

# A hybrid multi-objective immune algorithm for predictive and reactive scheduling

Iwona Paprocka<sup>1</sup> · Bożena Skołod<sup>1</sup>

Published online: 26 September 2016  
© The Author(s) 2016. This article is published with open access at Springerlink.com

**Abstract** The high productivity of a production process has a major impact on the reduction of the production cost and on a quick response to changing demands. Information about a failure-free machine operation time obtained in advance allows the users to plan preventive maintenance in order to keep the machine in a good operational condition. The introduction of maintenance work into a schedule reduces the frequency of unpredicted breaks caused by machine failures. It also results in higher productivity and in-time production. The foregoing of this constitutes the main idea of the predictive scheduling method proposed in the paper. Rescheduling of disrupted operations, with a minimal impact on the stability and robustness of a schedule, is the main idea of the reactive scheduling method proposed. The first objective of the paper is to present a hybrid multi-objective immune algorithm (H-MOIA) aided by heuristics: a minimal impact of disrupted operation on the schedule (MIDOS) for predictive scheduling and a minimal impact of rescheduled operation on the schedule (MIROS) for reactive scheduling. The second objective is to compare the H-MOIA with various methods for predictive and reactive scheduling. The H-MOIA + MIDOS is compared to two algorithms, identified in reference publications: (1) an algorithm based on priority rules: the least flexible job first (LFJ) and the longest processing time (LPT) (2) an Average Slack Method. The H-MOIA + MIROS is compared to: (1) an algorithm based on priority rules: the LFJ and LPT and (2) Shifted Gap-

Reduction. This paper presents the research results and computer simulations.

**Keywords** Predictive · Reactive scheduling · Multi-objective optimization · Immune algorithm · Preventive maintenance

## 1 Introduction

The term 'scheduling' refers to the allocation of a set of jobs, materials, tools and workers to predefined number of machines in order to optimize one or more objective functions. All input data are known in advance in deterministic scheduling problems. For example, when a set of tasks are assigned using a specific number of machines, processing times of operations are constant, and there is no disruption during a scheduling horizon on a shop floor (Wosik and Skołod 2009). Due to disruptions which may appear on the shop floor (a machine failure, job priority changes, material unavailability etc.), it is more realistic to build a predictive schedule (PS). In this paper, the term 'disruption' represents a machine failure. The paper presents the hybrid multi-objective immune algorithm (H-MOIA) aided by heuristic: minimal impact of disrupted operation on the schedule (MIDOS) for predictive scheduling. The advantage of this method is in the introduction of maintenance work into a schedule, taking into consideration a predicted time of failure. The introduction of maintenance into a schedule reduces the frequency of unpredicted breaks caused by a machine failure and enables higher productivity, as well as in-time production. In general, predictive schedules (PS) are robust if they absorb the effects of disturbances which occur (Liu et al. 2007; Al-Hinai and ElMekkawy 2011).

✉ Iwona Paprocka  
iwona.paprocka@polsl.pl  
Bożena Skołod  
bozena.skolud@polsl.pl

<sup>1</sup> Institute of Engineering Processes Automation and Integrated Manufacturing Systems, Silesian University of Technology, ul. Konarskiego 18a, 44-100 Gliwice, Poland

A reactive schedule (RS) is generated if effects of a disturbance are excessively large and disruption-affected tasks need to be rescheduled (Hasan et al. 2011; Turkcan et al. 2009; Grabowik and Kalinowski 2011). The paper presents the hybrid multi-objective immune algorithm (H-MOIA) aided by heuristic: minimal impact of rescheduled operation on the schedule (MIROS) for reactive scheduling. The advantage of this method is in introducing as few changes as possible after a disturbance in a schedule. This benefit can be achieved using two criteria for the evaluation of reactive schedules, ie stability and robustness.

Having reviewed the relevant literature, the authors continued the classification process proposed by Goren and Sabuncuoglu (2008). We have classified predictive and reactive scheduling algorithms and the problems the algorithms are applied to. Predictive and reactive scheduling algorithms are applied to various scheduling problems, such as a single machine scheduling problem (Liu et al. 2007; Pan et al. 2012; Goren and Sabuncuoglu 2008), a flow shop, permutation flow shop, job shop (Hasan et al. 2011; Al-Hinai and ElMekkawy 2011; Wosik and Skołod 2009; Paprocka et al. 2014), a general shop, parallel (Turkcan et al. 2009; Duenas and Petrovic 2008) and an open job shop scheduling problem (Smutnicki 2002) (Table 1). This paper focuses on the search for the PS and RS for flow shop and job shop scheduling problems. The literature proposes two scheduling methods to deal with static and dynamic problems. Deterministic or static (off line) scheduling takes place where a schedule is generated in advance. Dynamic (on line) scheduling refers to situations when a schedule is generated or a control decision is made after a disruption (Guilherme et al. 2003) (Table 1). This paper is concerned with static scheduling, in the case of a predictive scheduling problem, and dynamic scheduling, in the case of a reactive scheduling problem.

A Genetic Algorithm (GA) is the most popular algorithm used for solving scheduling problems (Table 1). The authors selected the GA as a method strong enough to escape from a local optimum (Liu et al. 2007). The comparison of the two algorithms, the GA and H-MOIA, leads to the conclusion that they offer a high diversity of solutions (Skołod and Wosik 2007). The migration mechanism (embedded in the evolution process of the GA) and a suppression mechanism (embedded in the H-MOIA) enable the search of various areas in the solution space.

The first objective of the paper is to present the H-MOIA aided by innovative methods, namely the MIDOS for predictive scheduling and the MIROS for rescheduling. The second objective is to compare the H-MOIA with methods for predictive and reactive scheduling which had already been identified in the related reference publications.

A critical review of the literature allowed the identification of benchmark approaches to predictive and reactive scheduling. Three algorithms identified for predictive scheduling

include (1) an algorithm generating a basic schedule and inserting a time buffer prior to a job with a disturbance prediction (2) an algorithm based on priority rules: the least flexible job first (LFJ) and the longest processing time (LPT) and (3) the Average Slack Method (ASM) (Table 1). The first algorithm was proposed by Lin et al. (2007). The basic schedule is generated regardless of a disturbance, optimizing one or more objective functions. Afterwards, a time buffer is inserted prior to a job with a disturbance prediction. The use of this method entails three problems, (a) for which machine a failure-free time should be predicted, (b) where in a schedule a time buffer should be inserted and (c) how long the time buffer should be. A predictive schedule generated with the use of this method can be inefficient when taking into account the machine utilization criteria. The model proposed by the authors of the paper provides the following solutions to the three problems mentioned above, (a) the prediction of a failure-free time for a bottleneck/s in a production system, (b) building a description of failure-free time using a probability distribution function based on historical data and the life stage of the bottleneck, and (c) insertion of a time buffer for maintenance in the PS only. The second algorithm was proposed by Duenas and Petrovic (2008), who handle the uncertainty by adding idle windows to the predefined processing times of the affected tasks with disturbance prediction. Uncertain input data are expressed using linguistic terms such as “the number of disruption occurrences is much higher than  $n$ ” and modelled using fuzzy logic. The parameters are specified imprecisely, although they could be obtained using the Manufacturing Execution Systems (MES) and based on maintenance records. Moreover, the evaluation of the stability and robustness of a schedule is still an open-ended issue as the starting and completion times of operations are unavailable before all tasks have been processed. Having obtained the processing times of the jobs with disturbance prediction, it is possible to apply two priority rules, those being the LFJ and LPT. The advantages of these priority rules include low computation complexity and easy implementation. The third algorithm was proposed by Goren and Sabuncuoglu (2008) who deal with the issue by proposing the initial schedule with the best performance in the event of a disruption. The performance of the schedule is computed by adding the initial performance measure of the schedule and the degradation in the performance measure (the slack) due to random disruptions. Input data are expressed using probability density functions known in advance. There is no explanation given for using different kinds of probability density functions. The parameters of distribution functions are arbitrarily selected in two cases, a long and a short failure-free time. Goren and Sabuncuoglu (2008) accept the value of parameters generating the most robust schedules. The scheduling algorithm consists of two parts, a sequence generator and a sequence evaluator. The Tabu Search algorithm

is applied to effectively scan the solution space. The advantage of the AMS is a high correlation between the stability and the average slack value of schedules.

Four algorithms identified for reactive scheduling include (1) Right Shifting (RSh), (2) Rescheduling Disrupted Operations to parallel machines available first (RDO), (3) rescheduling based on priority rules: the LFJ and LPT and 4) Shifted Gap-Reduction (SG-R) (Table 1). The first method maintains the same sequence as the initial schedule and consumes little computational power (Liu et al. 2007; Wu et al. 1993; Al-Hinai and ElMekkawy 2011; Goren and Sabuncuoglu 2008). The second method, proposed by Jai and Elmaraghy (1997), reschedules only disruption-affected operations in order to maintain the stability of the existing schedule and provide quick solutions. However, this may result in a less effective overall schedule if compared with the complete rescheduling. Furthermore, this method does not plan a disruption (maintenance) or re-scheduling in advance, and does not collect experience from past circumstances. The third method was proposed by Duenas and Petrovic (2008). The authors have built a new schedule with all the tasks that have not been processed yet, considering the moment of material shortage to be the earliest possible starting time of all the affected jobs. Building a new schedule using the LFJ and LPT rules prove to be better for efficiency objectives (such as makespan), but not for stability objectives. The fourth method was proposed by Hasan et al. (2011). In the SG-R algorithm, a disrupted operation fills a gap, provided that the gap is large enough to accommodate the operation, without creating infeasibility. The gap can also be filled (if it is shorter than the operation duration by a predefined tolerance limit) by shifting the operation to the right of the gap. The authors re-scheduled the affected tasks, for the remaining operations, from the point of breakdown. When applying this algorithm, the authors noticed that the impact of a breakdown was smaller where a machine failure-free time was known in advance rather than when the breakdown was sudden. The problem of reactive scheduling is complex when a breakdown is unpredictable. If a breakdown occurs at an early stage of a schedule, the affected tasks can be rescheduled with only a small increase in the makespan. The SG-R algorithm dominates the RSh if the makespan minimization is a criterion.

This paper focuses on the development of a new algorithm for predictive and reactive scheduling (the H-MOIA). The H-MOIA consists of three stages, i.e. the first for basic schedule generation, the second for predictive scheduling (H-MOIA+MIDOS) and the third for reactive scheduling (H-MOIA+MIROS). The determination of uncertain input data is still an open-ended issue. This is because data are expressed on the basis of vague or imprecise knowledge within the literature (Duenas and Petrovic 2008). In the method proposed by the authors of this paper, uncertain input data for the bottleneck, such as the MTTR and the MTTF,

are predicted on the basis of maintenance records related to historical failure-free times and repair times. Data based on maintenance records are used to fit a shape of the probability distribution, of both busy time and repair duration of the machine.

In the MIDOS, a maintenance task is introduced into a schedule, taking into consideration the predicted time of failure occurrence. The evaluation of the stability and robustness of the schedule remains a problematic issue. The starting and completion times of operations are unavailable before all tasks have been executed in the method which applies the insertion of idle time prior to each job with a disturbance prediction (Liu et al. 2007; Duenas and Petrovic 2008).

The MIROS chooses between two rescheduling methods, the RSh and RDO, in order to select a better solution for stability and robustness. The elaboration of the method that provides efficient schedules, not only for the makespan criterion, but also for the stability and robustness also is complex (Duenas and Petrovic 2008).

The authors of the paper intend to develop the existing studies by other methods based on immunology and heuristics. On the basis of overview of reference publications, the following research points have been identified:

- (1) methods of dealing with uncertainty and enabling the obtainment of best performance schedules, when the objective is to minimize makespan, flow time, total tardiness and idle time.
- (2) methods dealing with predicted or unexpected interruptions in an efficient manner, when the objective is to maximize stability and robustness.

The efficiency of the solution achieved using the H-MOIA + MIDOS is compared to the efficiency of solutions achieved using (1) LFJ and LPT and (2) ASM. The efficiency of the solution achieved using the H-MOIA + MIROS is compared to the efficiency of solutions achieved using (1) the algorithm based on priority rules, such as the LFJ and LPT and (2) SG-R.

The paper is organized as follows: a production scheduling problem is described in the next Section. The first stage of the H-MOIA, for basic scheduling, is presented in Sect. 3. The second and third stages of the H-MOIA, for predictive and reactive scheduling, are presented in Sect. 4. Methods for predictive and reactive scheduling, identified in the related literature, are also presented in Sect. 4. The criteria for predictive and reactive scheduling are described in Sect. 5. The flow shop and job shop scheduling problems, together with interruptions for experimental study, are presented in Sect. 6. Section 7 contains an exemplary flow shop scheduling problem generated to illustrate the steps of the proposed predictive and reactive algorithm. Section 8 contains necessary analyses and experimental test results related to the research on the

**Table 1** Literature review related to predictive and reactive scheduling algorithms

Paper	[1]	[6]	[7]	[8]	[5]	[9]	[10]
Shop floor	Single machine	Job shop with parallel machines	Single multi-functional machine	Parallel machine	Job shop	Parallel machines	Single machine
S/D	Dynamic(D)	Static (S)	S	S	S	S	S
Stochastic/deterministic	Stochastic	Stochastic	Deterministic	Stochastic	Stochastic	Stochastic	Stochastic
Method	Two-stage multi-population GA	Two-stage GA	Priority rule-minimal job tardiness	Linear-relaxation-based algorithm	GA with local search procedure	Dispatching rules: LFJ and LPT	Tabu search-based algorithm based on ASM
When	ADNA	ADNA		ADNA-after a disturbance not absorbed by the PS			
Scheme	Off line	Off Line		Off line	Off line	Off line	Off/on line
Rescheduling policy	Right shifting	Right shifting		Complete rescheduling and controllable processing times estimation	Shifted gap-reduction	Left shift for disrupted job or complete rescheduling using LFJ and LPT rules	Right shifting
Nouber of objectives	Bi-criteria, SORS-scalar objective function of robustness and stability		Single minimizing the maximum tardiness of a job	Bi-criteria, convex function of processing times (cost) and total weighted earliness and tardiness	M-makespan	M for predictive scheduling, M and stability for rescheduling	Robustness, stability and SORS
Robustness	Minimizing total weighted tardiness of jobs	Bi-criteria, resource utilization and flexibility to adopt changes					M or flowtime or total tardiness
Stability	SAD-sum of absolute deviations between planned job completion times and the ones realized			SAD, and cost of changing between planned schedule and RS		Sum of absolute deviations between job start times in PS and RS	SAD
Disturbance	MF-machine failure		MF, degradation	MF, order cancellation, arrival of a high priority job, changing of due date	MF	Material shortage	MF

application of predictive and reactive algorithms in job shop and flow shop scheduling problems. The paper concludes with a brief summary of the results (Sect. 9).

## 2 Problem formulation

The job shop production system is described as follows:

- number of jobs  $J, j = 1, 2, \dots, J$  to be executed on a number of machines  $W, w = 1, 2, \dots, W$ . Each job consists of a number of operations  $V_j, v_j = 1, 2, \dots, V_j$ ,
- each operation  $v_j$  can be executed on one machine from a set of parallel machines,
- the bottleneck can be broken,
- each job consists of a number of operations,
- operations must be processed in a pre-specified order,
- jobs are non-preemptive and non-reentrant,
- production batch of a job is predefined,
- deadlines related to the execution of individual batches of jobs are predefined.

In the flow shop production system, the operations of each task are processed in the same sequence on all machines. The remaining conditions are the same as in the job shop production system.

Such production systems are monitored using the MES which allows, the downloading of information about production processes directly from machines. Acquired data include a mode of disruption, availability time, disability time (the time when a machine is incapable of work due to a disturbance) and the number of disturbances (Ćwikła 2014; Janik and Gendarz 2011). Having historical data concerning failure-free times and repair times of a bottleneck, it is possible to predict the Mean Time of Repair (MTTR) and the Mean Time To Failure/To First Failure (MTTF/MTTFF) of the machine. The MTTR and the MTTF/MTTFF are described using probability density functions. Parameters of the functions are estimated using the Empirical Moments approach, the Renewal Theory based approach and the Maximum Likelihood approach (Skołud et al. 2011; Kempa et al. 2014). The mathematical description of the production model is presented by Paprocka and Kempa (2012).

For the production system (flow shop or job shop), the basic schedule is generated using the H-MOIA. Having predicted the values of the MTTF and MTTR, the predictive schedule is generated using the H-MOIA+MIDOS.  $Td$  is the real time of the bottleneck failure. The reactive schedule is generated using the H-MOIA+MIROS.

The problem is concerned with generating a predictive schedule, robust for disturbance, created on the basis of historical data about failure-free times and repair times of the bottleneck. If the predictive schedule cannot absorb the effect

of a disturbance, the problem is the generation of a reactive schedule. The reactive schedule should introduce as few changes as possible into the previous schedule.

The efficiency of the basic and predictive schedules is evaluated using such criteria as makespan  $C_{\max}$ , flow time  $F$ , total tardiness  $T$  and idle time  $I$ . The efficiency of reactive schedules is evaluated using stability and robustness criteria described in Sect. 4.3 (6,7).

## 3 The basic schedule generation

The basic schedule is generated by applying the hybrid multi-objective immune algorithm (H-MOIA). Phenomena of an immune system, adopted in the H-MOIA, includes a pathogen representing a scalar objective function (1), an antibody corresponding to a solution of the problem—the schedule with the minimal value of the scalar objective function. Priority rules such as the LPT (Longest Processing Times), RIPS (Random Insertion Permutation Scheme) and EDD (Earliest Due Date) aid the processes of searching for a good quality basic schedule.

Input data to the H-MOIA can be divided into two groups; the first group consists of information about a production system, whereas the second group is composed of the H-MOIA parameters:

1. number of evaluation criteria  $O, o = 1, 2, \dots, O$ ; priority rule of an objective function  $w_o$ ; objective function  $f_o(y)$  used for the evaluation of a schedule, the number of tasks (jobs)  $J, j = 1, 2, \dots, J$ ; the number of machines  $W, w = 1, 2, \dots, W$ ; operation time  $a_{w,v_j}$ ; the number of operations of each job  $j$  performed using the machine  $w, v_j$ ; routes of a job  $j$ ; the batch size of a job  $j, s_j$ ; the due date of a job  $j, d_j$ . A decision-maker can select criteria from makespan  $f_1(y) = C_{\max}$ , flow time  $f_2(y) = F$ , total tardiness,  $f_3(y) = T$  and idle time  $f_4(y) = I$ .
2. size of a sub-population for a single objective scheduling problem,  $z$ ; the size of the initial population for the multi-criteria scheduling problem,  $Y = z \cdot O, y = 1, 2, \dots, Y$  ( $y$ -an antibody); the number of iterations *termcon* (terminal condition); the maximal number of genes undergoing mutation in a hypermutation process; the affinity threshold *affthres* (used for defining if one antibody is similar to another); the stimulation threshold *stimthres* (used for defining the number of similar solutions which can exist in a population).

The H-MOIA is mainly based on an iterative cycle of six steps, i.e. from 3 to 8. The steps of the H-MOIA are as follows:

1. DNA Library generation (*DNAL*). In the *DNAL*, each gene represents the number of a job in a scheduling problem. The number of genes is equal to the number of jobs. An example of the *DNAL* for  $J = 5$  is  $\{1, 2, 3, 4, 5\}$ .
2. Initial population generation (*IP*). The *IP* of antibodies is randomly generated by giving priorities to jobs. The size of *IP* equals  $Y$ . An example of antibody coding is  $\{2,3,5,4,1\}$  and means that job 2 is first in the antibody chromosome and has the highest priority in the schedule. Next, job 3 is scheduled, etc.
3. Fitness function ( $FF_y$ ) evaluation. After decoding (generating the schedule), antibody  $y$  is evaluated using a scalar objective function (1)

$$FF_y = \sum_{o=1}^O f_{s_o}(y) \cdot w_o \quad (1)$$

$$f_{s_o}(y) = f_o(y)/f_o(y^*) \quad (2)$$

where  $y$ —a schedule,  $f_{s_o}(y)$ —the scalar sub-function of schedule  $y$ ;  $o$ —the number of the scalar sub-function,  $o = 1, 2, \dots, O$ ;  $f_o(y^*)$ —the maximal value of scalar sub-function  $o$  of schedule  $y^*$ ;  $w_o$ —priority of objective function  $o$ .

4. Sub-population generation. The *IP* is copied into two sub-populations: the first ( $SCP_o$ ) is evaluated using a single objective function from  $o = 1, 2, \dots, O$ , the second one (*MCP*) is evaluated using the fitness function (1). In the subpopulation of antibodies evaluated using objective function  $f_1(y)$ , the antibody with the lower affinity to the pathogen (a schedule with a lower value of makespan criterion) dies and a new one is generated according to the *LPT* rule. In the subpopulation of antibodies evaluated using objective function  $f_3(y)$ , the schedule with the lower value of the total tardiness criterion is replaced with a new one generated according to the *Earliest Due Date (EDD)* rule. In the subpopulation of antibodies evaluated using the fitness function, the schedule with the lower value of (1) is replaced with a new one generated according to the rules of the *LPT* and the *Random Insertion Perturbation Scheme (RIPS)* (Ponnambalam et al. 2004).
5. Evolution of populations ( $SCP_o$ ),  $o = 1, 2, \dots, O$ .
  - (a) In these sub-populations, new antibodies are generated using *job\_based\_crossover* and *displacement\_mutation* (Cheng et al. 1999). Each time a new antibody is generated, an elite selection process between parent and offspring solutions is performed.
  - (b) Multi-criteria population (*MCP*) evolution. In this sub-population new antibodies are generated using a *hypermuation* (better solutions undergo mutation with lower frequency). After generating a new antibody, the elite selection is performed between the parent and offspring solution.

6. Affinity and stimulation. In order to prevent a premature convergence of the *H-MOIA* to a local optima, it is necessary to use *affihres*. The degree of affinity between antibodies from *SCP* is calculated using the Hamming distance. If the degree of affinity between two antibodies is greater than the *affihres*, one antibody stimulates another as both antibodies are similar. An antibody is deleted from *SCP* if it is stimulated by a number of antibodies more than *stimthres*.
7. Elite selection. The elite selection is performed between antibodies from two sub-populations, i.e. the *SCP* and the *MCP*, in order to create a new *IP*.
8. Immune memory (*IM*). In each iteration, the antibody with the best affinity to the pathogen (a schedule with the best fitness function) is selected from the *IP* and memorized in the *IM*.  
Go to step 3, unless a terminal condition (*termcon*) is met.
9. Local search of the *IM*. In the *IM*, each antibody  $y$  is searched locally in order to achieve the Pareto optimal solution. The neighbourhood of each antibody  $N(x)$  is generated using four mutation procedures, those being *switching mutation*, *insertion mutation*, *reciprocal exchange mutation*, *displacement mutation* (Cheng et al. 1999). An undominated solution  $y'$  is obtained if  $f_o(y) \leq f_o(y')$ ,  $\forall_{o \in O}$ . From the Pareto set of optimal solutions, the best schedule is selected using (1). The structure of the *H-MOIA* is presented in Fig. 1.

## 4 Generation of predictive and reactive schedules

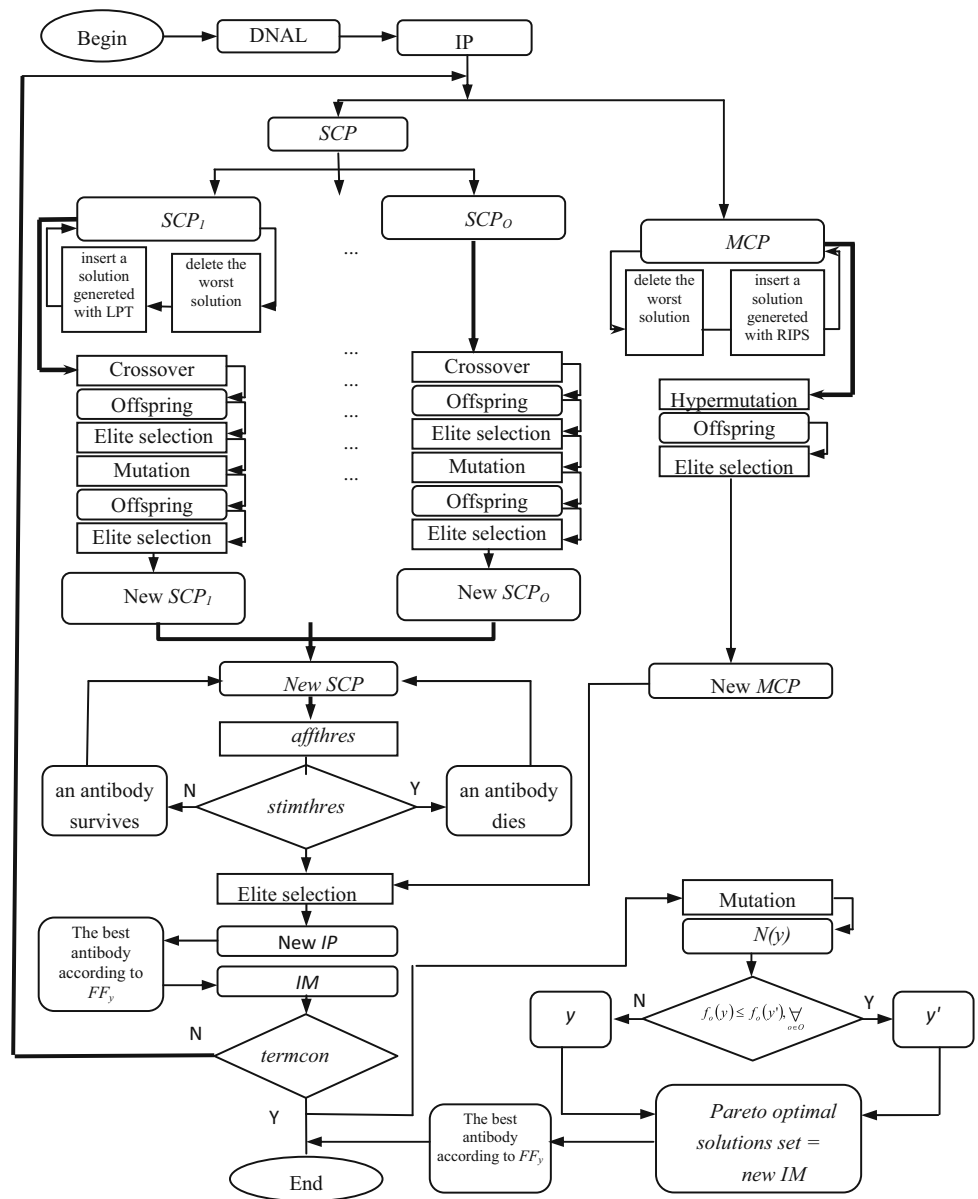
A predictive schedule (*PS*) has two functions. The first function relates to the allocation of jobs to resources in order to optimize one or more objective functions. The second function is to serve as an overall plan under the conditions of external disturbances. A reactive schedule (*RS*) is generated if the effect of a disruption is excessively large.

### 4.1 Predictive scheduling methods

In the related literature there are, mainly, three basic methods of generating a *PS*. The advantages and disadvantages of the methods are described in the Introduction.

1. The basic schedule is generated regardless of a disturbance, by optimizing one or more objective functions and, next, inserting a time buffer prior to a job with disturbance prediction (Liu et al. 2007).
2. The Average Slack Method (*ASM*) can also be applied to generate a robust schedule. The *ASM* performs the evaluation of a schedule using the same criterion as the criteria used for the initial schedule evaluation. However, in this

**Fig. 1** Architecture of the H-MOIA



case, this criterion is increased by the value of deterioration of the criterion due to a disturbance. In the tabu search method, the sequence of jobs is generated and evaluated taking into account the effects of disturbances (Goren and Sabuncuoglu 2008). The best sequence, achieved in an iteration, constitutes the input data for the next iteration.

3. The PS is generated using two dispatching rules, the least flexible job first (LFJ) and the longest processing time (LPT) for parallel machines predictive scheduling problems (Duenas and Petrovic 2008). The LFJ rule identifies the job which can be performed using the lowest number of parallel machines and assigns it to the first available machine. If at least two jobs have the same priority rule, the job with the longest processing time (the LPT rule) has the highest priority. The PS is generated by increasing

a job processing time by a time approximately equalling a repair time.

In the approach proposed by the authors of the paper, the basic schedule is generated using the H-MOIA. The basic schedule constitutes the input data for predictive scheduling. The remaining input data for the second stage of the H-MOIA + MIDOS are as follows:

- the number of machines which can be disrupted-*bottle\_necks*;
- predicted failure-free time—MTTF;
- predicted repair time—MTTR;
- $a$  is estimated on the assumption that the probability of the failure-free time of the *bottle\_neck* is higher than an equalling 40% (Paprocka and Skofud 2013);

- $b$  is estimated on the assumption that the probability of the failure-free time of the *bottle\_neck* is less than  $b$  equalling 70% (Paprocka and Skołod 2013).

A technical inspection of the bottleneck is planned at a time period  $[MTTF, MTTF + MTTR]$ . At the time period  $[a, b + MTTR]$  operations are scheduled according to the rule of the minimal impact of disrupted operation on the schedule (MIDOS).  $[MTTF, MTTF + MTTR]$  ( $MTTF$  mean time to failure,  $MTTR$ —mean time of repair) belongs to  $[a, b + MTTR]$ . Batch  $\hat{s}_j$  of job  $j$ , for which operation is scheduled at the bottleneck at the time of the predicted disturbance  $[a, b + MTTR]$ , is removed from the basic schedule. Time points  $a, b, MTTF, MTTR$  are estimated on the basis of the probability theory (Paprocka and Kempa 2012; Skołod et al. 2011; Kempa et al. 2014). The MIDOS is computed if the operation can be executed on the bottleneck which is a parallel machine. The operation of job  $\hat{j}$ , which is the most flexible and whose disruption causes the smallest number of changes in the schedule, is firstly assigned at the bottleneck

at a time period  $[a, b + MTTR]$ . For the remaining operations of job  $\hat{j}$ , backward and forward scheduling is applied. In the backward scheduling procedure, the operation of job  $\hat{j}$  is scheduled on the first machine available from a set of parallel machines, starting from time point  $a$ . Components of the  $MIDOS_{\hat{v}_j}$  (1) are the following: (1) the number of changes needed to be performed after the disturbance of the operation  $\hat{v}_j$  performed at the bottleneck,  $R_{\hat{v}_j}$ .  $R_{\hat{v}_j}$  equals the number of job  $\hat{j}$  operations which have to be rescheduled; (2) the number of machines on which the disrupted operation can be performed alternatively,  $F_{\hat{v}_j}$ .  $F_{\hat{v}_j}$  equals the number of parallel machines.

$$MIDOS_{\hat{v}_j} = R_{\hat{v}_j} + (W - F_{\hat{v}_j}) \rightarrow \min \quad (3)$$

The PS with the maximum solution and quality robustness can be obtained using  $MIDOS_{\hat{v}_j}$ .

The second stage of the H-MOIA is compared with the modified algorithm based on the priority rules of the LFJ and the LPT (Duenas and Petrovic 2008). The LFJ and LPT are

- For every job  $j$ 
  - For every operation executed at the *bottle\_neck*
    - compute the LFJ
    - compute the LPT
- Forward scheduling according to the LFJ priority rule; if at least two jobs have the same priority, it is necessary to select the job with the longest processing time (according to the LPT rule)

**Fig. 2** Pseudocode of the LFJ/LPT predictive algorithm

- Generate randomly a tabu-solution
- Evaluate the tabu-solution using  $FF_{y^0}$  for the first iteration
- For the number of iteration:
  - Generate a neighborhood of the tabu-solution using: switching mutation; insertion mutation; reciprocal exchange mutation and displacement mutation;
  - For each neighboring solution
    - Decode the solution by generating schedule  $y$ ;
    - Evaluate the schedule using  $FF_y$
    - Generate reactive schedule  $y^\#$  in the case of the bottleneck disturbance occurrence using one heuristic from the following;
      - RSh (HI) – reschedule the disrupted operation of the disrupted job  $\hat{j}$  using heuristic: move to the right. Reschedule successors of the disrupted operation of job  $\hat{j}$  in order to meet the non-preemptive conditions
      - RDO II (HII) - reschedule the disrupted operation of job  $\hat{j}$  on a parallel machine available first. Reschedule successors of the disrupted operation of job  $\hat{j}$  on parallel machines available first in order to meet the non-preemptive conditions
    - Evaluate the reactive schedule using  $FF_{y^\#} = ASM$
  - Select the schedule with the minimal value of the ASM from the Neighborhood,
  - if the selected  $ASM < FF_{y^0}$ , the new tabu-solution is obtained and  $FF_{y^0} \leftarrow ASM$ .

**Fig. 3** Pseudocode of the ASM predictive algorithm



computed for operations which are performed at the bottleneck and can be disturbed. The pseudocode of the LFJ/LPT predictive algorithm is presented in Fig. 2.

The second stage of the H-MOIA is also compared with the modified ASM (Goren and Sabuncuoglu 2008). Figure 3 presents the tabu search method based on the ASM.

### 4.2 Reactive scheduling methods

There are three main strategies to generate the RS, namely after the occurrence of a disturbance, after a rescheduling point (periodic) (Al-Hinai and ElMekkawy 2011) and after the occurrence of a disturbance which is not absorbed by the PS (Table 1). The frequency of rescheduling influences the nervousness of a production flow shop. If the nervousness measure of a schedule is high, the stability of the schedule is low (Wu et al. 1993). Therefore, rescheduling after a disturbance is more commonly used in practice than after a rescheduling point. Four methods for reactive scheduling are identified in the related literature. The advantages and disadvantages of the methods are described in the Introduction.

1. The most popular method referred to in the reference publications is the right shifting (RSh) of jobs affected by a disturbance (Liu et al. 2007; Wu et al. 1993; Al-Hinai and ElMekkawy 2011; Goren and Sabuncuoglu 2008) (Table 1).
2. The second method is rescheduling disrupted operations to parallel machines available first (RDO) (Jai and Elmaraghy 1997).
3. The third approach consists of removing disturbance-affected jobs from a schedule. The remaining jobs undergo the Left Shift (LS) procedure, whereas removed jobs are rescheduled using the LFJ and the LPT dispatching rules (Duenas and Petrovic 2008).
4. The fourth method identifies “time windows” before shifting jobs to the right. Jobs executed after the occurrence of a machine failure are not affected if the RSh of the jobs does not affect successive jobs. Affected jobs are rescheduled using the Shifted Gap-Reduction (SGR) applied in order to re-optimize the makespan criterion. The SGR heuristic identifies any “time window” between any two consecutive operations performed on a machine. If the “time window” is larger, or equals the processing time of the affected operation, the operation is inserted into the time window without violating the preceding constraints of successive job operations. The operation can also be put into the “time window” if the duration of the “time window” is shorter than the operation time, but within the limits of a given tolerance, provided that the algorithm improves the makespan criterion (Hasan et al. 2011).

In the approach proposed by the authors of the paper, the PS constitutes the input data for reactive scheduling at the third stage of the H-MOIA. The real time of the bottleneck failure is the input data for the third stage of the H-MOIA. After the bottleneck failure occurrence, disrupted operation  $\hat{v}_j$  of job  $\hat{j}$  and successive operations of job  $\hat{j}$  are deleted and repair works are performed. Undisrupted jobs are performed according to the PS. The disrupted job is rescheduled according to the following heuristics: I) moving to the right (RSh) provided that the non-preemptive condition has been met, II) reschedule operation to a parallel machine first available (RDO) provided that the non-preemptive condition has been met. Taking into account the quality robustness (QR) criterion (see next Sect. 7), the RSh obtained using the RDO proves more convenient. Keeping in mind the solution robustness (SR) criterion (see next Sect. 6), the RS obtained using the RSh is always more convenient. The best schedule is selected according to the rule of the Minimal Impact of Rescheduled Operation on the Schedule (MIROS). The MIROS rule consists of two sub-criteria, the SR and the QR. Depending on priorities (weights) of criteria SR and QR in a scalar fitness function (4), a single solution is selected.

$$\text{MIROS}(y) = \sum_{p=1}^2 f_{sr_p}(y) \cdot w_p \tag{4}$$

$$f_{sr_p}(y) = f_{r_p}(y) / f_{r_p}(y^*) \tag{5}$$

$y$ —a schedule,  $f_{sr_p}(y)$  — a scalar sub-function after rescheduling;  $p$ — the number of scalar sub-functions,  $p = 1, 2$ ;  $f_{r_p}(y)$ — the value of scalar sub-function  $p$  reached for schedule  $y$ ;  $f_{r_p}(y^*)$ —the maximal value of scalar sub-function  $p$ ;  $w_p$ —the priority of scalar sub-function  $p$ .

The RSh and the RDO are embedded in the third stage of the H-MOIA. Therefore, only the algorithm based on the priority rules of the LFJ and LPT and algorithm SGR are applied in the comparative analysis. The algorithm based on the priority rules, the LFJ and LPT, is modified in order to enable its application in the job shop scheduling problem with batch production. The pseudocode of the LFJ/LPT reactive algorithm is presented in Fig. 4. The pseudocode of the SGR algorithm is presented in Fig. 5.

### 5 Evaluation criteria for predictive and reactive schedules

The previous section presents the second stage of the H-MOIA for predictive scheduling and the third stage of the H-MOIA for reactive scheduling. This section contains a description of evaluation criteria for predictive and reactive scheduling.

- Delete each disrupted operation and successive operations of the disrupted batch  $\hat{s}_j$  of disrupted job  $\hat{j}$
- Left shifting for undisrupted jobs  $j$
- For each disrupted batch  $\hat{s}_j$  of disrupted job  $\hat{j}$ 
  - Apply the LFJ/LPT predictive algorithm described in Fig.2

**Fig. 4** Pseudocode of the LFJ/LPT reactive algorithm

- Delete each disrupted operation, and the successive operations of the disrupted batch  $\hat{s}_j$  of disrupted job  $\hat{j}$ 
  - A) Right shifting for the disrupted operation of job  $\hat{j}$  until the moment at which the machine becomes idle
  - B) If the machine is not idle for a sufficient time, it is necessary to perform right shifting of successive operations to provide a sufficient idle time for rescheduling of the disrupted operation
- Evaluate solutions A and B with the use of computing criteria selected by a decision-maker from:  $C_{max}$ ,  $F, T, I$
- Select the best solution from the two options: A and B

**Fig. 5** Pseudocode of the SGR reactive algorithm

Predictive scheduling has two functions, (1) searching for the allocation of jobs to machines, for which the best value of an objective criterion is achieved and (2) serving as an overall plan in the event of disruption. After the disruption, a schedule becomes infeasible. Therefore, the objective is to generate the PS which can absorb disruption without affecting executed jobs and maintaining the high performance of the shop floor (Liu et al. 2007). The first function of predictive scheduling is to achieve efficient performance of the shop floor and can be measured using, for example, makespan, flow time, total tardiness and machine utilization criteria. Every schedule modification can affect values of the criteria. The second function of predictive scheduling is to obtain stable and robust schedules for the shop floor and can be measured using solution robustness and quality robustness criterion respectively.

A schedule robustness means that the performance of the schedule is insensitive to the disturbance (Leon et al. 1994; Policella et al. 2004). A schedule stability can be understood as the schedule nervousness. Schedule stability can be measured using the number of revisions or changes which are done for a schedule (Wu et al. 1993). Building a stable schedule requires the performance of a sensitivity analysis in order to answer the question “what” happens to the schedule “if” a disturbance appears (Goren and Sabuncuoglu 2008). Leon et al. (1994) noticed that the stability of a schedule is, in fact, the definition of the flexibility of a schedule. In other words, a schedule is flexible if it can respond efficiently to changing circumstances (Policella et al. 2004). The most popular criterion used for stability is the sum of absolute deviations between completion/start times of planned jobs and those performed (Table 1). In the approach proposed by the authors of this paper, the criterion SR (6) is used as well—in this case

for operations. The model involves the use of the criterion QR (7) for robustness evaluation. In the QR it is possible to use any combination of criteria popular in the related literature:  $C_{max}$ ,  $F$ ,  $T$ . Moreover, the use of machine utilization  $I$  criterion can be used as well.

$$fr_1(x) = SR = \sum_{j=1}^J \sum_{v_j=1}^{V_j} |st_{j,v_j}(PS) - st_{j,v_j}(RS)| \quad (6)$$

$st_{j,v_j}(PS)$ —start time of operation  $v_j$  of job  $j$  in the PS;  
 $st_{j,v_j}(RS)$ —start time of operation  $v_j$  of job  $j$  in the RS;

$$fr_2(x) = QR = \sum_{j=1}^N |f(x)_{PS} - f(x)_{RS}| \quad (7)$$

$f(x)_{PS}$ —scalar objective function of the PS;  $f(x)_{RS}$ —scalar objective function of the RS.

## 6 Flow shop and job shop with interruptions

This section presents a flow shop (FS) and a job shop (JS) scheduling problem with interruptions for experimental study, as well as a schedule of research.

Predictive and reactive methods are applied to two scheduling problems, 8 jobs have to be performed on 5 machines (5x8) in the FS and 5 jobs have to be performed on 10 machines (5x10) in the JS. The complexity of the flow shop scheduling problem (5x8) equals 40320 possible schedules and is sufficient to perform a comparative analysis. The input data for the FS scheduling problem (5x8) are presented in Fig. 6. The objective is to obtain a solution

**Fig. 6** Input data of the flow shop scheduling problem (5×8)

	MPR0 1 2 3 4 5	MPM0 1 0 1 0 1	MPM20 1 0 1 0 1
	MPR1 1 2 3 4 5	MPM1 0 1 1 0 0	MPM21 0 1 1 0 0
	MPR2 1 2 3 4 5	MPM2 1 0 1 0 0	MPM22 1 0 1 0 0
	MPR3 1 2 3 4 5	MPM3 0 1 1 1 0	MPM23 0 1 1 1 0
	MPR4 1 2 3 4 5	MPM4 0 1 0 0 1	MPM24 0 1 0 0 1
	MPR5 1 2 3 4 5	MPM5 1 0 0 1 1	MPM25 1 0 0 1 1
	MPR6 1 2 3 4 5	MPM6 0 1 1 1 0	MPM26 0 1 1 1 0
	MPR7 1 2 3 4 5	MPM7 1 0 1 1 0	MPM27 1 0 1 1 0
		MPM8 0 1 1 1 0	MPM28 0 1 1 1 0
	MOT0 5 2 1 4 1	MPM9 0 0 1 0 1	MPM29 0 0 1 0 1
	MOT1 5 3 2 4 1	MPM10 1 1 1 0 0	MPM30 1 1 1 0 0
	MOT2 5 4 1 4 1	MPM11 1 1 1 0 0	MPM31 1 1 1 0 0
	MOT3 5 2 4 4 1	MPM12 0 0 1 0 1	MPM32 0 0 1 0 1
	MOT4 5 1 2 4 1	MPM13 0 0 1 1 1	MPM33 0 0 1 1 1
	MOT5 4 1 1 4 2	MPM14 0 0 1 1 1	MPM34 0 0 1 1 1
	MOT6 5 1 2 4 1	MPM15 1 0 0 1 1	MPM35 1 1 1 0 0
	MOT7 4 1 1 4 2	MPM16 0 1 1 0 1	MPM36 1 1 1 0 0
		MPM17 0 1 1 0 1	MPM37 0 0 1 0 1
	VBS = [2 2 1 1 2 1 1 1]	MPM18 1 0 0 1 1	MPM38 0 0 1 1 1
	VDD = [110 100 50 110 110 50 110 110]	MPM19 1 0 1 0 1	MPM39 0 0 1 1 1

**Fig. 7** Input data of the job shop scheduling problem (5×10)

	MPR0 1 2 3 4 5	MPM0 1 0 1 0 1	MPM25 1 0 0 1 1
	MPR1 1 2 3 4 5	MPM1 0 1 1 0 0	MPM26 0 1 1 1 0
	MPR2 1 2 3 4 5	MPM2 1 0 1 0 0	MPM27 1 0 1 1 0
	MPR3 1 2 3 4 5	MPM3 0 1 1 1 0	MPM28 0 1 1 1 0
	MPR4 1 2 3 4 5	MPM4 0 1 0 0 1	MPM29 0 0 1 0 1
	MPR5 1 2 3 4 5	MPM5 1 0 0 1 1	MPM30 1 1 1 0 0
	MPR6 1 2 3 4 5	MPM6 0 1 1 1 0	MPM31 1 1 1 0 0
	MPR7 1 2 3 4 5	MPM7 1 0 1 1 0	MPM32 0 0 1 0 1
	MPR8 5 2 3 1 4	MPM8 0 1 1 1 0	MPM33 0 0 1 1 1
	MPR9 3 4 2 5 1	MPM9 0 0 1 0 1	MPM34 0 0 1 1 1
		MPM10 1 1 1 0 0	MPM35 1 1 1 0 0
	MT0 5 4 1 4 1	MPM11 1 1 1 0 0	MPM36 1 1 1 0 0
	MT1 5 2 4 4 1	MPM12 0 0 1 0 1	MPM37 0 0 1 0 1
	MT2 5 1 2 4 1	MPM13 0 0 1 1 1	MPM38 0 0 1 1 1
	MT3 4 1 1 4 2	MPM14 0 0 1 1 1	MPM39 0 0 1 1 1
	MT4 5 1 2 4 1	MPM15 1 0 0 1 1	MPM40 1 1 1 1 1
	MT5 4 1 1 4 2	MPM16 0 1 1 0 1	MPM41 1 1 1 1 1
	MT6 5 1 2 4 1	MPM17 0 1 1 0 1	MPM42 1 1 1 1 1
	MT7 4 1 1 4 2	MPM18 1 0 0 1 1	MPM43 1 1 1 1 1
	MT8 5 2 1 4 1	MPM19 1 0 1 0 1	MPM44 1 1 1 1 1
	MT9 5 3 2 4 1	MPM20 1 0 1 0 1	MPM45 1 1 1 1 1
		MPM21 0 1 1 0 0	MPM46 1 1 1 1 1
	VBS = [2 2 1 1 2 1 1 2 2]	MPM22 1 0 1 0 0	MPM47 1 1 1 1 1
	VDD = [110 100 50 110 110 50 110 110 120 120]	MPM23 0 1 1 1 0	MPM48 1 1 1 1 1
		MPM24 0 1 0 0 1	MPM49 1 1 1 1 1

for two objective functions:  $C_{max}, T$  with equal priorities:  $w_1 = 0.5, w_2 = 0.5$ . The input data for the JS scheduling problem (5×10) are presented in Fig. 7. The objective is to obtain a solution to four objective functions:  $C_{max}, F, T, I$  with priorities:  $w_1 = 0.3$  for  $C_{max}, w_2 = 0.2$  for  $F, w_3 = 0.3$  for  $T, w_4 = 0.2$  for  $I$ .

In production systems, a number of jobs  $J, j = 1, 2, \dots, J$  have to be performed on a number of machines  $W, w = 1, 2, \dots, W$ . Each job consists of a number of operations  $V_j, v_j = 1, 2, \dots, V_j; a_{w,v_j}$  denotes the execution time of operation  $v_j$  of job  $j$  on machine  $w$ . The execution times of operations  $a_{w,v_j}$  are predefined in the Matrix of Operations Times  $MOT$ . A production route is described in the Matrix of Processes Routes  $MPR$ . The deadline  $d_j$  of job  $j$  is predefined and described in the Vector of Due Dates  $VDD$ . Batch size  $s_j$  of job  $j$  is predefined and described in the Vector of Batch

Size  $VBS$ . The interpretation of matrices and vectors was presented by [Paprocka and Kempa \(2012\)](#). Each operation  $v_j$  can be executed on a machine, from a set of parallel machines described in the Matrix of Parallel Machines  $MPM[c_{jwv_j,w^*}]$  with dimensions:  $J \cdot W \times w^*$ .  $w^*$  is a machine parallel to the basic machine  $w, w^* = 1, 2, \dots, W. W^* = W. c_{jwv_j,w^*}$  represents the possibility of performing operation  $v_j$  on parallel machine  $w^*$ , previously assigned to machine  $w$  in the basic schedule (according to the processes routes described in  $MPR$ ).  $c_{jwv_j,w^*} = \{0, 1\}$ , 1 indicates that operation  $v_j$  can be performed on machine  $w^*$ , 0 means that operation  $v_j$  cannot be executed on machine  $w^*$ . Each row of the  $MPM$  below describes a set of parallel machines for a single operation  $v_j$  primarily assigned to machine  $w$ , and is calculated from  $(j - 1) \cdot W + w_v$ , (see Sect. 7).

**Table 2** The research schedule

	Methods	Scheduling problem		Criteria
		(5 × 8)	(5 × 10)	
Basic scheduling	H-MOIA	$C_{max}, T, FF_y$	$C_{max}, F, T, I, FF_y$	
Predictive scheduling	H-MOIA + MIDOS			
	LFJ, LPT			
	ASM			
Reactive scheduling	H-MOIA + MIROS	$C_{max}, T, FF_y, SR, QR$	$C_{max}, F, T, I, FF_y, SR, QR$	
	LFJ, LPT			
	SGR			

**Table 3** Evaluation of basic schedules achieved using H-MOIA, applied to the FS scheduling problem

No	The priority rule	$C_{max}$	$T$
1	{8, 3, 1, 6, 2, 4, 7, 5}	61	0
2	{1, 6, 3, 8, 4, 2, 5, 7}	61	0
3	{1, 8, 6, 3, 4, 7, 2, 5}	61	0

The schedule of the research is presented in Table 2. There are three stages of the experiment, i.e. first—the generation of a basic schedule, second—the generation of PS and third—the generation of RS for two scheduling problems (described in Figs. 6, 7) and the given criteria. The basic schedule is generated in order to define which machine constitutes the bottleneck. The PS is generated in order to meet the deadline. The RS is generated if an unpredicted disturbance happens.

### 7 Approach to a solution

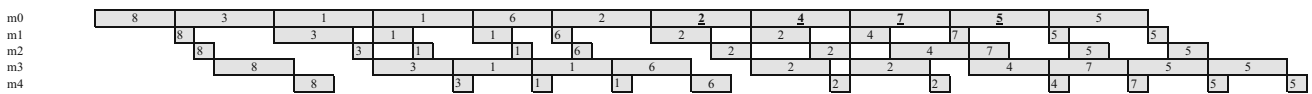
This section presents the application of the MOIA in solving the FS scheduling problem and discusses the influence of failure-free time on the quality of the generated PS. The mock problem with machines and job parameters presented in Fig. 6 was generated to illustrate the proposed predictive and reactive algorithms.

Three runs of simulation were generated for the FS problem (5 × 8) using the first stage of the H-MOIA. As a result, three different basic schedules with the same quality:  $C_{max} = 61, T = 0$  were obtained. In the first simulation, the best basic schedule was generated according to the rule of {8,3,1,6,2,4,7,5}. The rules of the best basic schedules and the quality of the achieved schedules are presented in Table 3. Afterwards, the issue was analysed using the example of the first rule. The schedule obtained is presented in Fig. 8. The machine  $w = 1$  is the bottleneck.

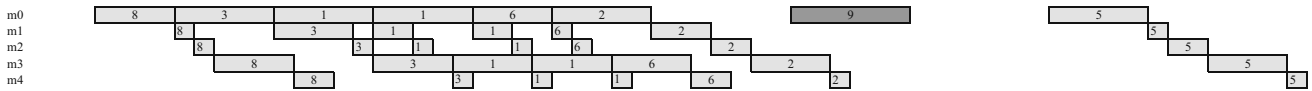
In the second stage of the H-MOIA, the MTTF of the bottleneck was predicted to ensure that deadlines of jobs were

not exceeded. The additional goal was to obtain a schedule of undelayed jobs with the minimal value of makespan criterion. Every unpredicted failure of the bottleneck may disturb the production process.

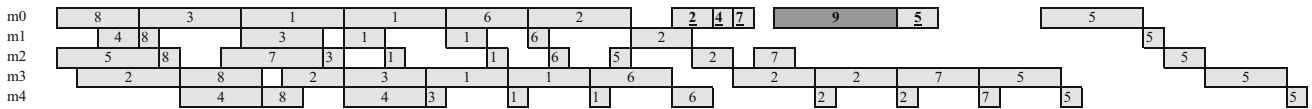
Next, the PS is generated using the second stage of the H-MOIA. The basic schedule generated according to the rule of {8,3,1,6,2,4,7,5} is the input data to the second stage of the H-MOIA + MIDOS. The authors made the assumption that  $a = 30$  and  $b = 42$  and  $MTTF = 36, MTTR = 6$  for machine  $w = 1$ . It was predicted that at the time period  $[a, b + MTTR]$ , the first operation  $\hat{v}_2 = 1$  of job  $\hat{j} = 2$  of batch  $\hat{s}_2 = 2$  would be disturbed. Therefore, it was expected that the following operations would be disturbed,  $\hat{v}_4 = 1$  of batch  $\hat{s}_4 = 1, \hat{v}_7 = 1$  of batch  $\hat{s}_7 = 1$  and  $\hat{v}_5 = 1$  of batch  $\hat{s}_5 = 1$  (see Fig. 8, highlighted jobs). Disrupted batches  $\hat{s}_j$  were deleted from the basic schedule (Fig. 9). The MIDOS was computed for operations of disrupted batches, having the bottleneck in a set of parallel machines. The set of parallel machines for operation  $\hat{v}_j$  first executed on machine  $w$  (in the basic schedule, Fig. 8) is described by a row  $(j - 1) \cdot W + w_v$  of the MPM (Fig. 6). In the case of job  $\hat{j} = 2$ , two operations can be performed on machine  $w = 1: v_2 = 1$  and  $v_2 = 3$  [first column and row 6 of the  $MPM_{(2-1) \cdot 5+1=6}$  and row 8 of the  $MPM_{(2-1) \cdot 5+3=8}$  (Fig. 6)]. Operation  $\hat{v}_2 = 1$  is performed on machine  $w_v = 1$  (the first column of row 1,  $MPR_{j-1=1}$  in Fig. 6). Operation  $\hat{v}_2 = 1$  involves three parallel machines columns no. 1, 4 and 5 (the sixth column of row  $MPM_{(2-1) \cdot 5+1=6}$  in Fig. 6). Thus, the number of machines on which disrupted operation  $\hat{v}_2 = 1$  can be alternatively performed equals  $F_{\hat{v}_2} = 3$ . The number of changes to perform after the occurrence of the disturbance of operation  $\hat{v}_2 = 1$  performed at the bottleneck equals  $R_{\hat{v}_2} = 5$ . Finally,  $MIDOS_{1_2} = 5 + (5 - 3) = 7$ . For operation  $v_2 = 3, MIDOS_{3_2} = 3 + (5 - 3) = 5$ . Since  $MIDOS_{3_2} = 5$  is less than  $MIDOS_{1_2} = 7$ , the operation  $v_2 = 3$  is the most flexible and its disruption causes the smallest changes in the schedule. The operation  $v_2 = 3$  is first assigned to the bottleneck at time period  $[a, b + MTTR = 30, 42 + 6]$ . For the remaining operations of job  $\hat{j} = 2$ , backward and forward scheduling is



**Fig. 8** Basic Gantt chart generated at the first stage of H-MOIA ( $j = \{1, 2, \dots, 7\}$ -no. of job), highlighted jobs  $j = \{2, 4, 7, 5\}$ -predicted to be disrupted



**Fig. 9** Gantt chart with deleted batches (predicted to be disturbed) ( $j = \{1, 2, \dots, 7\}$ -no. of job, 9-the machine predicted to be unavailable)



**Fig. 10** PS obtained at the second stage of H-MOIA in the first iteration ( $j = \{1, 2, \dots, 7\}$ -no. of job, 9-predicted technical inspection at time 36)

**Table 4** MIDOS for disrupted jobs after (a) first (b) second and (c) third iteration related to the checking of the condition leading to the starting time violation

(a) MIDOS <sub><i>i</i><sub><i>j</i></sub></sub>	<i>i</i> = 1	<i>i</i> = 2	<i>i</i> = 3	<i>i</i> = 4	<i>i</i> = 5
$\hat{j} = 2$	7	0	5	0	0
$\hat{j} = 4$	7	0	0	4	3
$\hat{j} = 7$	7	6	0	0	0
$\hat{j} = 5$	7	0	6	0	0
(b) MIDOS <sub><i>i</i><sub><i>j</i></sub></sub>	<i>i</i> = 1	<i>i</i> = 2	<i>i</i> = 3	<i>i</i> = 4	<i>i</i> = 5
$\hat{j} = 2$	7	0	5	0	0
$\hat{j} = 4$	7	0	0	4	0
$\hat{j} = 7$	7	6	0	0	0
$\hat{j} = 5$	7	0	6	0	0
(c) MIDOS <sub><i>i</i><sub><i>j</i></sub></sub>	<i>i</i> = 1	<i>i</i> = 2	<i>i</i> = 3	<i>i</i> = 4	<i>i</i> = 5
$\hat{j} = 2$	7	0	5	0	0
$\hat{j} = 4$	7	0	0	4	0
$\hat{j} = 7$	7	6	0	0	0
$\hat{j} = 5$	7	0	0	0	0

applied (Fig. 10). MIDOS<sub>*i*<sub>*j*</sub></sub>

are computed for operations of disrupted job  $\hat{j}$  and disrupted batch  $\hat{s}_j$  having the bottleneck in a set of parallel machines (see Table 4a). In the case of job  $\hat{j} = 4$ , operation  $i = 5$  is the most flexible and is first assigned to the bottleneck at the time period [30,48]. Next, the condition of the starting time violation is checked. Since operation  $i = 2$  of job  $\hat{j} = 4$  starts at time 4 and there is no place on machine  $w = 1$  to perform operation  $i = 1$ , the starting time condition is not satisfied (see Fig. 10). Therefore, MIDOS<sub>5<sub>4</sub></sub> is deleted, in the second iteration of predictive scheduling (Table 4b). In the third iteration of predictive scheduling MIDOS<sub>3<sub>5</sub></sub> is also deleted. Finally, the PS obtained by the second stage of the H-MOIA is pre-

sented in Fig. 11. The quality of the PS obtained at the second stage of H-MOIA is  $C_{max} = 59, T = 0$  and  $FF_y = 29.5$ .

The authors made the assumption that real MTTF of the bottleneck equals 45. In H-MOIA + MIROS, if the bottleneck fails, disrupted jobs are rescheduled according to two heuristics, i.e. the RSh and the RDO. From the two schedules ( $y = I, y = II$ ) generated using two heuristics, i.e. the RSh and the RDO (Figs. 12, 13), the best solution is selected using the MIROS rule (4). There are two sub-functions, when  $p = 1$  is QR and  $p = 2$  is SR, with equal priorities:  $w_{p1} = 0.5, w_{p2} = 0.5$ . Since the quality of the schedule generated using the RSh is  $fr_1(I) = QR = 4.5, fr_2(I) = SR = 90$  (Table 6), and the quality of the schedule generated using RDO is  $fr_1(II) = QR = 2, fr_2(II) = QR = 100$  the following equations are obtained:

$$MIROS(I) = \frac{4.5}{4.5} \cdot 0.5 + \frac{90}{90} \cdot 0.5 = 1 \tag{8}$$

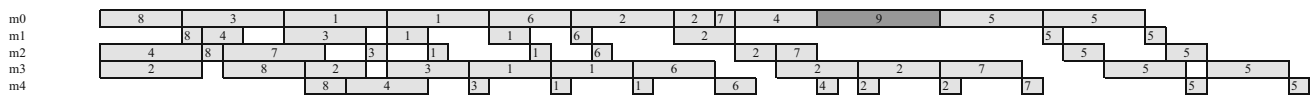
$$MIROS(II) = \frac{0}{4.5} \cdot 0.5 + \frac{20}{90} \cdot 0.5 = 0.11 \tag{9}$$

Since  $MIROS(II) < MIROS(I)$ , the best schedule is generated using the RDO. The quality of the RS generated using the H-MOIA + MIROS is  $C_{max} = 59, T = 0, FF_y = 29.5, SR = 20, QR = 0$  (Fig. 13).

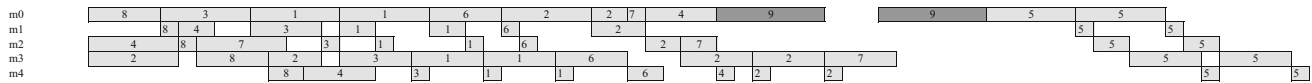
### 8 Experimental study and results

The objective of the experimental study is to determine which method of predictive and reactive scheduling provides better results considering the following sets of objectives:

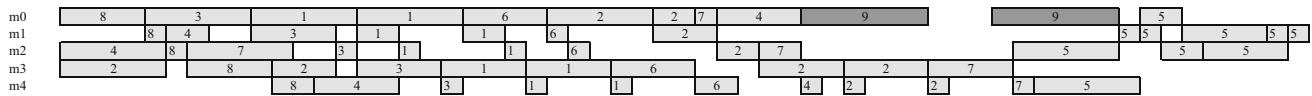
- makespan, flow time, total tardiness and idle time for predictive scheduling,



**Fig. 11** PS obtained at the second stage of H-MOIA +MIDOS after the third iteration related to the check of the condition which causes the starting time violation ( $j = \{1, 2, \dots, 7\}$ -no. of job, 9-predicted technical inspection at time 36)



**Fig. 12** RS obtained using RSh (predicted  $a = 30$ , predicted MTTF = 36, MTTR = 6, real MTTF = 45,  $j = \{1, 2, \dots, 7\}$ -no. of job). Input data consists of the PS generated by the H-MOIA +MIDOS



**Fig. 13** RS obtained using RDO (predicted  $a = 30$ , predicted MTTF = 36, MTTR = 6, real MTTF = 45,  $j = \{1, 2, \dots, 7\}$ -no. of job). Input data consists of the PS generated by the H-MOIA +MIDOS

– solution robustness and quality robustness for reactive scheduling,

for the FS and JS.

**8.1 The flow shop scheduling problem**

In the previous section, the experiment for the FS scheduling problem was run using the H-MOIA. The quality of the PS obtained at the second stage of the H-MOIA is  $C_{max} = 59$ ,  $T = 0$ . Next, the PSs using algorithms identified in the literature, ASM, LFJ/LPT, were generated.

The permutation flow, according to the rule of  $\{5, 2, 1, 6, 8, 4, 3, 7\}$  constitutes the input data to the ASM searching procedure. The PS generated using the ASM is a permutation flow according to the rule of  $\{5, 2, 6, 8, 3, 1, 4, 7\}$ . The PS generated using the LFJ/LPT is the permutation flow according to the rule of  $\{8, 7, 6, 5, 4, 3, 2, 1\}$ . The quality of the PSs generated for the scheduling problem ( $5 \times 8$ ) using various algorithms is summed up in Table 5.

Taking into account two criteria,  $C_{max}$ ,  $T$ , the best solution is obtained at the second stage of the H-MOIA. The technical inspection of the bottleneck planned at time 36 and performed in accordance with the method proposed in the paper can generate failure-free production. The question arising, on this occasion, is concerned with what happens if the failure of the bottleneck occurs, even though the maintenance has been performed? In order to assess the predictive scheduling algorithms performance, two breakdown cases were set, (1) real MTTF = 36 and (2) real MTTF = 45. RSs were evaluated using the solution robustness criterion SR (6) and quality robustness QR (7).

In the case where real MTTF = 36, the PS generated by the MOIA +MIDOS is robust and stable. When the machine

unavailability is known in advance, the effect on the RS is very small compared with the RS generated by the ASM or the LFJ/LPT. Thus, the experiment was run for the real MTTF = 45. For each PS obtained using three methods, H-MOIA +MIDOS, ASM and LFJ/LPT, the RSs were generated using the methods of the H-MOIA +MIROS, HI, HII, LFJ/LPT and SGR (Table 2).

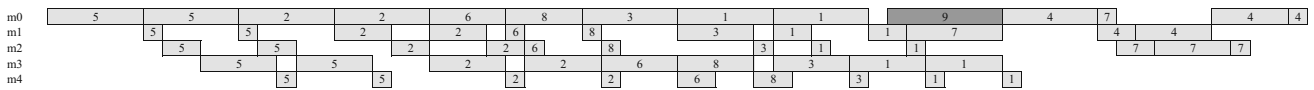
As indicated in Sect. 7, the quality of the RS generated using the H-MOIA +MIROS is  $C_{max} = 59$ ,  $T = 0$ ,  $FF_y = 29.5$ ,  $SR = 20$ ,  $QR = 0$  (Fig. 13). The quality of RSs achieved using LFJ/LPT and SGR is presented in Table 5.

The next step involved the generation of RSs using the methods of the LFJ/LPT and SGR based on the PS generated using the ASM. The PS generated using the ASM is the permutation flow according to the rule of  $\{5, 2, 6, 8, 3, 1, 4, 7\}$ . The detailed results achieved using different methods are presented in Table 5. Taking into account the criteria of SR and QR, the best RS is achieved using the MIROS (Fig. 14).

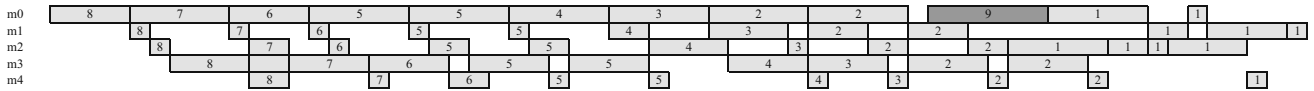
The subsequent step was concerned with the generation of the RSs using the methods of LFJ/LPT, H-MOIA +MIROS and SGR based on the PS generated using the LFJ/LPT. The PS generated using the LFJ/LPT is the permutation flow according to the rule of  $\{7, 6, 5, 4, 3, 2, 1\}$ . The detailed results achieved using different methods are presented in Table 6. Taking into account the criteria of SR and QR, the best RS is generated using the MIROS. The best RS is presented in Fig. 15.

**Table 5** Evaluation of predictive scheduling methods for a flow shop problem

	Flow shop ( $5 \times 8$ )		
	$C_{max}$	$T$	$FF_y$
H-MOIA	59	0	29.5
LFJ, LPT	62	0	31
ASM	63	0	31.5



**Fig. 14** RS obtained using the MIROS (predicted  $a = 30$ , predicted MTTF = 36, MTTR = 6, real MTTF = 45,  $j = \{1, 2, \dots, 7\}$ -no. of job). Input data consists of the PS generated by the ASM



**Fig. 15** RS obtained using the MIROS (predicted  $a = 30$ , predicted MTTF = 36, MTTR = 6, real MTTF = 45,  $j = \{1, 2, \dots, 7\}$ -no. of job). Input data consists of the PS generated by the LFJ/LPT

The quality of the RSs generated for scheduling problem  $(5 \times 8)$  using different algorithms is summed up in Table 6. Three different PSs are generated by three different algorithms. Taking into account the criteria of  $C_{max}$  and  $T$ , the best PS is generated using the MOIA + MIDOS. Although the additional maintenance task is introduced in the PS (Fig. 11), the MOIA achieves a better quality solution than the tabu search algorithm ASM and the algorithm based on heuristic LFJ/LPT. Taking into account the criteria of SR and QR, the best RSs are generated using the MIROS. The MIROS selects which RS is better for a given set of priorities of criteria, from the two schedules generated by the RSh and the RDO. The disadvantage of the SGR is the fact that the algorithm is based mainly on right shifting. Indeed, the gap can be filled even if it is smaller than the operation duration provided that the missing time is restricted within predefined tolerance limits. The operation(s) following the gap is/are shifted to the right by the missing time. However, the method may prove problematic due to the difficulty related to the determination of the tolerance limit.

**8.2 The job shop scheduling problem**

This section is concerned with the job shop scheduling problem  $(5 \times 10)$  described in Fig. 7. The search was focused on a schedule, which is robust enough for disturbance and with good quality with criteria  $C_{max}$ ,  $F$ ,  $I$ ,  $T$ . Machine  $w = 1$  is

**Table 7** Evaluation of basic schedules achieved after running three simulations of H-MOIA applied to the JS scheduling problem

No	The priority rule	$C_{max}$	$F$	$I$	$T$
1	{8,7,4,3,10,6,2,1,5,9}	72	218	154	4
2	{10,3,6,1,2,7,4,5,8,9}	72	222	154	0
3	{7,4,10,8,3,2,6,1,5,9}	72	225	154	4

the most loaded one in the job shop scheduling problem. The assumption is as follows  $a = 30$  and  $b = 42$  and MTTF = 36, MTTR = 6 for machine  $w = 1$ .

In order to achieve the basic schedule for the job shop problem  $(5 \times 10)$  using the H-MOIA, three simulations were generated. The best basic schedules and the quality of the achieved schedules are presented in Table 7. The best basic schedule was generated in the second simulation. Therefore, the issue was analysed using the example of the second rule.

PSs were generated using the methods of H-MOIA + MIDOS, ASM and LFJ/LPT.

First, the PS was generated using the second stage of the H-MOIA. The basic schedule generated according to the rule of {10,3,6,1,2,7,4,5,8,9} was the input data to the second stage of H-MOIA + MIDOS. It was predicted that in the time period  $[a, b + MTTR]$  operations performed on machine  $w = 1$  could be disturbed. Disrupted batches  $\hat{s}_j$  were deleted from the basic schedule. First, the most flexible operation of

**Table 6** Evaluation of reactive scheduling methods for the flow shop problem  $(5 \times 8)$

Predictive method	Reactive scheduling	$C_{max}$	$T$	$FF_y$	SR	QR
H-MOIA + MIDOS	LFJ, LPT	68	0	34	90	4.5
	MIROS	59	0	29.5	20	0
	SGR	68	0	34	90	4.5
ASM	LFJ, LPT	71	0	35.5	66	4
	MIROS	66	0	33	42	1.5
	SGR	70	0	35	70	3.5
LFJ, LPT	LFJ, LPT	68	6	37	76	6
	MIROS	63	6	34.5	45	3.5
	SGR	68	6	37	76	6

**Table 8** Evaluation of predictive schedules achieved after running three simulations of ASM

No	The priority rule	$C_{\max}$	$F$	$I$	$T$
1	{1, 2, 7, 10, 3, 6, 9, 4, 5, 8}	89	285	239	15
2	{10, 3, 6, 9, 1, 2, 7, 4, 5, 8}	90	242	244	0
3	{10, 3, 7, 4, 5, 8, 6, 9, 1, 2}	94	258	264	10

each deleted job in the time period  $[a, b + \text{MTTR}]$  was scheduled. For the remaining operations, backward and forward scheduling algorithms were applied. In the PS, in the time period  $[\text{MTTF}, \text{MTTF} + \text{MTTR}]$ , the technical inspection of the bottleneck was scheduled. The quality of the PS obtained at the second stage of the H-MOIA is  $C_{\max} = 82$ ,  $F = 264$ ,  $I = 203$  and  $T = 0$ .

The following step involved the generation of PSs using the algorithms of ASM, LFJ/LPT.

In order to achieve the PS using the ASM, three simulations were generated (the number of iteration = 30 and was the same as in the first stage of the MOIA). The best predictive schedules and the quality of the achieved schedules are presented in Table 8. The best PS was generated in the second simulation. Thus, the issue was analysed using the example of the second rule.

Afterwards, the efficiency of two algorithms, the MOIA and the ASM in searching a solution space was compared. Therefore, the input data to the neighbourhood searching heuristic (ASM) was the permutation of jobs obtained at the first stage of the H-MOIA. After running the experiment, the PS generated using the ASM is the flow according to the rule of {10,3,6,1,2,7,4,5,8,9}. The quality of the PS obtained by the ASM is presented in Table 9. The ASM does not achieve a better quality solution than that generated at the first stage of the H-MOIA.

The PS generated using the LFJ/LPT is the flow according to rule of {8,7,6,5,4,3,2,1,10,9}. The detailed results achieved using LFJ/LPT are presented in Table 9.

Taking into account the criteria of  $C_{\max}$ ,  $F$ ,  $I$ , and  $T$  (Table 9), the solutions generated using the LFJ/LPT are better than those generated using the MOIA + MIDOS. However, it should be noted that the algorithms of ASM and LFJ/LPT do not insert the maintenance task into a schedule. The basic schedule generated using the MOIA enables the achievement of solutions close to the best quality basic schedule:  $C_{\max} = 72$ ,  $F = 233$ ,  $I = 154$ ,  $T = 0$ , excluding the technical inspection of the bottleneck at time 36. However, undelayed jobs are still generated in the PS using the MOIA + MIDOS. The main advantage is the minimization of the probability of the bottleneck breaking down due to the insertion of the additional task, i.e. planned technical inspection into the basic schedule.

**Table 9** Evaluation of predictive scheduling methods for a job shop problem

	$C_{\max}$	$F$	$T$	$I$	$FF_y$
H-MOIA	82	264	0	203	118.3
LFJ, LPT	72	279	0	154	108.2
ASM	90	242	0	244	124.2

The question which arises in such a situation is what happens if the bottleneck fails even though the maintenance has performed?. In order to answer the question, for each PS obtained using three methods, i.e. the H-MOIA + MIDOS, ASM and LFJ/LPT RSs were generated using the methods H-MOIA + MIROS, LFJ/LPT and SGR (Table 2). RSs were evaluated using the solution robustness criterion SR (6) and the quality robustness QR (7).

For the assumption that the real MTTF of the bottleneck equals 45, the detailed results generated for job shop scheduling problem ( $5 \times 10$ ) using different algorithms are presented in Table 10

Taking into account criteria of  $C_{\max}$ ,  $F$ ,  $T$  and  $I$ , the best PS seems to be generated by the LFJ/LPT (Table 9). However, further analysis indicated that the PS generated by the H-MOIA + MIDOS absorbs the effect of the bottleneck failure more efficiently (Table 10). The PS generated by the H-MOIA + MIDOS is robust and the most stable with  $SR = 35$  and  $QR = 0.2$ .

Taking into consideration the criteria of SR and QR (Table 10), the best RSs are generated using the MIROS regardless of the method used for predictive scheduling.

The LFJ/LPT is less effective than the H-MOIA and the ASM comparing results achieved for two scheduling problems, the flow shop and the job shop (Tables 6 and 10). This is because the LFJ/LPT is based on heuristics and no solution space is searched. The information that jobs least flexible and jobs with the longest processing time should be performed first proves insufficient to generate robust and stable schedules.

In the ASM the uncertainty is handled by proposing the initial schedule with the best performance in the case of a disruption. In the MOIA the most flexible operations and those whose disruptions cause the least changes are scheduled for the bottleneck at time period of increased probability of a failure. Comparing the performance of schedules generated by two algorithms, the H-MOIA and the ASM applied to two scheduling problems (Table 6 and 10) it can be noticed that the H-MOIA performs better when the MIROS is applied as the rescheduling procedure. The ASM performs better when the rescheduling procedure based on right shifting (LFJ/LPT and SGR) is applied. Therefore, in the case of unpredicted failure, it is more convenient to apply the MIROS rule.



**Table 10** Evaluation of reactive scheduling methods for the job shop problem

Predictive method	Reactive scheduling	$C_{max}$	$F$	T	$I$	$FF_y$	SR	QR
H-MOIA +MIDOS	LFJ, LPT	92	314	0	245	139.4	333	21.1
	MIROS	83	271	0	197	118.5	35	0.2
	SGR	92	297	0	242	135.4	280	17.1
LFJ, LPT	LFJ, LPT	89	264	78	234	149.7	319	41.5
	MIROS	70	306	0	138	109.8	253	1.6
	SGR	90	265	80	238	151.6	329	43.4
ASM	LFJ, LPT	89	212	0	235	116.1	326	8.1
	MIROS	83	204	0	203	106.3	93	17.9
	SGR	87	189	0	223	108.5	175	15.7

### 9 Conclusion

The paper presents an innovative scheduling algorithm consisting of three stages. The first stage is used for the generation of the basic schedule, the second stage is used for predictive scheduling, whereas the third stage is used for reactive scheduling. The objective of the first stage of the H-MOIA is to obtain a good quality basic schedule which is modified to create a robust schedule. The robust schedule is generated using the novel heuristic, the minimal impact of disrupted operation on the schedule (MIDOS) presented in the paper. The minimal impact of rescheduled operation on the schedule (MIROS) is used for reactive scheduling.

The approach presented is efficient if a real failure-free time belongs to a time period  $[a, b]$ . Technical inspections of the bottleneck planned in the time period  $[a, b]$  can result in failure-free production. In order to assess the predictive scheduling algorithms performance, two breakdown cases were set, (1) the real MTTF belonging to  $[a, b]$  and (2) the real MTTF not belonging to  $[a, b]$ . Reactive schedules were evaluated using the solution robustness and the quality robustness criterion. The algorithm presented in the paper obtains of the predictive schedules which are robust and the most stable for both scheduling problems.

The MIDOS rule for predictive scheduling was compared with the algorithms identified in the related literature, (1) the algorithm based on priority rules, i.e. the least flexible job first (LFJ) and the longest processing time (LPT) and (2) the Average Slack Method. The MIROS was compared with algorithms (1) based on priority rules, i.e. the LFJ and LPT and (2) the Shifted Gap-Reduction. Related analyses were performed for two scheduling problems, i.e. the flow shop and the job shop.

The LFJ/LPT has proved to be a less effective predictive scheduling method than the H-MOIA and the ASM. This can be ascribed to the fact that the LFJ/LPT is based on heuristics and no solution space is searched. Information that

jobs least flexible and with the longest processing time should be performed first is insufficient to generate robust and stable schedules.

In the ASM the uncertainty is handled by proposing an initial schedule with the best performance in the case of a disruption. In the MOIA the most flexible operations, and those whose disruptions cause the least changes, are scheduled for the bottleneck at a time period of increased probability of a failure. The H-MOIA performs better when the MIROS is applied as the rescheduling procedure. The ASM performs better when rescheduling procedures based on right shifting (LFJ/LPT and SGR) are applied.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

### References

Al-Hinai, N., & ElMekkawy, T. Y. (2011). Robust and flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics*, *132*, 279–291.

Cheng, R., Gen, M., & Tsujimura, Y. (1999). A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: Hybrid genetic search strategies. *Computers & Industrial Engineering*, *36*, 343–346.

Ćwikła, G. (2014). Methods of manufacturing data acquisition for production management—A review. *Advanced Materials Research*, *837*, 618–623.

Duenas, A., & Petrovic, D. (2008). An approach to predictive-reactive scheduling of parallel machines subject to disruptions. *Annals of Operations Research*, *159*, 65–82.

Goren, S., & Sabuncuoglu, I. (2008). Robustness and stability measures for scheduling: Single-machine environment. *IIE Transactions*, *40*, 66–83.

Grabowik, C., & Kalinowski, K. (2011). Object-oriented models in an integration of CAD/CAPP/CAP systems. *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial

- Intelligence and Lecture Notes in Bioinformatics), 6679 LNAI (PART 2), pp. 405–412.
- Guilherme, E. V., Herrmann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1), 35–58.
- Hasan, Kamrul S. M., Sarker, R., & Essam, D. (2011). Genetic algorithm for job-shop scheduling with machine unavailability and breakdowns. *International Journal of Production Research*, 49(16), 4999–5015.
- Jai, A. K., & Elmaraghy, H. A. (1997). Production scheduling/rescheduling in flexible manufacturing. *International Journal of Production Research*, 35, 281–309.
- Janik, W., & Gendarz, P. (2011). Disassembly and aggregation in computer aided overhaul preparation. *Journal of Achievements in Materials and Manufacturing Engineering*, 44(2), 187–197.
- Kempa, W. M., Paprocka, I., Kalinowski, K., & Grabowik, C. (2014). Estimation of reliability characteristics in a production scheduling model with failures and time-changing parameters described by Gamma and exponential distributions. *Advanced Materials Research*, 837, 116–121.
- Leon, V. J., Wu, S. D., & Storer, R. H. (1994). Robustness measures and robust scheduling for job shops. *IIE Transactions*, 26, 32–43.
- Liu, L., Han-yu, G., & Yu-geng, X. (2007). Robust and stable scheduling of a single machine with random machine breakdowns. *The International Journal of Advanced Manufacturing Technology*, 31, 645–656.
- Paprocka, I., & Kempa, W. M. (Eds.). (2012). Estimation of reliability characteristics in a production scheduling model with the renewal theory application. In *Information systems architecture and technology. The use of IT models for organization management* (pp. 59–68). Wrocław: Wrocław University of Technology.
- Paprocka, I., Kempa, W. M., Kalinowski, K., & Grabowik, C. (2014). A production scheduling model with maintenance. *Advanced Materials Research*, 1036, 885–890.
- Paprocka, I., & Skołod, B. (2013). Robust scheduling, a production scheduling model of failures. *Applied Mechanics and Materials*, 307, 443–446.
- Policella, N., Oddi, A., Smith, S. F., & Cesta, S. (2004). Generating robust partial order schedules. In *Principles and practice of constraint programming—CP*, pp. 496–511.
- Ponnambalam, S. G., Ramkumar, V., & Jawahar, N. (2004). A TSP-GA multi-objective algorithm for flowshop scheduling. *The International Journal of Advanced Manufacturing Technology*, 23, 909–915.
- Skołod, B., & Wosik, I. (Eds.). (2007). Multi-objective genetic and immune algorithms for batch scheduling problem with dependent setups. In *Recent developments in artificial intelligence methods* (pp. 185–196). Poland, Gliwice: Silesian University of Technology.
- Skołod, B., Wosik, I., Kempa, W. M., & Kalinowski, K. (Eds.) (2011). Estimation of reliability characteristics in a production scheduling model with time-changing parameters—second part, numerical example. In *Management and control of manufacturing processes* (pp. 19–30). Lublin: Lubelskie Towarzystwo Naukowe.
- Smutnicki, Cz. (2002). *Jobs scheduling algorithms (in Polish)*. Wrocław, DC: Wrocław University of Technology.
- Turkcan, A., Akturk, M. S., & Storer, R. H. (2009). Predictive/reactive scheduling with controllable processing times and earliness-tardiness penalties. *IIE Transactions*, 41, 1080–1095.
- Wosik, I., & Skołod, B. (2009). Tuning of a fuzzy system for controlling searching process in multi objective scheduling immune algorithm. *Journal of Machine Engineering*, 9(1), 130–143.
- Wu, S. D., Storer, R. H., & Chang, P. C. (1993). One-machine rescheduling heuristics with efficiency and stability as criteria. *Computers & Operations Research*, 20, 1–14.