The influence of algorithms for basic-schedule generation on the performance of predictive and reactive schedules

I Paprocka¹, C Grabowik¹, W M Kempa², D Krenczyk¹ and K Kalinowski¹
¹Silesian University of Technology, Faculty of Applied Mathematics, Institute of Mathematics, Kaszubska 23, 44-100 Gliwice, Poland
²Silesian University of Technology, Faculty of Mechanical Engineering, Institute of Engineering Processes Automation and Integrated Manufacturing Systems, Konarskiego 18A, 44-100 Gliwice, Poland

E-mail: iwona.paprocka@polsl.pl

Abstract. Lately, a great deal of effort has been spent developing methods to generate robust schedules. The Multi-Objective Immune Algorithm (MOIA) is one of the methods dealing with an uncertainty [1]. The basic schedules, obtained by the MOIA, are modified using the rule of the Minimal Impact of Disturbed Operation on the Schedule (MIDOS) in order to generate predictive schedules. If the effect of a disruption is too large, in order to generate reactive schedules, the rule of the Minimal Impact of Rescheduled Operation on the Schedule (MIROS) is applied. This paper is the continuation of the searching process for an algorithm which achieves good quality basic schedules that influence on the quality of predictive and reactive schedules. In [2,3], the two immune algorithms are compared, the MOIA and Clonal Selection Algorithm (CSA) when investigating the influence of basic schedules on the obtainment of stable and robust schedules with the application of the MIDOS and MIROS. The two algorithms are applied for a multi criteria job shop scheduling problem. In this paper, the genetic algorithm (GA) is applied for the same scheduling problem and the achieved schedules are compared.

1. Introduction
Since the performance of predictive and reactive schedules is sensitive to the quality of basic schedules, we apply Genetic Algorithm (GA) to a job shop scheduling problem. Searching for an algorithm which achieves good quality basic schedules, for the multi criteria job shop scheduling problem, GA is compared with immune algorithms. In [2,3], the two immune algorithms were compared, the Multi-Objective Immune Algorithm (MOIA) and Clonal Selection Algorithm (CSA) when investigating the influence of basic schedules on the obtainment of stable and robust schedules with the application of two heuristics. The basic schedules, obtained by the MOIA and CSA, are modified using the rule of the Minimal Impact of Disturbed Operation on the Schedule (MIDOS) in order to generate predictive schedules. If the effect of a disruption is too large, the rule of the Minimal Impact of Rescheduled Operation on the Schedule (MIROS) is applied in order to generate reactive schedules. In this paper, the genetic algorithm (GA) is applied for the same scheduling problem, next predictive and reactive schedules are generated using the heuristics: MIDOS and MIROS. The quality of achieved predictive and reactive schedules is compared.
Due to the combinatorial nature of the job shop scheduling problem and multi-dimensional searching space, no method has been found which solves the problem in polynomial time. The most commonly used approach to solve the problem is the application of GA-s. The GA, presented in the paper, bases on genetic mechanisms e.g. selection, crossover and mutation. Moreover, the phenomenon of the migration is embedded in the GA. In biology, if the changing conditions are not to adopt by a population of any species, the population is in the danger of dieing. In a deadlock, the population migrates to another region.

In the GA, an individual represents a solution of the scheduling problem, and a scalar fitness function is used to estimate the quality of the solution. Thus, the following research points have been identified:

1) methods of generating basic schedules enabling the obtainment of the best performance of the job shop system when the objective is to minimize the fitness function consisting of the four components: makespan, flow time, total tardiness and idle time.

2) methods of generating predictive schedules enabling the obtainment of stable and robust schedules when the objective is to minimize the fitness function consisting of the two components: solution robustness (SR) and quality robustness (GR) [1].

The proposed method is dedicated for maintenance scheduling. The method gives robust solutions in the condition of disruption. But, it should be aided by methods of understanding the maintainability process and innovative tools supporting reliability that positively impacts all phases of production [11, 12].

The paper is organized as follows: the genetic algorithm for basic scheduling is presented in the next Section. Section 3 presents a job shop scheduling problem with interruptions for experimental study. Predictive schedules are generated using the MIDOS rule and reactive schedules are generated using the MIROS rule in order to evaluate the stability and robustness performance, the results are presented in Section 4. The paper concludes with the estimation the correlation between the quality of basic schedules and the quality of predictive and reactive schedules.

2. A basic schedule generation using the Genetic Algorithm

The GA consists of the following modules: data interface, individuals coding, multi criteria genetic optimisation and selection and individuals decoding. The pseudo code of the GA is presented in figure 1. The parts of the algorithm are explained in following subsections.

2.1. Encoding and decoding

In a job shop scheduling problem, job-based representation is common scheme to encode a solution. Each individual represents a precedence feasible order of production tasks. The order of a production task is randomly generated between 1 and a total number of production tasks accepted for realization. By scanning the permutation from left to right, the occurrence of production task number (gene) indicates the priority of the task. To transform the individual to a feasible solution a production task (according to the permutation) is scheduled at the earliest feasible time according to the precedence and resource constraints.

2.2. Initialization

Genes, stored in the DNA Library, represents tasks accepted in the production system for realization. A set of randomly generated solutions (permutation representations of tasks) serves as the initial population $\eta$. In GA, an individual represents a solution of the scheduling problem – a schedule, while a fitness function is a measure used to estimate the quality of the schedule. The fitness function $FF(p_\eta)$ of individual $p_\eta$ is a linear function of four criteria: makespan $C_{max} \rightarrow \min$; total tardiness $T \rightarrow \min$; flow time $F \rightarrow \min$ and idle time $I \rightarrow \min$. A weight is attached to each criterion. The criteria weights are arranged according to the hierarchy of criteria validity. Constant weights assigned to the criteria make the search direction constant in the multidimensional criteria space. The fixed weights assigned to the criteria make it possible to compare solutions achieved by the GA and MOIA or CSA.
Step 1: Genes coding in the DNA Library
Step 2: Generation of initial population $\eta$

While number of iterations $L (l=0, \ldots, L)$ is higher than 0 in the genetic selection process do

Step 3: Evaluate individuals $p_\eta$ from $\eta$ using the fitness function,

$$FF(p_\eta), \text{ evaluate average fitness function } \bar{FF}(\eta) \text{ for population } \eta,$$

Increase the migration threshold if $FF_{l-1}(\eta) \leq \bar{FF}_{l}(\eta)$

Step 4: clone initial population $\eta$ in order to create matting population $C$

Step 5: Set matting individual $c_k$ from $C$ for each $p_\eta$

Step 6: Crossover in order to create offspring population $C'$

Step 7: $p_\eta \leftarrow c_k^*$ if $FF(c_k^*) < FF(p_\eta)$ in the selection process

Step 8: Shift Mutation in order to create offspring population $C^{**}$, Apply Displacement Mutation if the migration threshold $> mn$,

Step 9: $p_\eta \leftarrow c_k^{**}$ if $FF(c_k^{**}) < FF(p_\eta)$ in the selection process

Step 10: The number of the best individuals $\lambda$ survive in $\eta$ for criteria: $C_{\max} \rightarrow \min; T \rightarrow \min; F \rightarrow \min$ and $I \rightarrow \min$. Remaining individuals are randomly selected in $\eta$

Figure 1. Pseudo code of the GA algorithm

Migration mechanism
The average quality of the initial population is computed in order to evaluate if the population has ability to adopt to changing environment. If the changing conditions are not followed by the population by some number of generations, the population is in the danger of dying out and is forced to the migration to another region. This phenomenon is adopted in the GA. The Displacement mutation (DM) is applied if the average quality of population does not improve after a number of generations $(nm)$, else, the Switching mutation is used. In the DM, a substring of genes is selected at random and inserted into a randomly selected position of the chromosome. By using the DM, the probability the genetic material losts is higher. In Shift mutation (SM) procedure, a task (gene) is randomly selected, then is swapped with the preceding gene. By using the SM, the emphasis of losing the genetic material is lower.

2.3 Chromosom differentiation and selection
The parents pool is created by copying the initial population and pariring of the most-matched individuals. The process of an individual selection is the same as proposed in [4]. The fitness function is transformed into a new fitness equation:

$$new_{-}fit(c_k) = 1 - \frac{FF(c_k)}{S}$$ (1)

$$S = \sum_{k=1}^{K} FF(c_k)$$

The new fitness parameter is converted into frequency of selection $fr(p_k)$:
The probability that individual $c_k$ survives and evolves depends on accumulation:

$$a(c_k) = \begin{cases} fr(c_k), & \text{if } k = 1 \\ a(c_{k-1}) + fr(c_k), & \text{else if } k > 1 \end{cases}$$

A number between 0 and 1 is randomly selected for each individual. Individual $c_k$ is the second parent if the following condition is met:

$$a(c_k) \leq l \geq a(c_{k+1})$$

The proposed procedure guarantees that the most-matched individual is selected. The most-matched individuals are involved in a reproduction procedure, and their children inherit the best features, coded in genes. The most-matched chromosomes have many copies, the worst of them are dying. Parents undergo the Order Crossover (OX) procedure in order to construct new solutions. OX procedure begins with the selection of the gene subsequence in the chromosome of the first parent. Offspring is produced by copying the selected gene upstream in the corresponding positions of its chromosome. The selected genes are removed from the second parent's chromosome. As a result, genes required to complete the offspring are achieved. Going from left to right, genes are copied according to the sequence resulting from the chromosome of the second parent [5,6]. The OX procedure ensures that there are no relative positions between genes, even between extreme positions. In the elite selection procedure, the best individual does not change from the pair: parent and child. The fitness function is evaluated for each individual, providing fitness values, which are then normalized. Normalization means dividing the fitness value of each individual by the maximum value from all fitness values, so that the resulting fitness values are never greater than 1. The parents undergo Displacement or Switching mutation. Also, the elite selection is repeated. In a generation, the best individuals are unchanged in the next generation.

### 2.4. Ordering selection procedure

In the ordering selection procedure, a certain number $\lambda$ of the best individuals for each criterion create a new initial population. The remaining individuals are randomly selected on a feasible solution space. High selection pressure is balanced with random generation of chromosomes in order to escape from a local optima.

### 2.5. Terminal condition

Running a number of iteration is a termination condition. The termination condition is the same as in the case of immune algorithms. This enables the comparison of the algorithms. The best solution, which is said to be the optimal or close to the optimal solution, is in the last generation.

### 3. A job shop scheduling problem

This Section presents a job shop (JS) scheduling problem with disruptions, for experimental study. 15 jobs have to be performed on 10 machines (15x10) in the JS scheduling problem. The input data for the JS scheduling problem (15x10) is presented in (Paprocka, 2016). The objective is to achieve a feasible schedule for four objective functions: makespan $C_{max} \rightarrow \text{min}$; flow time $F \rightarrow \text{min}$; total tardiness $T \rightarrow \text{min}$ and idle time $I \rightarrow \text{min}$. In order to make possible the comparison of the two algorithms, a decision maker defined the priorities of criteria. The priority (weight) of 1st and 3rd criterion equals 0.3, the priority of 2nd and 4th criterion equals 0.2.

The first machine is the most heavily loaded. The increased probability of the bottleneck failure occurs in time horizon: $[a, b+MTTR]$ where: $a = 60$ and $b = 72$ and, $MTTR = 6$. The failure free time of
the bottleneck MTTF equals 66. After a disturbance each operation \( v_j \) can be rescheduled on a machine from a set of parallel machines described in the Matrix of Parallel Machines (MPM) (Paprocka, 2016).

The question arising in such a situation is: do the methods of generating basic schedules influence over the stability and robustness of predictive and reactive schedules generated in the second and third step of the H-MOIA? In the next Section, the results of computer simulations with the application of the GA, for the basic scheduling problem are presented. The influence of the quality of basic schedules over the quality of predictive and reactive schedules is investigated. The objective is to find a stable and robust schedule in the event of a machine failure.

4. Results of computer simulations

Input parameters of the GA are: the number of criteria used to estimate schedules, \( O \); size of the initial population (\( \eta = z \cdot O \), \( O \), \( z \) is the number of individuals), a number of generations necessary to start the migration, \( nm \); number of individuals copied into the next generation, \( \lambda \); number of iterations \( L \), \( (l = 0, \ldots, L) \), number of crossover points, \( nc \) (represents the number of genes in the subsequence of the chromosome). Computer simulations were run for the size of the initial population \( (\eta = z \cdot O = 6 \cdot 4 = 24) \) and number of iterations \( L = 30 \). For comparison, in the MOIA, the size of the initial population and the number of iterations were the same. In the CSA, the number of iterations equalled to 30 and the population of clones equalled to 40. Six simulations were run for the MOIA and CSA, for the above input data and for an affinity threshold equalled 8 and 80. In the GA, three computer simulations are run for each combination of input data: number of generations necessary to start the migration, \( nm = 4, 8 \); number of individuals copied into the next generation, \( \lambda = 2, 4 \); number of crossover points, \( nc = 2, 4 \). Thus, 24 computer simulations are needed.

| No simulation | The priority rule of the basic schedule | The quality of the schedule | Normalized FF | FF |
|---------------|----------------------------------------|-----------------------------|---------------|----|
| \( nc=2 \)    |                                        | \( C_{max} \)                | \( F \)       | \( I \) | \( T \) |               |               |
| 1             | 13 5 2 14 8 1 6 11 7 0 4 9 10 12 3     | 121                          | 438           | 678   | 11     | 0.63604       | 262.8         |
| 2             | **12 14 2 10 6 5 1 9 0 11 7 4 3 8 13** | **117**                      | **492**       | **638** | 0     | 0.608        | **261.1**     |
| 3             | 2 5 14 1 6 11 0 3 8 4 7 10 9 12 13     | 117                          | 506           | 638   | 0      | 0.6501        | 263.9         |
| \( nc=4 \)    |                                        |                             |               |       |        |               |               |
| 1             | 5 3 2 10 9 1 4 6 7 0 12 4 1 11 8 13    | 118                          | 525           | 648   | 1      | 0.6365        | 270.3         |
| 2             | 5 3 2 10 9 1 4 6 7 0 12 4 1 11 8 13    | 118                          | 525           | 648   | 1      | 0.6365        | 270.3         |
| 3             | 6 5 7 14 10 2 0 3 12 4 1 11 9 8 13     | 117                          | 517           | 638   | 4      | 0.6639        | 267.3         |

| No simulation | The priority rule of the basic schedule | The quality of the schedule | Normalized FF | FF |
|---------------|----------------------------------------|-----------------------------|---------------|----|
| \( nc=2 \)    |                                        | \( C_{max} \)                | \( F \)       | \( I \) | \( T \) |
| 1             | 12 14 6 5 2 0 10 1 8 4 11 7 3 9 13    | 117                          | 498           | 638   | 0     | 0.6653        | 262.3         |
| 2             | 5 2 1 14 4 6 7 10 0 8 3 9 11 12 13     | 117                          | 505           | 638   | 0     | 0.6722        | 263.7         |
| 3             | 1 1 4 2 5 6 1 0 1 2 3 7 0 4 9 8 11 13 | 117                          | 533           | 638   | 0     | 0.6805        | 269.3         |
In order to achieve the basic schedule for JS problem (10x15) using the 

| No simu | C max | F | I | T | Normalized FF | FF |
|---------|-------|---|---|---|----------------|----|
| 1       | 117   | 504 | 638 | 0 | 0.6302         | 263.5 |
| 2       | 117   | 498 | 638 | 3 | 0.6201         | 263.2 |
| 3       | 117   | 502 | 638 | 0 | 0.6430         | 276.6 |

Table 3. The best basic schedules achieved by the GA and for 

\lambda = 24, L = 30, nm = 8, \lambda = 2, nc = \{2, 4\}. 

| No simu | C max | F | I | T | Normalized FF | FF |
|---------|-------|---|---|---|----------------|----|
| 1       | 117   | 504 | 638 | 0 | 0.6302         | 263.5 |
| 2       | 117   | 498 | 638 | 3 | 0.6201         | 263.2 |
| 3       | 117   | 502 | 638 | 0 | 0.6430         | 276.6 |

Table 4. The best basic schedules achieved by the GA and for 

\lambda = 24, L = 30, nm = 8, \lambda = 4, nc = \{2, 4\}. 

In order to achieve the basic schedule for JS problem (10x15) using the GA, three simulations were run for input data: \lambda = 24, L = 30, nm = 4, \lambda = 2, nc = \{2\} and for nc = 4 . Achieved solutions are presented in table 2. In the first simulation, the best basic schedule was generated according to the rule of 

\{13 5 2 14 8 1 6 11 7 0 4 9 10 12 3\}. The quality of the priority list based schedule is C max = 121, F = 438, I = 678 and T = 11. Since, the value of Normalized FF depends on the maximal quality value achieved in a population, we use fitness function FF in order to compare the solutions achieved for various simulations. The fitness function of the first solution equals 262.8. Achieved results for computer simulations are presented in tables 2-4.

Using the GA, the best basic schedule was generated in the second simulation for input data: \lambda = 24, L = 30, nm = 8, \lambda = 4, nc = 4 (table 4). The best basic schedule was generated according to the rule of 

\{2 13 10 14 5 1 4 7 6 0 8 3 9 11 12\}. The quality of the best basic schedule equals to FF = 259.1, for the components: C max = 121, F = 436, I = 678 and T = 0.
In the next Section, the experimental results of computer simulations using the second and third step of the H-MOIA are given. Predictive schedules are generated using two best rules: \{2 13 10 14 5 1 4 7 6 0 8 3 9 11 12\} (table 4) and \{12 14 2 10 6 5 1 9 0 11 7 4 3 8 13\} (table 1).

4.1. MIDOS
First, the PS is generated using the second stage of the H-MOIA. Input data to the second stage of H-MOIA+MIDOS constitute the best two basic schedules generated by the GA. It was predicted that operations performed on machine \(w = 1\) can be disturbed in the time period \([a, b+MTTR]\). Disturbed batches \(\tilde{s}_j\) is deleted from the basic schedule.

### Table 5. The influence of the basic scheduling method over the quality of predictive schedules achieved for the JS problem (15x10).

| No | Basic scheduling method | Predictive scheduling method | The priority rule of the basic schedule | C\(_\text{max}\) | F | T | I | FF |
|----|-------------------------|-----------------------------|-----------------------------------------|-------------|---|---|---|----|
| 1  | GA                      | MIDOS                       | 2 13 10 14 5 1 4 7 6 0 8 3 9 9 11 12   | 123         | 520| 0 | 692| 279.3 |
| 2  | MIROS                   |                             | 12 14 2 10 6 5 1 9 0 11 7 4 3 8 13    | 109         | 614| 0 | 552| 265.9 |

### Table 6. The influence of the basic scheduling method over the quality of reactive schedules achieved for the JS problem (15x10) (where: PS - predictive scheduling, RS - reactive scheduling).

| No | Basic + PS method | RS method | The priority rule of the basic schedule | C\(_\text{max}\) | F | T | I | FF | SR | QR |
|----|-------------------|-----------|-----------------------------------------|-------------|---|---|---|----|----|----|
| 1  | GA + MIDOS        | MIROS     | 2 13 10 14 5 1 4 7 6 0 8 3 9 9 11 12   | 123         | 520| 0 | 689| 278.7 | 6 | 0 |
| 2  |                   |           | 12 14 2 10 6 5 1 9 0 11 7 4 3 8 13    | 109         | 614| 0 | 549| 265.3 | 2 | 0 |

In the PS, the technical inspection of the bottleneck is scheduled at the time period \([\text{MTTF}, \text{MTTF}+\text{MTTR}]\) = \([66, 66+6]\]. Next, the most flexible operation of each deleted job is scheduled in the time period \([60, 72+6]\) \(\notin [\text{MTTF}, \text{MTTF}+\text{MTTR}]\). For the remaining operations, backward and forward scheduling algorithms are applied. The quality of predictive schedules obtained at the second stage of the H-MOIA, is presented in table 5.

4.2. MIROS
The question which arises in such a situation is what happens if the bottleneck fails before the planned maintenance work?. In order to answer the above question, reactive schedules were generated using the MIROS for the predictive schedules (obtained using the GA+MIDOS). Reactive schedules are evaluated using the criteria: solution robustness SR and quality robustness QR [1]. For the assumption that the real MTTF of the bottleneck equals 63, the detailed results achieved for the job shop scheduling problem (15x10) using different basic schedules are presented in table 6.

Taking into account the fitness function of criteria of \(C_{\text{max}}, F, T\) and \(I\), the best basic schedule is according to the rule of \{2 13 10 14 5 1 4 7 6 0 8 3 9 11 12\} (table 4). However, further analysis indicated that the predictive schedules generated by the MIDOS and for the rule of \{12 14 2 10 6 5 1 9
0 11 7 4 3 8 13} absorbs the effect of the bottleneck failure more efficiently, taking into account the quality robustness SR (table 6). The best predictive schedule achieved by the GA+MIDOS is presented in figure 2.

![Figure 2. The best predictive schedule achieved by the GA+MIDOS.](image)

5. Results

Using the MOIA, the best basic schedule was achieved for the priority list \{10 14 2 5 3 1 0 4 7 8 9 6 11 12 13\}, the quality of the schedule equals to $FF = 261.3$ [3]. Using the CSA, the best basic schedule was achieved for the rule of \{7 4 2 0 6 5 8 1 11 12 14 3 9 13 10\}, the quality of the schedule equals to $FF = 291.2$ [3]. Using the GA, the best basic schedule was achieved for the rule of \{2 13 10 14 5 1 4 7 6 0 8 3 9 11 12\}, the quality of the schedule equals to $FF = 259.1$. Comparing the meta-heuristic algorithms for the basic schedule generation, the GA achieved better solution than immune algorithms.

Evaluating predictive schedules, achieved using genetic (table 5) and immune algorithms (table 5 [3]), using the fitness function of criteria of $C_{\text{max}}, F, T$ and $I$, the best predictive schedule was achieved by the GA+ MIDOS. The best predictive schedule was achieved for the rule of \{12 14 2 10 6 5 1 9 0 11 7 4 3 8 13\} (table 5). The quality of the schedule equals to $FF = 265.9$. Thus, applying a better basic schedule to the MIDOS procedure does not always allow us to achieve better predictive schedule.

Evaluating reactive schedules, achieved using genetic (table 6) and immune algorithms (table 6 [3]), using the fitness function, SR and QR, the best reactive schedule was achieved by the GA+ MIDOS+MIROS. The quality of the schedule is: $FF = 265.3$, SR=2 and QR=0. The predictive schedules, generated by the GA+MIDOS+MIROS, are more stable and robust than schedules generated by the immune algorithms.

The following research point needs to be investigated: which construction of algorithms achieves better solutions: creating a schedule which is robust for a disturbance (predictive scheduling) or creating a schedule which is the result of studies on the influence of rescheduling policies on the performance of a dynamic manufacturing system (predictive-reactive scheduling). The immune and genetic algorithms were investigated for the job shop scheduling problem. Also the algorithms should be compared with other meta-heuristic method [7] and should be applied for other manufacturing cells [8, 9, 10].

6. References

[1] Paprocka I and Skolud B 2017 J. Sched. 20 (2) 165-182 
[2] Paprocka I and Kempa W M 2016 Searching for a Method of Basic Schedules Generation which influences over the Performance of Predictive and Reactive Schedules. Information Systems Architecture and Technology: Proceedings of 37th International Conference on
Information Systems Architecture and Technology – ISAT 2016 – Part IV Advances in Intelligent Systems and Computing 524 233-242

[3] Paprocka I 2016 On the Quality of Basic Schedules Influencing over the Performance of Predictive and Reactive Schedules. Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology – ISAT 2016 – Part IV Advances in Intelligent Systems and Computing 524 243-253

[4] Liao G-Ch 2006 Short-term thermal generation scheduling using improved immune algorithm, Electric Power Systems Research 76

[5] Liaw Ch-F 2000 A hybrid genetic algorithm for the open shop scheduling problem European Journal of Operational Research 124 28-42

[6] Arroyo J E C and Armentano V A 2005 Genetic local search for multi-objective flowshop scheduling problems European Journal of Operational Research 167

[7] Kalinowski K and Skolud B 2017 The Concept of Ant Colony Algorithm for Scheduling of Flexible Manufacturing Systems. International joint conference SOCO’16- CISIS’16-ICEUTE’16 Book Series: Advances in Intelligent Systems and Computing 527 408-415

[8] Skolud B, Krenzcyk D, Kalinowski K, Ćwikla G and Grabowik C 2017 Integration of manufacturing functions for SME. Holonic based approach. [in] [ed.: Janusz Kacprzyk] Advances in Intelligent Systems and Computing 527 Springer 464-473

[9] Sękala A, Gwiazda A, Foit K, Banaś W, Hryniewicz P and Kost G 2015 Agent-based models in robotized manufacturing cells designing. IOP Conf. Ser.: Mater. Sci. Eng. 95 012106

[10] Monica Z 2015 Virtual modelling of components of a production system as the tool of lean engineering IOP Conf. Series: Materials Science and Engineering 95 012109

[11] Zolkiewski S 2014 Diagnostics and transversal vibrations control of rotating beam by means of Campbell diagrams. Key Engineering Materials 588 91-100

[12] Płaczek M, Buchacz A, Wróbel A 2015 Use of piezoelectric foils as tools for structural health monitoring of freight cars during exploitation, Eksploatacja i Niezawodność-Maintenance and reliability 17 (3) 443-449