APPLIED COMPUTER SCIENCE

The Journal is a peer-reviewed, international, multidisciplinary journal covering a broad spectrum of topics of computer application in production engineering, technology, management and economy.

The main purpose of Applied Computer Science is to publish the results of cutting-edge research advancing the concepts, theories and implementation of novel solutions in computer technology. Papers presenting original research results related to applications of computer technology in production engineering, management, economy and technology are welcomed.

We welcome original papers written in English. The Journal also publishes technical briefs, discussions of previously published papers, book reviews, and editorials. Especially we welcome papers which deals with the problem of computer applications in such areas as:

- manufacturing,
- engineering,
- technology,
- designing,
- organization,
- management,
- economics,
- innovations,
- competitiveness,
- quality and costs.

The Journal is published quarterly and is indexed in: BazTech, Cabell's Directory, CNKI Scholar (China National Knowledge Infrastucture), ERIH PLUS, Index Copernicus, J-Gate, Google Scholar, TEMA Technik und Management.

Letters to the Editor-in-Chief or Editorial Secretary are highly encouraged.

CONTENTS

Applied Computer Science, vol. 14, no. 2, pp. 5–19
doi:10.23743/acs-2018-09

Submitted: 2018-05-14
Revised: 2018-06-03
Accepted: 2018-06-21

time-dependent orienteering problem with time-windows, evolutionary algorithm, public transport network, tourist trip planning

Krzysztof OSTROWSKI[*]

AN EFFECTIVE METAHEURISTIC FOR TOURIST TRIP PLANNING IN PUBLIC TRANSPORT NETWORKS

Abstract

The Time-Dependent Orienteering Problem with Time Windows (TDOPTW) is a combinatorial optimization problem defined on graphs. Its real life applications are particularly associated with tourist trip planning in transport networks, where travel time between two points depends on the moment of travel start. In the paper an effective TDOPTW solution (evolutionary algorithm with local search operators) was presented and applied to generate attractive tours in real public transport networks of Białystok and Athens. The method achieved very high-quality solutions in a short execution time.

1. INTRODUCTION

The Time-Dependent Orienteering Problem with Time Windows (TDOPTW) belongs to the Orienteering Problem (OP) family. The classic OP is defined on a weighted graph with nonnegative profits associated to vertices and nonnegative costs associated to edges. The goal of the OP is to find a path between given two vertices that maximizes total profit of visited vertices and its total cost does not exceed a given limit. The OP solution does not have to contain all vertices (usually it is impossible because of cost limit) and each vertex can be visited only once.

The Time-Dependent Orienteering Problem with Time Windows (TDOPTW) is a generalization of the OP defined for time-dependent graphs. Edge costs are identified with travel times, which depend on the moment of travel start (edge

[*] Faculty of Computer Science, Białystok University of Technology, Wiejska 45A, 15-001 Białystok, Poland, k.ostrowski@pb.edu.pl

weights are time-dependent functions). In addition each vertex has a visit time and a time-window. Arriving too early means waiting for the time-window to open while arriving too late makes it impossible to visit a given vertex.

Transport networks are examples of time-dependent graphs are. Travel time between two points depends on traffic intensity (i.e. longer during rush hours) and timetables (in case of public transport networks). Problems from the TDOPTW family have many practical applications including tourist trip planning (Garcia, Vansteenwegen, Arbelaitz, Souffriau & Linaza, 2013) and transport logistics. In tourist trip planning each tourist attraction (point of interest – POI) has some profit (i.e. dependent on its popularity), visit time and opening hours (time-window). Finding an attractive tour of a limited duration in a time-dependent transport network is equivalent to solving the TDOPTW.

The paper is organized as follows. In section 2 a mathematical formulation of the TDOPTW is given. In section 3 a literature review is presented. Section 4 describes public transport network as a time-dependent graph. Section 5 describes methods applied. In section 6 experimental results are given. Section 7 is the conclusion of the paper.

2. PROBLEM DEFINITION

Let $G = (V, E)$ be a directed, weighted graph. Each vertex i has a nonnegative profit p_i, a nonnegative visit time τ_i and a time-window $<o_i, c_i>$. Travel time between vertices i and j ($i, j \in V$) is a nonnegative function $w_{ij}(t)$ dependent on the moment of travel start t. The goal of the TDOPTW is to find a path from vertex s to vertex e starting at time t_0 which maximizes total profit of visited vertices without exceeding total travel time (T_{max}) and without violating time-windows of visited vertices. TDOPTW can be formulated as a Mixed Integer Programming (MIP) problem. Let's introduce three additional variables. Variable x_{ij} is 1 if a path contains a direct travel from vertex i to vertex j and 0 otherwise. Let ta_i and td_i be a time of arrival at vertex i and a time of departure from it – these functions are defined only for vertices included in the path. It's assumed that vertices s and e have no profit, zero visit time and no time-window. The purpose of the TDOPTW is to maximize formula 1 without violating constraints 2–8:

$$\max \sum_{i,j \in V} p_i \cdot x_{ij} \tag{1}$$

$$\sum_{j \in V} x_{sj} = \sum_{i \in V} x_{ie} = 1 \tag{2}$$

$$\bigvee_{k \in V \setminus \{s,e\}} \left(\sum_{i \in V} x_{ik} = \sum_{j \in V} x_{kj} \leq 1 \right) \tag{3}$$

$$td_s = t_0 \tag{4}$$

$$\bigvee_{i,j \in V} \left(x_{ij} = 1 \Rightarrow ta_j = td_i + w_{ij}(td_i) \right) \tag{5}$$

$$\bigvee_{i \in V \setminus \{s,e\}} (td_i = \max(ta_i, o_i) + \tau_i) \tag{6}$$

$$ta_e - t_0 \leq T_{\max} \tag{7}$$

$$\bigvee_{i \in V \setminus \{s,e\}} (ta_i \leq c_i) \tag{8}$$

Equation 2 guarantees that the solution starts at vertex *s* and ends at vertex *e* while formula 3 means that each vertex can be visited only once. Formula 4 forces travel to start at time t_0 while formulas 5 and 6 defines a relation between arrival and departure times at subsequent vertices (based on travel times, visit times and time-windows). Assuming that travel times between different vertices are positive, formulas 4–6 guarantee that there are no sub-cycles in the path. Constraints 7 and 8 are associated with maximum travel time and time-windows.

3. LITERATURE REVIEW

Problems from the Orienteering Problem family are NP-hard (Golden, Levy, Vohra, 1987) and exact algorithms can be very time-consuming for larger graphs. For this reason most papers are devoted to metaheuristics. Various approaches for the OP were based i.a. on greedy and randomized construction of solutions (Campos, Marti, Sanchez-Oro & Duarte, 2014), local search methods (Chao, Golden & Wasil, 1996; Vensteenwegen, Souffriau, Vanden Berghe & Oudheusden, 2009), tabu search (Gendreau, Laporte & Semet, 1998), ant-colony optimization (Schilde, Doerner, Hartl & Kiechle, 2009) and genetic algorithms (Tasgetiren, 2001).

Most papers about Time-Dependent versions of the Orienteering Problem were published in recent years and emphasize practical aspects of the problem, especially tourist trip planning in transport networks. Garcia et al. (2013) presented the first paper describing its application in POI and public transport network of San Sebastian. To solve the problem the authors proposed Iterated Local Search method (ILS). However, they performed computations on average daily travel times and assumed periodicity of public transport timetables.

Gavalas et al. (2015) proposed an approach which uses real time-dependent travel times in a transport network of Athens. The authors introduced two fast heuristics (TD_CSCR and TDSlCSCR), which based on ILS and vertex clustering, and made comparisons of a few methods.

Verbeeck at al. (2014) developed new benchmark instances for the TDOP, which model street traffic. The authors proposed an ant-colony approach, which achieved high quality results in a short execution time. Gunawan et al. (2014) modified Verbeeck's benchmarks (discretization of time) and compared a few approaches (adaptive ILS proved to be the most effective of them).

The author's previous papers were devoted to metaheuristics for problems from the OP family. Methods developed by the author (composition of evolutionary algorithms and local search heuristics) proved successful on the OP (Ostrowski, Karbowska-Chilinska, Koszelew & Zabielski, 2017; Ostrowski, 2015) as well as TDOP benchmark instances (Ostrowski, 2017). The algorithms achieved results close to optimal and outperformed other methods: GRASP (Campos et al, 2014), GLS (Vensteenwegen et al., 2009), ACS (Verbeeck at al 2014) and Adaptive ILS (Gunawan et al, 2014). The purpose of this work was to adapt the TDOP algorithm to the TDOPTW, apply it for tourist trip planning in real public transport and POI network and verify quality of its solutions.

4. PUBLIC TRANSPORT AND POI NETWORK AS A TIME-DEPENDENT GRAPH

It's assumed that a tourist uses means of public transport (buses in the city of Białystok) when travelling between attractions (POIs). Travel time between POIs depends on bus timetables. For this reason a network of POIs connected by public transportation is a time-dependent graph. Here are assumptions made by the author:

1. A travel between two POIs can consist of two kinds of edges: walk links and bus connections.
2. Walk links have a limited length (D_{max}) and walk times are determined by assuming that walking speed is 3 km/h. A tourist can walk directly between POIs, between POIs and bus stops and between bus stops (during bus transfers).
3. During a travel between a pair of POIs a tourist can use up to k bus transfers (k+1 bus connections).
4. To compensate for deviations of bus arrival times from timetables a minimum waiting time at a bus stop (3 minutes) was introduced.
5. Each graph weight $w_{ij}(t)$ is the duration of the shortest travel from POI i to POI j starting at time t. To compute weights it's necessary to execute shortest path algorithm in multimodal time-dependent graph.

6. Time is discrete (resolution of 1 minute), which is consistent with time-tables. For this reason there are $1440n^2$ graph weights (*n* is number of POIs and there are 1440 minutes in a day).

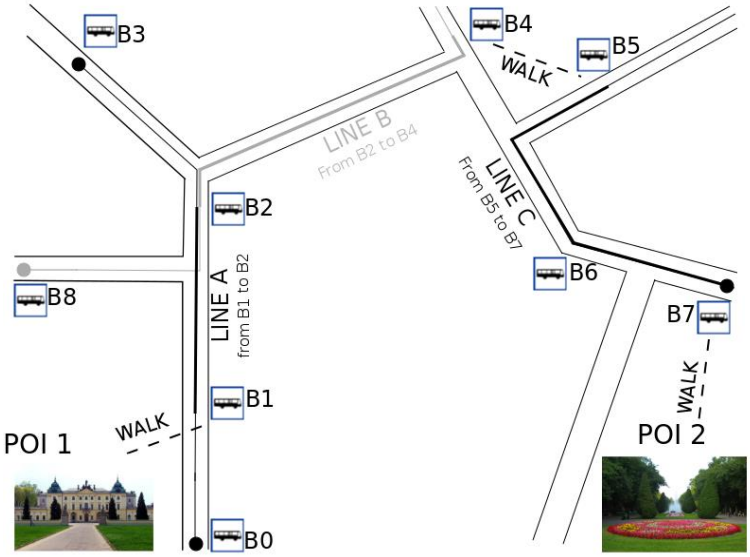


Fig. 1. An exemplary travel between two POIs

In fig. 1. there is an exemplary travel between two POIs consisting of 3 bus connections (2 transfers) and 3 walk links. A tourist leaves POI 1, walks to bus stop B1 and gets on a bus (line A). He gets off at bus stop B2 and waits there for another bus (line B). Afterwards he goes by bus B to bus stop B4. From there he walks to a nearby bus stop B5, where he gets onto another bus (line C) and travels to bus stop B7. From there he walks to his destination (POI 2). Computation of such shortest paths is necessary to get time-dependent weights, which will be used by the metaheuristic solving the Time-Dependent Orienteering Problem.

5. METHOD DESCRIPTION

To realize tourist trip planning in a public transport network of Białystok two tasks should be done:

1. Computation of time-dependent weights based on bus timetables and POIs location.
2. Execution of the TDOP algorithm, which operates on time-dependent weights and generates attractive tours.

5.1. Precomputation of weights

During its execution the TDOP metaheuristic refers to graph weights millions of times. Computing shortest path in a time-dependent network so many times can cause an additional time overhead. For this reason the author decided to precompute and save all $1440n^2$ weights. Precomputed weights are stored in a 3-dimensional array. Thanks to the precomputation step the TDOP algorithm has access to all weights in constant time.

Shortest travels from a given POI at a given start time to all other POIs

In order to compute all weights efficiently the author decided to use modified Ford-Bellman algorithm and optimize some precomputation steps. The basic Ford-Bellman procedure computes shortest travels starting at time t_0 from a given start POI s to all other POIs and bus stops. The algorithm has k+1 main iterations (k – number of bus transfers). It enables to efficiently compute shortest paths consisted of a limited number of bus connections. During the first iteration bus connections starting at bus stops not farther than max. walk distance (D_{max}) from s are considered. Only earliest possible buses of given lines are considered. Afterwards all possible bus stops, where a tourist can get off the bus (exit bus stops), are analysed. From there destination POIs (within walking distance from exit bus stops) are checked. In this way all shortest travel times (consisting of one transport connection) are computed. Analogically the second iteration computes all shortest travels consisted of at most two transport connections (one bus transfer): when analysing an exit bus stop the algorithm searches for other bus connections leaving from the current bus stop and from other neighbouring bus stops (transfer step). The same steps are performed for subsequent iterations. For optimization purposes the algorithm only analyses those bus stops, for which travel time improved in the previous iteration.

Shortest travels from a given POI for all start times

The purpose of precomputation is to compute shortest travels for all 1440 start times (minutes) in a day. Instead of executing the same shortest path algorithm (described above) 1440 times an optimization can be done. It arises from a simple observation (known as FIFO property): $t + w_{ij}(t) \leq (t+1) + w_{ij}(t+1)$. The formula means that earlier travel start implies not later travel finish – in the most pessimistic case travel starting at time t will use the same bus connections as travel starting at time t+1 (the only difference is one more minute of waiting at the first bus stop). Thanks to this property shortest travels for consecutive start times can be computed much faster. When computing shortest travels starting at time t the algorithm uses shortest travel times previously computed for start time t+1. The algorithm considers only those bus connections, which were impossible

to catch one minute later. This significantly reduces precomputation time, especially when frequency of bus connections is less than a minute (which is common in public transport networks – see fig. 2). In the below figure there is an example of time-dependent arrival times: the function is nondecreasing (and constant in intervals). Inside these intervals an equality $t + w_{ij}(t) = (t+1) + w_{ij}(t+1)$ holds, which usually means that the same set of connections is used and computation time can be reduced. What is more, the FIFO property in time-dependent networks implies existence of polynomial time shortest-paths algorithms (Dean, 2004), which made it possible to develop fast precomputation procedure.

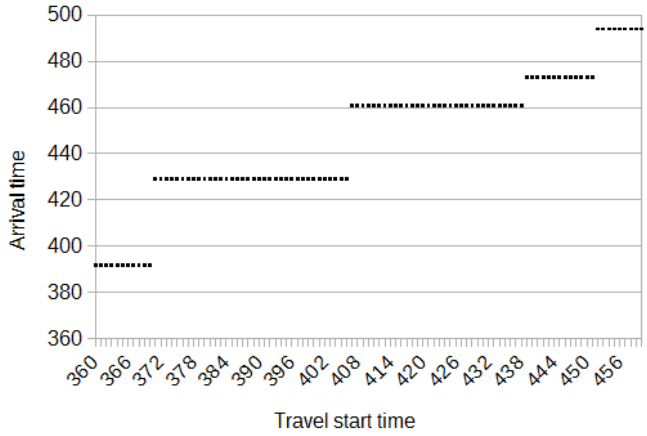


Fig. 2. Arrival time at POI *j* as a function of departure time from POI *i*

Shortest travels between all POIs for all start times

To compute all possible weights for all 1440 start times the above procedures are executed for all starting POIs. Thanks to the described optimization precomputation time was significantly reduced.

5.2. TDOPTW metaheuristic for tourist trip planning

To solve the trip planning problem the author used an evolutionary algorithm, which is based on the method solving the TDOP (Ostrowski, 2017). The author adapted the method to the presence of time-windows. It uses both random and local search operators, 2-point heuristic crossover, disturb operator and deterministic crowding as selection mechanism. What is more, infeasible solutions (too long paths) are present in the population (penalized by the fitness function). A path representation is used – subsequent genes in a chromosome correspond to successive vertices in a path. After a random initialization the algorithm continues computations until a given generations limit (N_g) is achieved or there was no solution improvement in the last C_g generations.

Evaluation

Fitness of a feasible solution s is equal to its profit. Otherwise it is described by the formula: $fitness(s) = p(s) \cdot [T_{max}/t(s)]^k$, where $p(s)$ and $t(s)$ are profit and travel time of solution s. Parameter k (penalty severity) is initially equal to 1 but it is adaptive and increases if more than α percent of paths in the population are infeasible.

Crossover

Parent selection is random. Crossover probability determines the fraction of population chosen for reproduction (selected individuals arranged in random pairs). The algorithm uses specialized 2-point heuristic crossover. Crossover procedure exchanges one pair of path fragments between consecutive common points of both parents. In fig. 3. an example of crossover is illustrated. There are three possible crossings (varying in exchanged fragments). Heuristic crossover chooses the option which maximizes fitness of the better child.

Selection

After crossover children compete with their own parents for places in the population – survivor selection in the form of deterministic crowding (Mahfoud, 1992). Distance metric used bases on the length of longest common subsequence of two solutions. This form of selection preserve population diversity for longer, which allows a more effective search of the solution space (Ostrowski, 2015).

Mutation

Mutation probability determines the fraction of individuals which are selected (randomly) for mutation. Initially, selected paths undergo 2-opt procedure, which tries to reverse a path fragment in order to reduce total travel time as much as possible. Afterwards a vertex insertion or vertex deletion is carried out (each with a probability of 0.5). Both *insert* and *delete* operators have two versions: local search and random. Local search *insert* from all options of inserting a new vertex chooses the one that maximizes profit to travel time increase ratio. Analogically *delete* searches for a vertex which minimizes profit to travel time reduction ratio. Random versions choose vertices arbitrary but insertion place is chosen in order to minimize travel time increase. Probability of local search during mutation is determined by a parameter (heuristic coefficient).

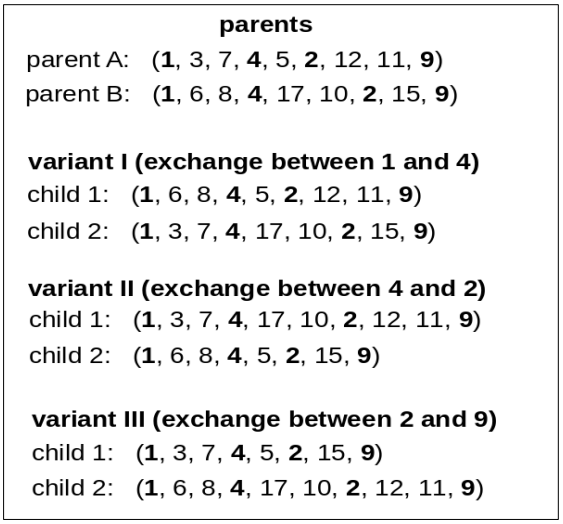


Fig. 3. An example of crossover

Disturb

Disturb procedure is another form of mutation, which applies bigger changes in individuals but is executed rarely. A small fraction of population (determined by disturb probability) is randomly chosen and a path fragment (no longer than 10 percent of vertices) is removed from each of them. Path fragment is chosen randomly or in a heuristic way (to minimize profit to travel time reduction ratio).

Time-Windows

All operators used in the algorithm were modified in order to take into account time-windows. Given a POI with a time-window $<o, c>$ and arrival time t, arriving too early ($t<o$) means that additional waiting time ($o-t$) was added to the tour duration. Arriving too late ($t>c$) made it impossible to visit a vertex and such cases weren't allowed by the algorithm operators.

6. EXPERIMENTAL RESULTS

Experiments were conducted on a computer with Intel Core i7 3.5 GHz processor and the algorithms were implemented in C++. First part of experiments was devoted to precomputation of travel times between POIs in public transport network of Białystok and in the second part the TDOPTW metaheuristic was applied for tourist trip planning in this network (using precomputed weights).

6.1. Precomputation and network properties

Public transport and POI network of Białystok consists of 74 POIs (museums, palaces, churches etc), 37 bus lines and 693 bus stops. Thanks to optimizations (described in the previous section) precomputation time was only a few seconds. Shortest travel times between all pairs of POIs for all start times (1440 minutes in a day) were stored in a 3-dimensional array (occupying 18 MB of RAM). To find out about interesting features of the network, precomputation was executed many times for different values of parameters: maximal walk distance (D_{max}) and maximal number of bus transfers (k). Tests were conducted for two D_{max} values (0.3 and 0.6 km) and four k values (0, 1, 2, 3).

In fig. 4. a dependency between percentage of connected POIs and pre-computation parameters is illustrated. It can be seen that large majority of POI pairs are connected when travel consists of at most one bus transfer (two transport connections) and there is no connectivity improvement for more than 2 bus transfers. Connection percentage improves for larger value of D_{max}. Longer walk links enable to reach larger number of bus stops, which naturally implies more connection options. One can see that for shorter walk links connectivity is always less than 100 percent (regardless of number of bus transfers). This is due to the fact that a few POIs were farther than 0.3 km from nearest bus stop. In fig. 5. it can be seen that most of shortest travels between POIs are very simple (no bus transfer or one transfer) and almost no paths consist of more than 2 transfers.

This is due to the fact that Białystok is a relatively compact city. Larger D_{max} value influences paths simplicity for the same reason as it influenced connections percentage.

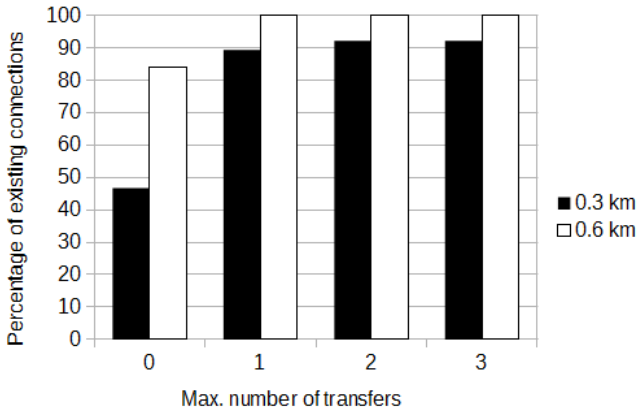


Fig. 4. Percentage of connected POI pairs depending on D_{max} and k

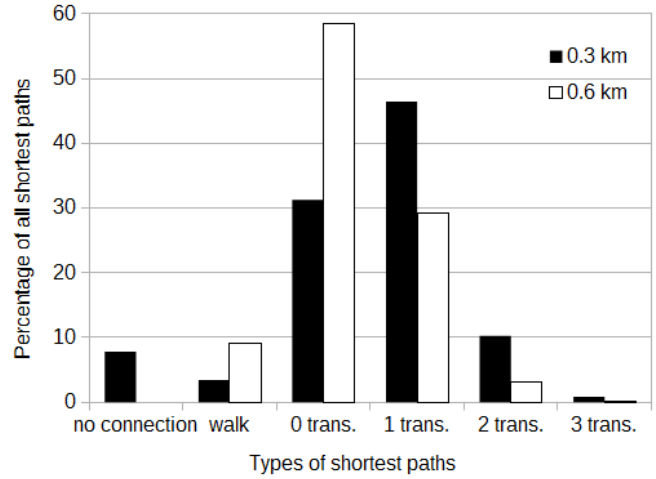


Fig. 5. Percentage of different types of shortest paths depending on D_{max}

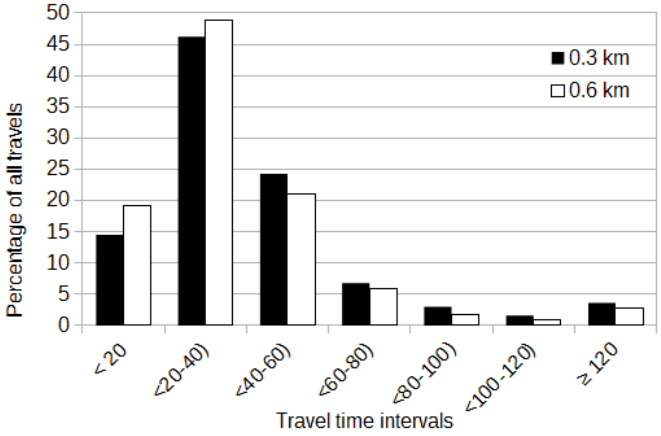


Fig. 6. Histogram of travel times between POIs for different D_{max} values (computed for daily hours 6:00–18:00).

In fig. 6. it can be seen that travel times of 20–40 minutes are most common (46–49 percent) and the majority of travels last less than an hour. In addition, there are more short travels and less long travels when increasing D_{max}. It is associated with the fact that for larger D_{max} value travel times are generally shorter (more bus connections are considered when searching for paths) – average daily travel time is 45 and 41 minutes (for D_{max} 0.3 and 0.6 accordingly).

6.2. Trip planning in Białystok

The tested network included 74 POIs and 2 start/end points. For each attraction a profit, a visit time and a time-window (opening hours) were assigned (link to network: http://p.wi.pb.edu.pl/krzysztof-ostrowski/node/1252). Trips were 3, 6 and 9 hours long (T_{max}) and started at 6:00, 9:00, 12:00 and 15:00 (t_0). Time unit used in test files as well as during computations was a minute i.e. 9:00 is 540 and 4 hours are 240 minutes. There were two variants of trips: starting and ending in the city centre ($s = e = 1$) and in the western part of the city ($s = e = 2$). The author tested two methods: the TDOPTW metaheuristic (evolutionary algorithm with local search heuristics) as well as an exact algorithm (composition of branch-and-bound and dynamic programming developed by the author). Thanks to the exact algorithm optimal solutions are known (up to a few hours of computation time for longest trips) and it's possible to access the quality of paths generated by the metaheuristic.

Algorithm parameter values were derived from EVO100 in the author's previous TDOP article (Ostrowski, 2017) with a small change: $m_h = 0.8$ and $c_h = 1$ instead of $m_h = 1$ and $c_h = 0.8$ (minor error in the article). For each test case the evolutionary algorithm was executed 30 times and average result was calculated. Gaps are given in percent and illustrate relative quality loss to optimal solutions. Execution times are given in seconds. The author's metaheuristic is marked as EVO while OPT indicate profits of optimal tours (expressed as the sum of attractiveness of visited POIs).

In tab. 1. results of trip planning are given (trips start and end in the city centre). One can see that EVO achieves optimal results in most cases and average gap is only 0.02 and 0.19 percent (for D_{max} = 0.3 and 0.6 km). High-quality results are achieved in short execution times (0.3–1.4 s). It can be seen that optimal trips are a few percent better for larger value of D_{max}. This is due to shorter travel times when using longer walk links (as described in the previous subsection), which enables to visit more POIs within a given time frame.

Tab.1. Trip planning results for $s = e = 1$

D_{max} = 0.3 km					D_{max} = 0.6 km				
		EVO					EVO		
T_{max}	t_0	Gap	Time	OPT	T_{max}	t_0	Gap	Time	OPT
3h	**6:00**	0.0	0.3	327	**3 h**	**6:00**	0.0	0.5	342
	9:00	0.0	0.4	475		**9:00**	0.0	0.5	475
	12:00	0.0	0.4	475		**12:00**	0.0	0.5	512
	15:00	0.0	0.5	485		**15:00**	1.6	0.5	506
6 h	**6:00**	0.0	0.9	741	**6 h**	**6:00**	0.1	0.8	769
	9:00	0.0	0.8	813		**9:00**	0.0	0.8	878
	12:00	0.1	0.9	850		**12:00**	0.0	0.7	900
9 h	**6:00**	0.1	1.4	1089	**9 h**	**6:00**	0.0	1.2	1144
	9:00	0.0	1.1	1178		**9:00**	0.0	1.2	1218
Average		**0.02**	**0.75**	–	**Average**		**0.19**	**0.75**	–

In tab. 2. analogical results are presented for trips starting and ending in the western part of the city ($s = e = 2$). Optimal solutions were obtained by EVO for all but 2 test cases in short execution times. Trips quality is lower than it the previous table because most attractions are located in the city centre and additional time is needed to get there.

In fig. 7. a trip generated by the algorithm is presented. It is short and all visited POIs are in the city centre so only one bus connection is needed. It worth noting that the algorithm usually chooses consecutive POIs which are close to each other in order to use the time budget effectively. For this reason travels between POIs found in solutions are usually short (only 5–15 minutes, compared to average of 40–45 minutes) and simple (a walk link or a single bus connection).

Tab. 2. Trip planning results for $s = e = 2$

***Dmax* = 0.3 km**					***Dmax* = 0.6 km**				
		EVO					**EVO**		
T_{max}	t_0	**Gap**	**Time**	**OPT**	T_{max}	t_0	**Gap**	**Time**	**OPT**
3 h	**6:00**	0.0	0.3	226	**3 h**	**6:00**	0.0	0.4	226
	9:00	0.0	0.4	338		**9:00**	0.0	0.4	350
	12:00	0.0	0.4	338		**12:00**	0.0	0.4	338
	15:00	0.0	0.4	358		**15:00**	0.0	0.4	375
6 h	**6:00**	0.0	0.7	659	**6 h**	**6:00**	0.0	0.6	677
	9:00	0.0	0.7	708		**9:00**	0.5	0.7	721
	12:00	0.0	0.8	786		**12:00**	0.0	0.7	805
9 h	**6:00**	0.0	1.1	996	**9 h**	**6:00**	0.1	1.2	1024
	9:00	0.0	1.1	1125		**9:00**	0.0	1.0	1162
Average		**0.0**	**0.65**	–	**Average**		**0.07**	**0.65**	–

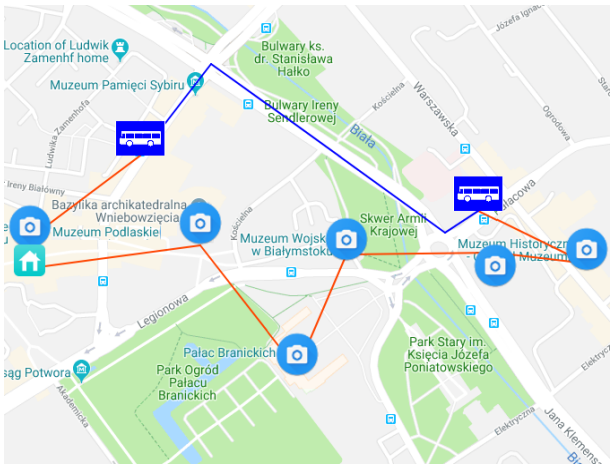


Fig. 7. A tour generated by the algorithm (Google Maps): walk links in red, bus connections in blue; icons: camera (POI), bus (bus stop), house (start/end), T_{max} = 3h, t_0 = 9:00.

6.3. Trip planning in the city of Athens

Additional tests were performed on public transport and POI network in Athens (tests created by Gavalas et al, 2015). The authors proposed 2000 different test cases (varying in topology and tourist preferences) and the presented results are average of all 2000 algorithm runs. The evolutionary algorithm was compared with the following heuristics:

1. Time-dependent heuristics: TD_CSCR, TD_SlCSCR (Gavalas et al, 2015) and their version working on average travel times (AvgCSCR).
2. ILS algorithm working on average travel times (AvgILS, Garcia et al, 2013) and its time-dependent version (TD_ILS).
3. Exact algorithm implemented by the author (OPT).

Compared methods were very fast (execution times of less than 0.1 s) and in order to achieve similar execution times the author tested another version of the evolutionary algorithm with reduced population size (EVO30). Population size and generation parameters were scaled (P_{size} = 30, N_g = 1500, C_g = 150).

In tab. 3. experimental results are presented. It can be seen that the proposed evolutionary algorithm (in both versions) achieves results very close to optimal. They are 0.6–2.0 percent better than the best of other metaheuristics (TD_SlCSCR and TD_CSCR). Gaps to other methods are much bigger (3.5–13.5 percent). It confirms effectiveness of the proposed method.

Tab. 3. Trip planning results for the city of Athens (T_{max} = 5h, t_0 = 10:00)

Method	Score	Gap	Method	Score	Gap
EVO100	344.57	0.02	**Avg_CSCR**	332.01	3.7
EVO30	344.34	0.1	**TD_ILS**	326.28	5.3
TD_SlCSCR	342.06	0.7	**Avg_ILS**	298.53	13.4
TD_CSCR	337.78	2.0	**OPT**	344.6	–

7. CONCLUSION

In this paper a metaheuristic solving the Time-Dependent Orienteering Problem with Time Windows (TDOPTW) was presented and applied to tourist trip planning in public transport networks. The algorithm was tested on public transport and POI networks of Białystok and Athens and in all cases obtained optimal or close to optimal solutions (tours) in short execution times. The composition of evolutionary algorithm and local search heuristics confirmed to be effective for the problems from the Orienteering Problem family (high-quality results were previously obtained by the author for the OP and TDOP benchmarks). Further research will concentrate on adaptation of the proposed method to the Time-Dependent Team Orienteering Problem with Time Windows (TDTOPTW). In this version of the problem *m* paths are generated (instead of one) and effective TDTOPTW solutions can be applied to planning multi-day tours.

REFERENCES

Campos, V., Marti, R., Sanchez-Oro, J., & Duarte, A. (2014). Grasp with Path Relinking for the Orienteering Problem. *Journal of the Oper. Res. Society*, *65*(12), 1800–1813. doi:10.1057/jors.2013.156

Chao, I., Golden, B., & Wasil, E. (1996). Theory and methodology - a fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, *88*(3), 475–489. doi:10.1016/0377-2217(95)00035-6

Dean, B.C. (2004). *Shortest paths in FIFO time-dependent networks: theory and algorithms.* Technical report, MIT Department of Computer Science.

Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W., & Linaza, M. T. (2013). Integrating public transportation in personalised electronic tourist guides. *Computers and Operations Research*, *40*(3), 758–774. doi:10.1016/j.cor.2011.03.020

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., & Vathis, N. (2015). Heuristics for the time dependent team orienteering problem: Application to tourist route planning. *Computers and Operation Research*, *62*, 36-50. doi:10.1016/j.cor.2015.03.016

Gendreau, M., Laporte, G., & Semet, F. (1998). A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, *106*(2–3), 539–545. doi:10.1016/S0377-2217(97)00289-0

Golden, B., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics*, *34*, 307-318. doi:10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0. CO;2-D

Gunawan, A., Yuan, Z., & Lau, H. C. (2014). A Mathematical Model and Metaheuristics for Time Dependent Orienteering Problem. In *PATAT 2014: Proceedings of the 10th International Conference of the Practice and Theory of Automated Timetabling, 26–29 August 2014* (pp. 202–217). Research Collection School Of Information Systems.

Mahfoud, S. W. (1992). Crowding and preselection revisited. In *Proceedings of the 2nd International Conference on Parallel Problem Solving from Nature (PPSN II), Brussels, Belgium, 1992* (pp. 27–36). Amsterdam: Elsevier.

Ostrowski, K. (2015). Parameters Tuning of Evolutionary Algorithm for the Orienteering Problem. *Advances in Computer Science Research*, *12*, 53–78.

Ostrowski, K., Karbowska-Chilinska, J., Koszelew, J., & Zabielski, P. (2017). Evolution-inspired local improvement algorithm solving orienteering problem. *Annals of Operations Research*, *253*(1), 519-543. doi:10.1007/s10479-016-2278-1

Ostrowski, K. (2017). Evolutionary Algorithm for the Time-Dependent Orienteering Problem. In K. Saeed, W. Homenda, & R. Chaki (Eds.), *Computer Information Systems and Industrial Management. CISIM 2017, Lecture Notes in Computer Science* (10244, pp. 50–62). Cham: Springer. doi:10.1007/978-3-319-59105-6_5

Schilde, M., Doerner, K., Hartl, R., & Kiechle, G. (2009). Metaheuristics for the biobjective orienteering problem. *Swarm Intelligence*, *3*(3), 179–201. doi:10.1007/s11721-009-0029-5

Tasgetiren, M. (2001). A genetic algorithm with an adaptive penalty function for the orienteering problem. *Journal of Economic and Social Research*, *4*(2), 1–26.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., & Oudheusden, D.V. (2009). A guided local search metaheuristic for the team orienteering problem. *European Journal of the Operational Research*, *196*(1), 118–127. doi:10.1016/j.ejor.2008.02.037

Verbeeck, C., Sörensen, K., Aghezzaf, E.H., & Vansteenwegen, P. (2013). A fast solution method for the time-dependent orienteering problem. *European Journal of Operational Research*, *236*(2), 419–432. doi:10.1016/j.ejor.2013.11.038

Applied Computer Science, vol. 14, no. 2, pp. 20–37
doi:10.23743/acs-2018-10

Submitted: 2018-03-22
Revised: 2018-06-01
Accepted: 2018-06-21

graphical representations, Petri nets, Control Flow Graph, Concurrent Process Systems, multithreaded applications

Damian GIEBAS[*]*, Rafał WOJSZCZYK*[*]

GRAPHICAL REPRESENTATIONS OF MULTITHREADED APPLICATIONS

Abstract

This article contains a brief description of existing graphical methods for presenting multithreaded applications, i.e. Control Flow Graph and Petri nets. These methods will be discussed, and then a way to represent multithreaded applications using the concurrent process system model will be presented. All these methods will be used to present the idea of a multithreaded application that includes the race condition phenomenon. In the summary, all three methods will be compared and subjected to the evaluation, which will depend on whether the given representation will allow to find the mentioned phenomenon.

1. INTRODUCTION

Applications written for modern computers are characterized by diversity and are used in almost every area of life. Many of these applications are single-threaded programs that perform tasks one by one. Along with the development of computer hardware and the introduction of processors enabling concurrent performance of tasks, multithreaded applications began to appear. Some programming languages, as in C and C++, were not created for multithreading. To complement these gaps for C language, a pthreads library was created in line with the constantly evolving standard (ISO/IEC, 2003). The C++ language has received support for multithreading in the form of an extension of the standard library, with the introduction of the C++ 11 standard (Hinnant, Dawes, Crowl, Garland & Williams, 2007).

[*] Faculty of Electronics and Computer Science, Koszalin University of Technology, 75-453 Koszalin, Śniadeckich 2, Poland, +48 94 34 78 706, damian.giebas@gmail.com; +48 94 3478 710, rafal.wojszczyk@tu.koszalin.pl

As there is a multitude of software created in these languages on the market and a lot of such software is still created, the mentioned languages have been selected to present examples of multithreaded programs.

Compared to previously used single-threaded programming, multithreaded programming has a number of advantages and a number of disadvantages (Torp, 2001). The most important of them are presented below:

Advantages

- Responsiveness – in the case of long tasks in programs with a graphical user interface, single-threaded programs undergo the so-called freezing. This problem does not occur in multi-threaded applications, as such tasks can be delegated to separate threads.
- Resource sharing – threads that run as part of a single process share computer resources. Everything happens within one address space. In the case of single threaded programs, tasks had to be delegated to separate processes and communication was done by copying values from one address space to another.
- Savings – multithreaded programs consume less memory than solutions that use several single-threaded applications.
- Scalability – multithreaded applications make much better use of the hardware capabilities of multithreading processors than a set of single-threaded applications that perform the same task together. At the same time, machines for multithreaded application states are much less complicated than machines of an analogous state of solution composed of single-threaded applications.

Disadvantages

- Complex application code – each application start-up may look different and depends on the current state of other system components. The programmer never knows how much time the scheduler has allocated to a given thread and does not know the order of their work. This state of affairs also affects:
- Debugging of such applications is very difficult because the debugging process itself can affect the behavior of the application.
- Testing the application is very difficult, because it is extremely hard to predict all possible states in which applications will be found.
- Deadlock – this phenomenon is also called jamming or blocking. A situation in which a process or thread in case of multithreaded applications orders access to resources and goes into a waiting state. It is possible that the pending process or thread will never change its state, because the resources it procures are held by other waiting processes (Silberschatz, Galvin & Gagne, 2005).

Deadlock and race condition were known before, as they occur not only in multithreaded applications but also in solutions in which single-threaded applications use shared resources.

Other known phenomena occurring in multithreaded applications are described in chapter 4 on atomicity violation and order violation (Lu, Park, Seo & Zhou, 2016), and the last one is not analyzed in this work. This work focuses on graphical representation of multithreaded applications, which will allow, above all, to reveal places where race condition is present. Control Flow Graph discussed in chapter 1 is the most well-known graphical representation that allowed to develop methods and build tools for error detection of multithreaded applications. Petri nets discussed in chapter 2 are another popular graphic representation. The graphic representations used today have a number of limitations, which affect the developed methods and tools that use them. These tools include:

- Helgrind – a tool from the Valgrind's Tool Suite to securely debug multithreaded programs that can detect any kind of problems related to parallel access to resources. On the creators' site there is information that they do not guarantee the correct operation of the application. Despite all the advantages, Helgrind does not have the possibility of remote debugging, which is necessary to work in a very large number of environments where C and C++ languages are used, e.g. in embedded systems. For more about Valgrind's Tool Suite go to http://valgrind.org/info/tools.html#others
- ThreadSanitizer – Google's tool based on Helgrind and also having its limitations. Both tools use the algorithm described in the Helgrind documentation. ThreadSanitizer is a tool included in the LLVM / Clang and GCC compilers package for the x86 platform. This tool, like Helgrind, is in beta phase and its authors do not guarantee correct operation. For more about ThreadSanitizer go to https://clang.llvm.org/docs/ThreadSanitizer.html
  - RacerX – a tool to detect race condition and deadlock phenomena described in the paper (Engler & Ashcraft, 2003) using static code analysis. Detection is carried out by creating a Control Flow Graf for the analyzed application and enriching it with a list of function calls, global variables used, pointers to variables passed as a parameter, and optionally a list of all local variables. This tool is currently not publicly available to anyone according to https://goo.gl/DgYzt5
  - Relay – a tool created at the University of California, San Diego for static code analysis to detect race condition. This tool worked on a similar principle to RacerX as described in the work (Voung, Jhala & Lerner, 2007). This tool was used to analyze the Linux kernel code version 2.6.15. The analysis was carried out on the number of 4.5 million of the code line and showed the presence of 53 places where the race condition occurred. This is the only such a detailed report on Linux kernel code analysis by static code analysis for this phenomenon. Although the tool is publicly available, it has not been developed since 2010.

The first two tools for error detection use dynamic techniques, i.e. they work with the compiled application code, while the next two work tools require the source code of the application, because they use static techniques by analyzing the source code. The use of Concurrent Processing Systems (CPS) for the detection of race condition and deadlock phenomena is an example of a static approach. The main advantage of the methods that analyze the source code is the fact that they are independent of the platform on which the application code is written, but they are unable to take into account the phenomena caused by aggressive optimization of the compiler (resulting, for example, in changing the variable's contents by a numerical constant). The phenomena caused by aggressive optimization are detected thanks to dynamic techniques, however tools using dynamic techniques are strongly related to the platform, and, for example, all aspects of Helgrind can be used only on x86 and AMD64 platforms.

C and C++ languages have already been expanded to allow for parallel work. The Cilk extension ("A Brief History of Cilk", 2017) for C and C++ was created in 1990 at MIT and commercialized as Cilk ++, and then sold to Intel, which develops them as CilkPlus. This extension has not gained much popularity and will only be used until 2018. Intel proposes migrating from CilkPlus to OpenMP framework or Intel Threading Building Block ("Intel Threading Building Blocks Documentation", 2017).

The mentioned OpenMP framework (Bull, Reid & McDonnell, 2012) was created for Fortran and C languages and then expanded for C++ 98 and is supported by the largest companies in the IT sector. The program is parallelized with OpenMP by using the appropriate preprocessor directives, which increase the complexity of the code and do not cooperate with the latest versions of C++.

Competitive solution for OpenMP, i.e. the Intel TBB library ("Intel Threading Building Blocks Documentation", 2017) for C++ is much better adapted to work with the latest versions of this language. Unfortunately, when using Intel TBB, a lot of code must be rewritten using its elements.

Charm++ ("Introduction to Charm++ Concepts", 2017) is a dedicated C++ language framework for creating applications with parallel processing. It introduces a new paradigm, i.e. object-oriented asynchronous message passing parallel programming paradigm, which decomposes the program into chares that communicate using objects called messages. The disadvantages of this solution were presented in the presentation "Charm++" (Aiken, 2017). The biggest disadvantage concerns easy to omit synchronization of chares work, which is required to avoid race condition. Another big disadvantage of Charm++ is making the programmer manage message memory. Wrong management can lead to very serious memory leaks when resources are allocated and they are not released.

The above solutions for C and C++ languages have one undesirable feature, i.e. a high level of code complexity written with their use. In the case of using the pthread library or the C++ 11 standard, this code is much more readable.

The next part of the work concerns the location of the race condition phenomenon found in the program code shown in Listing No. 1 by means of graphical representation of multithreaded applications. The program was written in C language using the pthreads library. The aim of the program from the listing below is to perform a million incrementing operations of the *balance* variable by each of the application threads. The result of the action should be a two million value. Unfortunately, incrementing operations on a shared resource are not synchronized, resulting in a race condition in the program. Synchronization should be done by using synchronization mechanisms called the mutexes provided with the pthreads library. Mutexes are abstract structures that use the mechanism of mutual exclusion to synchronize work on selected resources. The word mutex is derived from the English words mutual exclusion.

```
#include <stdio.h>
#include <pthread.h>
static volatile int balance = 0;
void* deposit(void *param) {
	// Block B
	char *who = (char*)param;
	int i;
	printf("%s: begin\n", who);
	for (i = 0; i < 1000000; i++)
	// Block C
	{
	 balance = balance + 1; // Place with uncontrolled access. Race condition place.
	}
	// Endblock C
	printf("%s: done\n", who);
	return NULL;
	// Endblock B
}
int main() {
	// Block A
	pthread_t p1, p2;
	char a[] = "A";
	char b[] = "B";
	printf("main() starts depositing, balance = %d\n", balance);
	pthread_create(&p1, NULL, deposit, a);
	pthread_create(&p2, NULL, deposit, b);
	// Endblock A
	// Block D
	pthread_join(p1, NULL);
	pthread_join(p2, NULL);
	printf("main() A and B finished, balance = %d\n", balance);
	return 0;
	// Endblock D
}
```

Listing. 1. Multithreaded application code containing the race condition phenomenon

Although it is possible to write multithreaded programs in C language, the lack of native language support causes that these programs often include race condition or deadlock phenomena. The following application code containing the race condition phenomenon will be transformed into subsequent graphical representations in which this phenomenon should be exposed, because graphic representations do not have C language limitations and are better adapted to presenting high-level ideas.

The use of graphic methods results from the need for a universal tool that will allow for code analysis and detection of the occurrence of the discussed phenomena. As already mentioned, the graphic methods are better for presenting high-level ideas than the C programming language. In addition, converting the source code into a graphical representation is a platform-independent solution to which the code is to be compiled. This transformation of the source code for graphical representation is part of the static code analysis.

2. CONTROL FLOW GRAPH

Control Flow Graph (CFG) is nothing more than a directed graph, which is one of the possible graphical representation of a multi-threaded application. CFG presented in the work (Allen, 1970) consists of nodes and edges that correspond to the next blocks of code and determine the order in which they are executed. CFG assumes the existence of 3 types of nodes. The first type of node is the entry node, which is characterized by the fact that it does not have an ancestor, but it has descendants. The second type of node is the exit node, which, similarly to the entry node, does not have descendants, but it has ancestors. The third type of nodes are nodes having both ancestors and descendants. These nodes can have at least one ancestor and at least one descendant. Ancestors and descendants can be both direct and indirect nodes. In other words, CFG is a directed G graph being a pair (B, E) where B is a set of nodes $b_1, b_2, b_3, \ldots, b_n$ while E is a subset of the set of all possible edges $\{(b_1, b_2), (b_1, b_3), \ldots, (b_m, b_n)\}$ occurring between these nodes.

Figure 1 presents CFG for an application whose code is on Listing 1. The code is divided into 4 logical blocks that allow its easy conversion into CFG. Block A is a fragment of the code preparing the application for working on threads, while block D is a fragment of the code terminating work on threads and ending the work of the application. Blocks B and C are a fragment of the application executed in parallel, therefore to emphasize this aspect of the application on CFG for one thread they have been labeled as B and C and for the other as B' and C'. In addition, block C is contained in block B and its work is repeated a million times.

With the exception of the main function, every logical block, i.e. any other function, the body of a loop, the body of control instructions, or any other block enclosed in braces will have its reflection in the form of a node. The main function in the C and C++ languages is the place of beginning and end of the application work, therefore it has been broken into the above-mentioned blocks A and D.

Figure 1 shows the CFG of the application whose code is on Listing 1. The diagram starts with node A in the main function. It precedes the creation of two application threads, which are represented as nodes B and B'.

Nodes C and C' are the nodes corresponding to the body of the for loop, so until the loop condition is always true, the block will still be executed, as indicated by the presence of edges *(C, C)* and edges *(C', C')*. After completing the loop operation, the control returns to the main body of the function, i.e. to blocks B and B'. The program already has only the D block responsible for the end of work and the corresponding node D finishes the graph.

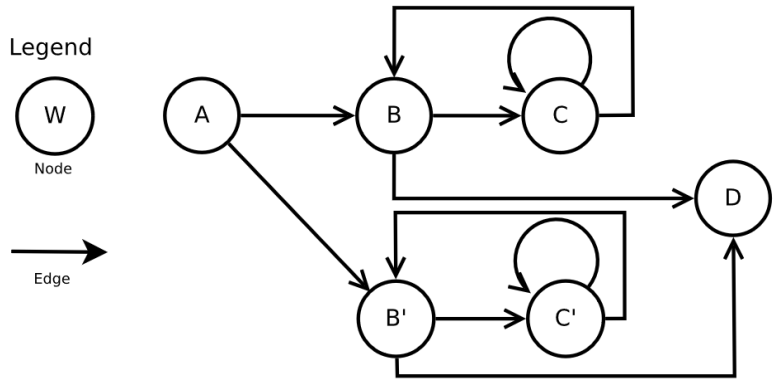


Fig. 1. Control Flow Graph of application from Listing 1

The created CFG reflects exactly the order of the blocks of code executed, however, the information about operations on shared resources cannot be read out of it. These operations occurring in block C of the application do not show their graphical representation, therefore the CFG diagram will be identical for both a properly functioning application and the one in which the race condition phenomenon is located. Hence, CFG is not a good enough notation to detect race condition and deadlock phenomena.

Another disadvantage is the fact that CFG does not allow showing blocks' nesting. Without an exact description, one can get the impression that after the block C exit and returning to block B, it is possible to return to block C again, which is not possible in the application.

Information that a given block is executed in two independent threads allows for the presumption that this is the place where the race condition may occur. In the case of systems where threads do not share any resources, it would also be necessary to check all such places. In a situation where the resource is shared by two threads that do not have common blocks, this mechanism is insufficient to locate the race condition phenomenon.

Control Flow Graf is used in tools to detect these phenomena, e.g. in the RacerX tool, each of the CFG nodes created by this tool is additionally enriched with lists of function calls, global variables used, pointers to variables passed as a parameter, and optionally a list of all local variables. Only when there is a set of all this information it is possible to detect the phenomenon of race condition.

3. PETRI NET

Petri Net (PN) is a formal information flow model designed to describe asynchronous systems in which tasks are carried out in parallel. Petri nets consist of places and transitions connected by directed edges (Peterson, 1977). The flow of information is demonstrated by moving tokens between places by passing through the edges. On the edges there are transitions, which are responsible for the permission to make the transition, and this happens when there are tokens at all the entry points of the transition. The simplest example of PN is shown in Figure 2. There are two places on it, p1 and p2 and one transition t1. The token located in place p1 will be moved along the edges to the place p2, because the transition condition is met, i.e. place p1 is the only entry point of the t1 transition and it has a token.

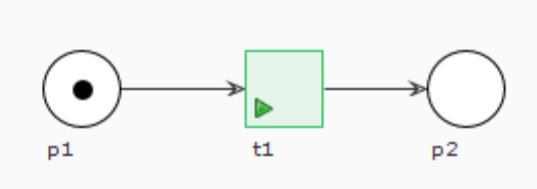


Fig. 2. Example of Petri net

Unlike CFG, PN is not built only on logical blocks of code. The construction should take into account such things as the initial state of some elements, i.e. the place reflecting the loop counter should have as many tokens as the iterations will be made by the loop, or information about the order of execution of individual tasks in case they can be done in parallel. Net from figure 3 showing the application of the mutual exclusion mechanism, which imposes the order of tokens shifting from place p1 through the transitions t1 and t2 is the example of the above.

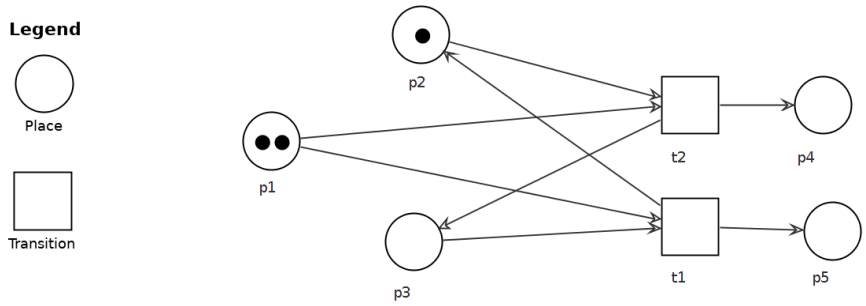


Fig. 3. Example Petri net with the mechanism of mutual exclusion

The possibility of using mutual exclusion mechanisms (it consists of the transitions t1, t2 and places p2, p3 in the above example) allows to control token shifts in the net and simulate application multithreading. However, it is not a realistic representation. In the case of multithreaded applications whose main purpose is the speed of the operation, the programmer does not impose their execution order. It is the scheduler to decide which thread is currently working while the situation of alternating thread work, as shown in the figure above, is unlikely.

In Figure 4, the Petri net is presented for the application under consideration. This net is built from 6 places and 4 transitions. Place p1 corresponds to block A of the selected application and means its start. Place p2 is the equivalent of the moment of starting both threads of the application. In the case of the Petri net, we can simulate the operation of the for loop, so block C in this case will consist of places (p3, p5, p8) and transitions (t2, t3) for the first thread, as well as places (p3, p4, p8) and transitions (t2, t4) for the second thread. Place p3 is a loop count, which should have a million tokens, because so many iterations are executed by each loop in the threads. This will enable every branch of the net to perform a million times as in every thread a million operations are performed on a shared resource.

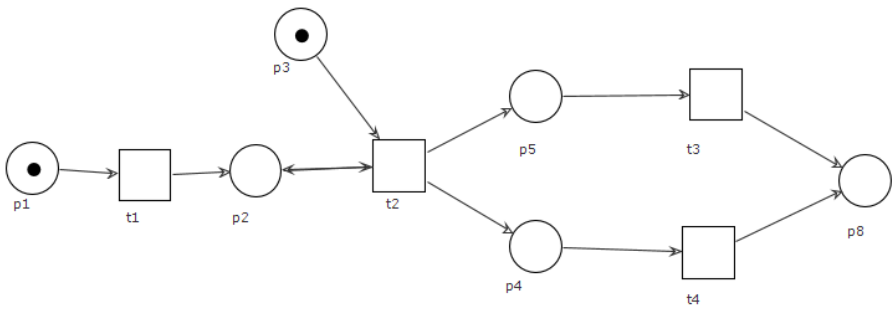


Fig. 4. Petri net of multithreaded application from Listing No. 1

Place p8 is the equivalent of application block D and ends the whole network, and the number of tokens corresponds to the value of the balance variable. If there were one million tokens in place p3, i.e. the maximum number of iterations of the loop in block C, then after the simulation there will be 2 million in the place p8.

The structure of the net does not allow the occurrence of a situation in which the race condition occurs, therefore the result of the net operation will be consistent with the expected result of the application but not with its real operation.

An additional disadvantage of such representation is that many net models can be built into one and the same application code. This situation causes that when a net model for the application is created, one can never be sure that all necessary information can be read from it to locate the information one is looking for.

In the case of using the mutual exclusion mechanisms in the net, it should always be determined which transition will have a priority resulting in a predetermined order of operation of the transition. This is not the case in applications. The programmer is never sure which thread will be given access to the resource first, because the work of the threads is set to execute tasks as quickly as possible and they are executed immediately when the scheduler assigns the processor time to the thread. Unlike nets, the mechanisms of mutual exclusion provided with the C language do not enforce the order of threads.

4. FORMULATING THE PROBLEM

The presented analysis of two widely known graphical representations of multithreaded applications allows to conclude that using them to find the location of the race condition phenomenon is very complex and in many cases requires the use of additional (redundant) control mechanisms.

A multithreaded application code written in C language using the pthreads library is available.

The limitations result from the syntax of the C language, its grammar and the fact that the calculations must be performed in parallel.

Therefore, the question is as follows: is the application code correct, i.e. are there no phenomena like:

- deadlock,
- race condition?

The two previously discussed graphical representations did not allow to determine whether the code on Listing 1 is free from these phenomena. In item 4, a representation using models of concurrent processes for this purpose will be presented. Two representations will be shown, where the phenomenon of race condition will be visible on the first one and the second one will present a solution to eliminate this phenomenon.

5. SYSTEM OF CONCURRENT PROCESSES (Polish: SWP) FOR MULTITHREADED APPLICATIONS

The process system is a set of processes $P=\{P_i \mid i=1, \ldots, ln\}$ performing operations based on a set of shared resources $R=\{R_k \mid k=1, \ldots, lm\}$. Concurrent execution of processes means that each successive operation of one process begins before the end of another process operation and is associated with limited access to shared resources (Banaszak, Majdzik & Wójcik, 2008). A specific case refers to the systems in which processes are carried out cyclically (i.e. process operations are repeated many times over fixed time periods). In this approach, the System of Concurrent Cyclic Processes (Polish: SWPC) is understood as a set of concurrent cyclic processes that are related to each other through the use of shared resources (Bocewicz, Banaszak & Wójcik, 2006; Bocewicz, 2013).

When talking about SWPC one should mention the conflicts of resource demands, which are a consequence of the occurrence of, among others, such phenomena as starvation and blockade. Similar phenomena can be found in multithreaded applications. Starvation occurs when one of the threads of the application over its entire duration does not release the specified resource and thus prevents access to other threads. Deadlock, on the other hand, occurs when two threads (or more) attempt to gain access to the resources they occupy, and so-called resource request cycle occurs. This situation causes that each thread waits for the remaining ones to release their resources, which never happens.

Another specific phenomenon of multithreaded applications refers to the race condition - a situation in which the status of a shared resource (e.g. the value of a variable represented by this resource) is changed by one of the threads when other threads perform operations with an already obsolete resource value. The consequence of such a phenomenon is the possibility of obtaining various results of applications (often difficult to predict) depending on the order of access of threads to shared resources.

Similarly to the CFG and PN models discussed in the previous sections, systems of concurrent cyclic processes can also be used to represent multithreaded applications. A set of graphic elements is used for its purpose (Figure 5), consisting of:

- shared resources representing an instance of any type that is shared among threads, e.g. by means of a pointer or as a global variable,
- internal resources of threads, which like resources are shared by instances of any type, and their period of life lasts as long as the life span of threads,
- cyclical processes representing the threads of the application,
- synchronization mechanism (mutex) ensuring mutual exclusion of processes on resources. In C language, a mutex is an algorithm implemented in the form of an object on which blocking and releasing operations can be performed.

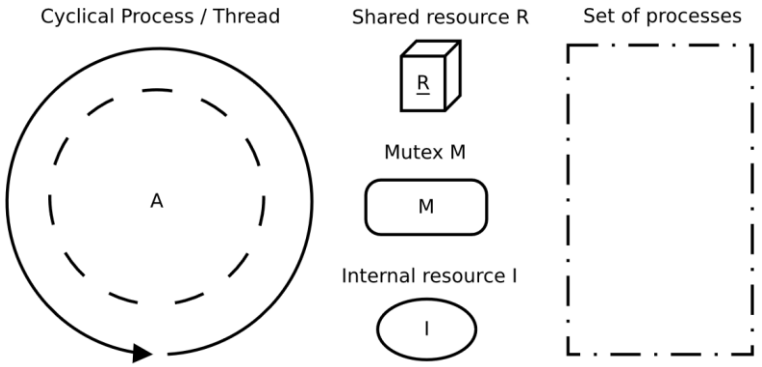


Fig. 5. SWPC elements used for multithreaded application modeling

Both resources and mutexes can be a base for cyclic processes performed (the names of these operations are given inside the resource).
These are, e.g.:

- Inc – resource increment operation,
- Lock – operation of placing a lock on a mutex object,
- Unlock – the operation of releasing the lock from the mutex object.

The proposed SWPC model (using the proposed set of elements), unlike the Petri net and CFG, hides many implementation details. It will highlight only those features of the application that are important to assess its correctness (in terms of the occurrence of phenomena leading to conflicts of resource demands). This approach should allow for accurate reproduction of the application from the model and at the same time indicate the places where race condition or deadlock can occur.

Fig no. 6 presents SWPC for the application from Listing No. 1. It differs significantly from the Petri net and CFG. The system includes a pair of processes (A, B) corresponding to both threads of the application under consideration. The A and B processes are within one set, as both threads work within one application. Both processes perform the operation of increasing the value of a shared resource called balance or increase the value of their internal resources in a similar way to the threads of the sample application. The remaining elements of the application, i.e. displaying information on the standard output, initializing variables or terminating the work of threads are hidden, because they are unnecessary in the process of detecting the phenomenon of race condition.

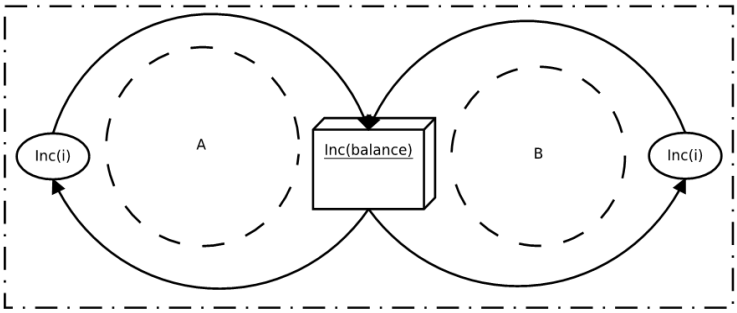


Fig. 6. SWPC model of a multi-threaded application from Listing No. 1

Although very general. the presented SWPC has the necessary information regarding the reconstruction of the application under consideration. The programmer receives a set of information that allows to recreate its code. It is easy to see in the figure that work on a shared resource is not synchronized, i.e. there is no mutex that ensures mutual exclusion of processes on a shared resource. This means that race condition on this resource may occur.

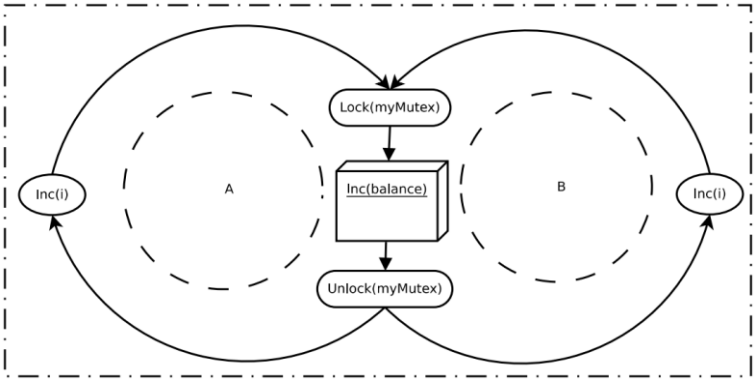


Fig. 7. The SWPC model of Listing 1 application without race condition error

Hiding unnecessary details about threads implemented in the application makes the model very clear. Omitting the implementation details does not affect the assessment of the correctness of the application. In contrast to PN and CFG, the SWPC model enhances the sensitive elements of the application, which translates into a better presentation of how the application works and allows to locate places where potential errors may occur.

Eliminating the error resulting from the occurrence of race condition is possible as a result of adding synchronization elements. Figure 7 shows the SWPC model with mutexes that eliminate race condition. In the presented solution, processes before sharing a shared resource block access to it (Lock (myMutex)) and then release it (Unlock (myMutex)) after completing operations on this resource.

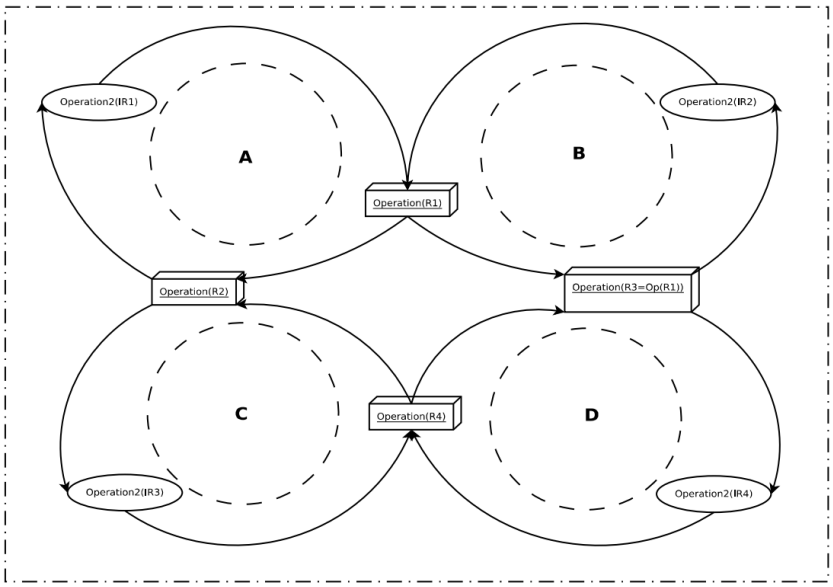


Fig. 8. Sample SWPC model of the exemplary application without synchronization elements

The application from Listing No. 1 is an example for which SWPC is not complicated. Figure 8 includes SWPC model for an exemplary application with four threads. The application has four shared resources R1, R2, R3 and R4, and each thread works with two of them and with its own internal resource. Additionally, in the B-thread, the operation on the R3 resource is dependent on the new R1 resource value (this relationship is expressed by the R3 = Op (R1) equation included in the graphic element). The figure clearly shows that the operations carried out on the shared resources are not synchronized, therefore, a race condition may occur. In addition to race condition, atomic violation is also present in the application. This phenomenon is a consequence of the relationship between R1 and R3. The state of the resource R1 affects the state of the resource R3. Before process B performs an operation on resource R3, the state of resource R1 can be changed by process A.

The elimination of the race condition comes down to placing 4 mutexes: mutex1, mutex2, mutex3, and mutex4 in application in order to ensure mutual exclusion of processes on shared resources – the appropriate SWPC is shown in Figure 9. Before each operation, a lock action on the corresponding mutex is performed on the shared resource, and after its execution, this mutex is released.

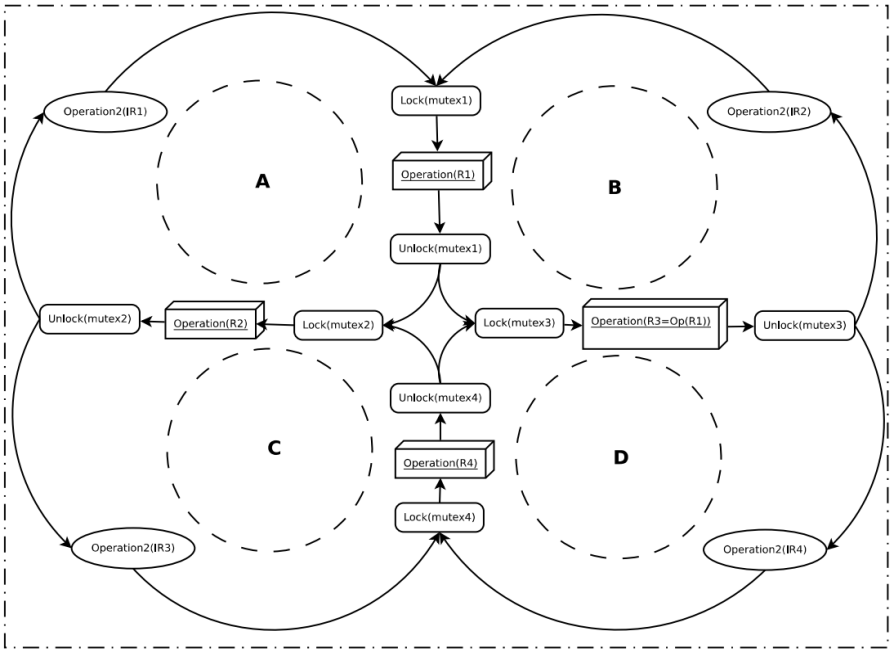


Fig. 9. An example of the SWPC model of an exemplary atomic violation application

Unfortunately, this approach does not eliminate the phenomenon of atomicity violation. This phenomenon is still present because thread B, after releasing mutex1, goes to the mutex3 blocking operation and unsecured by thread B R1 resource can be changed by thread A. One of the acceptable ways to eliminate this phenomenon is to introduce an additional mutex, which in the B-thread will control working on both resources, and in the A-thread it will control only operations on the R1 resource. A model with such a mutex is presented in Figure 10. Eliminating the phenomenon by adding another mutex increases the risk of blocking but there is a better solution, i.e. without adding mutex5.

The solution that will eliminate the phenomenon of atomicity violation without adding mutex5 is shown in Figure 11. The role of the element synchronizing the work of the B thread was received by mutex1 so that the excess mutex could be removed. The moment thread B starts work, it blocks the possibility of working on shared resources for threads A and D until it finishes the work.

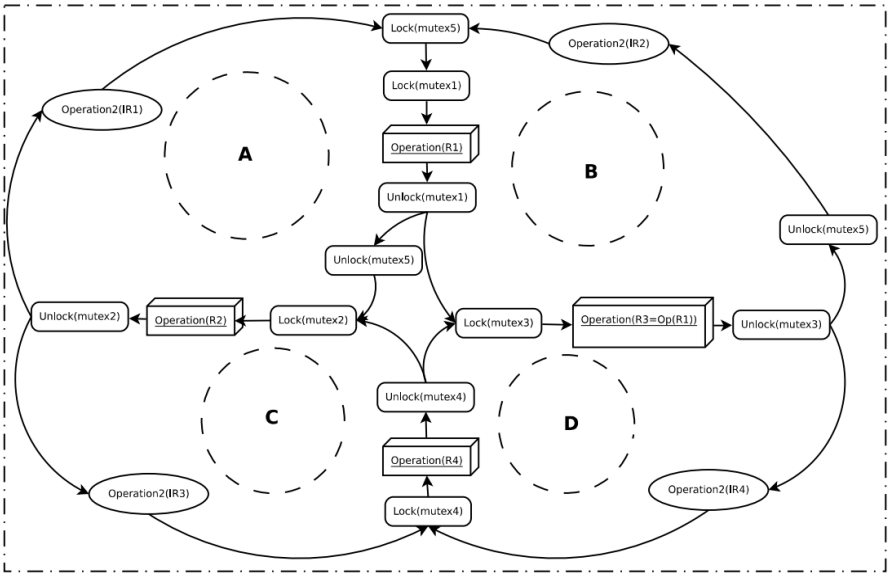


Fig. 10. An example of the SWPC model of an exemplary application with atomic violation over redundant mutex

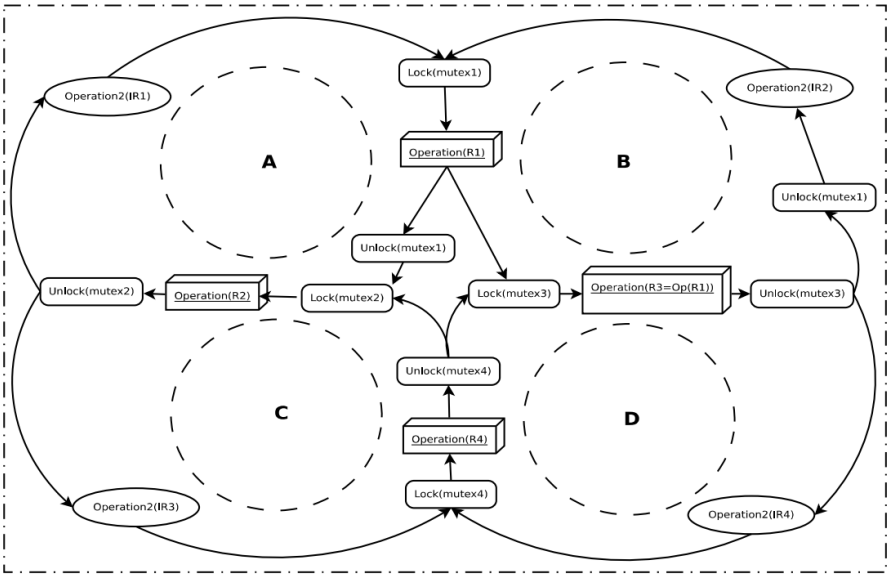


Fig. 11. An example of the SWPC model of an exemplary application with a solution for atomicity violation with a minimum number of mutexes

The models presented for the example application show that thanks to SWP, it is very easy to locate not only the phenomenon of race condition but also the phenomenon of atomicity violation. An additional advantage of SWP lies its readability, which allowed for the optimization consisting in the removal of the excess mutex. This operation will affect the speed of the application, as there are less blocking and unblocking operations, which can be very expensive.

6. CONCLUSION

All three presented representations have their advantages and disadvantages. In terms of multithreaded applications, CFG should be used when the objects of interest include the number of logical blocks and the order in which they are executed. Unfortunately, CFG is a very general graphical representation and is not suitable for analyzing relationships between threads without additional information about individual code blocks that are presented as nodes.

Petri nets are a much more sophisticated tool. They show the mechanism of mutual exclusion and the flow of information. However, the complexity of the net will increase with the complexity of the application, and an attempt to optimize it may hide important details. For each multithreaded application, it is also possible to create many different PNs. Each of the nets can work exactly as assumed by the multi-threaded application concept, while none of them will work as a real application when the application has race condition.

The method using SWP models seems to be a much better solution than the two previous methods. It hides most of the implementation details, highlighting those places where race condition, atomicity violation or deadlock may occur, which like atomicity violation is a phenomenon resulting from incorrect setting of mutexes. Interpretation of the SWP model is much simpler than in the case of PN or CFG and the extension of the notation allowed to locate the place of the race condition error in the example application. In addition, a small change in the SWP model showed how to solve the race condition in the example application or atomic violation in the example application in chapter 4. The method using SWP models is disadvantageous because they were not created for multithreaded applications. For the purposes of this article, it was necessary to extend the standard notation so that it could express all the necessary elements of a multithreaded application.

REFERENCES

A Brief History of Cilk. (n.d.). Retrieved September 16, 2017, from https://www.cilkplus.org/cilk-history

Aiken, A. (October 28, 2017). *Charm++*. Retrieved from https://web.stanford.edu/class/cs315b/lectures/lecture11.pdf

Allen, F. E. (1970). *Control Flow Analisys*. Retrieved July 5, 2017, from http://sumanj.info/secure_sw_devel/p1-allen.pdf

Banaszak, Z., Majdzik, P., & Wójcik, R. (2008). *Procesy współbieżne. Modele efektywności funkcjonowania.* Koszalin: Wydawnictwo Uczelniane Politechniki Koszalińskiej.

Bocewicz, G. (2013). *Modele multimodalnych procesów cyklicznych.* Koszalin: Wydawnictwo Uczelniane Politechniki Koszalińskiej.

Bocewicz, G., Wójcik, R., & Banaszak, Z. (2006). Harmonogramowane pracy wózków samojezdnych w warunkach ograniczonego dostępu do współdzielonych zasobów ESW (Model logiczno-algebraiczny). In *Postępy robotyki: Systemy i współdziałanie robotów*. Warszawa: WKiŁ.

Bull, J. M., Reid, F., & McDonnell, N. (2012). A Microbenchmark Suite for OpenMP Tasks. In: B. M. Chapman, F. Massaioli, M.S. Müller, M. Rorro (Eds), *OpenMP in a Heterogeneous World. IWOMP 2012. Lecture Notes in Computer Science* (271–274). Berlin, Heidelberg: Springer.

Engler, D., & Ashcraft, K. (2003). RacerX: effective, static detection of race conditions and deadlocks. *ACM SIGOPS Operating Systems Review*, *37*(5), 237–252. doi:10.1145/1165389.945468

Hinnant, H. E., Dawes, B., Crowl, L., Garland, J., & Williams, A. (June 24, 2007). *Multi-threading Library for Standard C++*. Retrieved from http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2007/n2320.html

Intel Threading Building Blocks Documentation. (n.d.). Retrieved September 16, 2017, from https://software.intel.com/en-us/tbb-documentation

Introduction to Charm++ Concepts (n.d.). Retrieved September 16, 2017, from http://charmplusplus.org/tutorial/CharmConcepts.html

ISO/IEC. (2003). Information technology - Portable Operating System Interface (POSIX) – Part 1: Base Definitions (9945-1:2003).

Lu, S., Park, S., Seo, E., & Zhou, Y. (2008). Learning from mistakes: a comprehensive study on real world concurrency bug characteristics. In *Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-XIII)* (pp. 329–339). New York: ACM.

Peterson, J. L. (1977). Petrie Nets. *ACM Computing Surveys (CSUR)*, *9*(3), 223-252.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2005). *Operating System Concepts.* USA: John Wiley & Sons

Torp, K. (November 19, 2001). *Multithreading*. Retrieved from http://people.cs.aau.dk/~torp/Teaching/E02/OOP/handouts/multithreading.pdf

Voung, J. W., Jhala, R., & Lerner, S. (2007). RELAY: static race detection on millions of lines of code. In Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering (pp. 205–214). New York: ACM.

Applied Computer Science, vol. 14, no. 2, pp. 38–47
doi:10.23743/acs-2018-11

Submitted: 2018-05-14
Revised: 2018-06-11
Accepted: 2018-06-20

ANN, support vector regressions, Intermittent Demand Forecasting

Gamze Ogcu KAYA[*]*, Ali TURKYILMAZ*[**]

INTERMITTENT DEMAND FORECASTING USING DATA MINING TECHNIQUES

Abstract

Intermittent demand occurs randomly with changing values and a lot of periods having zero demand. Ad hoc intermittent demand forecasting techniques have been developed which take special intermittent demand characteristics into account. Besides traditional techniques and specialized methods, data mining offers a better alternative for intermittent demand forecasting since data mining methods are powerful techniques. This study contributes to the current literature by showing the benefit of using data mining methods for intermittent demand forecasting purpose by comprising mostly used data mining methods.

1. INTRODUCTION

Companies need forecast values regarding different variables related to their business which may be a general variable like population forecast of the country or more specific variables like machine availability, costs, profit, inventory, and lead times. Demand forecasting is a specific type of forecasting which a very significant forecast needed by companies. All departments of the company plan themselves according to demand forecast since demand forecast is the prediction of sales values which is the primary revenue source for the company.

If the demand of item to be forecasted is smooth and continuous, then it is successfully predictable with traditional forecasting methods. On the other hand, when the demand does not occur at every forecasting period and has changing values, then it is hardly predictable with traditional forecasting methods. This kind of items are called intermittent demand items.

[*] Sampoerna University, Department of Industrial Engineering, Jakarta, Indonesia, gamzeogcu@gmail.com
[**] Nazarbayev University, School of Engineering, Astana, Kazakhstan, ali.turkyilmaz@nu.edu.kz

Data mining methods are powerful techniques which can solve complicated practical problems in various areas and are becoming more and more widespread day by day. They can be applied to time series forecasting problems effectively due to their distinguished properties. Data mining methods can handle many difficulties in the modeling of the time series like non-stationary. Moreover applying data mining methods is quite easy since there is no need for any complex mathematical formulations.

There is a lack of effective forecasting methods in intermittent demand forecasting area. This study shows the ability of data mining methods for accurate intermittent demand forecasting. Famous data mining methods which are artificial neural networks, support vector regressions and decision tree techniques are applied to simulated intermittent demand data as well as Croston's method which is the main ad-hoc intermittent demand forecasting method. The study is organized as follows: Section 2 includes the literature review, the following section contains experimental results, conclusions, and final comments are made in Section 4.

2. LITERATURE REVIEW

Intermittent demand is common in spare parts like in the automotive and aircraft industry, military maintenance and high-priced capital goods. Intermittent demand is commonly stated as spare parts demand in the literature (Hua & Zhang, 2006; Syntetos, 2007; Regattieriet, Gamberi, Gamberini & Manzini, 2005, Bacchettive & Saccani, 2012). The reason for this statement comes from the intermittent manner of spare parts items since they are only demanded when a maintenance problem occurs. It is clear that management of spare parts demand is crucial from technical and economic perspectives which make intermittent demand forecasting a very significant need. In case of a poor demand forecast, there might be difficulties in supplying the proper spare part item which leads to production interruptions and also technical difficulties. Kennedy stated that production interruptions result with direct economic losses since customer demand cannot be satisfied due to lost production and quality costs and also indirect economic losses due to customer dissatisfaction. So the aim of inventory planners is to reduce inventory costs while achieving high customer service levels (Kennedy, Patterson & Fredendall, 2002).

Although the availability of spare parts in inventory is very important due to economic and technical handicaps, two properties of spare parts items make it difficult to store high amounts of the items in inventory. The first property of spare parts is that: they aren't of the type which is called "general purpose" which makes them needed only if their specific requirement is realized. So it is not reasonable to store a high amount of spare parts in inventory rather than general purpose items with limited inventory storage capacity. The second property of spare parts items

is their high unit values. Since spare parts have technical characteristics, their acquisition and also storage are costly for companies which store them in low quantities.

Intermittent demand forecasting is not an easy task since there are a lot of time periods with zero demand. Since intermittent demand is not stationary, traditional forecasting methods not applicable for intermittent demand forecasting. Traditional forecasting techniques are not capable of forecasting intermittent demand successfully since they assume stationary which is not the case in items having intermittent demand structure. This situation reveals the need for specialized techniques for intermittent demand forecasting.

Simple exponential smoothing method is the most common traditional method which is applied by companies dealing with intermittent demand. But in the work of Croston, it was first shown that exponential smoothing generally results with inappropriate stock levels (Croston, 1972). Croston's method corrected by Rao is a widely used approach for intermittent demand forecasting and inspires a lot of researchers (Rao, 1973).

In his work, Croston handles intermittent demand data as a combination of two elements which are the non-zero demand size Z_t and the inter-demand intervals P_t. He used two separate forecasts for the size of demand and for the demand arrival rate. Croston's forecasting method is superior to traditional single exponential smoothing forecast when applied to intermittent demand. It is pointed that single exponential smoothing predicts average demand higher than the real average which results in inappropriate stock levels, the reason was intermittent and zero demand occurrences at many time periods.

Nomenclature:

Z_t – demand in period t,

$\hat{Z}_t$ – the estimate of the mean size of a nonzero demand,

α – smoothing parameter, the value is between 0 to1,

$\hat{P}_t$ – the estimate of the mean interval between nonzero demands,

D_t – forecast value at time t.

Croston forecasts $\hat{P}_t$ separately the time between consecutive transaction. P_t and the magnitude of the individual transactions Z_t.

If $Z_t = 0$,

$$\hat{Z}_t = \hat{Z}_{t-1} \text{ and } \hat{P}_t = \hat{P}_{t-1}, \text{ forecast values remain unchanged} \quad (1)$$

If $Z_t \neq 0$,

$$\hat{Z}_t = \alpha Z_t + (1-\alpha)\hat{Z}_{t-1} \quad (2)$$

$$\hat{P}_t = \alpha P_t + (1-\alpha)\hat{P}_{t-1} \quad (3)$$

The forecast of demand per period at time t is found as a division of demand value by inter-arrival time:

$$D_t = \frac{\hat{Z}_t}{\hat{P}t} \tag{4}$$

where: Z_t – demand in period t, $\hat{Z}_t$ – demand forecast in period t for the forecast of next demand size, α – smoothing parameter taking a value between zero and one, P_t – time between consecutive transactions, $\hat{P}_t$ – forecast of the demand interval, D_t – the final forecast value.

Data mining methods are powerful techniques presenting an alternative approach to conventional techniques. They are good at solving complicated practical problems in various areas, thus increasing their popularity day by day. Data mining is the non-trivial process of extracting implicit, previously unknown and potentially useful information from data which can also be referred to as knowledge discovery in the database. Data mining techniques can be used for a number of objectives such as clustering, classification, data summarization, discovering dependency networks, and detecting anomalies as stated by Punjari (2006).

Data mining techniques differ from other methods with their learning ability. Once they are trained they can perform prediction with high accuracy and high speed. Moreover, they have the ability to handle noisy data and non-linearity which make them applicable to most of the problems. Data mining methods consist of many models like artificial neural networks, expert systems, genetic algorithms, support vector regressions, fuzzy logic, decision tree and hybrid systems which are formed as a combination of two or more methods. In this study, we employed artificial neural networks, support vector regressions and decision tree techniques since they are commonly used data mining methods.

Artificial neural networks are formed as an imitation of the brain and its associated nervous systems. The model mainly contains an input layer, an output layer, and one or more hidden layers. The layers include interconnected neurons each of which is connected to other neurons in the subsequent layer. The neurons at hidden layer and output layer use a non-linear transfer function for processing their layer inputs by summing the product of each input by its weight in order to get a result (Chau, 2006). Figure 1 demonstrates neural network architecture with an input layer of three neurons, a hidden layer of two neurons and an output layer of one neuron.

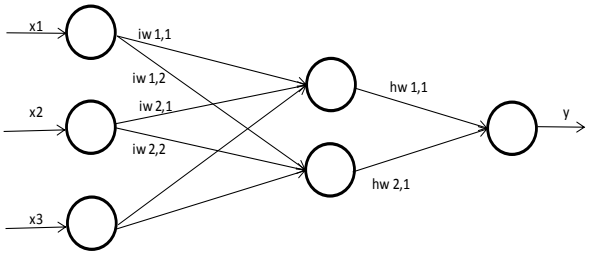


Fig. 1. Illustration of artificial neural network architecture

The second data mining method used in this study is Support vector regression which is the application of Support vector machines to time series forecasting. Support vector machine was introduced by (Vapnik, 1995). It is a powerful classification method that separates two classes by a hyperplane. Support vector machines are based on structured risk minimization principle since the method tries to minimize upper bound of error by obtaining a separating hyperplane as wider as possible. This property of the method is the source of its strength over traditional methods. Figure 2 illustrates the simplest case of two linearly separable classes where empty circles represent –1 and full circles represent +1. The hyper-plane separating two classes has the maximum margin between the samples of the two classes.

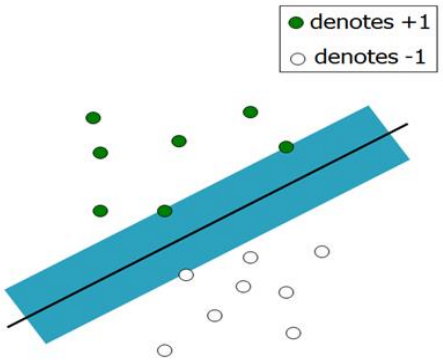


Fig. 2. The simplest illustration of support vector machines

The third data mining method used in this study is decision tree which aims to predict the value of a target variable based on several input variables. Decision tree models can also be learned like artificial neural network and support vector regression by splitting the source set. Mainly decision tree method uses a tree structure to build the classification models. Leaf nodes represent a decision and groups of instances that receive the same class label. Based on feature values of instances, they are classified in the decision tree according to labels of leaves. Each node in the tree represents a feature of an instance and each branch represents a value.

Figure 3 is a representation of a simple decision tree (Mitchell, 1997). There are three nodes in the decision tree which are outlook, humidity, and wind. The branches of the attributes show the possible values that the node can take. For this example, wind node can take on the values strong and weak. The leaf nodes at the bottom of the tree are the possible labels that can be made by classifying an instance using the tree. In this decision tree, an instance can be labeled as either a yes or no.

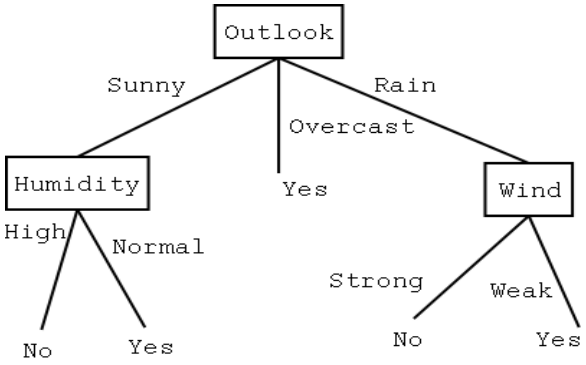


Fig. 3. Simplest illustration of decision tree

There are many variances of decision trees such as ID3, C4.5, and C5. But in this study, C5 is used since it has superior performance over others. For instance, C5 is faster than C4.5, it gets smaller decision trees than C4.5. Moreover, C5 is a better alternative than ID3 for handling the training data with missing attribute values (Pandya & Pandya, 2015).

3. EXPERIMENTAL RESULTS

This section includes the application of forecasting methods on simulated intermittent data and the performance results and comparisons. Common data mining methods which are artificial neural networks, support vector regressions and decision tree techniques are employed. Moreover, Croston's method with a smoothing parameter value of 0.2 is applied to the same data as the benchmark method since it is a very significant method in intermittent demand forecasting area since it is the first and main ad-hoc intermittent demand forecasting method.

3.1. Intermittent data

Since intermittent demand characteristics are not common for most of the fast moving products, there is limited real data. This situation leads us to generate artificial intermittent data in order to assess the performance of methods. In order

to generate artificial demand data, the intermittent package of R software is used. 400 intermittent demand series are generated each of which is composed of 300 data points. Statistical details regarding the simulated intermittent data are presented in Table 1.

Tab. 1. Descriptive statistics of the simulated data set

Statistics	Demand Size	Demand Interval
Mean	4.98	3.01
Std. Dev.	0.17	1.98
Minimum	4.36	1.03
1. Quartile	4.88	1.05
Median	4.98	2.65
3. Quartile	5.08	4.97
Maximum	5.66	6.05

In Figure 4, a simulated intermittent demand time series data is demonstrated. As can be seen from the figure, there is an interval between demand occurrences with changing values and demand values are also changing in value.

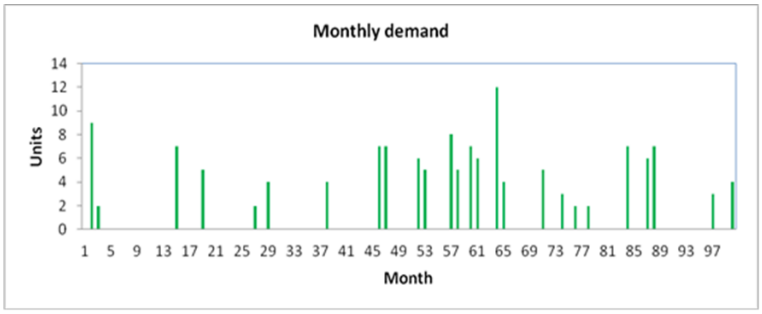


Fig. 4. An example of intermittent data

3.2. Accuracy evaluation and comparison

The prediction performance of the methods are evaluated using the following statistical metrics, namely, the root mean squared error (RMSE), mean absolute deviation (MAD), and alternative mean absolute percentage error (A-MAPE). For intermittent demand forecasting performance evaluation, the original MAPE cannot be used since actual demand values take zero value for most of the time periods which results in division by zero error. So, we used A-MAPE introduced by Hoover which is the ratio of mean absolute deviation to mean value. The employed metrics are calculated as follows (Hoover, 2006):

Let Y_t denote the observation at time t and F_t denote the corresponding forecast value Then define the forecast error as $e_t = Y_t - F_t$. Then:

$$\mathrm{RMSE} = \left({}^{1}\!/\!_{\mathrm{n}} \sum_{t=1}^{n} (Y_t - F_t)^2 \right)^{1/2} \tag{5}$$

$$MAD = {}^{1}\!/\!_{\mathrm{n}} \sum_{t=1}^{n} | Y_t - F_t | \tag{6}$$

$$A - MAPE = \frac{{}^{1}\!/\!_{\mathrm{n}} \sum_{t=1}^{n} | Y_t - F_t |}{{}^{1}\!/\!_{\mathrm{n}} \sum_{t=1}^{n} Y_t} \tag{7}$$

3.3. Forecast results

All four methods are applied to intermittent demand data. Since data mining methods have a learning process, data is divided into two as the training set and testing set where the method learns the structure in training stage. Training set includes the first 75% time periods for each intermittent time series (215 data points) and the last 25% time periods form the testing set (75 data points). In the following table, performance results for the training set are presented.

Tab. 2. Results for the training set

Performance	ANN	SVM	C5	Croston
RMSE	3.11	1.76	4.61	5.26
MAD	2.27	0.83	3.11	4.10
A-MAPE	0.76	0.28	1.04	1.37

As can be seen from the table, use of data mining method is very promising for intermittent demand forecasting since they have superior performance over Croston's method based on all performance metrics. This is an important result since Croston's method is a specialized intermittent demand method which has shown better performance over traditional methods. Moreover, Support vector machine is the best performing method among the artificial methods. Performance results for the testing set are presented in Table 3.

Tab. 3. Results for the testing set

Performance	ANN	SVM	C5	Croston
RMSE	2.66	1.35	2.93	4.38
MAD	2.17	0.67	2.34	3.16
A-MAPE	0.69	0.21	0.75	1.01

The performance results for the testing set are in line with the results for the training set. All data mining method has lower error values than the Croston's method with respect to all performance metrics.

4. CONCLUSIONS AND PERSPECTIVES

Due to peculiar characteristics of intermittent demand, traditional forecasting methods perform poorly. Traditional methods assume demand to be stationary where intermittent demand data is not stationary. More complicated methods like data mining methods are more appropriate to be used for intermittent demand forecasting since they have been shown to perform successfully in most of the research.

Considering both the importance and the difficulty of accurate demand forecasting for intermittent demand items, this study focuses on applying data mining methods since they are powerful methods which are able to deal with non-stationary data. Since intermittent demand time series have many time periods with zero demand and have variable demand sizes, they possess non-stationary pattern.

Common data mining methods which are artificial neural networks, support vector regressions and decision tree techniques and Croston's method which is developed specifically for intermittent demand forecasting are applied to simulated intermittent demand data. Results showed that use of data mining methods are beneficial with their high-performance values and support vector machine is the most appropriate one since it resulted with the lowest error value among all techniques.

Finally, considering the lack of effective forecasting methods for intermittent demand, it would be interesting to compare the employed data mining methods with other methods such as genetic programming, fuzzy logic and hybrid methods. These perspectives are let as future works.

REFERENCES

Bacchetti, A., & Saccani, N. (2012). Spare parts classification and demand forecasting for stock control: Investigating the gap between research and practice. *Omega*, *40*, 722–737.

Chau, K. W. (2006). A review on the integration of data mining into coastal modeling. *Journal of Environmental Management*, *80*, 47–57.

Croston, J. F. (1972). Forecasting and stock control for intermittent demands. *Operational Research Quarterly*, *23*, 289–304.

Hoover, J. (2006). Measuring Forecast Accuracy: Omissions in Today's Forecasting Engines and Demand-Planning Software. *International Journal of Applied Forecasting*, *1*, 32–35.

Hua, Z., & Zhang, B. (2006). A hybrid support vector machines and logistic regression approach for forecasting intermittent demand of spare parts. *Applied Mathematics and Computation*, *181*, 1035–1048.

Kennedy, W. J., Patterson, J. W., & Fredendall, L. D. (2002). An overview of recent literature on spare parts inventories. *Int. J. of Production Economics*, *76*, 201–215.

Mitchell, T. (1997). *Machine Learning.* Boston: McGraw-Hill.

Pandya, R., & Pandya, J. (2015). C5.0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning. *International Journal of Computer Applications*, *11*, 718–21.

Punjari, A. K. (2006). *Data Mining Techniques*. Universities Press Private Limited.

Rao, A. V. (1973). A comment on: forecasting and stock control for intermittent demands. *Operational Research Society*, *24*, 639–640.

Regattieri, A., Gamberi, M., Gamberini, R., & Manzini, R. (2005). Managing Lumpy Demand for Aircraft Spare Parts. *Journal of Air Transport Management*, *11*, 426–431.

Syntetos, A. A. (2007). A note on managing lumpy demand for aircraft spare parts. *Journal of Air Transport Management*, *13*, 166–167.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* New York: Springer.

Applied Computer Science, vol. 14, no. 2, pp. 48–59
doi:10.23743/acs-2018-12

Submitted: 2018-05-25
Revised: 2018-06-05
Accepted: 2018-06-25

neural networks, gypsum-polymers, rubber regranulate

Grzegorz KŁOSOWSKI[*], *Tomasz KLEPKA*[**], *Agnieszka NOWACKA*[**]

NEURAL CONTROLLER FOR THE SELECTION OF RECYCLED COMPONENTS IN POLYMER-GYPSY MORTARS

Abstract

This study presents research on the development of an intelligent controller that allows optimal selection of rubber granules, as an admixture recycling component for polymer-gypsy mortars. Based on the results of actual measurements, neural networks capable of predicting the setting time of gypsum mortar, as well as determining the bending and compressive strength coefficients were trained. A number of simulation experiments were carried out, thanks to which the characteristics of setting times and strength of mortars containing different compositions of recycling additives were determined. Thanks to the obtained results, it was possible to select the rubber admixtures optimally both in terms of the percentage share as well as in relation to the diameter of the granules.

1. INTRODUCTION

This paper presents the original method of selecting the amount of admixture of rubber granulate and the diameter of granules for the gypsum mixture. Thanks to the use of artificial neural networks, a neural controller has been developed that allows optimal selection of gypsum mixture parameters. Three output variables were optimized: setting time [min], bending strength coefficient (BSC) and compressive strength coefficient (CSC).

[*] Lublin University of Technology, Lublin, Poland, +48 81 538 45 67, g.klosowski@pollub.pl
[**] LUT – Department of Technology and Polymer Processing, Lublin, Poland, +48 81 538 47 66

Calcium sulphates in semi-aquatic form have been used for a long time as building binding materials, so-called gypsum binders (Osiecka, 2005). These binders are ecological materials, obtained from natural raw materials, which allow building elements of various sizes and shapes to be quickly and easily (Chłądzyński, 2008). Modern binders are increasingly being created by adding ingredients in the form of fillers from various materials (Hooton, 2015). Due to the size of production and the possibility of processing into valuable building materials, materials originating from waste are becoming more and more important as binder additives, among which the ones of polymer materials are of the greatest importance (Di Mundo, Petrella & Notarnicola, 2018). The basic advantage which should have a new gypsum binder containing a semi-hydrate, deciding its usefulness in construction, is the short setting time and obtaining a good hardness (Serna, del Rio, Palomo & González, 2012). The binder can thus be a polymer-mineral mixture (polymer-cement or gypsum and polymer-lime) as well as a mortar with the addition of only polymer fillers (Aslani, Ma, Wan & Muselin, 2018).

In construction mobile connections, which include expansion joints and larger assembly gaps, the joint material must provide adjacent structures with the possibility of mutual movement in a specific plane, with a clearly defined range of changes (Claudiu, 2013). The vibrations of load-bearing elements, caused by external loads, e.g. shocks, uneven settlement of objects, dynamic loads from vehicle traffic, as well as dimensional changes caused by temperature changes must also be compensated. The sealing materials must therefore have appropriate physical and chemical properties as well as resistance to bending, compressive and shearing forces (Pedro, De Brito & Veiga, 2012). Gypsum-polymer mortars suitable as a binder can work well with virtually all other building materials.

Nowadays, the most pursued route to enhanced elastic properties is the formation of C-S-H organic hybrids on the nanoscale by incorporating soft matter into and/or in between the C-S-H nanoplates (Bergström, 2015; Seto 2012). The two necessary main features for achieving strong specific C-S-H/polymer interactions are negatively charged groups for Ca^{2+} mediated electrostatic interactions and hydrophilic residues preferentially with alcohol or amide groups for hydrogen bond interactions (Picker, 2017). The bond between the C-S-H nanoplates and polymers can be obtained by mixing a resin with aggregate (Kou & Poon, 2013). The most commonly used resins for PC are unsaturated polyester resin, epoxy resin, furan resin, polyurethane resin, and urea formaldehyde resin (Gorninski, 2017). The matrix is given by the polymer which acts as binder of the components, that can be different types of natural aggregates, powders, fibers, nano-materials, etc. (Sosoi & Barbuta, 2018). Polymers, such as styrene-butadiene rubber (SBR) latex and polyvinyl acetate (PVA) emulsion have been commonly used as admixtures in concrete practice (Konar, Das, Gupta & Saha, 2011). This polymer admixtures are known to not only increase the workability but also modify the physical properties of cement pastes by reducing macro voids and improving

the bond strength of the polymer cement mortars to aggregates (Al Menhosh, Wang, Wang, & Augusthus-Nelson, 2018). Recent studies on recycled tire polymer fibers (RTPF) have shown that the addition of this type of fibers in fresh concrete mixes has a positive effect on volume deformations at an early age and mitigates the explosive spalling at high temperatures without affecting the residual mechanical properties of concrete (Baricevic, Jelcic, Rukavina & Pezer, 2018). Based on non-destructive tests (NDT), it was found that increasing the coarse aggregate size and polymer ratio reduced the porosity of specimens. This is attributed to the decreased surface area to volume ratio with increasing particle size, which allowed the polymer to completely coat the surface of aggregates (Jafari, Tabatabaeian & Joshaghani, 2018). However, prior to their utilization in construction products, such materials require characterization which will demonstrate their properties and assure positive long-term behavior, especially when exposed to aggressive environmental conditions (Serdar, 2015).

Such environmental factors could lead to a decrease in performance, thus decrease the durability of concrete. Exposure to high temperatures is one of these physical factors that have an impact on concrete (Sahmaran, 2009). Recent studies shown that structures exposed to high temperature can be used the polymer-phosphazene concrete. The filling of the voids with the polymer was ensured so that the physical binding of polymer molecules and strong adhesion of these molecules to each other was obtained (Tanyildizi & Asilturk, 2018). This strategy could greatly benefit future construction processes because fracture toughness and elasticity of brittle cementitious materials can be largely enhanced on the nanoscale.

Gypsum-polymer mortars can be used to supplement cavities, leveling walls and floors by increasing the elasticity of the mixture (Forrest, 2014). Thanks to this, after solidifying, the weld section made is resistant to deformation that occurs during the period of operation of the analyzed object (Benosman et al., 2017). Gypsum-polymer mortar can be products with good functional properties, easy to use. Proper selection of the additive, e.g. regranulate, will allow to obtain an improved flexible mortar, which can be evaluated on the basis of tests of bending strength or compressive strength, etc. (Jarosiński, Żelazny & Nowak, 2007; Herrero, Mayor & Hernández-Olivares, 2013).

The aim of the research was to develop an intelligent controller that would allow optimal selection of rubber granulate parameters as an admixture for gypsum-polymer mortar with addition of hydrated lime and admixtures in various amounts from rubber regranulate. Parameters such as: time of setting components of the gypsum mixture (minimization of time) and mechanical properties of the obtained mortar (maximization of strength) were optimized.

2. MATERIALS AND METHODS

During the research it was found that due to the complex, non-linear dependence of 2 independent variables (input):

1. Diameter of granules,
2. Percent of admixture,

regressive neural networks can be an appropriate tool to train the controller (Lorrentz, 2015). Based on 120 real measurements, 3 separate neural networks were trained. Each of the networks had at the entry 2 of the above-mentioned values. The network outputs were: setting time [min], bending strength factor (BSC) and compressive strength index (CSC). Both strength coefficients are expressed in Newton [N].

In order to collect training data for the neural network, research was conducted on polymer-gypsum mortars. The tests were based on samples similar in shape and size to welds used in real conditions. Characteristic indicators that determined the weld's usefulness are the time of binding of the gypsum-polymer mixture as well as the bending and compression strength factors.

$CaSO_4 \cdot ½H_2O$ construction gypsum was used for the developed material composition. The gypsum bond fulfilled the requirements of EN 13279-1-A1. The content of calcium sulphate ($CaSO_4$) was above 50%, the graining of the material was covered by a 1.0 mm sieve residue, no more than 0.5% and a 0.2 mm sieve, which was equal to or lower than 15%. Hydrated lime was added to the mixture, its share in mixtures with gypsum did not exceed 2%. Rubber granulate derived from the recycling of tires with several grain diameters from ø 0.1 mm to ø 0.63 mm was used as an additive to increase flexibility.

The components of the mixtures were introduced in various proportions to the binder when mixing with water. They consisted of gypsum, hydrated lime and polymer regranules. When choosing the right regranulate for the tests, the amount of polymer incorporated into the mixture was also significant, the appropriate amount being determined based on the adopted research program.

Fig. 1 presents molds for the preparation of gypsum-polymer mortar beams. Fig. 2 presents a sample of a set plaster mortar containing 20% of a rubber granulate with a granule diameter of 0.1 mm.

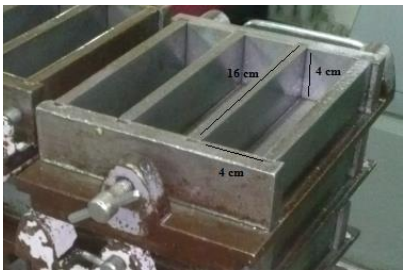


Fig. 1. Molds for the preparation of mortar beams

Fig. 2. Gypsum mix with 20% rubber granulate with a grain diameter of ⌀ 0.1 mm

To determine the bending and compressive strength of the samples, a hydraulic press with a 16 cm^2 jaw area was used, compressing at the standard speed, with a jaw pressure from 0–200 kN. The studies of setting time of gypsum mixtures were made using the Vicat apparatus.

Fig. 3 shows a schematic of a neural network. The input vector consists of 2 elements. It is a multi-layer perceptron with one hidden layer containing 10 neurons (Thomas & Thomas, 2011). The hidden layer transition function has a sigmoidal waveform. One neuron is located in the output layer. The transition function has a linear course here. The output is a real number, so we are dealing with a regression model.

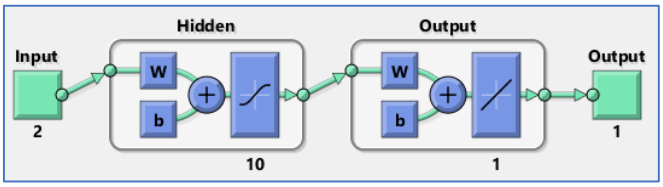


Fig. 3. Diagram of neural network used in research

Table 1 shows the results of training the neural network. The historical data set consisted of 120 cases, divided into 3 sets: training set, validation set and testing set, in the ratio of 70:15:15. The main measures of network quality are mean squared error (MSE) and regression (R).

Tab. 1. Results of neural network training for individual sets

Sets	Samples	MSE	R
Training	84	$6.51330 \cdot 10^{-4}$	0.997353
Validation	18	$1.66109 \cdot 10^{-3}$	0.994454
Testing	18	$5.97817 \cdot 10^{-4}$	0.998574

The formula based on which the MSE error is calculated is presented using the dependence (1).

$$MSE = \frac{1}{n} \cdot \sum_{i=1}^{n} (ref_i - x_i)^2 \tag{1}$$

where: n – number of cases in the set (training, validating or testing),
ref – reference values (patterns),
x – predicted values of a model.

The high quality of the network is demonstrated by low MSE and high R. The best indicator reflecting the ability of the neural network for generalization is MSE for testing set. Testing set contains cases that did not participate in the network training process, hence MSE for testing set is usually higher than for training set. In the present case, the MSE values for testing set and training set are very similar. Both values are very low, which is a good sign of the quality of the trained neural network. Regression for testing set is 9.986 which is a high indicator and also confirms the effectiveness of the solution obtained.

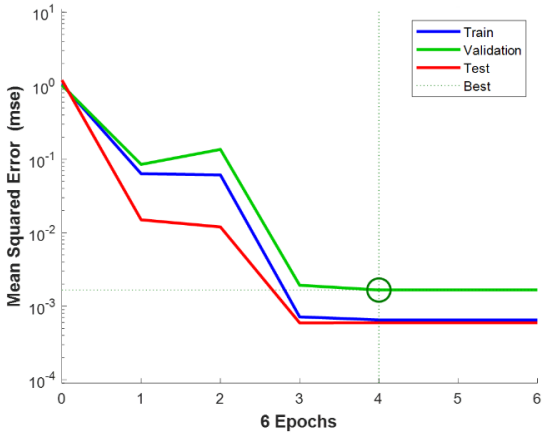


Fig. 4. Plot of neural network performance according to MSE

Fig. 4 shows the course of training the neural network based on the MSE values in individual iterations (epochs). The shape of the curve resembles a bit of hyperbole.

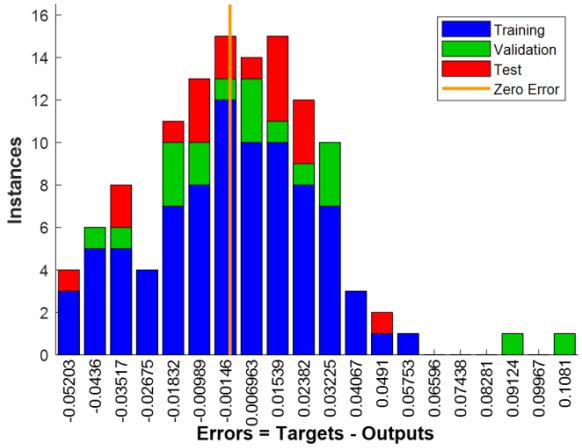


Fig. 5. Error histogram of the neural network training process

There are not many fluctuations on the chart. These features testify to the lack of network overtraining. On the line of the validation error, the place where the learning process ends was marked. It falls into epoch 4. Best validation performance (MSE) equals 0.0016611. The task of the validation set is to stop training the neural network, which occurs when after six successive iterations MSE does not fall.

Fig. 5 shows a histogram of MSE errors for sets: training, validation and testing. The whole range of errors has been divided into 20 bins. The most errors had absolute values close to zero. The shape of the graph resembles a normal distribution curve, which also well shows the quality of the obtained neural network.

Quantitatively, the most registered MSE errors concerned training set. This is due to the fact that all historical cases were divided in such a way that as many as 70% of them fell on the test set. The remaining 30% were divided into validation and test sets. Fig. 5 reflects this division.

Fig. 6 presents regression statistics of the obtained neural network. As you can see, regression is close to 1 for both the test set (left picture) and for all cases including training, validation and testing (right picture).

Regression is a measure of matching the network response to patterns. If it is close to 1 even for test cases that do not participate in the network learning process, it means that the network has the ability to generalize responses. This is a key feature thanks to which the network deserves to be called intelligent.

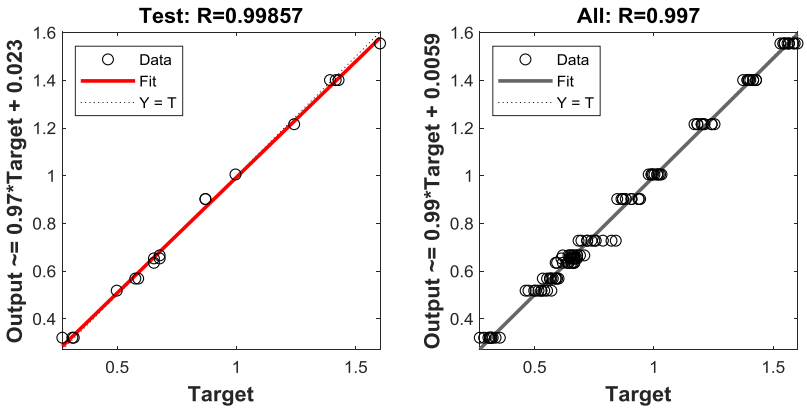


Fig. 6. Regression statistics of the neural network training process

Fig. 7 shows a simulation model of the system for the selection of recycled components in polymer-gypsy mortars. The configuration of the presented model assumes that the diameter of the rubber granulate is constant and amounts to ⌀ 0.35 mm. The percentage of admixture varies in the given range. Several variants of simulation models were tested, in which the constant diameter of granules assumed the following values: ⌀ 0.1, 0.2, 0.35 and 0.5 mm. For each given diameter of granules, the percentage of admixture was smoothly changed.

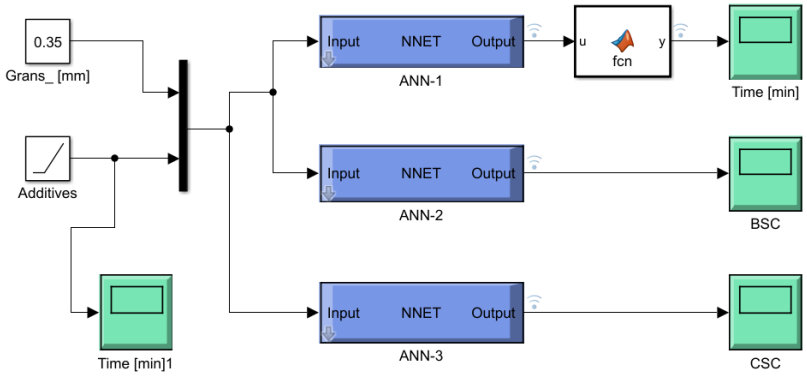


Fig. 7. Simulation model including three separate neural networks

Three separate neural networks were trained in which each had a different output. In the first network (ANN-1) the dependent variable was the binding time, in the second network (ANN-2) it was the bending strength factor (BSC) and in the third network (ANN-3) the bending strength index (CSC). In the case of ANN-1 it was necessary to convert the real number to minutes, hence an additional module visible in Fig. 7 was placed to the right of ANN-1.

3. RESULTS

Based on the designed neural models, a number of simulation experiments were carried out, including various variants of intelligent controller settings. Variants were tested in which the diameter of the granules was predetermined (ø 0.1 mm, ø 0.2 mm etc.), while the proportion of admixture of rubber granulate was smoothly variable. The results of the tests are shown in Fig. 8 and Fig. 9.

The drawings are presented in pairs, in a system allowing to compare two cases together with different diameter of granules next to each other with a smooth change in the amount of additives. In Fig. 8, the diameters of granules ø 0.1 mm and ø 0.2 mm were compared. The first pair of graphs in Fig. 8 shows the duration of the binding time expressed in minutes. The next pairs of drawings show the waveforms of BSC and CSC values expressed in Newtons. The drawings in Fig. 9 are presented in an analogous manner, but the analyzed cases concerned different granule diameters: ø 0.35 mm and ø 0.5 mm.

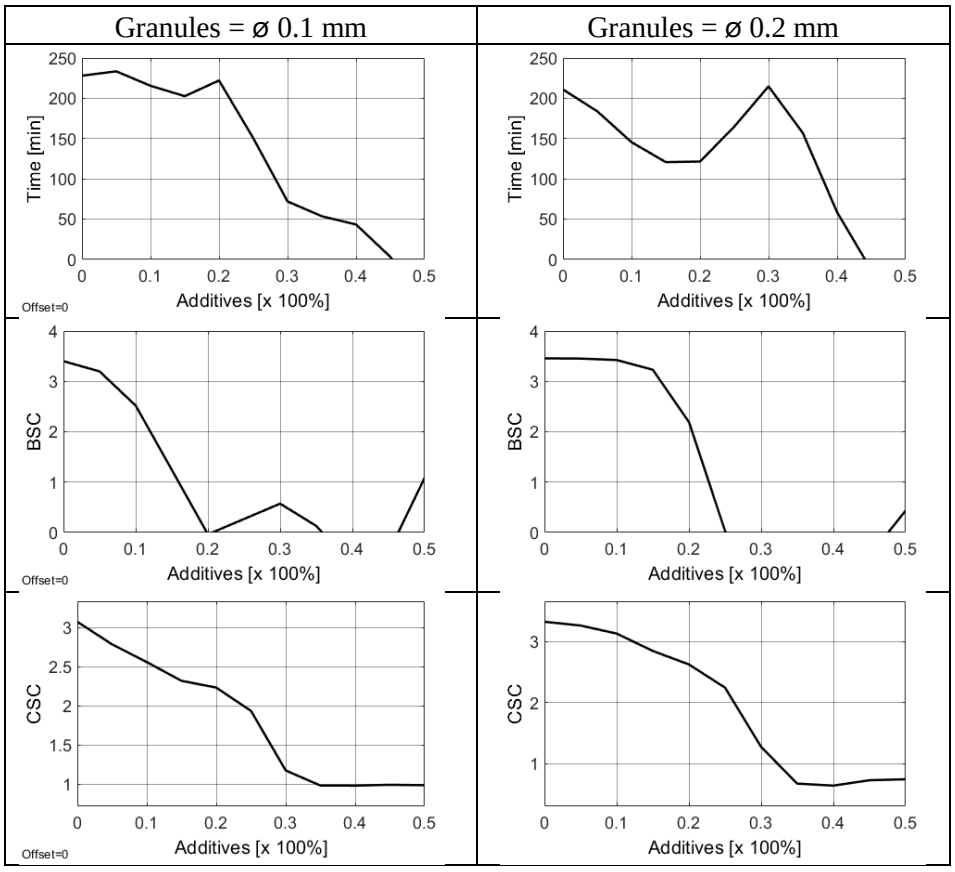


Fig. 8. Prediction results with constant granule diameters (Ø 0.1 mm and Ø 0.5 mm)

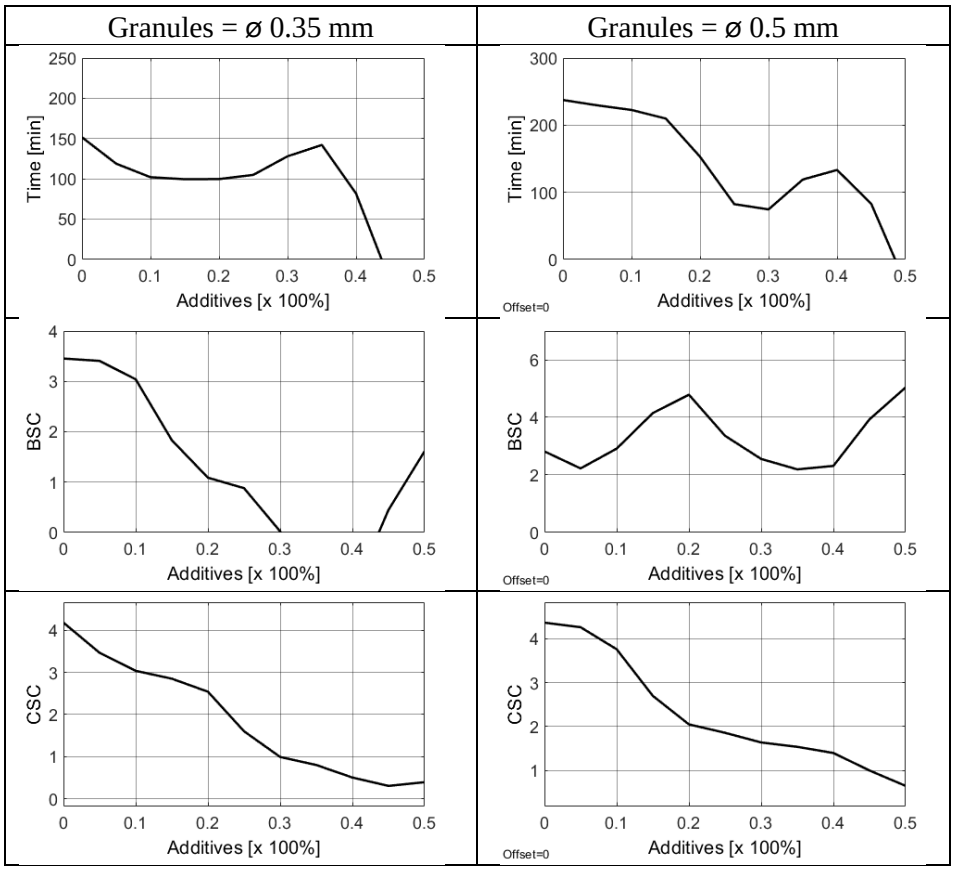


Fig. 9. Prediction results with constant percentage inclusions (10% and 40%)

4. CONCLUSIONS

Analyzing the results of the experiments shown in Figs. 8 and 9, it can be seen that in order to select the optimal parameters of admixture of rubber granules, multiple graphs should be analyzed simultaneously. For example, some waveforms reach zero values, which might seem a mistake. However, it must be taken into account that ranges of input variables (predictors) exceed limits for some output parameters. Therefore, when analyzing the results, one should look for such values of the horizontal axis, for which the strength parameters (CSC and BSC) presented on the vertical axis are as high as possible. Most often, this kind of choice will be a kind of trade off.

It can be concluded that the optimal level of admixture of rubber in CSC and BSC should not exceed 10%. In turn, the diameter of the granules should oscillate from ø 0.35 mm to ø 0.5 mm. With these parameters, the admixtures, CSC and BSC

values remain at relatively high levels. If we add to the target beam also minimizing the time of setting the gypsum mixture, the optimal mixture should contain 10% admixture with a granule diameter of ø 0.35 mm.

To analyze the increase in load bearing capacity of joints filling expansion joints a different gypsum sample size would be used. In turn, to simulate the parameters of welds in smaller building gaps, it is likely that other cross-section shapes should be used. On the basis of the research results presented above, it can be concluded that addition of polymer admixtures with different percentage to the gypsum mixture increases both the flexibility of the mortar and load transfer without damaging the weld structure, which increases the range of use of ceramic and architectural elements with such admixtures.

Further studies are required to evaluate and distinguish influence of crumb rubber inclusions from fibers' contribution, as recent studies have indicated a significant potential of recycled tire polymer fibers (RTPF) in the construction industry.

REFERENCES

Aslani, F., Ma, G., Wan, D. L. Y., & Muselin, G. (2018). Development of high-performance self-compacting concrete using waste recycled concrete aggregates and rubber granules. *Journal of Cleaner Production*, *182*, 553-566. doi:doi.org/10.1016/j.jclepro.2018.02.074

Baricevic, A., Jelcic Rukavina, M., & Pezer, M. (2018). Influence of recycled tire polymer fibers on concrete properties. *Cement and Concrete Composites*, *91*, 29–41.

Benosman, A. S., Taïbi, H., Senhadji, Y., Mouli, M., Belbachir, M., & Bahlouli, M. I. (2017). Plastic Waste Particles in Mortar Composites: Sulfate Resistance and Thermal Coefficients. *Progress in Rubber, Plastics and Recycling Technology*, *33*(3), 171.

Bergström, L., Sturm (née Rosseeva), E. V., Salazar-Alvarez, G., & Cölfen, H. (2015). Mesocrystals in biominerals and colloidal arrays. *Acc. Chem. Res.*, *48*, 1391–1402. doi:10.1021/ar500440b

Chłądzyński, S. (2008). *Spoiwa gipsowe w budownictwie*. Warsawa: Dom wydawniczy Medium.

Aciu, C. (2013). Possibilities of Recycling Rubber Waste in the Composition of Mortars. *ProEnvironment Promediu*, *6*(15).

Di Mundo, R., Petrella, A., & Notarnicola, M. (2018). Surface and bulk hydrophobic cement composites by tyre rubber addition. *Construction and Building Materials*, *172*, 176–184. doi:10.1016/j.conbuildmat.2018.03.233

Forrest, M. (2014). *Recycling and re-use of waste rubber*. Shropshire: Smithers Rapra.

Gorninski, J. P., Dal Molin, D. C., & Kazmierczak, C. S. (2007). Strength degradation of polymer concrete in acidic environments. *Cem. Concr. Compos.*, *29*(8), 637–645. doi:10.1016/j.cemconcomp. 2007.04.001

Herrero, S., Mayor, P., & Hernandez-Olivarez, F. (2013). Influence of proportion and particle size gradation of rubber from end-of-life tires on mechanical, thermal and acoustic properties of plaster-rubber mortars. *Materials & Design*, *47*, 633–642. doi:10.1016/j.matdes.2012. 12.063

Hooton, R. D. (2015). Current developments and future needs in standards for cementitious materials. *Cement and Concrete Research*, *78*, 165–177. doi:10.1016/j.cemconres.2015. 05.022

Jafari, K., Tabatabaeian, M., Joshaghani, A., & Ozbakkaloglu, T. (2018). Optimizing the mixture design of polymer concrete: An experimental investigation. *Construction and Building Materials*, *167*, 185–196. doi:10.1016/j.conbuildmat.2018.01.191

Jarosiński, A., Żelazny, S., & Nowak, A. (2007). *Warunki otrzymywania spoiwa gipsowego z produktu odpadowego pochodzącego z procesu pozyskiwania koncentratu cynku*. Kraków: Czasopismo techniczne 1/Ch-2007 Wydawnictwo Politechniki Krakowskiej.

Konar, B., Das, A., Gupta, P. K., & Saha, M. (2011). Physicochemical characteristics of styrene-butadiene latex- modified mortar composite vis-à-vis preferential interactions. *J. Macromol. Sci*., *48* (9), 757–765. doi:10.1080/10601325.2011.596072

Kou, S.-C., & Poon, C.-S. (2013). A novel polymer concrete made with recycled glass aggregates, fly ash and metakaolin. *Constr Build Mater.*, *41*, 146–151. doi:10.1016/j.conbuildmat. 2012.11.083

Lorrentz, P. (2015). *Artificial Neural Systems: Principle and Practice*. Bentham Science Publishers. doi: 10.2174/9781681080901115010l

Al Menhosh, A., Wang, Y., Wang, Y., & Augusthus-Nelson, L. (2018). Long term durability properties of concrete modified with metakaolin and polymer admixture. *Construction and Building Materials*, *172*, 41–51. doi:10.1016/j.conbuildmat.2018.03.215

Osiecka, E. (2005). *Materiały budowlane – tworzywa sztuczne*. Warszawa: Oficyna Wydawnicza Politechniki Warszawskiej.

Pedro, D., De Brito, J., & Veiga, R. (2012). Mortars made with fine granulate from shredded tires. *Journal of Materials in Civil Engineering*, *25*(4), 519–529. doi:10.1061/(ASCE)MT. 1943-5533.0000606

Picker, A., Nicoleau, L., Burghard, Z., Bill, J., Zlotnikov, I., Labbez, C., Nonat, A., & Cölfen, H. (2017). Mesocrystalline calcium silicate hydrate: A bioinspired route toward elastic concrete materials. *Science Advances*, *11*(3), 37–49. doi:10.1126/sciadv.1701216

Sahmaran, M., & Li, V. C. (2009). Durability properties of micro-cracked ECC containing high volumes fly ash. *Cem. Concr. Res.*, *39*, 1033–1043. doi:10.1016/j.cemconres.2009.07.009

Seto, J., Ma, Y., Davis, S. A., Meldrum, F., Gourrier, A., Kim, Y.-Y., Cölfen, H. (2012). Structure-property relationships of a biological mesocrystal in the adult sea urchin spine. *Proceedings of the National Academy of Sciences*, *109*(10), 3699.

Serdar, M., Baricevic, A., Jelcic Rukavina, M., Pezer, M., & Bjegovic, D. (2015). Shrinkage behaviour of fibre reinforced concrete with recycled tyre polymer fibres. *Int. J. Polym. Sci*., 145918. doi:10.1155/2015/145918

Serna, Á., del Rio, M., Palomo, J. G., & González, M. (2012). Improvement of gypsum plaster strain capacity by the addition of rubber particles from recycled tyres. *Construction and Building Materials*, *35*, 633–641. doi:10.1016/j.conbuildmat.2012.04.093

Sosoi, G., Barbuta, M., Serbanoiu, A. A., Babor, D., & Burlacu, A. (2018). Wastes as aggregate substitution in polymer concrete. *Procedia Manufacturing*, *22*, 347–351. doi: 10.1016/j.promfg.2018.03.052

Tanyildizi, H., & Asilturk, E. (2018). High temperature resistance of polymer-phosphazene concrete for 365 days. *Construction and Building Materials*, *174*, 741–748. doi:10.1016/j.conbuildmat. 2018.04.078

Thomas, P., & Thomas, A. (2011). Multilayer perceptron for simulation models reduction: Application to a sawmill workshop. *Engineering Applications of Artificial Intelligence*, *24*(4), 646-657. doi:10.1016/j.engappai.2011.01.004

Applied Computer Science, vol. 14, no. 2, pp. 60–68
doi:10.23743/acs-2018-13

Submitted: 2018-05-29
Revised: 2018-06-10
Accepted: 2018-06-25

renovation, continuous casting line, rollers, SAW, wear

Janette BREZINOVÁ[*], Ján VIŇÁŠ[*], Dagmar DRAGANOVSKÁ[*], Anna GUZANOVÁ[*], Jakub BREZINA[*]

POSSIBILITIES OF RENOVATION FUNCTIONAL SURFACES OF EQUIPMENTS IN THE MECHANICAL ENGINEERING INDUSTRY

Abstract

The paper analyzes the possibilities of increasing the lifespan of rollers in continuous steel casting line. There are analyzed the causes of the surface rollers damage and the impact of degrading factors in metallurgical production. Three types of welding consumable electrodes designed for restoration layers formation applied by SAW surfacing technology were analyzed. There were analyzed microstructure, microhardness and properties of weld clads in tribological conditions.

1. INTRODUCTION

The technology of continuous casting of steel slabs was introduced for full range of dead-melted steels, deep-drawing, so-called pseudodead-melted steels and also effervescing steels, which replaced the traditional method of making slabs by rolling ingots (Brezinova et al., 2017; Blaskovits & Čomaj, 2006). Continuous casting line is in generally one-strand or two-strand (Nadooshan, Saeedi, Rasooli, Izadi & Poursina, 2009; Olson, Dixon & Liby, 2012). The rollers have been made from the material C-Cr-Mo, or C-Cr-Mo-V type, they are solid (forged or split), alternately warmed-up by hot cast slab and cooled by

[*] Technical University of Košice, Faculty of Mechanical Engineering, Masiarska 74, 040 01 Košice, Slovakia, +421 55 602 3542, janette.brezinova@tuke.sk; jan.vinas@tuke.sk; dagmar.draganovska@tuke.sk; anna.guzanova@tuke.sk; jakub.brezina@tuke.sk

external water-vapour-air cooling system (Lorincova 2012; Mikuš, Szabó, Drlička & Bakajová, 2014). In the upper part of continuous casting line temperature of slab is approximately of 1250°C and in the bottom part about 800°C (Vinas, Brezinová, Guzanová & Lorincová, 2011; Vinas, Brezinová, Guzanová & Svetlík, 2013).

The rollers of continuous casting line are exposed to combined abrasive-adhesive load, connected with high temperature cyclic fatigue stress in corrosion environment, what results in changing of material structure and properties of functional surface (Houldcroft, 2014; Wu, Xiang, He, Chen & Hu, 2015). Application of cladding layer created using high-alloy welding wires in combination with flux enables creating functional roller surface with higher lifespan than new rollers (Vinas, Greš & Vaško, 2016; Paulicek, Kotus, Daňko & Žúbor, 2013, Kotus, Andrássyová, Cico, Fries & Hrabe, 2011). For renovation by hard surfacing are generally used at least three cladding layers (Toyserkani, Khajepour & Corbin, 2004; Lekhov, Mikhalev, Bilalov & Shevelev, 2017).

Thermal treatment of the roller with clad applied shows Fig. 1.

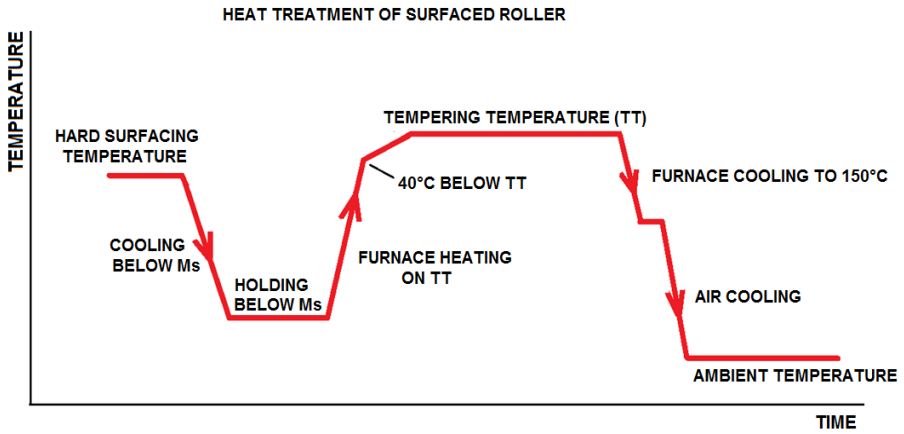


Fig. 1. Thermal treatment of the roller

2. EXPERIMENTAL PROCEDURE

In the experimental part of work, the twice renovated roller with diameter of 180 mm, made from forged steel X20Cr13 EN 10088-3 was used. The chemical composition of the roller is given in Table 1.

Tab. 1. Chemical composition of the roller X20Cr13 in [%]

C	Cr	Si	Mn	P	S	Fe
0,16–0,25	12,00–14,00	max. 1,00	max. 1,50	max. 0,040	max. 0,015	bal.

Wear size of roller was assessed by visual check. Worn layer (with thickness of 7 mm) was removed from roller surface by turning. Turned surface was checked visually again. The presence of inner defects was checked by ultrasonic test. Considering chemical composition of roller, there was necessary to preheat it before cladding on temperature of 350°C–400°C. Chemical composition of welding wires used is given in Table 2**.**

Tab. 2. Chemical composition of weld wires [in wt. %], balance of Fe

Wire	C	Si	Mn	Ni	Cr	Mo	Nb	V
W 3	0.1	0.6	1.0	2.5	12.2	0.8	0.15	0.15
W 5	0.25	0.6	1.0	0.25	9.0	2.0	–	–
W 8	0.3	0.6	1.0	–	12.2	0.75	–	0.15

For cladding was used universal flux AWS A5 17-89 EM 13K, which chemical composition can be found in Table 3.

Tab. 3. Chemical composition of universal flux AWS A5 17-89 EM 13K

$\frac{S_iO_2 + T_iO_2}{20}$	$\frac{C_aO + M_gO}{38}$	$\frac{AL_2O_3 + M_nO}{17}$	$\frac{C_aF_2}{19}$

Welding wire W3 (W3-WLDC 3) is commercially used for cladding of continuous casting rollers. Welding wires W8 (W8-WLDC 8) and W5 (W5HT-WLDC 5) were still not used for cladding of continuous casting rollers. The roller was renovated using welding equipment Weldclad GU125LZ, COREWIRE. The parameters of cladding are listed in Table 4. After cladding, roller was cooled down in electric furnace in isothermic wrap. Cooling speed was 40°C per hour up to room temperature. Then it was tempered in furnace at 500°C / 8 hours.

Tab. 4 Parameters of cladding

Wire	Diameter [mm]	Voltage [V]	Current [A]	Oscillations [mm]
W 3	3.2	28	450	45
W 5	3.2	26	600	47
W 8	3.2	26	450	50

After the surfacing and heat treatment the roller was turned to specified diameter. Quality of weld layers was assessed by non-destructive testing according to STN EN ISO 23277 and STN EN ISO 11666. Microhardness tests were realized by Vickers diamond pyramid, load applied: 980.7 mN (HV 0.1), time of indentation: 15 s, distance of indentations: 1–16 mm (with step of 1 mm). Test samples were subjected to cyclic thermal load in high-temperature chamber furnace. Test samples were heated to temperature of 400°C and 900°C. Heating temperature 400°C was chosen to verify the resistance of weld deposits against temperatures, to which the roller is exposed during its normal operation in continuous steel casting line. Heating temperature 900°C was chosen to verify the resistance of weld deposits against thermal load when continuous casting line is stopped due to emergency reasons. After reaching the sample temperature of and remaining at stated temperature for 5s, the samples were rapidly cooled by water with temperature of 20°C. The cycle of heating and cooling of the samples was repeated 20 times.

Erosive wear of weld clads was simulated by mechanical abrasive blasting process. Brown corundum with grain size 0.71 mm was used as blasting abrasive, velocity of abrasive grains was 70.98 mps, impact angle was 45° and 90°. Abrasive grains impinged the cover layer of cladding. Wear resistance of weld clads was evaluated by mass loss of the cladding.

3. RESULTS AND DISCUSSION

Microstructure of cover layer, first cladding layer and HAZ are presented in Table 5 and in Fig. 2. At a depth of 16 mm below the surface, microstructure of the base material consists of low-carbon martensite.

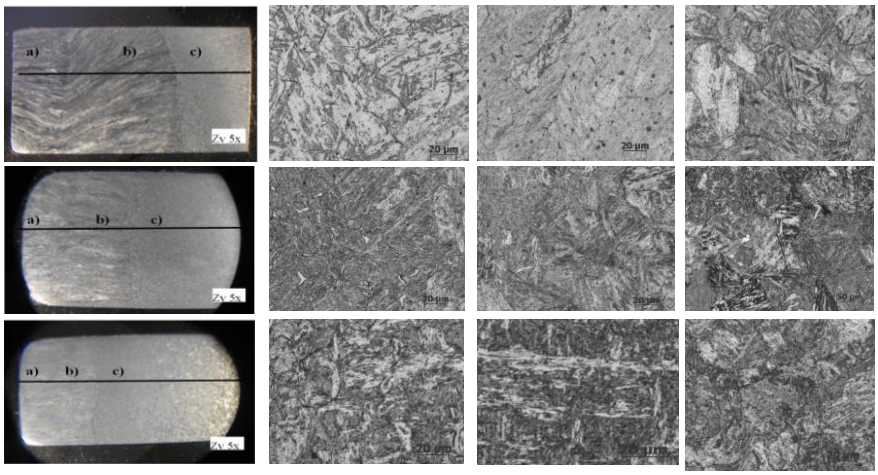


Fig. 2. Macro- and microstructure of weld clads (a – cover layer, b – first layer, c – HAZ)

Tab. 5 Microstructure of weld clads

	Cover layer	First layer	HAZ
W 3	low-carbon high-tempered martensite (sorbite)	low-carbon high-tempered martensite, carbide phases	low-carbon tempered martensite without carbids
W 5	martensite with islands of ferrite	low-carbon tempered martensite	low-carbon tempered martensite
W 8	tempered martensite with small carbide particles	of low-carbon tempered martensite	low-carbon tempered martensite

Fig. 3 shows the microhardness of particular weld layers and HAZ of weld deposits W3, W5 and W8. Higher microhardness was found in clads made using welding wires W5 and W8, from the cover layer to the base material the hardness decreased.

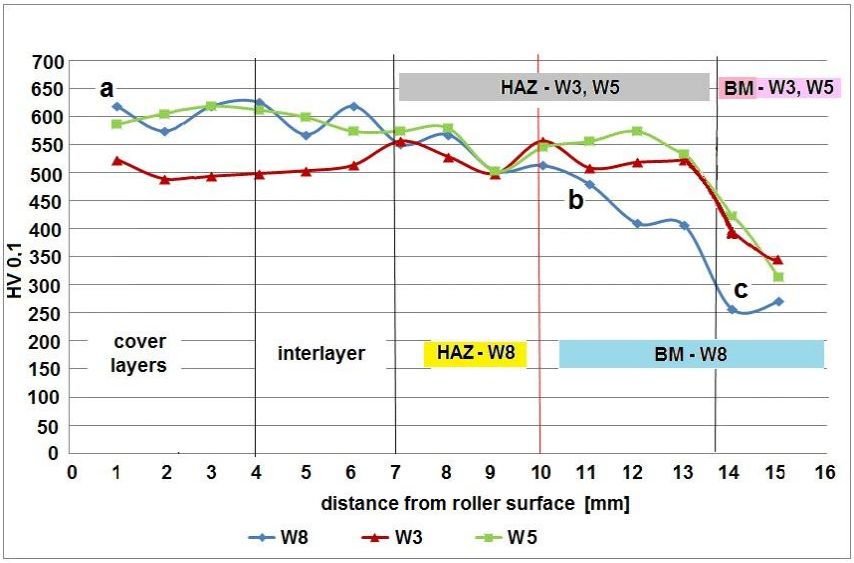


Fig. 3. Microhardness of particular layers of weld clads

At temperature of 400°C weld deposits were loaded up to 20 thermal cycles and there were still not observed breach of integrity, but when samples were thermally loaded at 900°C, after the 5th cycle thermal cracking of W5 weld deposit occurred, Fig. 4. In weld deposit W3 and W8 cracks on test specimens were not recorded.

W 3 W 5 W 8

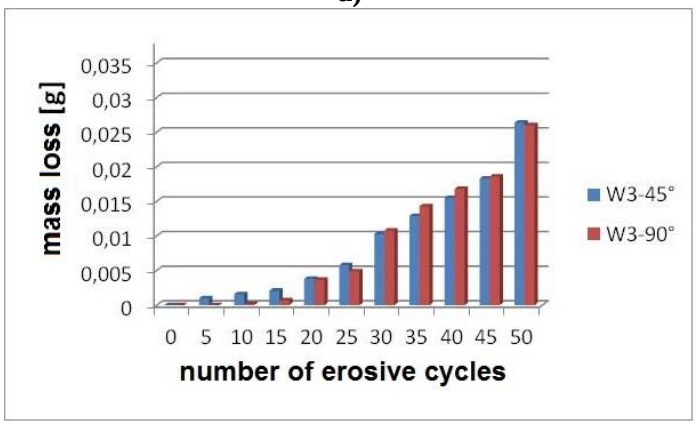
Fig. 4. Cracks on test samples due to thermal cyclic loading after thermal cycles at 900°C

The structural analysis of the test samples after different stages of thermal stress points to an increased incidence of carbide particles and grain refinement after the thermal load in the cladding metal. HAZ showed no significant change in microstructure.

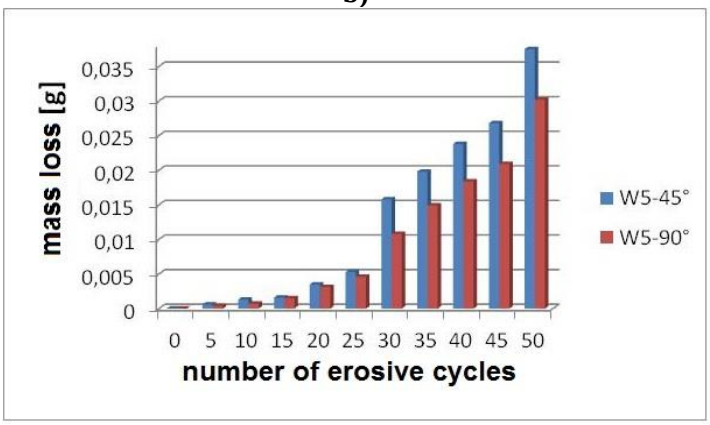
Mass loss of weld deposits at two impact angles of abrasive is presented in Fig. 5.

Weld deposits W5 and W8 showed greater mass loss after 50 erosive cycles compared to W3. Forging effect of incident grains caused exhaustion of weld clad plasticity. It led to fatigue failure of surface of the weld deposit and subsequently release microparticles of material from the surface. Weld deposits W5 showed greater mass loss at blasting angle 45°, which can be caused by chemical composition of the weld, especially lower Cr content compared to other evaluated weld clads. Effect of the blasting angle on wear resistance of weld deposit W3 had not significant impact.

a)



b)



c)

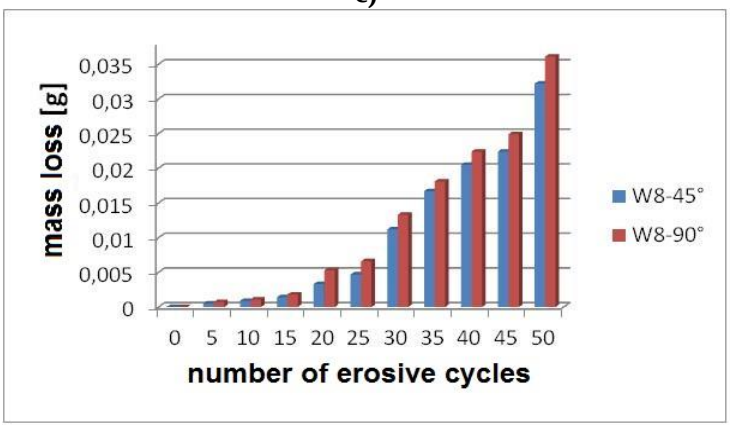


Fig. 5. Mass loss of weld deposits a) W3, b) W5 and c) W8 at two impact angles

4. CONCLUSIONS

Based on the realized experimental work can be concluded, that the highest value of microhardness before thermal cycles were found in the cover layer of weld deposit made of welding wire W8-WLDC 8, 620 HV 0.1. This value is consistent with the chemical composition of the welding wire used, especially with the highest carbon content among the assessed filler materials and also with high chromium content.

Based on the evaluation of results obtained by simulation of degradation phenomena affecting the continuous casting rollers during their operation, newly developed welding wire W8-WLDC 8 can be designated as the best material for renovation of continuous steel casting rollers.

Acknowledgment

This work was supported by scientific grant agency of the Ministry of Education of the Slovak Republic VEGA No.1/0424/17.

REFERENCES

Blaškovitš, P., & Čomaj, M. (2006). *Renovation by welding and thermal spraying*. Bratislava: STU.

Brezinová, J., Viňáš, J., Maruschak, P. O., Guzanová, A., Draganovská, D., & Vrabeľ, M. (2017). *Sustainable renovation within metallurgical production*. Lüdenscheid: Verlag.

Houldcroft, P. T. (2014). *Submerged-Arc Welding*, Elsevier.

Kotus, M., Andrássyová, Z., Cico, P., Fries, J., & Hrabe, P. (2011). Analysis of wear resistent weld materials in laboratory conditions. *Research in Agricultural Engineering, 57*, 74–78.

Lekhov, O. S., Mikhalev, A. V., Bilalov, D. K., & Shevelev, M. M. (2017). Cyclic-deformation system for the reduction of continuous-cast slabs. *Steel in Translation 4(47)*, 225–228. doi: 10.3103/S0967091217040064

Lorincová, D. (2012). *Analysis of quality of renovation layers from aspect of degradation phenomena* (doctoral dissertation). Technical University of Kosice: Kosice.

Mikuš, R., Szabó, V., Drlička, R., & Bakajová, J. (2014). Hardness and microstructure analysis of deposit material based on Fe-Cr-Ni. *Advanced Materials Research, 1059*, 35–42. doi: 10.4028/www.scientific.net/AMR.1059.35

Nadooshan, A. A., Saeedi, A., Rasooli, M., Izadi, J., & Poursina, M. (2009). Calculation of possibility for parameter design in changing central cooling system to revolver cooling system for continuous casting rollers. *World Academy of Science, Engineering and Technology, 49*, 252–257.

Olson, D. L., Dixon, R. D., & Liby, A. L. (2012). *Welding: Theory and Practice, Materials Processing: Theory and Practices.* Elsevier.

Pauliček, T., Kotus, M., Daňko, M., & Žúbor, P. (2013). Resistance of hard-facing deposit created by laser surfacing technology. *Advanced Materials Research, 801*, 117–122. doi:10.4028/www.scientific.net/AMR.801.117

Toyserkani, E., Khajepour, A., & Corbin, S. F. (2014). *Laser Cladding*. CRC Press.

Viňáš, J., Brezinová, J., Guzanová, A., & Lorincová, D. (2011). Properties of renovation layers applied by submerged arc welding. *Chemické listy, 105(17),* 858–859.

Viňáš, J., Greš, M., & Vaško, T. (2016). Cladding of wear-resistant layers in metallurgy and engineering. *Materials Science Forum, 862*, 41–48.

Viňáš, J., Brezinová, J., Guzanová, A., & Svetlík, J. (2013). Degradation of renovation layers deposited on continuous steel casting rollers by submerged arc welding. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 227(12)*, 1841–1848.

Wu, Y., Xiang, Q., He, Y., Chen, H., Hu, B. (2015). Main technical route of renovation heat-treatment of aero-engine turbine blades. *Jinshu Rechuli/Heat Treatment of Metals, 11(40)*, 176–178.

Applied Computer Science, vol. 14, no. 2, pp. 69–81
doi:10.23743/acs-2018-14

Submitted: 2018-05-20
Revised: 2018-06-12
Accepted: 2018-06-22

diesel engine, opposed piston engine, two-stroke engine, injection timing

Paweł KARPIŃSKI[*]

THE INFLUENCE OF THE INJECTION TIMING ON THE PERFORMANCE OF TWO-STROKE OPPOSED-PISTON DIESEL ENGINE

Abstract

The performance of the engine strongly depends on the parameters of the combustion process. In compression ignition engines, the fuel injection timing has a significant influence on this process. The moment of its occurrence and its duration should be chosen so that the maximum pressure value occurs several degrees after TDC. In order to analyze the effect of the fuel injection timing on the performance of the tested two-stroke opposed-piston diesel engine, a zero-dimensional model was developed in the AVL BOOST program. Next, a series of simulations were performed based on the defined calculation points for maximum continuous power, which resulted in power, specific fuel consumption and mean in-cylinder pressure. Finally, the engine map was made as a function of the start of combustion angle.

1. INTRODUCTION

Two-stroke opposed piston engines have a number of advantages over four-stroke engines. Due to the lack of a timing system, there are less thermal losses. The lack of timing system translates into greater reliability, cheaper operation and simpler service. In addition, in this type of engines one stroke per revolution of the crankshaft is produced, and thus more power is obtained compared to a four-stroke engine with the same capacity. The advantage of opposed piston engines is good balance, the possibility of using different fuels and simple

[*] Department of Thermodynamics, Fluid Mechanics and Aviation Propulsion Systems, Faculty of Mechanical Engineering, Lublin University of Technology, Nadbystrzycka 36, 20-618 Lublin, Poland, pawel.karpinski@pollub.edu.pl

injectors (Pirault, 2010). Regner et al. (2011) and Regner, Koszewnik, and Venugopal (2014) presented a number of advantages of the proposed two-stroke opposed piston diesel engine. However, four-stroke engines are better in terms of the emission of toxic substances as well as the heat load of the components of the crankshaft and the wear of the cylinder liner and the rings.

So far, these types of engines have been used on ships, in military vehicles and special machines. The wide range of high-quality catalysts and lubricants currently available on the market creates new opportunities for the use of two-stroke opposed piston engines, especially in land and air transport.

Many tests of this type of engines have been performed. An example is work written by Ma et al. (2015), in which the combustion and heat dissipation process in an opposed piston diesel engine was studied. Abani et al. (2017) presented a number of assumptions and design solutions applied in the design of a two-stroke opposed piston diesel engine that achieves 55% thermal efficiency. Another example is the study of a two-stroke opposed piston diesel engine for propulsion of light aircraft performed in the work written by Cantore, Mattarelli, and Rinaldini (2014).

In self-ignition engines, the injection advance angle is a parameter significantly affecting the process of formation of a fuel-air mixture (propagation of fuel drops after combustion, evaporation and mixing with air) and the combustion process (speed and amount of heat produced). Its value depends on the moment when the pressure in the cylinder increases dramatically in relation to the piston position in the TDC. In order to obtain the highest efficiency, this angle should be selected so that the maximum pressure occurs several degrees CA after the piston TDC.

The injection advance angle is clearly linked to the angle of the start of combustion. An earlier fuel injection results in an earlier formation of a fuel-air mixture, and therefore an earlier initiation of the combustion process. In the work written by Katarasnik, Trenc, and Opresnik (2006), the criterion for determining the start of combustion based on the analysis of the third derivative of the mean pressure signal in the cylinder as a function of the angle of rotation of the crankshaft has been presented.

Agarwal et al. (2013) conducted experimental studies of a single-cylinder diesel engine, in which the impact of injection advance and injection pressure on selected engine performance parameters (including cylinder pressure, rate of heat release, specific fuel consumption, exhaust emissions) were analyzed.

In recent years, many works have been published devoted to the study of dual-fueled engines. The examples are works written by Sayin, and Canakci (2009) and Sayin, Ilhan, Canakci, and Gumus (2009), in which the influence of the injection advance angle on the performance of a single-cylinder diesel engine fueled by a mixture of diesel with ethanol and diesel with methanol, respectively. The tests have shown that increasing the injection advance angle in the engine running on the test mixture significantly reduces the emission of toxic substances (carbon monoxide and unburnt hydrocarbons).

Research was also performed on the influence of ignition timing on the performance of CI engines fueled by unconventional types of oils: used edible oil (Bari, Yu & Lim, 2004) plastic oil (Mani & Nagarajan, 2009) and bio-diesel (Ganapathy, Gakkhar & Murugesan, 2011).

Another important parameter correlated with the angle of advance of injection is the ignition delay. This is the period between the moment the fuel appears on the injector and the first self-ignition spots appear (Heywood, 1988). There are many studies devoted to the study of this parameter in diesel engines.

An example is work written by Assanis, Filipi, Fiveland, and Syrimis (2003), which shows the correlation of the ignition delay with physical and chemical processes based on pressure, temperature and stoichiometry.

Research performed by Kobori, Kamimoto, and Aradi (2000) showed that the ignition delay can be shortened by increasing the temperature and pressure inside the cylinder. In addition, increasing the injection pressure, the number of cetane fuel or reducing the diameter of the hole in the injector results in a shorter ignition delay.

The change of fuel type affects the ignition delay, as shown in the research performed in the publication written by Saho, and Das (2009). The study investigated the use of selected vegetable oils as a fuel in diesel engines and their effect on the combustion process.

As part of this work, a model of a two-stroke opposed piston diesel engine was presented. Based on the developed model, simulation calculations were made to investigate the effect of the injection advance angle on the performance of the designed opposed piston diesel engine.

2. ENGINE MODEL

In order to perform simulation tests on the influence of the injection advance angle on the performance of the designed engine, a zero-dimensional model was developed using the AVL BOOST program (Fig. 1).

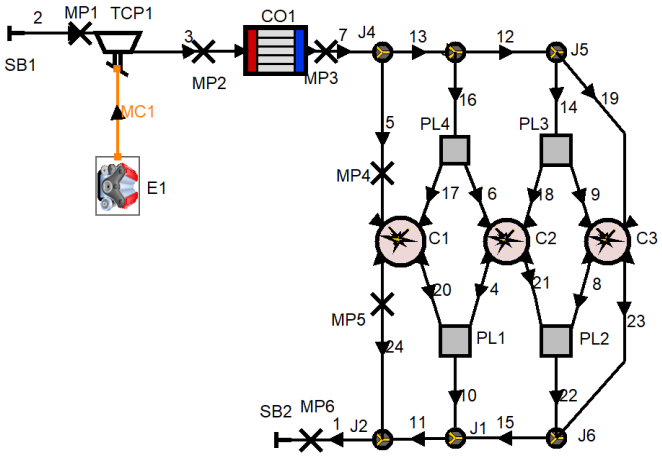


Fig. 1. Model of tested engine created in the AVL BOOST program

In the zero-dimensional model, the flame propagation method is not included, and all thermodynamic parameters are averaged in the working volume of the cylinder. This model is based on the first law of thermodynamics. The independent variable is the time, and the mass burning rate is assumed based on literature data (Heywood, 1988).

Figure 2 shows the basic settings for the calculation cycle defined in the AVL BOOST program.

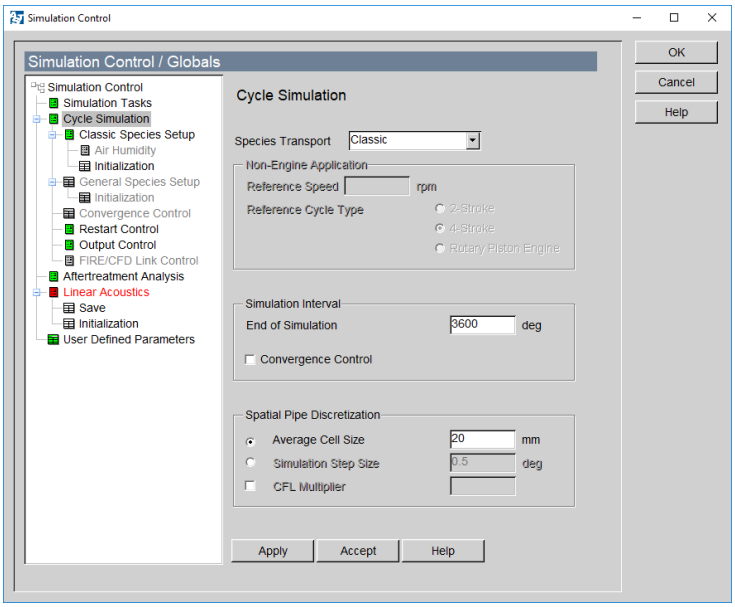


Fig. 2. Basic settings of the calculation cycle defined in AVL BOOST program

The Viebe function was used to describe the combustion process, which describes the rate of heat release (*ROHR*) as a function of the crank angle. After its integration, a part of the burnt mass (mass fraction burned, *MFB*) is obtained as a function of the crank angle. The Viebe function depends on four basic parameters: start of combustion (*SoC*), combustion duration (*CD*), curve shape parameter m and parameter a, depending on the burned part of the fuel injected (Stiesch, 2003). This function is the ratio of the heat released after a given angle of rotation of the crankshaft $Q(\alpha)$ and the total amount of heat released at the end of combustion Q_c. It has the form (1):

$$\frac{Q(\alpha)}{Q_c} = 1 - \exp\left(- a \left(\frac{\alpha - SoC}{CD} \right)^{m+1} \right) \tag{1}$$

where: *SoC* – start of combustion,
CD – combustion duration,
m – shape parameter,
a – parameter depending on the burned part of the fuel injected.

In the developed engine model, the following parameters of the combustion model were adopted:

- start of combustion *SoC*, from –10° to +10° relative to TDC, every 2°,
- combustion duration *CD* = 40°,
- curve shape parameter *m* = 0.7,
- parameter *a* = 6.9.

Fig. 3 shows the heat release coefficient (*ROHR*) and the mass fraction burned (*MFB*) as a function of the crank angle determined in the AVL BOOST program based on the defined parameters of the combustion model.

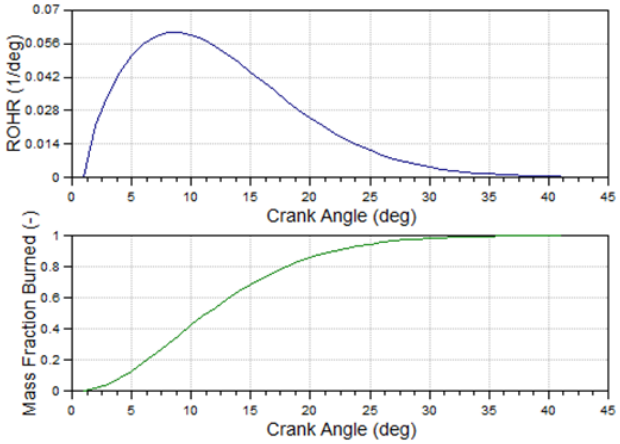


Fig. 3. ROHR and MFB as a function of the crank angle

In the model it was also necessary to define the model of heat exchange through the walls. The Woschni 1978 standard model was adopted, based on the Newton's law of cooling, in the form of (2):

$$\frac{dQ_w}{dt} = hA\left(T_w - T_g\right) \tag{2}$$

where: Q_w – heat flow from gas to walls,
h – heat exchange coefficient,
A – surface area,
T_w – average temperature of walls,
T_g – average gas temperature. (Stiesch, 2003)

In the heat exchange model, the surface area of the piston and cylinder was defined based on the design assumptions of the engine. In addition, based on data from work written by Pulkrabek (1997), temperatures were set on their surfaces. In order to take into account the effect of the load turbulence in the cylinder on the heat exchange coefficient h, the swirl ratio SR = 1.5 defined as the speed of the load rotation in relation to the engine speed was defined. Table 1 presents the basic technical parameters of a given engine.

Tab. 1. Basic technical parameters of a tested engine

Engine type	Two-stroke opposed piston diesel engine
Scavenging method	Mechanical compressor Rotrex C30-64 with intercooler
Cylinders numer	3
Piston bore d	65.5 mm
Piston stroke S	72 mm
Compression ratio ε	22:1
Maximum power N_{max}	100 kW (4200 rpm)
Offset O	14°

In order to improve the filling of the cylinders with a fresh charge in the two stroke opposed-piston engines, the difference in the working phases of the crankshafts is used. Crankshaft from the side of the inlet ports get ahead of the shaft from the side of outlet ports. This angle is called offset. For the tested engine model an offset equal to O = 14° was assumed.

The AVL BOOST program does not have the option of directly designing the opposed piston engine, therefore simplification has been applied, consisting in doubling the piston area, resulting in a calculated piston diameter of $d_{calc} = d\sqrt{2}$ (Fig. 4).

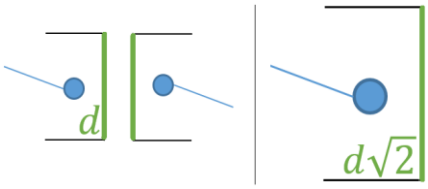


Fig. 4. The method of determining the calculated piston diameter

A number of geometrical parameters have been introduced to the model: the calculated diameter of the piston d_{calc}, the piston stroke S, the compression ratio ε, the length of the connecting rod l.

The model assumes a classic model of transport of chemical compounds (classic species transport). As a fuel, diesel oil with a calorific value of W_u = 42.8 kJ/kg and stoichiometric air-fuel ratio AFR = 14.7 kg of air per kg of fuel was adopted. The firing order was defined as follows: cylinder 1 – 0°, cylinder 2 – 120°, cylinder 3 – 240°. The engine model includes the friction model developed by Patton, Nitschke, and Heywood (1989).

The load was exchanged with a Rotrex C30-64 mechanical compressor driven mechanically from the engine shaft. Operating characteristics of the compressor was introduced to the model based on a map from the manufacturer ("Rotrex Technical Datasheet C30 Range", 2018). In addition, an air-to-air intercooler was used to improve the filling of the engine with a fresh charge behind the compressor. The scavenging model defined as scavenging efficiency as a function of the delivery ratio was introduced based on literature data (Blair, 1996). Figure 5 shows the defined scavenging model in cylinder settings window.

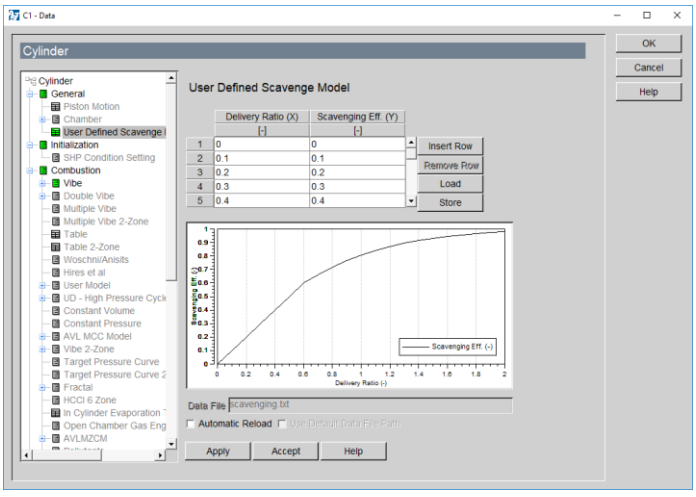
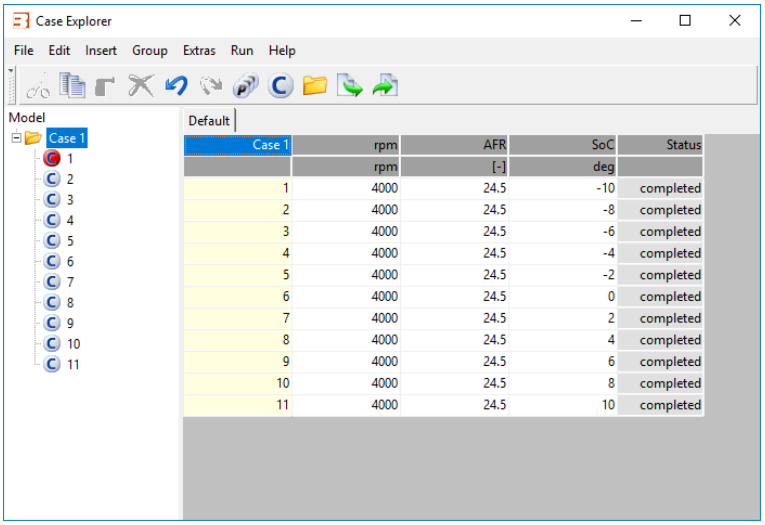


Fig. 5. Scavenging model defined in cylinder settings window in AVL BOOST program

In addition, inlet and outlet ports characteristics were introduced based on the engine design assumptions.

3. RESEARCH PLAN

During the tests, the influence of the injection timing on the performance of the analyzed engine (power, specific fuel consumption, mean cylinder pressure) was examined. The moment of the beginning of combustion depends on the injection timing, therefore the value of the angle at which the start of combustion (*SoC*) occurs is taken as the parameter. The tests were performed for a rotational speed of n = 4000 rpm and an air-fuel ratio equal to AFR = 24.5 kg air / kg fuel. The SoC parameter varied from –10 ° to +10 ° relative to TDC, every 2°. The defined measuring points are shown in Fig. 6.



Case 1	rpm	AFR	SoC	Status
	rpm	[-]	deg	
1	4000	24.5	-10	completed
2	4000	24.5	-8	completed
3	4000	24.5	-6	completed
4	4000	24.5	-4	completed
5	4000	24.5	-2	completed
6	4000	24.5	0	completed
7	4000	24.5	2	completed
8	4000	24.5	4	completed
9	4000	24.5	6	completed
10	4000	24.5	8	completed
11	4000	24.5	10	completed

Fig. 6. Defined measuring points in Case Explorer window in AVL BOOST program

4. RESULTS AND ANALYSIS

As a result of the performed calculations, the effective power N (Fig. 7), the value of specific fuel consumption g_e (Fig. 8) and the mean cylinder pressure as a function of crank angle (Fig. 9) were obtained for defined measurement points. The results of calculations are presented in Table 2.

Tab. 2. Calculation results

SoC (°)	*N* (kW)	g_e (g/kWh)	p_{max} (MPa)	$\alpha(p_{max})$ (°)
–10	84.85	222.05	17.62	5
–8	85.83	219.35	16.70	6
–6	86.31	217.91	15.77	7
–4	86.29	217.56	14.73	8
–2	85.52	218.48	13.65	9
0	85.19	219.74	12.62	11
2	83.85	222.49	11.54	12
4	82.41	226.06	10.54	14
6	80.45	230.66	9.54	15
8	78.64	235.82	9.39	0
10	76.66	241.85	9.39	0

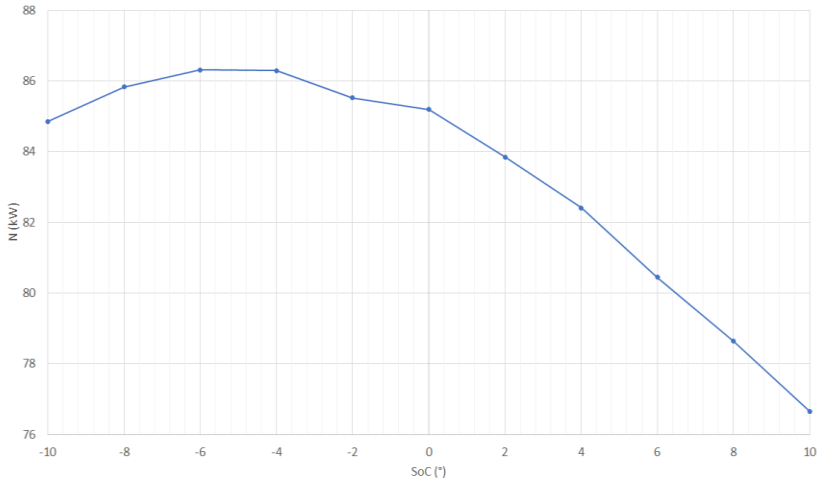


Fig. 7. Effective power as a function of SoC

The effective power has the highest value equal to 86.31 kW and 86.29 kW respectively for *SoC* = –6° and *SoC* = –4° angles. As the *SoC* angle increases or decreases, the effective power decreases. for an angle greater than *SoC* = 0°, the decrease is rapid.

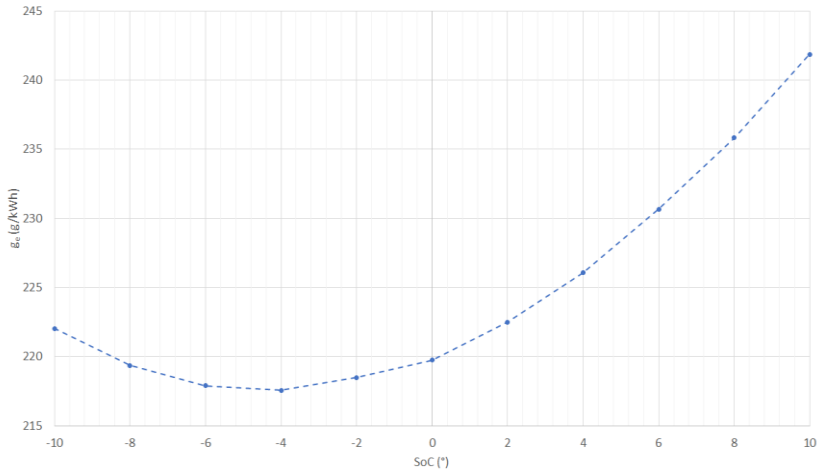


Fig. 8. Specific fuel consumption as a function of SoC [source: own study]

The specific fuel consumption has the smallest value equal to 217.91 g/kWh and 217.56 g/kWh for *SoC* = –6 ° and *SoC* = –4 ° respectively. With the increase or decrease of the *SoC* angle, the specific fuel consumption increases.

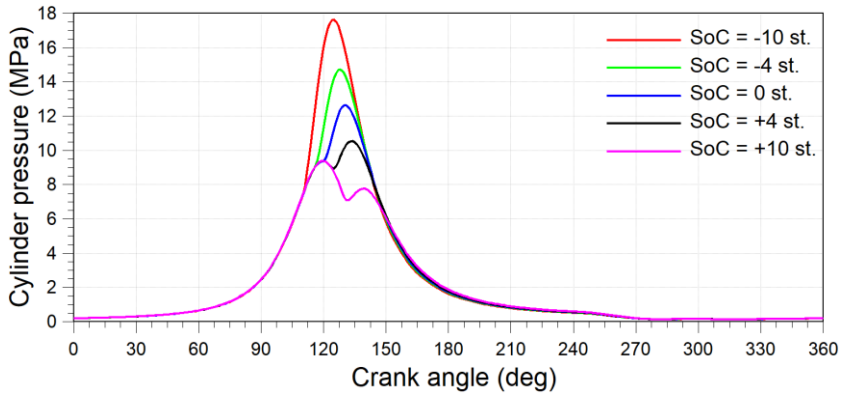


Fig. 9. The mean pressure in 2nd cylinder as a function of the crank angle for selected values of the SoC angle

As the value of the *SoC* angle decreases, the maximum pressure value p_{max} increases and the area under the plot increases (the indicated work increases). For positive values of the *SoC* angle, the indicated work decreases, whereas for the large values of this angle (10°) the second peak of pressure appears, which indicates an incorrect combustion process. The maximum pressure for the angle of *SoC* = –4° is for the crank angle *CA* = 128°, and therefore 8° for the TDC of the piston.

On the basis of the cylinder pressure as a function of crank angle, the maximum pressure value p_{max} and the angle of occurrence of the maximum pressure after the TDC of the piston $\alpha(p_{max})$ were read from the defined measurement points (Fig. 10). For *SoC* = 8° and *SoC* = 10°, the value of $\alpha(p_{max})$ was 0, because in these cases there were two pressure peaks, the highest of which corresponded to the compression pressure in the TDC of the piston.

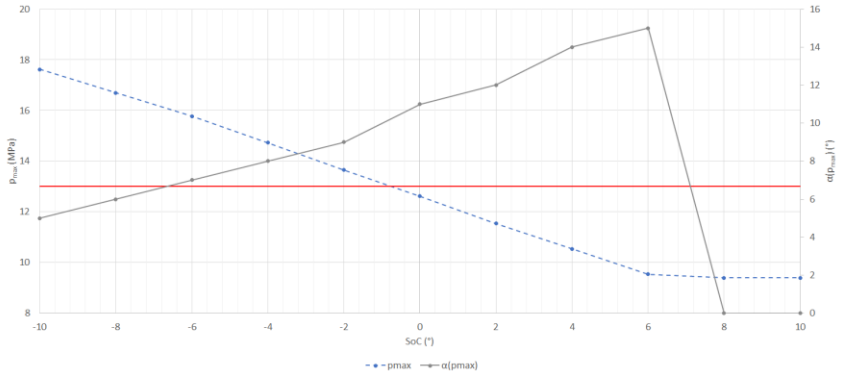


Fig. 10. Maximum cylinder pressure and the angle of occurrence of the maximum pressure after the TDC as a function of SoC angle

For reasons of strength of the crank and piston system, the maximum cylinder pressure should not exceed a certain limit value. Based on the design assumptions of the engine design, its value was assumed at 13 MPa. The value of the maximum pressure in the cylinder of the tested engine decreases with the increase of the value of the start of combustion angle (Fig. 10). It follows that the *SoC* angle for the engine under test should not be less than 1°, so as not to exceed the set maximum pressure limit in the cylinder. As the value of the *SoC* angle increases, the angle of occurrence of the maximum pressure after TDC of the piston $\alpha(p_{max})$ increases.

5. CONCLUSIONS

The fuel injection process depends on the injection timing and the combustion process. These processes have a significant impact on the engine performance. In addition, the injection timing translates into the moment of the start of combustion SoC, which determines the moment of rapid pressure increase in the cylinder. The calculations made it possible to analyze how the start of combustion SoC (and thus indirectly the injection timing) affects the effective power N, the specific fuel consumption g_e and the mean cylinder pressure p as a function of the crank angle. In addition, the limit value of *SoC* angle for the tested engine was specified.

For an optimum *SoC* angle value, the effective power generated reaches its maximum, while the specific fuel consumption assumes the minimum value. The optimum *SoC* angle for the designed engine ranges from –6° to –4°. For the *SoC* angle –4°, the effective power is approximately 86.29 kW, and the specific fuel consumption is 217.56 g/kWh. However, as the *SoC* angle decreases, the maximum pressure in the cylinder increases. This pressure should not exceed the permissible value resulting from the strength properties of the crank-piston system. It follows that the *SoC* angle for the designed engine should not be less than –1°.

Acknowledgment

This work has been realized in the cooperation with The Construction Office of WSK "PZL-KALISZ" S.A." and is part of Grant Agreement No. POIR.01.02.00-00-0002/15 financed by the Polish National Centre for Research and Development.

REFERENCES

Abani, N., Chiang, M., Thomas, I., Nagar, N., Zermeno, R., & Regner, G. (2017). Developing a 55+ BTE Commercial Heavy-Duty Opposed-Piston Engine without a Waste Heat Recovery System. In *Heavy-Duty-, On-und Off-Highway-Motoren* 2016 (pp. 292–310). Wiesbaden: Springer Vieweg. doi:10.1007/978-3-658-19012-5_17

Agarwal, A. K., Srivastava, D. K., Dhar, A., Maurya, R. K., Shukla, P. C., & Singh, A. P. (2013). Effect of fuel injection timing and pressure on combustion, emissions and performance characteristics of a single cylinder diesel engine. *Fuel*, *111*, 374–383. doi:10.1016/j.fuel.2013.03.016

Assanis, D. N., Filipi, Z. S., Fiveland, S. B., & Syrimis, M. (2003). A predictive ignition delay correlation under steady-state and transient operation of a direct injection diesel engine. *Journal of Engineering for Gas Turbines and Power*, *125*(2), 450–457. doi:10.1115/1.1563238

Bari, S., Yu, C. W., & Lim, T. H. (2004). Effect of fuel injection timing with waste cooking oil as a fuel in a direct injection diesel engine. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *218*(1), 93–104. doi:10.1243/095440704322829209

Blair G.P. (1996). *Design and simulation of two-stroke engines.* Warrendale: Society of Automotive Engineers Inc.

Cantore, G., Mattarelli, E., & Rinaldini, C. A. (2014). A new design concept for 2-Stroke aircraft diesel engines. *Energy Procedia*, *45*, 739-748. doi:10.1016/j.egypro.2014.01.079

Ganapathy, T., Gakkhar, R. P., & Murugesan, K. (2011). Influence of injection timing on performance, combustion and emission characteristics of Jatropha biodiesel engine. *Applied energy*, *88*(12), 4376-4386. doi:10.1016/j.apenergy.2011.05.016

Heywood, J. B. (1988). *Internal combustion engine fundamentals.* New York: McGraw-Hill Education.

Katrašnik, T., Trenc, F., & Oprešnik, S. R. (2006). A new criterion to determine the start of combustion in diesel engines. *Journal of engineering for gas turbines and power*, *128*(4), 928-933. doi:10.1115/1.2179471

Kobori, S., Kamimoto, T., & Aradi, A. A. (2000). A study of ignition delay of diesel fuel sprays. *International Journal of Engine Research*, *1*(1), 29-39. doi:10.1243/1468087001545245

Ma, F., Zhao, C., Zhang, F., Zhao, Z., Zhang, Z., Xie, Z., & Wang, H. (2015). An experimental investigation on the combustion and heat release characteristics of an opposed-piston folded-cranktrain diesel engine. *Energies*, *8*(7), 6365-6381. doi:10.3390/en8076365

Mani, M., & Nagarajan, G. (2009). Influence of injection timing on performance, emission and combustion characteristics of a DI diesel engine running on waste plastic oil. *Energy*, *34*(10), 1617-1623. doi:10.1016/j.energy.2009.07.010

Patton, K. J., Nitschke, R. G., & Heywood, J. B. (1989). *Development and Evaluation of a Friction Model for Spark-Ignition Engines.* SAE Technical Paper, 890836. doi:10.4271/890836

Pirault, J.-P., & Flint M. (2010). *Opposed piston engines: evolution, use, and future applications.* SAE International.

Pulkrabek, W. W. (1997). *Engineering fundamentals of the internal combustion engine* (No. 621.43 P8). New York: Pearson.

Regner, G., Herold, R. E., Wahl, M. H., Dion, E., Redon, F., Johnson, D., Callahan, B. J., & McIntyre, S. (2014). *The achates power opposed-piston two-stroke engine: performance and emissions results in a medium-duty application* (2011-01-2221, pp. 2726–2735). *SAE International*. doi:10.4271/2011-01-2221

Regner, G., Koszewnik, J., & Venugopal, R. (2014). *Optimizing combustion in an opposed-piston, two-stroke (OP2S) diesel engine*. San Diego: Achates Power, Inc.

Rotrex Technical Datasheet C30 Range. (n.d.). Retrieved April 17, 2018, from Rotrex website, http://www.rotrex.com/Home/Technology/Product_Technical_Data

Sahoo, P. K., & Das, L. M. (2009). Combustion analysis of Jatropha, Karanja and Polanga based biodiesel as fuel in a diesel engine. *Fuel*, *88*(6), 994-999. doi:10.1016/j.fuel.2008.11.012

Sayin, C., & Canakci, M. (2009). Effects of injection timing on the engine performance and exhaust emissions of a dual-fuel diesel engine. *Energy conversion and management*, *50*(1), 203-213. doi:10.1016/j.enconman.2008.06.007

Sayin, C., Ilhan, M., Canakci, M., & Gumus, M. (2009). Effect of injection timing on the exhaust emissions of a diesel engine using diesel–methanol blends. *Renewable Energy*, *34*(5), 1261–1269. doi: 10.1016/j.renene.2008.10.010

Stiesch, G. (2003). *Modeling Engine Spray and Combustion Processes*. Berlin: Springer-Verlag.

Applied Computer Science, vol. 14, no. 2, pp. 82–95
doi:10.23743/acs-2018-15

Submitted: 2018-05-23
Revised: 2018-06-06
Accepted: 2018-06-28

High Speed Machining, milling, thin-walled construction

Paweł BAŁON[*], Edward REJMAN[**], Robert SMUSZ[**], Janusz SZOSTAK[***], Bartłomiej KIEŁBASA[*]

HIGH SPEED MILLING IN THIN-WALLED AIRCRAFT STRUCTURES

Abstract

Aircraft structures are designed to mainly consist of integral elements which have been produced by welding or riveting of component parts in technologies utilized earlier in the production process. Parts such as ribs, longitudinals, girders, frames, coverages of fuselage and wings can all be categorized as integral elements. These parts are assembled into larger assemblies after milling. The main aim of the utilized treatments, besides ensuring the functional criterion, is obtaining the best ratio of strength to construction weight. Using high milling speeds enables economical manufacturing of integral components by reducing machining time, but it also improves the quality of the machined surface. It is caused by the fact that cutting forces are significantly lower for high cutting speeds than for standard machining techniques.

1. INTRODUCTION

Aeroplane structures are exposed to many loads during their working lifespan. Every particular action made during a flight is composed of a series of air movements which generate various aeroplane loads. One rigorous requirement which modern aeroplane structures must meet is that they be of high durability and reliability. This requirement involves taking many restrictions into account during the aeroplane design process. The most important factor is the structure's

[*] SZEL-TECH Szeliga Grzegorz, Wojska Polskiego Street 3, 39-300 Mielec, Poland, p.balon@szel-tech.pl, bartek.kielbasa@gmail.com
[**] Rzeszów University of Technology, Powstańców Warszawy Avenue 9, 35-959 Rzeszów, Poland, erejman@prz.edu.pl, robsmusz@prz.edu.pl,
[***] AGH University of Science and Technology, Mickiewicza Avenue 30-B4, 30-059 Kraków, Poland, szostak@imir.agh.edu.pl

overall mass, which has a crucial impact on both utility properties and cost-effectiveness. This makes aeroplanes one of the most compound results of modern technology.

Almost every currently produced aeroplane structure, or to be more precise, aeroplane core structure, is manufactured as a thin-walled composition which perfectly meets the requirements concerning the structure's mass minimisation. Some compositions have coverages reinforced with longitudinal and transverse elements which provide the required stiffness and strength of the whole composition. Local buckling of the coverage is allowed under load conditions, but exceeding the critical load levels of the elements such as frames or longitudinals virtually guarantees the structure's destruction.

The methodology mentioned above forces constant improvements in both design methods and aeroplanes' structural solutions. Development of material science and technological processes allows for the fabrication of geometrically compound integral structures. These structures enable the utilization of material characteristics in a more reasonable way, and also a significant increase in their core structure mechanical properties. The most important advantage of using the integral structures is the costs saving from the reduction or elimination of assembly operations.

Close-ribbed covering elements increase strength parameters and reduce the weight of the core structure. It is possible to achieve a structure with significantly higher critical loads by reducing covering thickness and simultaneously, by implementing adequately close-packed and stiffened longitudinal elements. As a result, more beneficial gradients and stress distribution will also be obtained, which directly increase the structure's fatigue life (Lundblad, 2002).

The machining of thin-walled elements generates a lot of technological issues related to deformation and elastic and plastic displacements of the workpiece. Due to displacements of the milled workpiece, vibrations can occur, and thus, geometric errors may surface in the structure of the workpiece. Furthermore, plastic deformation can also cause shape problems and be a source of internal stresses in the surface layer, which are very difficult to remove and lead to deformation of the workpiece after machining. Consequently, this leads to an increase in the manufacturing costs of machining operations, especially of thin-walled elements, due to shortages and increased manufacturing time.

It is recommended that multiple methods for minimizing machining errors be utilized to improve the quality of thin-walled elements, such as:

- optimization of the machining strategy,
- increasing the cutting speed v_s,
- optimization of cutting parameters, especially feed per tooth f_z and the thickness of the cut layer a_e due to the minimization of the cutting force component perpendicular to the surface of the milled wall (Mativenga & Hon, 2005).

In the aviation industry, the geometry of an aircraft is designed according to the laws of aerodynamics. Then, the particular aircraft assemblies, subassemblies and parts are designed on the basis of the developed geometry. The designer has a limited area to design a part within, and he/she needs to conform it within the constraints of that area using the best (the highest) strength-to-weight ratio. The strength-to-weight ratio is the basis for selecting material types to be used in the construction of the aircraft. The parts are designed and the appropriate material types (aluminium, titanium, steel, composite) are selected depending on the loads which occur in the aircraft. Full block material, a forging, or a casting is used for blanks. The most commonly used materials for aircraft constructions are composites, aluminium alloys, and titanium alloys. In recent years, high competition in the aviation industry has caused very rapid development of modern manufacturing processing. Currently, a lot of aircraft parts are manufactured ”ready-made” from the full quantity of the material. The integral aircraft parts typically require removal of up to 95% of the material over the course of the production process. In order to realise such large volume machining processes, it is ncessary that highly efficient methods be used so that production can be cost-effective. High Speed Milling technology makes this possible. Moreover, the manufactured parts are homogenoeus and have better physical properties. The lack of riveted joints results in a lighter part structure with a higher strength-to-weight ratio (Adamski, 2010). A comparison of the ”buy-to-fly” ratio (the total weight of the purchased materials to the part mass of the finished aircraft) of some aircraft elements is presented in Table 1.

Tab. 1. A comparison of ”buy-to-fly” ratios for various manufacturing technologies

Manufacturing technology type	„Buy-to–fly„ ratio
Machining from the forged block	30 : 1
Machining from the sections	12 : 1
Die forgings	8 : 1
Laser machining	3 : 1
Form casting	1.4 : 1
Pressure die-casting	1.2 : 1
Additive Manufacturing	1.2 : 1

The results show that the High Speed Milling technology is one of the most material-consuming methods, resulting in a high percentage of material turned into chips. However, the remaining advantages, especially the reduction of the part’s production time resulted in further development of this technology. For instance, the manufacturing time of an F-15 aircraft’s aerodynamic brake has been reduced from about 3 months to 12 hours. An analysis of HSM technology usage in the aviation industry requires consideration of technical and economic factors significantly influencing the application of this method. The most important of these are:

- Market requirements – increasing market competition presents greater challenges. Shortening procurement deadlines and reducing costs makes the HSM an effective tool to meet the demands of tightening procurement deadlines and cost reduction programs.
- Materials – implementing new materials (which are more difficult for machining) increases the need for developing new machining solutions.
- Quality – the requirement of parts and product quality influencens safety and this is caused by competition. Implementing HSM offers many new solutions in this field, for example reducing manual machining.
- Production process – implementing HSM techniques causes a shortening of the production cycle by decreasing the number of operations and simplifying logistics.
- Design and development – HSM technology enables rapid implementation of new products by linking HSM technology and the design process.
- Product complexity – increasing demands relating to the functions of products, especially in the aviation industry, forces their accuracy and shape complexity, which can be ensured with the highly efficient HSM technology.
- Manufacturing capabilities – implementing CAD/CAM systems related to HSM technology increases the manufacturing capabilities of companies making them more competitive on the market.

The main effect of implementing HSM is a significant reduction in machining time. Additional effects from using HSM are:

- Simplified mounting,
- Lower cutting forces,
- More clean tool edges (lack of build-up on edges), which results in fewer deformations during the machining process,
- Smoother surfaces – a finishing machining is not required,
- Less wear on tools.

The CAM systems not only generate a tool path but are also used for its verification and optimization in order to reduce the error amount, possibly even to zero. The nature of the aviation industry is small-volume production which requires flexibility even at the stage of technological production preparation where the integrated CAD and CAM systems are especially useful.

The planning strategy for machining of thin-walled compound structures used in the aviation industry requires taking into account both the principles recommended for CNC machine programming and also the specific features of thin-walled structures with a high ratio of wall height to wall thickness, especially when using HSM. For HSM technology, it is important to choose an appropriate machining strategy, especially for smooth aircraft thin-walled structures such as frames, ribs, etc. In those cases, the typical parameter is the wall's ratio of height-to-wall thickness. Three cases can be distinguished:

- low height-to-thickness ratio < 15:1,
- moderate height-to-thickness ratio < 30:1,
- high height-to-thickness ratio > 30:1.

This ratio reflects the elastic stiffness of the workpiece and, consequently, the deformations which occur during machining. In order to reduce wall bending, the appropriate number of tool passes needs to be used. Moreover, the time of tool-workpiece contact needs to be reduced by using high cutting speeds and a low ratio of cutting depth a_p to cut layer width a_e. The stability of the tool and the machined wall has a great impact in the process. Milling in reverse should be used in the case of less rigid support of the thin-walled object. For a height-to-weight ratio < 15:1, only one side of the wall should be machined in non-overlapping passes. Milling needs to be repeated for the opposite side and the allowance needs to be left for finishing machining (Kuczmaszewski, Pieśko & Zawada-Michałowska, 2017).

2. EXPERIMENTAL PROCEDURE

The aim of the technological and geometric research studies was the aircraft frame, made of an aluminium alloy 7075 (EN 10204) with a chemical composition as follows: Al 89.72%, Cu 0.09%, Mn 0.11%, Mg 1.6%, Zn 0.03%, Si+Fe 2.5%. The mechanical properties of the 7075 T6 alloy measured in the temperature of 21°C for a velocity tensile of 0,001 s^{-1} are as follows: tensile strength R_m = 562 MPa, yield strength $Rp_{0,2}$ = 518 MPa, Young's modulus E = 71,7 GPa, Poisson's ratio ν = 0,33, elongation = 13 %, and hardness = 180 HV1 (Fig. 1). Two 60 x 1190 x 1215 mm blocks with a total weight of 243kg were used as semi-finished products and were delivered in a solution state.

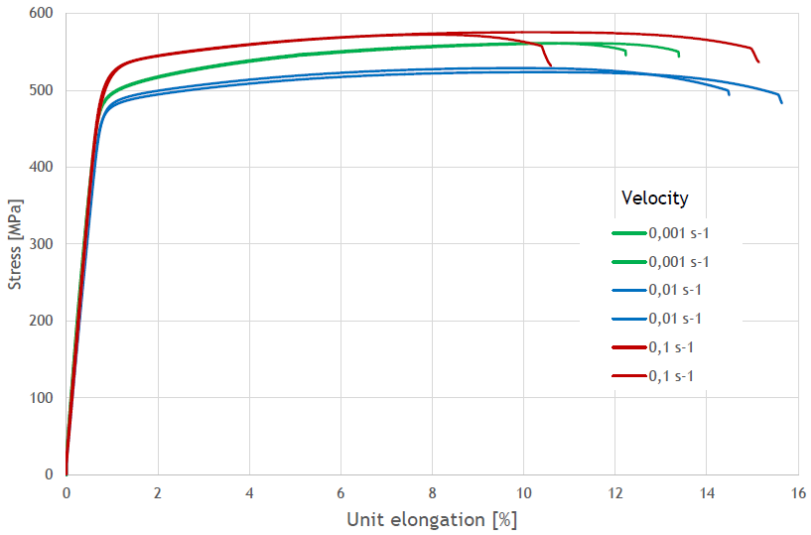


Fig.1 Stress-strain curve for aluminium alloy 7075

The overall dimensions of the workpiece were 1012 x 1354 x 55.8 mm. The walls were 1, 2 and 3 mm thick. The largest free flat surfaces (without ribs) were 393 x 385.4 x 1 mm. The ratio of the ribs height to its thickness h/g was in the range of 10–12.9. The passes radiuses between the walls and the ribs were 2 mm. The frame was machined by the two DMU 75 machines (roughing). The machining was performed simultaneously at the two machines using different manufacturing technologies. The roughing was done with a spindle rotational speed of n = 15000 rpm and this was intended to prepare the machining datum surfaces for subsequent operations. The workpiece was mounted to the machine table by pressure clamps.

The machining was performed in two phases: shaping and finishing. Two workpieces were machined simultaneously on the two machines, while the symmetrical and asymmetrical machining of the pockets were used. In the first case, the frame pockets were oriented symetrically to the workpiece axis and they were machined in an alternate way. For asymmetrical machining, the pockets were milled alternately in relation to the symmetrical axis of the frame. For such treatment, integral deformations of the frame were estimated after the machining process.

Manufacturing of the double-sided frame pockets required preparing the machining bases and the bases needed to set and mount the workpiece on the milling machine. Therefore, a flange with cylindrical holes was made from the semi-finished material. The flange was a machining datum surface and the holes enabled mounting of the workpiece on the machine table (Fig. 2). Those bases were removed in the final frame machining operation after being done ”on the ready”.

According to the assumptions of the construction documentation, the linear dimensions tolerance of the frame is determined by the norm BN-85/3813-79 – Deviations of non-tolerated dimensions, shape and location for aviation products. For the frame walls, the tolerances are:
g = 1–3mm, the deviations are: 0; 0.25 mm
For the rib height:
h = 20–25.8 mm, the deviations are: 0; –0.52 mm
For the overall dimensions:
1012 x 1354 mm, the dimensions are: 0; –2.6 mm

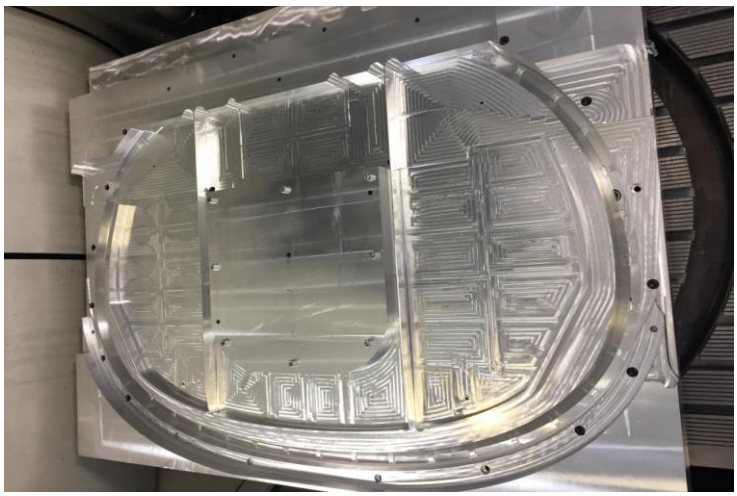


Fig. 2. The frame mounting by technological bases and additional holes during the pocket machining

Mechanical machining of the frame was performed in two stages: pre-machining and finishing. A number of passes in all cases was determined by the wall dimensions and the axial cutting depth. The basic requirement in the machining process was to ensure as little deformation of the ribs as possible and also ensuring appropriate surface roughness during the finishing machining. In the machined frame, 2 mm finishing allowances were used for large face surfaces and about 0.2 mm on the ribs walls with a thickness of 1–3 mm. In order to reduce the wall deformation, the contact time between the tool and the workpiece had to be shortened. It was obtained by using high speed cutting (high tool rotational speeds). The subsequent treatment which ensured high machining precision of smooth frame walls (the height-to-thickness ratio was approximately 13) was to maintain a proper machining strategy. The most effective strategy was alternate pocket machining with the change of machining sides and with machining to the same levels in each pocket. During machining of thin walls it is recommended to mill in reverse.

For the most reasonable cutting parameters, shank cutters with sintered carbide pads were used and the variable depths and widths of the milled layer were also used. The parameters are presented in Table 2. The best results of the cutting process efficiency were obtained using the Φ16 mm cutter and the lowest surface roughness and waving were obtained using the Φ8mm cutter.

Tab. 2. The cutting parameters used during the frame machining

No.	Cutter diameter Φ mm	Cutting speed v_c m/min	Cutting depth a_p mm	Cut layer width a_e mm	Feed rate f_z mm/tooth
1.	6	250	1	0.05–0.075	0.03
2.	8	300	2	0.05–0.075	0.055
3.	16	350	5	0.05–0.075	0.07

3. RESULTS AND DISCUSSION

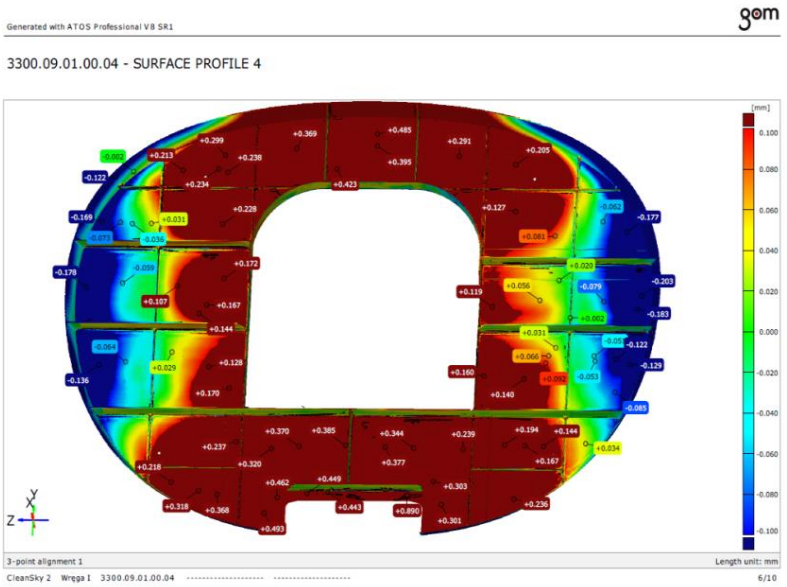
It can be stated that the implemented technology can be used for thin-walled structures manufacturing where the chips mass can be up to 95% of the initial material. It should be emphasized that the frame weight is 3.85 kg. As a result of the machining, the frame compatible with the structural documentation was obtained (Fig. 3).

The weight of the finished frame was 3.15 kg and the weight of the initial material was 243 kg, so the product weight was 1.3% of the initial material. 98.7% of the semi-finished material was changed into chips. Therefore, the ”buy-to-fly” ratio (weight of the initial material to the weight of the final product) was 77.1. The maximum contact time between a cutter and a workpiece during the milling process was: a) $t = 3{,}75 * 10^{-3} s$; b) $t = 6{,}38 * 10^{-4} s$.

Fig. 3. The finished frame after mechanical machining (shaping)

The frame was optically measured by the GOM scanner in order to verify its shape, dimensions, and spatial displacements (Fig. 4, Fig. 5). This research allowed to determine linear dimensions of the particular frame elements and to create digital and contour maps showing the frame stereometry. Those maps mostly allowed evaluation of the deformation potential of the large frame surface. The deformations can be caused by internal stresses generated in the structure during mechanical processing (Shih &Yang, 1993; Shet & Deng, 2003). The dimensions analysis shows that the deviations of the linear dimensions fall within the predicted tolerances. The block is mainly deformed towards the shorter axis of symmetry and the displacements are symmetrical towards the longer axis of symmetry. The highest relative displacement values are 0.57 mm. The GOM scanner ensures the precise measurements with a margin of error of +/– 0.25 mm. However, this method shows in a reliable way what deformations should be expected in the machining of products of a similar structure.

a)



b)

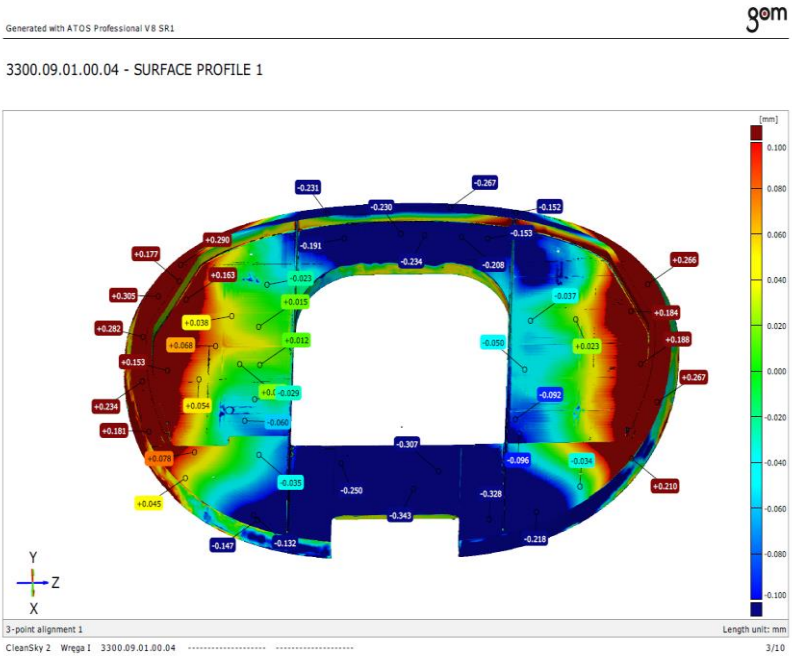
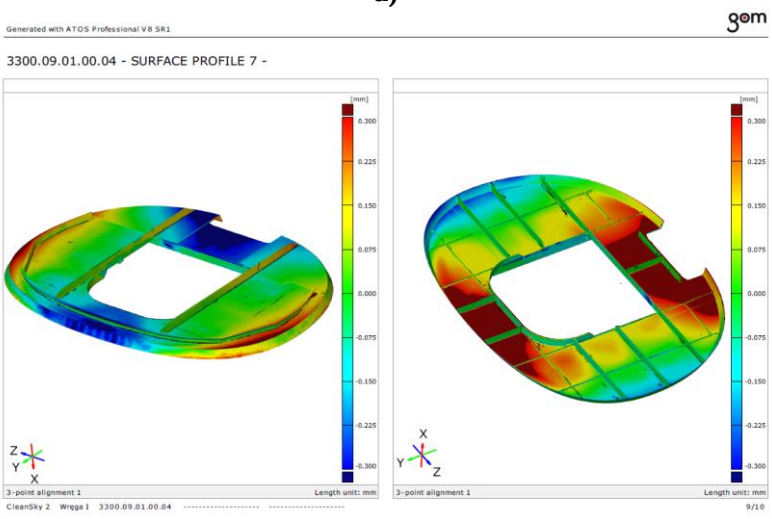


Fig. 4. The maps of the linear dimensions and displacements of the frame elements after the mechanical machining: a) top side, b) bottom side

a)



b)

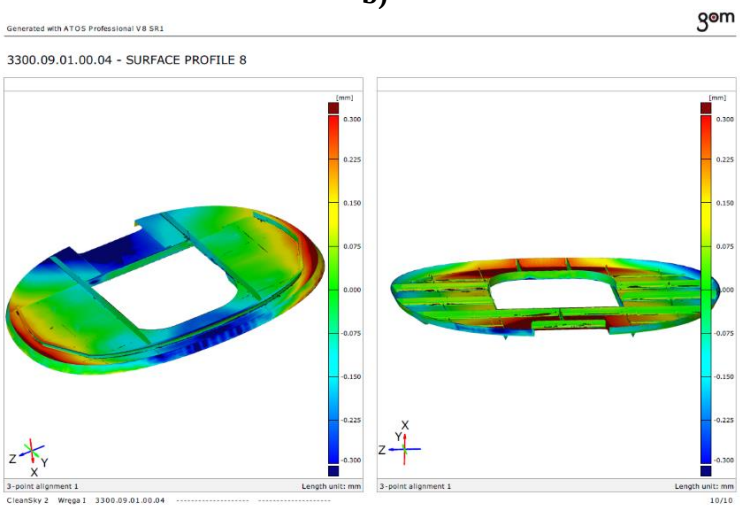


Fig. 5. The maps of the linear dimensions and displacements of the frame elements after the mechanical machining

For a precise evaluation of the ribs' wall thicknesses, which are the most significant when considering construction strength, they were measured using a micrometer. The measurements were done at the ribs' points which were expected to be the most vulnerable to deformations during machining. The measurements were done at the base of the rib and at the top. The layout of the measurement points is presented in Figure 6, while the measurement results are presented in Table 3.

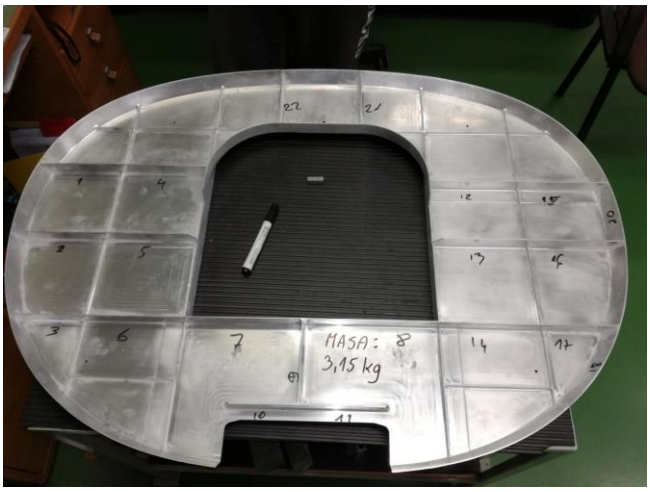


Fig. 6. The measurement points layout (the part after finishing)

Tab. 3. Thickness dimensions of the particular frame walls

Point no.	Dimension at the base	Dimension at the top	Point no.	Dimension at the base	Dimension at the top	Point no.	Dimension at the base	Dimension at the top
1.	2.00	2.02	11	3.01	3.01	21	2.01	2.02
2.	2.01	2.02	12	2.05	2.01	22	2.05	2.02
3.	2.01	1.95	13	2.01	2.01	23	1.99	2.00
4.	1.95	2.02	14	2.01	2.01	24	1.05	–
5.	2.00	2.02	15	2.00	2.02	25	1.07	–
6.	2.05	2.00	16	2.00	2.00	26	1.06	–
7.	1.96	1.95	17	2.00	2.00	27	2.00	2.02
8.	1.95	1.96	18	1.98	–	28	2.00	2.01
9.	2.00	2.05	19	2.00	–	29	2.01	2.02
10.	3.01	3.01	20	1.95	–	30	2.01	2.01

In order to estimate the shape, dimensions and spatial displacements of the frame as a finished subassembly, it was checked on a control device which is used to control the frames currently used in the aircraft (Fig. 7). The research has proved that the frame dimensions fall within the tolerances predicted in the technical documentation. The dimensions also conformed to the norm BN-85/3813-79, which is used for verifying non-tolerated deviations, shape, and location for aviation products.

Fig. 7. The finished frame mounted in the device used for checking the aircraft frames

4. FINAL REMARKS AND CONCLUSIONS

The use of High Speed Milling technology for machining of thin-walled aircraft frames is possible due to the precision of the workpiece and the machining capabilities of the 7075 aluminium alloys. Appropriate cutting parameter selection ensures good surface roughness and waviness.

In the case of the walls machining having a height-thickness ratio of less than 15, the influence of the structure deformation on the workpiece's dimensional precision is not important and it falls within the tolerance of the milling machine. For larger walls slenderness, this influence starts to be visible and needs to be minimized by the cutting speed selection (feed rate, cutting depth).

Providing dimensional-shape precision during the machining requires an appropriate selection of the technological bases and the mounting method of the workpiece. In compound spatial structures some additional bases ensuring mounting stiffness should be implemented and then removed during the final operations.

The final HSM technology application allows the producer to reduce the workpiece machining time. The manufacturing time of the frame (including software preparation) is 320 hours. The frame manufacturing time can be reduced to 15 hours in the series production by using the proven control program and the experience gained during the prototype manufacturing. The roughing time of the frame is about 18 hours, and the finishing is about 5 hours, which overall takes about 23 hours to manufacture the frame using the HSM technology. When comparing the HSM method to conventional machining methods, it needs to take about 141 hours to manufacture the frame by conventional milling.

The completed part is an integral construction replacing the current methods of manufacturing the same element by plastic processing of individual parts (30 or more parts) and then joining them by fasteners. The overall manufacturing time of individual elements of the frame was more than 6 times more hours than the manufacturing time of the proposed HSM method. Simultaneously with the benefits resulting from the reduction of the manufacturing time of the frame, the benefit of the proposed technology is also the quality and accuracy of the manufacturing as well as the roughness class of the surfaces.

Acknowlegments

The works were carried out within the project "699757/SAT-AM" – Work programme topic: JTI-CS2-2015-CPW02-AIR-02-07. More Affordable Small Aircraft Manufacturing; Airframe itd. Grant Agreement No: CS2-AlR-GAM-2014-2015-01 (annex III) co-financed by Horizon 2020 Clean Sky 2.

Special thanks go to the teams from: Instytut Lotnictwa, SZEL-TECH Szeliga Grzegorz, Zakłady Lotnicze Margański&Mysłowski, PZL Mielec, CIRA, Eurotech, Metrol and Ultratech.

REFERENCES

Lundblad, M. (2002). Influence of Cutting Tool Geometry on Residual Stress in the Workpiece. In *Proc. Third Wave AdvantEdge User's Conferece* (Paper 7). Atlanta, GA.

Shet, C., & Deng, X. (2003). Residual Stresses and Strains in Orthogonal Metal Cutting. *Int. J. Machine Tools Manuf., 43*(6), 573-587. doi:10.1016/S0890-6955(03)00018-X

Shih, A. J., & Yang, H. T. Y. (1993). Experimental and Finite Element Predictions of Residual Stresses Due to Orthogonal Metal Cutting. *Int. J. Num. Meth. Eng., 36*, 1487–1507. doi:10.1002/nme.1620360905

Adamski, W. (2010). Manufacturing development strategies in aviation industry. *Advances in Manufacturing Science and Technology*, *34*(3), 73–84.

Mativenga, P. T., & Hon, K. K. B. (2005). An experimental study of cutting force in high speed end milling and implications for dynamic force modelling. *Journal of Manufacturing Science and Engineering, 127*(2), 251-261. doi:10.1115/1.1863254

Kuczmaszewski, J., Pieśko, P., & Zawada-Michałowska, M. (2017). Influence of Milling Strategies of Thin-walled Elements on Effectiveness of their Manufacturing. *Procedia Engineering, 182*, 381-186. doi:10.1016/j.proeng.2017.03.117

Applied Computer Science, vol. 14, no. 2, pp. 96–111
doi:10.23743/acs-2018-16

Submitted: 2018-03-08
Revised: 2018-03-20
Accepted: 2018-06-21

induction motor, wavelet transformation, backlash zone, neural networks

Marcin TOMCZYK[*]*, Barbara BOROWIK*[**]*, Bohdan BOROWIK*[***]

IDENTIFICATION OF THE MASS INERTIA MOMENT IN AN ELECTROMECHANICAL SYSTEM BASED ON WAVELET–NEURAL METHOD

Abstract

This paper presents the results of testing of a complex electromechanical system model. These results have been obtained for accepted in simulations the method of identifying an inertia moment of reduced masses on shaft of induction motor drive during the changes of a backlash zone width. The effectiveness of correct diagnostic conclusions enables coefficients analysis of testing signals wavelet expansion as well as weights of a supervised learning neural network. The earlier fault detection of five important state variables, which describe physical quantities of chosen complex electromechanical system has been verified for its correctness during the backlash zone width monitoring in the early stage of its gradual rise. The proposed here algorithm with mass inertia moment changes has proved to be an effective diagnostic method in the area of system changeable dynamic conditions and this has been shown in the resulting changes of backlash zone width.

1. INTRODUCTION

Diagnostics of electromechanical processes deals with the identification of changes in their states, what has been presented in the form of a sequence of intentional actions performed by means of the set of available machines and devices for a fixed period of time. After exceeding a certain value, the damage must be detected and identified. A diagnostic algorithm must detect and identify

[*] Electrical School No. 1 in Krakow, Kamieńskiego 49, 30-644 Kraków, tomczykmarcin@poczta.fm
[**] Cracow University of Technology, Warszawska 24, 31-155 Kraków, cnborowi@cyf-kr.edu.pl
[***] The University of Bielsko-Biala, Willowa 2, 43-309 Bielsko-Biała, bo@borowik.info

in a relatively short time a fault that occurs early in the development phase (Korbicz, 2002). In mechanical connections containing backlashes, non-linear resilient-absorbing elements or damaged bearings – it is necessary to classify signals simultaneously in the time domain and depending on frequencies using transformation methods, to be able to study its spectral properties (Duda, 2007). The adaptive, time-frequency distribution of signals waveform processing has a number of important, scalable properties, relating both to time and to frequency, analyzing the relationship between the function being studied and its transformation coefficients (Doniec, 2010).

During the period of the last several decades more and more scientific works have been appearing in the literature presenting methods of industrial structures damage diagnostics using a time-frequency analysis and neural networks. It is worthy to mention some of them:

- presentation of diagnostics of various types of faults of induction motor by means of packet wavelet analysis (Kowalski, 2006),
- presentation of a new detection technique and the method of classifying faults of induction motors by means of current and dispersed stator stream analysis (Ishkova & Vitek, 2016),
- using a discrete wavelet transform of the current stator envelope for detection of spiral short circuits in an induction motor, in the initial phase of failure (Wolkiewicz & Kowalski, 2015),
- application of statistical features of wavelet distribution coefficients to a neural network training, using a backpropagation algorithm (Yayakumar, Thangavel & Elango, 2015),
- detection of a stator winding fault, using a combination of a wavelet transform and a neuro-fuzzy identifier (Farronato et al., 2005).

2. METHODOLOGY AND RESEARCH OF THE DIAGNOSTIC ALGORITHM FOR FAULT IDENTIFICATION

Diagnostic tests have been carried out for the nominal conditions of an induction motor whose model has been built in a stationary coordinate system related to the stator (model α, β, 0). It is assumed that the load of the induction motor is a working machine of the form of a dynamic mass-absorbing-resilient element. Figure 1 shows in a simplified form a diagram of the connection of a working machine with the induction motor. The tested width of the backlash zone occurs between the rod of the induction motor drive and a working machine drive wheel. The backlash results from the line slip of the dynamic-absorbing-resilient dynamic element on the surface of the working machine drive wheel.

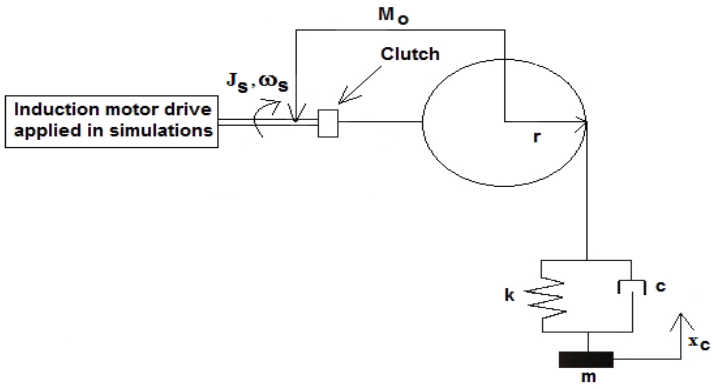


Fig. 1. Diagram of the dynamic mass-absorbing-resilient element, which has been connected to the used in the tests induction motor. The diagram shows the inertia moment of the masses reduced on the motor shaft

The tests have been carried out within the MATLAB / Simulink environment, using the following parameters of the induction motor (parameters of its substitute circuit are expressed in relative units): circuit stator relative resistance r_s = 0.059 [Ω], circuit rotor relative resistance r_w = 0.048 [Ω], relative reactance of the dispersion circuit stator x_s = 1.92 [Ω], relative reactance of the dispersion circuit rotor x_w= 1.92 [Ω], relative reactance of the dispersed circuit x_m= 1.82 [Ω], $w = x_s * x_w - x_m * x_m = 0.374$, mechanical time constant T_m = 0.86 [s].

3. IDENTIFICATION TESTS OF THE INERTIA MOMENT VALUES IN THE ELECTROMECHANICAL SYSTEM CONTAINING VISCOUS FRICTION

The tests have been carried out in four test groups, with the following four different values of apparent viscosity coefficient η_k: 0.0125 [Pa·s^{n_1}], 0.025 [Pa·s^{n_1}], 0.0375 [Pa·s^{n_1}] i 0.05 [Pa·s^{n_1}]. Each test group contained six cases with different backlash zone width values. Results of simulations for all physical quantities and for every case of change of inertia moment of reduced masses and connected stiffly with the induction motor drive rotor J_s – have been written in the matrix $M_{[7,2048]}$. Elements of the matrix M have been written for each consistency coefficient value η_k (i.e. for apparent viscosity). Value of an inertia moment J_s have been determined as the percentage in relation to its nominal value J_s, up with value A% and down with value C%. Formal changes of inertia moment J_s have been written in matrix "*moments*" in the following order: *moments* = [nominal value of inertia moment (J_s = 0.862), A=2.5% (J_s = 0.884), A=5% (J_s = 0.905), A=8% (J_s = 0.932), A=21% (J_s = 1.045), C=2.5 % (J_s = 0.841), C=5% (J_s = 0.819)].

In each of the six cases of changes of backlash zone width have been carried out simulation tests for seven inertia moment J_s values. The backlash zone width values have been taken in sequence from the epsilon matrix, in the following order: *epsilon* = [0.0025, 0.00375, 0.005, 0.0075, 0.009, 0.01]. All tests have been carried out for the creep index n_1 value equal 0.93.

The wavelet type and its order has been selected in such a way that the shape of the basic wavelet approximately would be adequate to the character of the transient course of the tested physical quantity, obtained as a result of a simulation for the case of the smallest backlash value. Based on the carried out tests the following selections of wavelets have been made for individual physical variables, with decomposition level 10:

a) linear acceleration of the induction motor drive a_s – sym5,
b) electromagnetic moment of the induction motor drive m_{el} – db6,
c) angular speed of the induction motor drive rotor ω_s – sym5,
d) linear acceleration of a mass a_c – db6,
e) linear speed of a mass v_c – sym5.

In these simulation tests, it has been assumed that the process of the electromechanical system dynamics testing in the backlash zone starts when the expression specified in the left part of the following inequality (1) is smaller than the right part of the below inequality:

$$\left|\alpha_1 - \alpha_2\right| < \frac{epsilon(i)}{r}; \quad i = 1,2..6, \tag{1}$$

where: r – radius of the drive wheel of a working machine [m],
$epsilon_{(i)}$ – value that has been taken sequentially from the "*epsilon*" matrix and corresponding to the given backlash value in mechanical connection,
i – index number within the "*epsilon*" matrix.

The location change angle α_1 for rod masses of the induction motor drive [rad/s] has been calculated using the formula: $\alpha_1 = \frac{x_1}{r}$, and the location change angle α_2 of the dynamic mass-absorbing-resilient element mass [rad/s], has been calculated using the formula: $\alpha_2 = \frac{x_c}{r}$, where x_c and x_1 – are linear distances, the first done by mass of the dynamic mass-absorbing-resilient element [m], and the second - done by rod mass of the induction motor drive [m].

After satisfying the condition determined by inequality described by formula (1) the load moment of dynamic mass-absorbing-resilient element is set to zero in the tested electromechanical system. For all tested physical quantities the matrix *M*

has contained 2048 samples chosen starting from the moment of obtaining backlash zone. Figure 2 presents the example of a course of the tested signal of the induction motor driver linear acceleration a_s in a backlash zone.

Samples chosen according to the scheme presented on figure 2 have been written in sequence to the matrix M for each executed simulation for a given backlash zone width.

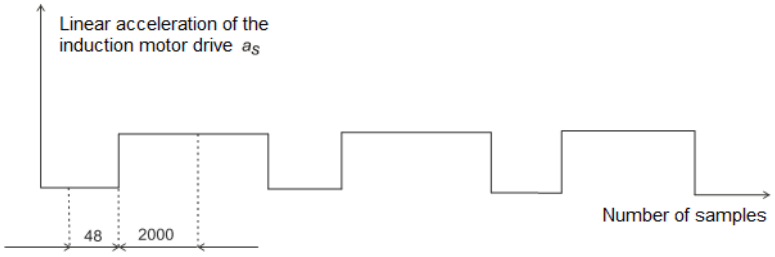


Fig. 2. Testing dynamics of the induction motor drive linear acceleration a_s in a backlash zone, carried out during selected time range choices (samples of the tested signal)

3.1. Processing of the three-layer neural network learned by the Levenberg-Marquardt algorithm with the use of a backpropagation method

Algorithm of a one-way neural network taught by means of the Levenberg-Marquardt algorithm using a backpropagation method and the cascade connections structure in this network – enables obtaining fast convergence at the relatively low computational complexity. In subsequent epochs of the network learning, if the calculated error value is smaller than the error in the previous (starting) point, then the coordinates of the new point will determine a new starting point with a corresponding set of weights that would have the ability to approximate the optimal parameters of the learned network.

Applied in the test neural network has contained three layers. To obtain the most profitable results of identification of backlash zone width in chosen physical quantities signals tests – an important element was to determine the number of neurons in the input layer, which represents the first layer of the applied neural network.

To obtain the appropriate number of samples, which determine input values as well as the proper output values of the neural network, there have been made many series of simulations and observations of results using different number of samples, which have been written in the matrix M. Finally, after the simulation tests execution, the last 400 samples have been chosen from the matrix M and have been written in a separate matrix M_1. In every case considered in the entire this section the value of the consistency coefficient η_k used in tests equals to 0.025. The same variables, which are used in many formulas, when their meaning stays the same, then it is explained only once during their first occurrence.

In case of all tested physical quantities every input value X in the first layer of the neural network has been transformed to the $[h_2, k_2]$ range of normalized values of the matrix M_1 according to the formula:

$$X_{(a)} = \left(\frac{\left(M_{1(e)(1,a)} - h_1 \right)}{k_1 - h_1} \right) \cdot \left(k_2 - h_2 \right) + h_2 ; \quad a = 1,2,...,400; \; e \in < 1,6 > \tag{2}$$

where: M_1 – values of the matrix M_1, registered for testing in which consistency's coefficient η_k = 0.025 and for assumed correct value of inertia moment J_s,
h_1 and k_1 – the minimal and maximal value (respectively) of the matrix M_1, for assumed correct value of inertia moment J_s,
h_2 and k_2 – the initial and the final value of the range, which contains normalized values of the matrix M_1,
e – an index of a column's number in the "*epsilon*" matrix.

The set values T of the neural network have been calculated in a similar way as input values X of the network. A transformation to the $[h_2, k_2]$ range has been carried out using the following formula:

$$T_{(a)} = \left(\frac{\left(k_1 - M_{1(e)(1,a)} \right)}{k_1 - h_1} \right) \cdot \left(k_2 - h_2 \right) + h_2 ; \quad \text{for } a = 1,2,...,400; \tag{3}$$

$$e \in < 1,6 >; \; i = 1,2...7;$$

The beginning as well as the end of the range, to which values of the matrix M_1 have been transformed – they represent the minimal ($h_2 = \min(h_3, k_3)$) and the maximal ($k_2 = \max(h_3, k_3)$) values, which have been determined for the two compared values of h_3 and k_4, where h_3 is the variable that has been determined using statistical parameters calculated for the matrix M_2 rows in testing, and k_3 is the variable determined using the difference between normalized median value of the tested matrix M row and the sum of values of variables h_3 and h_4.

Therefore the value of the variable h_3 has been calculated in the following way: $h_3 = \min\left(m_1, \frac{h_5}{k_5} - m_1 \right)$, where: m_1 is the average value of the matrix M_2, h_5 is the minimal value of the matrix M_2, and k_5 the maximal value of the matrix M_2.

Statistical parameters m_1, h_5 and k_5 have been calculated for matrix M_2 rows for testing, for assumed backlash zone width. For all tested physical quantities values of matrix M_2 contains medians of rows in matrix M_3. Matrix M_3 contains sorted in the ascending order values of rows of matrix M for testing, in which assumed consistency coefficient η_k has been set to 0.025.

To determine the value of the parameter k_3, it is necessary to perform its normalization, i.e. median of matrix's tested row M must be decreased by sum of parameter values h_3 and h_4. The normalization formula is as follows:

$$k_3 = \begin{cases} \dfrac{(m_2 - h_5)}{(k_5 - h_5)} - (h_3 + h_4); & for \ \max(m_5, m_6) = m_5 \\ \dfrac{(k_5 - m_2)}{(k_5 - h_5)} - (h_3 + h_4); & for \ \max(m_5, m_6) = m_6 \end{cases} \tag{4}$$

where: m_2 – the value of median of matrix's tested row M,
m_5 – the average value of matrix M_5 for test,
m_6 – the average value of matrix M_6 for test.

Value of the parameter h_4 is determined as the following minimum: $h_4 = \min\left(m_2, \dfrac{h_5}{k_5} - m_2\right)$.

For all tested physical quantities value of median m_2 is calculated for values of matrix M_4. This matrix contains sorted in the ascending order values of matrix M.

Values of arithmetic means m_5 and m_6 were calculated for rows in matrices respectively M_5 and M_6 for testing. In case of all tested physical quantities values of matrices M_5 and M_6 are located in the range [0, 1] and represent the result of the normalization process of M_2 matrix, carried out according to the below formulas:

$$M_{5(e)(i)} = \frac{(M_{2(e)(i)} - h_5)}{(k_5 - h_5)}; \ e \in <1,6>; i = 1,2...7 \tag{5}$$

$$M_{6(e)(i)} = \frac{(k_5 - M_{2(e)(i)})}{(k_5 - h_5)}; e \in <1,6>; i = 1,2...7 \tag{6}$$

where: M_2 – matrix containing sorted in the ascending order values of matrix M rows for testing, in which consistency's coefficient η_k =0.025,
h_5 and k_5 – are respectively the minimal and maximal value of the matrix M_2.

The initial values of weights $W_{11}, W_{12}, W_{21}, W_{31}$ and W_{32} – have been calculated in the following way: $W_{11(i)} = M_{7(i)}$, $W_{12(i)} = M_{7(i)}$, $W_{21(i,j)} = M_{7(i)}$, $W_{31(i)} = M_{7(i)}$, $W_{32(i)} = M_{7(i)}$, for indexes i,j = 1,2,...,7, where M_7 – matrix contains maximum values of rows respectively matrix M_8 as well M_9.

Therefore values of the matrix M_7 are calculated for test executed with consistency's coefficient η_k =0.025, according to the following formula:

$$M_{7(i)} = \max\left(\max\left(M_{8(i,j)}\right), \max\left(M_{9(i,j)}\right)\right); \quad i = 1,2...7; \; j = 1,2...2048 \tag{7}$$

The initial value of W_{13} weight represents the arithmetic mean m_7, calculated using M_7 matrix: $W_{13} = m_7$.

Values of matrices M_8 and M_9 are located in the range [0, 1] and represent values of matrix M, have been obtained as a result of the normalization process of all considered physical quantities used in testing, according to the below formulas:

$$M_{8(i,j)} = \frac{\left(M_{(e)(i,j)} - h_6\right)}{\left(k_6 - h_6\right)}; \quad e \in <1,6>; i = 1,2,...7; \; j = 1,2,...2048; \tag{8}$$

$$M_{9(i,j)} = \frac{\left(k_6 - M_{(e)(i,j)}\right)}{\left(k_6 - h_6\right)}; \quad e \in <1,6>; i = 1,2,...7; \; j = 1,2,...2048; \tag{9}$$

where: h_6 and k_6 – are respectively the minimal and the maximal value of the matrix M used for testing.

In executed simulations some parameters have been calculated especially only for testing, in which assumed consistency coefficient η_k has been set up to the value 0.025. It was applied for the following parameters: h_1, h_3, h_5, h_6, k_1, k_5, k_6, m_1, m_5 and m_6.

The initial values of matrix of biases for neural network first and hidden layer have been set to 1, i.e. $B_{(f)(i)}$=1; f = 1,2; i =1,2,…,7, where f is the number of neural network layer applied in test. It has been assumed that the initial value of bias B_3 in the output layer was also set to 1. Output signal Y_1 from the first layer of the neural network have been calculated as follows:

$$Y_{1(a,i)} = X_{1(a)} * W_{11(i)} + B_{1(i)}; \; a = 1,2,...,400; \; i = 1,2,...,7; \tag{10}$$

where B_1 – is the matrix of biases in the neural network input layer.

During executed tests a linearity of neurons activation function has been assumed. The output signals Y_2 from the hidden (i.e. the second) layer of the neural network, have been calculated as follows:

$$Y_{2_{(a,i)}} = \sum_{j=1}^{7} Y_{1(a,j)} * W_{21(i,j)} + X_{1(a)} * W_{12(i)} + B_{2(i)}; \; a = 1,...400; \; i = 1,...7, \qquad (11)$$

where B is the matrix of biases in the hidden layer of the neural network. The output signals Y_3 of the output layer in the neural network have been obtained in the similar way, as in the previous layers, i.e.:

$$Y_{3_{(a)}} = \sum_{i=1}^{7} Y_{2(a,i)} * W_{32(i)} + \sum_{i=1}^{7} Y_{1(a,i)} * W_{31(i)} + X_{1(a)} * W_{13} + B_3; \qquad (12)$$

$$for \; a = 1,...,400,$$

where B_3 is the value of bias in output layer of neural network.

Adaptation of weights and biases have been carried out for each epoch of neural network learning by means of updating values in matrix W_4, containing all weights and biases that have been obtained using the following formula:

$$W_4 = \begin{bmatrix} \left[W_{4(a)} = W_{11(i)}; \; a = 1, \; 2 \ldots 7; \; i = 1,2 \ldots 7\right] \left[W_{4(a)} = B_{1(i)}; \; a = 8, \; 9 \ldots 14;\right] \\ \left[W_{4(a)} = W_{12(i)}; \; a = 15,16 \ldots 21; \; i = 1,2 \ldots 7\right] \\ \begin{bmatrix} W_{4(a)} = \mathrm{W}_{21(i,1)}; \; a = 22,23 \ldots 28; \; i = 1,2 \ldots 7 \\ W_{4(a)} = W_{21(i,2)}; \; a = 29,30 \ldots 35; \; i = 1,2 \ldots 7 \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ W_{4(a)} = W_{21(i,7)}; \; a = 64,65 \ldots 70; \; i = 1,2 \ldots 7 \end{bmatrix} \\ \left[W_{4(a)} = B_{2(i)}; \; a = 71, \; 72 \ldots 77; \; i = 1,2 \ldots 7\right] \\ W_{4(a)} = W_{13}; \; a = 78 \\ \left[W_{4(a)} = W_{31(i)}; \; a = 79, \; 80 \ldots 85; \; i = 1,2 \ldots 7\right] \\ \left[W_{4(a)} = W_{32(i)}; \; a = 86, \; 87 \ldots 92; \; i = 1,2 \ldots 7\right] \\ W_{4(a)} = B_3; \; a = 93 \end{bmatrix} \qquad (13)$$

Values of matrix W_4 have been changed in the following way (Rusiecki, 2007):

$$W_{4(i)} = W_{4(i)} - \left[H_{(i,j)}\right]^{-1} \cdot \nabla G_{(i)}; \; a = 1,\ldots 400; \; i = 1,\ldots 93; \; j = 1,\ldots 93, \qquad (14)$$

where ∇G – is a matrix of neural network gradient, and H – is the approximated Hessian matrix, applied to the Levenberg-Marquardt algorithm.

The gradient matrix ∇G is determined according to the formula (Rusiecki, 2007):

$$\nabla G_{(i)} = J_{(a,j)^T} \cdot E_{(a)^T}; \quad a = 1,2,\ldots,400; \quad i = 1,2,\ldots,93, \tag{15}$$

The Hessian matrix H is calculated in the following way (Rusiecki, 2007):

$$H_{(i,j)} = J_{(a,i)}{}^T \cdot J_{(a,i)} + l \cdot I_{(i,j)}; a = 1,2\ldots400; i = 1,2\ldots93; \; j = 1,2\ldots93 \tag{16}$$

where: I – identity matrix assumed as diagonal,
J – Jacobian matrix of neural network,
l – coefficient of neural network learning, and belongs to the [0,1] range.

Values of the neural network Jacobian matrix J have been calculated according to following formula (Rusiecki, 2007):

$$J_{(a,i)} = \begin{bmatrix} \frac{\partial E_{(1)}}{\partial W_{4(1)}} & \frac{\partial E_{(1)}}{\partial W_{4(2)}} & \cdots & \frac{\partial E_{(1)}}{\partial W_{4(93)}} \\ \frac{\partial E_{(2)}}{\partial W_{4(1)}} & \frac{\partial E_{(2)}}{\partial W_{4(2)}} & \cdots & \frac{\partial E_{(2)}}{\partial W_{4(93)}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial E_{(400)}}{\partial W_{4(1)}} & \frac{\partial E_{(400)}}{\partial W_{4(2)}} & \cdots & \frac{\partial E_{(400)}}{\partial W_{4(93)}} \end{bmatrix}; \tag{17}$$

where: E – matrix containing values of the neural network errors,
W_4 – matrix of the neural network weights and biases.

In order to provide the appropriate approximation of the Hessian matrix H, the following diagonal identity matrix I has been was provided:

$$I_{(i,j)} = \begin{cases} 1 \text{ for } i = j \\ 0 \text{ for } i \neq j \end{cases} \text{for } i = 1,2,...,93; \; j = 1,2,...,93; \tag{18}$$

The coefficient of neural network learning l changes in every given epoch of this network learning process depending on the value of the mean square error (*MSE)* and this is shown by the below formula:

$$l_{(g+1)} = \begin{cases} l_{(g)} * l_1; \text{ for } MSE_{(g)} > MSE_{(g-1)} \\ l_{(g)} * l_2; \text{ for } MSE_{(g)} < MSE_{(g-1)} \end{cases}, \tag{19}$$

where: g – the number of a given epoch of this neural network learning process,
l_1 – the first beginning coefficient, set to 10,
l_2 – the second coefficient set to 0.1.

In the output layer of the neural network values of errors on neurons have been stored in matrix E. They have been calculated on the basis of the difference between the output values Y_3 and set values T, i.e.: $E_{(a)} = (Y_{3(a)} - T_{(a)})$, where $a = 1, 2, \ldots, 400$. The mean square error (MSE) has been calculated according to the formula: $MSE = \frac{\sum_{a=1}^{400} \left(E_{(a)}\right)^2}{400}$ and compared with δ, i.e. with the earlier experimentally assumed value, used for stopping the learning process of the neural network. In other words, when the condition of $MSE < \delta$ has been satisfied, the neural network learning process has been finished.

For the earlier assumed value of consistency coefficient $\eta_k = 0.025$ and backlash zone width there have been created pattern matrices W_w (for every group of tests one matrix). Each such pattern matrix with set up earlier value of $\eta_k = 0.025$ and backlash zone width – was accepting changing respectively values of the inertia moment J_s. In this way has been obtained a matrix with a dimension 7 by 7, and this can be expressed by the formula:

$$W_{w(e)(i,j)} = \frac{W_{32(j)}}{h_3}; \quad for\ e \in <1,6>\ and\ i,j = 1,2,\ldots,7, \tag{20}$$

where: h_3 – is the value of the variable necessary to determine the range, in which would be placed calculated values of matrix X and T.

In the same way, with the earlier set up a backlash zone width and for $e \in <1,6>$ and $j = 1,2,...,7$, it has been determined values of the matrix W_b of dimensions 1 by 7: $W_{b(e)(j)} = \frac{W_{32(j)}}{h_3}$.

Correct identification of the value of the inertia moment in the executed test for every physical quantity is possible for having assumed earlier values of these quantities can be obtained using the values of the matrix G calculated according to the formula:

$$G_{(i)} = \sum_{j=1}^{7} \left| W_{b(e)(j)} - W_{w(e)(i,j)} \right|; \quad e \in <1,6>; i = 1,2,..., 7; \tag{21}$$

Index nr_6 (for $nr_6 \in <1,7>$) in matrix G determines the column number in the matrix "*moments*", which contains the correct value of inertia moment. The value of index nr_6 ($nr_6 \in <1,7>$) for i = 1,2,...,7 in matrix G has been determined using the following minimum function: $G_{(nr_6)} = \min(G_{(i)})$. Therefore, the number i of a column in the matrix *moments* refers to the corresponding to it index nr_6 ($i = nr_6$).

3.2. Simulation results of the algorithm of the identification of the moment of inertia value while changing the width of the backlash zone in the electromechanical system, using a three-layer neural network

In the below tables, using the bold font have been presented the resulting final results of calculations of the matrix G, while in the column named *Test parameters* there have been placed the assumed set up earlier values, like the widths of the backlash zone, adopted in the process of identifying the moment of inertia of the reduced masses and connected stiffly with the rotor of the induction motor driver. The values of the G matrix presented in the Table 1 through 4 – are the correct results obtained finally in the process of identifying the fault number. Pattern matrices W_w have been created for analyzes, and in them it was assumed both, the specific value of the learning coefficient l of the neural network, as well as the value δ – as the value of the accepted error, which has allowed to stop the neural network learning process. On the basis of the executed simulations and analyzes of its results it is noticeable that obtainment of correct results of the identification number of a fault for all tested physical quantities is possible, while the following condition is satisfied: execution of a simulation with the neural network learning coefficient l changing within the range from 0.1 to 0.9, with the value of δ equal to 10^{-4}, i.e. the parameter controlling the stopping time of the neural network learning process. Results that have been shown in table 1 through 4 illustrate this fact independently from the obtained number of epochs of the neural network processing.

Tab. 1. Exemplified results of tests in matrix *G* for angular speed of the rotor of the induction motor drive ω_s

Test parameters	Results	Test parameters	Results
inertia moment J_s = 0.819, backlash zone = 0.0025, η_k = 0.05, epochs = 7, l = 0.9, $\delta = 10^{-5}$	0.0014 0.0032 0.0053 0.0084 2.8045 0.0007 **0.0001**	inertia moment J_s = 0.819, backlash zone = 0.0025, η_k = 0.05, epochs = 6, l = 0.9, $\delta = 10^{-4}$	0.0019 0.0040 0.0072 0.0137 4.8939 0.0007 **0.0005**
inertia moment J_s = 0.932, backlash zone = 0.0075, η_k = 0.0375, epochs= 7, l = 0.9, $\delta = 10^{-5}$	0.0063 0.0053 0.0034 **0.0004** 1.6930 0.0070 0.0076	inertia moment J_s = 0.932, backlash zone = 0.0075, η_k = 0.0375, epochs = 6, l = 0.1, $\delta = 10^{-4}$	0.0011 0.0009 0.0005 **0.0001** 3.5692 0.0012 0.0013

Tab. 2. Exemplified results of tests in matrix *G* for linear acceleration of the mass a_c

Test parameters	Results	Test parameters	Results
inertia moment J_s = 1.045, backlash zone = 0.009, η_k = 0.0375, epochs = 6, l = 0.9, $\delta = 10^{-5}$	1.4902 1.4776 1.4566 0.5744 **0.0864** 1.4958 1.5023	inertia moment J_s = 1.045, backlash zone = 0.009, η_k = 0.0375, epochs = 5, l = 0.9, $\delta = 10^{-4}$	4.5547 4.5166 4.4511 4.2691 **0.0482** 4.5754 4.5910
inertia moment J_s = 0.884, backlash zone = 0.01, η_k = 0.0125, epochs = 7, l = 0.9, $\delta = 10^{-5}$	0.0129 **0.0052** 0.0216 1.6877 1.4747 0.0254 0.0300	inertia moment J_s = 0.884, backlash zone = 0.01, η_k = 0.0125, epochs = 6, l = 0.1, $\delta = 10^{-4}$	0.0037 **0.0001** 0.0082 0.0270 0.0418 0.0064 0.0078

Tab. 3. Exemplified results of tests in matrix *G* for linear acceleration of the induction motor drive a_s

Test parameters	Results	Test parameters	Results
inertia moment J_s = 0.884, backlash zone = 0.0075, η_k = 0.0125, epochs = 7, l = 0.9, δ=10^{-5}	0.0526 1.0286 0.1911 3.2197 0.0094 0.0170 **0.0023**	inertia moment J_s = 0.884, backlash zone = 0.0075, η_k = 0.0125, epochs = 6, l = 0.9,δ = 10^{-4}	0.0506 2.2530 1.8502 8.0211 0.0075 0.0165 **0.0037**
inertia moment J_s = 0.905, backlash zone = 0.009, η_k = 0.05, epochs = 6, l = 0.9, δ = 10^{-5}	3.5903 3.1856 **0.6751** 10.8113 2.6810 2.7129 2.6956	inertia moment J_s = 0.905, backlash zone = 0.009, η_k = 0.05, epochs= 4, l = 0.1, δ = 10^{-4}	3.0191 5.2248 **0.0643** 13.9348 3.0097 3.0134 3.0110

Tab. 4. Exemplified results of tests in matrix *G for* Electromagnetic moment of the induction motor drive m_{el}

Test parameters	Results	Test parameters	Results
inertia moment J_s = 0.841, backlash zone = 0.005, η_k = 0.05, epochs = 7, l = 0.9, δ = 10^{-5}	0.0182 0.0454 1.5448 1.0588 0.0864 **0.0011** 0.0081	inertia moment J_s = 0.841, backlash zone = 0.005, η_k = 0.05, epochs = 6, l = 0.9, δ = 10^{-4}	0.0198 0.0638 0.1773 3.0455 0.1373 **0.0020** 0.0101
inertia moment J_s = 1.045, backlash zone = 0.0025, η_k = 0.0375, epochs = 6, l = 0.9, δ = 10^{-5}	1.7720 1.7384 1.6812 0.3638 **0.0269** 1.7914 1.8023	inertia moment J_s = 1.045, backlash zone = 0.0025, η_k = 0.0375, epochs = 5, l = 0.1, δ = 10^{-4}	5.2338 5.2311 5.2255 0.0258 **0.0063** 5.2351 5.2363

4. CONCLUSIONS

The presented paper describes a fault detection system containing a neural network trained using the Levenberg-Marquardt algorithm together with the adapting back error propagation method. The system has been applied to the identification of the inertia moment of reduced masses and connected stiffly with the induction motor drive rotor J_s.

The process of the identification was executed for certain time periods. These ranges have been obtained as the result of changes of the width of the backlash zone by means of time-frequency methods with the multistage decomposition of the signal for the complex electromechanical system, connected with a dynamic mass-absorbing-resilient element.

The application of the wavelet-neuron method strongly affects the efficiency of the analysis of non-stationary signals in the executed researches, effectively limiting the dangerous consequences of appearing fault in the initial phase of its development.

On the basis of the executed tests, we can notice that ensuring appropriate changes in parameters in a system containing non-zero backlash zones causes results to be much more effective in the detection and identification of the fault. These parameters are: coefficient of neural network learning and the fixed stopping value of the neural network learning process.

REFERENCES

Doniec, R. (2010). *Wykorzystanie metod sztucznej inteligencji do regulacji poziomu insuliny w organizmie człowieka* (doctoral dissertation). Wydawnictwo Politechniki Śląskiej, Gliwice.

Duda, J. T. (2007). Pozyskiwanie wzorców diagnostycznych w komputerowych analizach sprawności urządzeń. In J. Korbicz, K. Patan, & M. Kowal (Eds.), *Diagnostyka procesów i systemów* (pp. 1–16). Warszawa: Akademicka Oficyna Wydawnicza EXIT.

Farronato, L., Monti A., Ponci, F., Ferrero, A., Cristaldi, L., & Lazzaroni, M. (2005). Virtual system Fault Models for Training Fuzzy-Wavelet Identifiers in Electrical Drive Diagnosis: an Experimental Validation. In *IMTC 2005 Proceedings of the IEEE. Instrumentation and Measurement Technology Conference* (pp. 2310–2315). Ottawa: IEEE. doi: 10.1109/ IMTC.2005.1604589

Ishkova, I., & Vitek, O. (2016). Detection and Classification of faults in induction motor by means of motor current signature analysis and stray flux monitoring. *Przegląd Elektrotechniczny, 92*(4), 166–170. doi: 10.15199/48.2016.04.36

Korbicz, J. (2002). *Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania.* Warszawa: WNT.

Kowalski, Cz. (2006). Zastosowanie analizy falkowej w diagnostyce silników indukcyjnych. *Przegląd Elektrotechniczny, 82*(1), 21–26.

Rusiecki, A. (2007). *Algorytmy uczenia sieci neuronowych odporne na błędy w danych* (doctoral dissertation). Politechnika Wrocławska, Wrocław.

Wolkiewicz, M., & Kowalski, Cz. (2015). Diagnostyka uszkodzeń uzwojeń stojana silnika indukcyjnego z wykorzystaniem dyskretnej transformaty falkowej obwiedni prądu stojana. *Maszyny elektryczne: zeszyty problemowe, 3*(107), 13–18.

Yayakumar, K., Thangavel, S., & Elango, M. K. (2015). Backpropagation Algorithm for Bearing Fault Detection of Induction Motor Drive Using Wavelet Packet Decomposition. *International Journal of Applied Engineering Research, 10*(10), 26191–26208**.**