocity in the presence of moving object. The desired path of the vehicle is marked with the ray outgoing from the point denoting the vehicle position in the time $t=0$. The real path of the object is drawn with circles while the predicted path with dots.

Figure 2a shows a situation when the vehicle movement strategy is determined using the object movement prediction. In this case the system trusts the model and does not take into account uncertainty of the model. The object changes its movement direction rapidly after the prediction is computed. This situation results in accurate tracking the desired path but leads to collision danger. In a second experiment (Fig. 2b) prediction uncertainty handling mechanism is applied. The vehicle movement parameters are computed using the object movement prediction but with Hurwicz criterion for $\alpha = 0$, assuming five various states of Nature. We can see that considering possible to happen scenarios results in much safer vehicle trajectory. This in turn caused the vehicle went off the path much more. The last experiment (fig. 2c) illustrates the same scenario but solved for $\alpha = 0.5$. That what can be observed is the tradeoff in the vehicle control. The vehicle gets off the path less, but the distance to the object increases. These experiments show that the method allows to take into account possible environmental changes and adapts the movement strategy to uncertain environmental conditions.

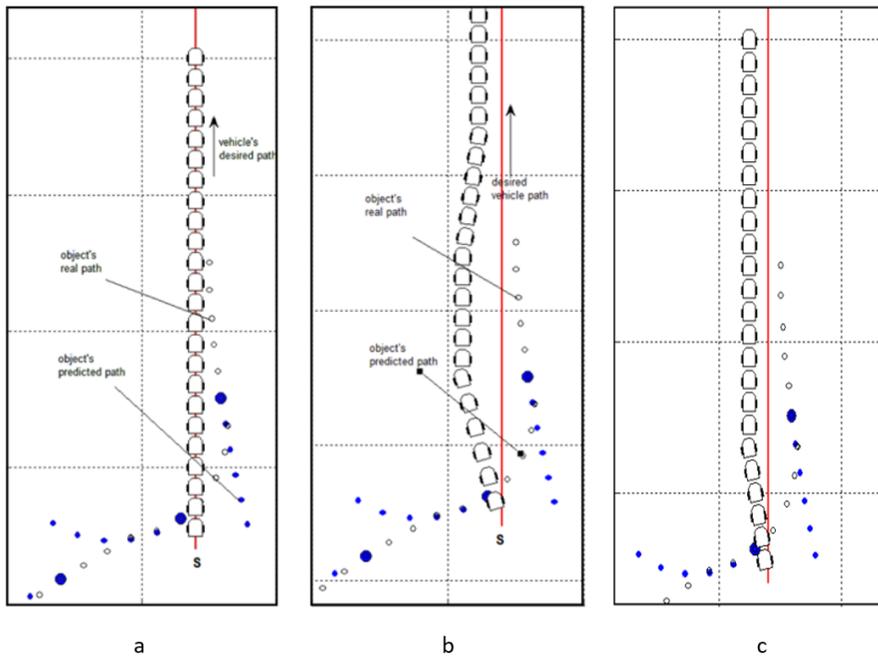


Figure 2: Predictive motion planning without uncertainty handling (a), using Hurwicz criterion for $\alpha = 0$ (b) and for $\alpha = 0.5$ (c)

7. Conclusions

In this paper the control system designed for navigating an automated vehicle is discussed. The vehicle is intended to stabilise a rectilinear movement in a direction chosen by the user. Moreover the navigation system is intended to avoid collisions with objects appearing on the course of the vehicle. Since the vehicle is assumed to navigate in populated, crowded spaces the control strategy has to follow the human-aware navigation rules. The main goal of this research is to verify the long term predictive control strategy as a method of generating the socially acceptable movement of the vehicle. The conception presented is simulated using a variety of scenarios. An application of game theory based methods to modelling the uncertainty of prediction model improved robustness of the navigation algorithm. The methodology of the uncertainty handling using G.A.N. seems to be promising and can be applied to many various problems in which the uncertainty modelling play an important role or is indispensable.

References

- [1] S. CHOI, S. KIM and S. OH: Real-time navigation in crowded dynamic environments using Gaussian process motion control. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, (2014).
- [2] D. DING and R.A. COOPER: Electric-powered wheelchairs: A review of current technology and insight into future directions. *IEEE Control Systems Magazine*, **25**(2), (2005), 22-34.
- [3] D. FEIL-SEIFER and M. MATARIC: People-aware navigation for goal-oriented behavior involving a human partner. *IEEE Int. Conf. on Development and Learning*, Germany, **2** (2011), 24-27.
- [4] A. GALUSZKA and A. SWIERNIAK: Non-cooperative game approach to multi-robot planning. *Int. J. of Applied Mathematics and Computer Science*, **15**(3), (2005), 359-367.
- [5] C. GAO, M. SANDS and J.R. SPLETZER: Towards autonomous wheelchair systems in urban environments. In *Proc. of Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Hamburg, Germany, (2012), 77-82.
- [6] L.S.DUTT and M. KURIAN: Handling of uncertainty - A Survey. *Int. J. of Scientific and Research Publications*, **3**(1), (2013).
- [7] J. MILNOR: Games Against Nature, in: R. Thrall, C. Coombs and R. Davis (Eds.), *Decision Processes*, Wiley, New York, 1954, 49-59.
- [8] M. MOUSSAID, D. HELBING and G. THERAULAZ: How simple rules determine pedestrian behavior and crowd disasters. *Proc. of the National Academy of Science of the United States of America*, **108**(17), (2011), 6884-6888.
- [9] D. SHI, E.G. COLLINS JR., A. DONATE and X. LIU: Human-aware robot motion planning with velocity constraints. *Int. Symp. on Collaborative Technologies and Systems*, Irvin, CA, USA, (2008), 490-497.
- [10] K. SKRZYPCZYK: Control of a team of mobile robots based on non-cooperative equilibria with partial coordination. *Int. J. of Applied Mathematics and Computer Science*, **15**(1), (2005), 89-97.
- [11] K. SKRZYPCZYK: Time optimal tracking a moving target by a mobile vehicle - game theoretical approach. *Przegląd Elektrotechniczny*, **86**(3), (2010), 211-215.