Prescriptive System for Reconfigurable Manufacturing Systems Considering Variable Demand and Production Rates

Catarina Baltazar, João Reis, Gil Gonçalves
SYSTEC, Research Center for Systems and Technologies
Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Email: {up201406435, jpcreis, gil}@fe.up.pt

***Abstract*—The current market is dynamic and, consequently, industries need to be able to meet unpredictable market changes in order to remain competitive. To address the change in paradigm, from mass production to mass customization, manufacturing flexibility is key. Moreover, current digitalization of the industry opens opportunities regarding real-time decision support systems allowing the companies to make strategic decisions, and gain competitive advantage and business value.**

The main contribution of this paper is a proof of concept Prescriptive System with a highly parameterizable simulation environment catered to meet the needs of Reconfigurable Manufacturing Systems allied with an optimization module that takes into consideration productivity, market demand and equipment degradation. With this system, the effects of different throughput rates are monitored which results in better recommendations to mitigate production losses due to maintenance actions while taking into consideration the health status of the remaining assets.

In the proposed solution the simulation module is modeled based on Directed Acyclic Graphs and the optimization module based on Genetic Algorithms.

The results were evaluated against two metrics, variation of pieces referred as differential and availability of the system. Analysis of the results show that productivity in all testing scenarios improves. Also, in some instances, availability slightly increases which shows promising indicators.

***Index Terms*—Reconfigurable Manufacturing Systems, Industry 4.0, Variable Throughput, Genetic Algorithm**

I. Introduction

Nowadays industries face constant changes as the result of unpredictable market trends. The challenge is to be flexible enough in order to respond in a timely manner to clients demand while maintaining a sustainable cost structure to remain competitive in a fierce business environment. For the purpose of attending markets needs, it is necessary to increase the efficiency of manufacturing processes in which machinery plays a fundamental role.

Reconfigurable Manufacturing Systems (RMS) arise to deal with uncertainty and individualized demand [1] by combining advantages of both Dedicated Manufacturing Lines and Flexible Manufacturing Systems [2]. Moreover, during the current industrial revolution, also referred as Industry 4.0, significant interest in the upgrade of Prognostics and Health Management (PHM) frameworks emerge as they allow improvements in reliability and reduction of costs associated with maintenance actions [3]. Advances in the Information and Communication Technologies domain enable the development of more sophisticated PHM tools, especially, based on Deep Learning methods as they simplify the process of feature learning and have superior performance. Deep Learning approaches represent a promising path towards a one-fits-all framework [4]. An effective PHM system should be able to timely predict failures by constantly monitoring health status of the equipment and also isolate and identify the faults [5]. Additionally, it must support decision-making systems to take full strategic advantage of the predictions provided by diagnosis and prognosis techniques [6]. While prognosis is related to failure prediction and tries to answer the questions "What will happen?" and "When will it happen?" [7], diagnosis consists in identifying and isolating the faults. Despite the intuitive relationship between predictions and prescriptions, and the undeniable benefits to gain competitive advantage, prescriptive systems' area is the field with less research [8]. These systems intend to recommend one or more courses of action based on predicted future and, therefore, allow to take proactive measures [7].

A thorough review of prescriptive systems is given by [8] where three categories were identified: production scheduling, life cycle optimization, supply chain management and logistics. For example, regarding inventory management, in both [9] and [10], spare parts are ordered based on equipment degradation. In the former, decisions regarding the purchase of spare parts are decided based on the levels of degradation observed during irregular inspections. In the latter, long short-term memory (LSTM) networks are employed to predict failure probability during different time windows. Then, based on the information provided by the prediction model, the appropriate options regarding maintenance and order of spare parts are chosen.

From the three categories identified, in an industrial context, maintenance scheduling is the more predominant one. In [11], a Genetic Algorithm (GA) is employed to optimize maintenance scheduling for manufacturing systems with a fixed structure. In this paper, it is assumed that the information regarding failure probabilities is available. Similarly, in [12], a GA is used to schedule maintenances based on machine degradation. However, in this case, the variables that are optimized are the throughputs of machines and possible maintenance actions instead of discrete time moments. In general, the proposed optimization procedure searches for the best trade-off between maintenance actions and throughput settings. Likewise, in [13] a continuous maintenance system based on real-time monitoring is proposed. The optimization module is also based on GA and assures production targets by searching the best sequence of machine throughputs taking into consideration equipment degradation. In contrast, in this paper, a Predictive Maintenance module is integrated and the GA helps in avoiding unexpected breakdowns based on constant condition monitoring in real-time. Solving scheduling problems is not limited to the application of GA but these algorithms represent the majority of the proposed solutions [14].

Few Prescriptive Systems are applied to RMS. In this context, the mitigation of production losses due to machines downtime can be achieved not only by tuning throughputs of different machines, but also by routing pieces to healthy assets. Accordingly, the main contributions of this paper are an optimization approach that shows good indicators in finding throughput sequences that balance productivity and maintenance actions in a RMS context, as well as a straightforward simulation module based on Directed Acyclic Graphs (DAG) that allows quick layout changes and easy parametrization of the shop-floor namely, scheduling of maintenance shifts, different types of failures and types of equipment.

The remainder of this paper is organized as follows. In section II both simulation module and optimization module are discussed. Then, in section III, the scenarios that are tested in order to validate the solution were presented. Additionally, some preliminary results are discussed. A more in depth analysis of the results presented in the previous section can be found in section IV and finally, in section V conclusions and future work are discussed.

II. Implementation

The proposed Prescriptive System is mainly composed of two modules: simulation module and optimization module. In the following subsections each module is further described and this current section concludes with the interactions between the two.

A. Simulation

The goal is to model manufacturing layouts such as the one presented in Fig. 1 so it allows easy changes in configurations in order to respond to different demands in the future. These configurations possess crossovers and all machines within the same stage execute the same tasks. Consequently, pieces in stage i can be transferred to any machine at the stage $i+1$. According to [15], these configurations are defined as Class II RMS.

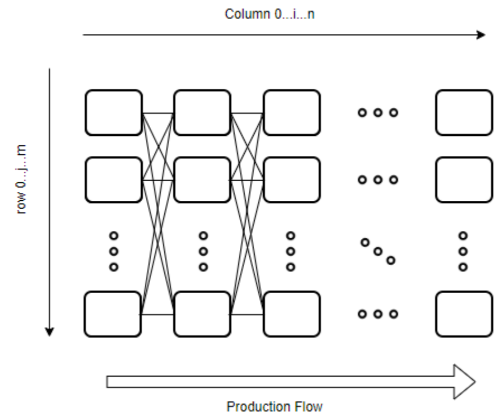

Fig. 1. Generic Manufacturing Layout

Accordingly, DAGs were chosen to model the system. This approach allows the rapid response in changing layouts configuration and the control of pieces flow in the manufacturing system. In order to implement it, the package Networkx, only available in Python, was chosen.

Each node of the graph represents a machine and the edges connections between machines that might be, for instance, conveyor belts. The edges are weighted and represent path priority. The lowest the weight the higher the priority. This approach allows to favour, for example, the shortest path when deciding to which machine should the piece be sent.

The machines are represented by the class Machine and each instance represents a node of the graph. This approach allows high parameterization of the equipment and the parameters can be separated in three main groups:

- Identifying Parameters: relate to the identification of the equipment
  - machine id;
  - type of machine;
  - age;
  - line;
  - stage;
- Operations Parameters: relate to the machine operation
  - available operations;
  - current throughput;
- Reliability-related parameters: relate to degradation of the equipment
  - mean time to repair (MTTR);
  - mean time between failures (MTBF);
  - types of failures.

Concerning to identifying parameters, line and stage correspond to the position of the machine in the layout, Fig. 1, while the remaining parameters in this category are related to specifications of the equipment. In respect of operation parameters, available operations relate to the range of operations that the machine can perform and current throughput identifies the production rate at which the equipment is operating. Lastly, regarding reliability-related parameters, this are of the utmost importance to simulate the degradation of the equipment. In terms of different types of failures, each machine can have associated different ones which will correspond to different MTTR and, as a result, maintenance actions will have different periods of time. Also, MTBF will be used as a mean to predict the failure.

In addition, in this case, the machines are also responsible to control the flow of production in the shop-floor. Each machine has a state machine associated as the one represented in Fig. 2.

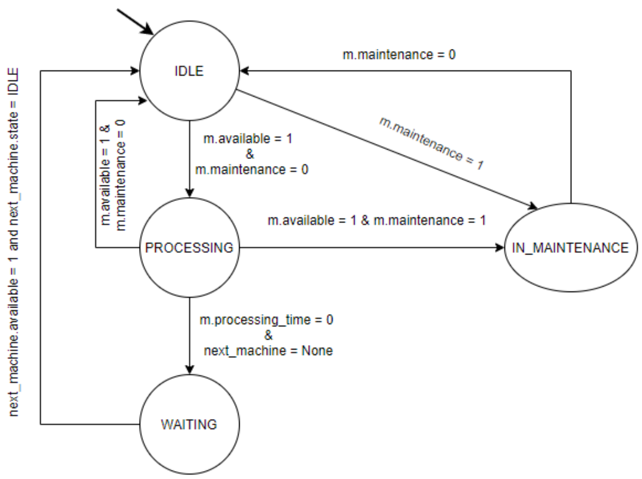

Fig. 2. State Machine associated with each machine

The machine has four states. It starts in its IDLE state and if the machine is not going to start any maintenance, maintenance = 0, and is available, the machine can receive pieces. Once the pieces are received they are processed. When the processing time ends, three things might happen: if the next machine is available the piece is dispatched and then the machine can return to its IDLE state or IN_MAINTENANCE state. Otherwise, it will transition to WAITING state. This transition happens when there are no available machines and the current machine behaves as a buffer until a possible machine becomes available. While in the WAITING state, the machine cannot receive any pieces. In the case that the machine does not have the respective tool, the piece experiences the same cycle, however, processing times are equal to zero. In short, the edges of the graph provides the different connections between machines and each connection is only admissible if green-lighted by the destination machine state.

In addition, not only machines can be parameterized but also other parts of the manufacturing environment. The simulation module developed in this paper takes into consideration, different simulation times, maintenance shifts and different sequences of operations to apply to different raw materials. Simulation times are related to how many seconds each tick (time unit in the simulation environment) worth and how many working weeks are being simulated. Also, it defines how many working days and working hours are considered. In regards to maintenance shifts, if one decides to integrate them in the simulation, the starting times and duration of said shifts can be defined. The only thing, which in some cases might be considered a limitation, is the fact that the maintenance shifts, by default, are periodic. Simply put, in every working day the shift starts at the same time and has the same duration. Additionally, different sequences of operations can be applied to the pieces in order to achieve different final products as long as the needed operations are available in the current machines and as long as the operations can be performed in a sequential manner as represented in Fig. 1. All these features allows the simulation of a wide variety of scenarios not only on time domain but also specification wise.

In this paper, it is assumed that the information regarding probability failures is known, as no predictive model is proposed. Recalling the parameters associated to each machine, namely, reliability-related ones, both MTBF and MTTR are known. In a simplified manner, MTTR refers to the average time to repair certain component and MTBF the forecasted time between failures [16]. Both these terms will allow to simulate degradation of the equipment as well as management of maintenance actions in order to implement the present system. As a result, the prediction of a pending failure will be calculated based on the difference between MTBF and current simulation time. If that difference is below a certain threshold, the failure will be signaled and maintenance scheduling takes place. Both Fig. 3 and Fig. 4 exemplify how the maintenance scheduling is handled. The difference between MTBF and current simulation time corresponds to a certain time window. This time window is the time to failure and is represented by the yellow area. If during that time window a shift takes place, blue area, then the maintenance of the respective equipment will occur when the shift starts (Fig. 4). Otherwise, an emergency maintenance is triggered (Fig. 3).

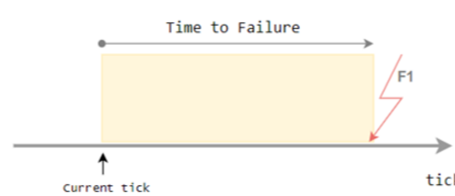

Fig. 3. Pending Failure that will result into an emergency maintenance

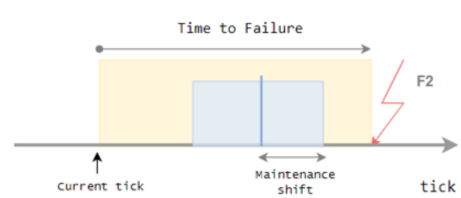

Fig. 4. Pending Failure that will result into a scheduled maintenance

Furthermore, different machines' throughputs have different impacts in degradation of the equipment. As stated in [12], when a machine decelerates it is expected that its degradation slowdown, and vice-versa if a machine increases its throughput. To simulate the degradation effects influenced by the chosen production rates, the MTBF will be inversely proportional to production rate. Similar to [13], five throughputs are available where mode 2 increases production rate two times in regards to baseline production, mode 1 production rate is 1.5 times higher, mode 0 corresponds to baseline throughput, mode -1 production rate decreases in 1.5 times and, lastly, mode -2 where production rate decreases 2 times.

B. Optimization

The optimization module is key to the implementation of the Prescriptive System as it is responsible for the compensation of production losses due to machines' downtime. A standard GA approach was chosen as its employment is well documented and produces near-optimal solutions [17]. GAs can be understood as an abstraction of the theory of evolution by natural selection by Darwin and are suitable to solve multi-objective problems [18]. The genetic variability within a population is simulated through mutation and crossover operators and the selection is done based on the survival of the fittest [19].

The optimization module can be triggered in two instances: when an emergency maintenance takes place, or when a maintenance does not finish during a maintenance shift. As a result, two types of maintenance can be identified:

- **Emergency Maintenance** - a maintenance that occurs outside a maintenance shift;
- **Scheduled Maintenance** - a maintenance that is allocated to a maintenance shift.

Emergency maintenances are more costly not only because of resources allocation, but also their impact in production. Even if a scheduled maintenance continues beyond the shift duration, the losses in production are lower because the downtime during maintenance shift is expected, which does not happen in a context of an emergency maintenance. When formulating the optimization problem, both types of maintenance are taken into consideration with different weights, as their impact is also different on production weekly goals.

The used approach follows very closely the one presented in [13]. The proposed formulation was applied to three parallel machines and can easily be applied to N parallel machines. However, other configurations require some fine-tuning in their weights and the addition of some terms depending on the problem. In summary, the goal is to extend the mentioned formulation to a more broad spectrum of layouts and adapt it to the RMS system considered in this Prescriptive System.

Every week, the production should comply with the customers orders so the GA optimizes a maximum of one week and once the current week ends, the throughputs return to their baseline unless new optimization takes place in that week and the process repeats itself once again. In this regard, each gene of the chromosome will represent the throughput of machine *i* at the day *j* as represented in Fig. 5.

$T_{1,1}$	$T_{1,2}$	...	$T_{1,j}$	$T_{2,1}$	$T_{2,2}$	...	$T_{2,j}$	...	$T_{i,j}$

Fig. 5. Chromosome Structure. Source: [13]

$T_{i,j}$ is an integer between -2 and 2 and corresponds to the machine *i* operation mode at the day *j*. Thus, the size of the chromosome is variable and equal to $i \times j$.

Companies' main goal is to attend customer's needs while remaining competitive and profitable. Therefore, it is crucial to meet production targets in the most efficient way. Accordingly, the fitness function (1) not only takes into consideration production targets but also machines' degradation.

$$F = min\Bigg[K_p\left(W-P\right)^2 + K_{sm}\sum_{i}^{N} F_{sm_i} + K_{em}\sum_{i}^{N} F_{em_i} + K_{nw}\sum_{i}^{N} F_{nw_i} + K_{ch}\sum_{i}^{N} C_{ch_i} + K_{sd}\sum_{i}^{N} S_i\Bigg] \quad (1)$$

subject to:
$$F_{sm_i}, F_{em_i}, F_{nw_i} = \{0,1,...,N\} \quad \forall_i$$
$$C_{ch_i} = \{0,1,...,d\} \quad \forall_i$$
$$S_i \geq 0 \quad \forall_i$$

The first term is the difference between production weekly target, W, and number of pieces produced, P, by the system, squared. In essence, it evaluates how far the system production is from the target and the square ensures that the algorithm does not favour solutions that exceedingly surpass the target, and the non-negativity of the values. The following three terms are regarding the different maintenances. Each type of maintenance is different and, as a result, also their weight in the fitness function. The second and third term is scheduled maintenance, sm, and emergency maintenance, em, respectively, and their different impacts were already stated. Throughout the formulation of the fitness function, initially there was no distinction between those two maintenances and the results were good so if a more broad approach is desired the maintenance might not be distinguished. However, the prescriptive system proposed has scheduled maintenance shifts integrated and the distinction between the two makes sense since they have different impacts in the production system. The fourth term is also related to maintenance, but it is regarding the first three days of the next week, nw. To increase production the throughput of some machines has to inevitably increase, which accelerates the degradation of those machines. So, this term is to prevent new failures in the beginning of the next week as it will affect the production goals of the next week.

The constant change of throughputs in a real production line is not practical. As a result, the last two terms are introduced to promote homogeneous solutions. The first term of the two, ch, corresponds to the number of changes in relation to the baseline, mode 0, and the second is the standard deviation, S, of the suggested throughputs to machine *i*.

Initially, the weights considered were the same as the ones presented in [13]. After several simulations, it was observed that the convergence of the solutions was not quite as desired. At the boundary of solutions that achieve the weekly targets and solutions with deficits, sometimes close to 2%, but with throughput rates more homogeneous, the latter were given priority (i.e., better fitness values). This behaviour was further proved by the conduction of a sensitivity analysis where the contributions from the different types of maintenance were considered constant and the remaining terms of the Equation (1) variable. As a term of comparison, margins of 1% in relation to production in regards to the desired targets were considered acceptable. So, the weights needed to be refined. Accordingly, based on the previous sensitivity analysis and additional simulations, the finals weights are as follows: $K_{\mathrm{p}} = 10, K_{\mathrm{sm}} = 900, K_{\mathrm{em}} = 1000, K_{\mathrm{nw}} = 300, K_{\mathrm{ch}} = 300$ and $K_{\mathrm{sd}} = 400$.

C. Prescriptive System

The proposed Prescriptive System involves the two modules explained above and an overview can be found in Fig. 6. Once a failure is detected and if the requirements regarding the conditions in which the maintenance will occur are met, the optimization module is triggered. As shown in Fig. 6, represented by blue rectangles, two instances of the simulation module are present: Manufacturing Environment Simulation and Simulation Module. The former corresponds to the simulation of the shop-floor of interest and the latter is an image of the former. However, in this case, its purpose is solely to feed the optimization module with the needed variables to evaluate the candidate solutions: pieces produced and number of maintenances during current week and the following one. These outputs are what allows the calculation of the solution fitness value represented by Equation (1). Additionally, in both these modules, a model to predict failures can be easily integrated. This cycle between optimization module and simulation model stops once the termination criteria is met. In this paper, the optimization stops when the maximum number of generations is exceeded. When the optimization module finishes, the best solution is recommended (white rectangle) and applied to the manufacturing environment simulation if the operator decides to.

III. System Validation and Verification

To evaluate the proposed system the testing was divided into two phases. Firstly, a set of tests are applied in order to analyze and validate the results provided by the GA as well as to prove that this system might be easily applied to configurations not fully connected or easily upgraded to handle failure in transport equipment. Secondly, scenarios that are more complex are investigated in order to check scalability. The simulation time in all tests is one working week. Also, there will be two shift changes per working day, where maintenance actions can be performed. One in the beginning of the day and other in the middle. The metrics used to assess the performance of the system are the variation of pieces produced in relation to target, named as differential, and an extension of availability per machine [20] to the whole system defined

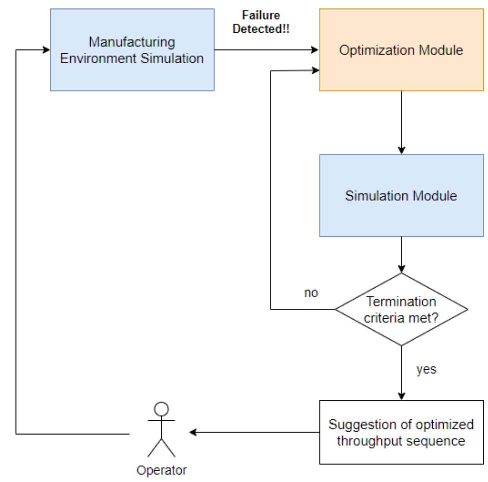

Fig. 6. Overview of the proposed Prescriptive System

by the ratio of total real operation time of all machines by the total theoretical operation time of all machines. Taking into consideration Fig. 1, the configurations will be referred as *nxm*, where *n* corresponds to the amount of stages and *m* the amount of production lines. The GA parameters were selected after several runs and set to:

- Population size = 100;
- Maximum generations = 100;
- Mutation Rate = 0.2;
- Crossover Rate = 1.0;
- Crossover Method: Single-point crossover;
- Selection Method: Elitism.

All tests were performed in a personal computer with the specifications: Intel core i5-3750 CPU @ 3.40GHz and 8.00 GB RAM.

A. First Set Scenarios

All tests were performed using a 3x2 configuration. In the first test, one of the machines is down a whole working day and another machine is on the verge of failing in the following week. In the second one, the same machine is down, however there is a second machine that fails in the middle of the week, during half-day. In the third and last test of this set, there are no broken machines but the connections from one of the machines are interrupted which isolates the equipment and, consequently, pieces processed by it have nowhere to flow to. The main goal of all tests is to understand, under different conditions, if the weekly target is achieved and how the algorithm deals with the different maintenance moments. However, the Test 3 is performed not only as a mean to study the previous statements but also as a tool to prove that this system might be applied to layouts different than the one presented in Fig. 1 where all stages are fully-connected. It may be applied, for example, to layouts where the stages have different number of machines. Also, it demonstrates that failures related to transportation equipment can be considered as long as the failure predictions are fed to the algorithm in order to trigger the optimization module.

In Table I, the effects of maintenances and connection interruptions during normal operation without optimization module are represented and summarized. Expected Production is the number of pieces produced by the system if no disturbances in the system occur. Pieces produced are the pieces that system manufactured under the conditions explained previously for each test without the intervention of the Prescriptive System. Also, differential and availability are the metrics previously explained taking into consideration that no optimization took place.

TABLE I
EFFECTS OF FAILURES IN THE SYSTEM WITHOUT OPTIMIZATION MODULE

	Expected Production	Pieces Produced	Differential	Availability
Test 1	796	731	-8,16%	96,3%
Test 2	796	698	-12,31%	94,4%
Test 3	796	607	-23,74%	92,0%

B. First Set Results

Each test was executed three times. In Table II, the averages of these three runs are presented, together with standard deviation, σ, of differentials.

TABLE II
RESULTS OF FIRST TESTING SET WITH OPTIMIZATION MODULE

	Pieces Produced	Differential	σ	Availability	Processing Times
Test 1	796	-0,044%	0,259%	96,3%	4,27h
Test 2	795	-0,084%	0,258%	94,4%	7,9h
Test 3	796	0%	0,000%	92,0%	2,87h

Recalling the conditions the test 1 was under, one of the possible outcomes could be the advancement of the failure that was scheduled to the beginning of the following week. However, this did not happen. In the second test, two optimization moments occurred, one per each failure. This is further supported by the fact that in both cases the availability did not change, which means that the downtime neither increased or decreased. In the third test, it is confirmed that the system can handle other types of situations and/or layouts. In this case, both differential and standard deviation are 0% because in all three runs the weekly target was scrupulously achieved.

C. Second Set Scenarios

Previous tests showed that the system behaves as expected so scenarios that are more complex were tested in order to investigate the scalability of the system. For each configurations tested, two types of situations were considered:

- Type A - weekly production target equal to expected production;
- Type B - weekly production target 1,2 times higher than expected production.

The purpose of type B tests is to explore situations where market demand increases and verify if the manufacturing system can still comply in those situations. Four different configurations were tested and Table III summarizes all the scenarios as well the effects of number of maintenances in the system without the optimization module. It was decided to increase the number of maintenances as the configurations increase in size in order to test similar levels of stress. This increase, in Table III, is referred as ”Number of maintenances”. Expected Production and Pieces Produced, as well as, differential and availability, have the same meaning as the presented in Table I. Each configuration has two different targets as they correspond two each type as stated before.

TABLE III
SCENARIO DEFINITION OF THE SECOND TESTING SET

Config.	Number of maintenances	Expected Production	Pieces Produced (Differential)	Availability	Target	Test name
3x3	1	1194	1113 (-6,78%)	97,5%	1194	Test1a
				97,5%	1433	Test1b
4x4	2	1532	1412 (-7,83%)	97,9%	1532	Test2a
				97,9%	1838	Test2b
7x7	5	1554	1490 (-4,12%)	97,5%	1554	Test3a
				97,5%	1865	Test3b
10x10	8	2030	1954 (-3,14%)	98,3%	2030	Test4a
				98,3%	2436	Test4b

D. Second Set Results

In all tests the target was achieved within 1% margin and, in some cases, the availability slightly increased. Those cases are marked in bold in Tables IV and V. In these instances, the increase in availability was because the algorithm “pushed” some failures to next week as a result of a reduction in the throughputs of the respective machines. In addition, this happened in higher order configurations, which indicates that is likely due to the higher redundancy in these systems.

TABLE IV
RESULTS FOR TESTS TYPE A

	Pieces Produced	Differential	σ	Availability	Processing Times
Test1a	1193	0%	0,181%	97,5%	3,0h
Test2a	1533	0,13%	0,134%	97,9%	8,7h
Test3a	1554	0%	0,273%	**98,0%**	30,9h
Test4a	2024	-0,279%	0,203%	**98,7%**	71,3h

TABLE V
RESULTS FOR TESTS TYPE B

	Pieces Produced	Differential	σ	Availability	Processing Times
Test1b	1434	0,07%	0,057%	97,5%	3,1h
Test2b	1838	0,108%	0,112%	97,9%	9,7h
Test3b	1864	-0,018%	0,241%	**97,8%**	29,5h
Test4b	2438	0,096%	0,102%	**98,5%**	77,3h

Still, in respect to the increase in availability, the comparison between Fig. 7 with Fig. 8 gives an insight of how the algorithm dealt with the different maintenance actions. These figures are related to Run 1 of test4b and its results can be found in Table VI. In Fig. 8, maintenance regarding machines J5 and G7 disappeared from the current week and the throughputs in those machines are, in general, lower than baseline. This is consistent with Equation (1) as maintenances in next week, F_{nw} are less penalizing than current week and the algorithm found a way of decreasing the fitness value by pushing the maintenance to next week without jeopardizing the achievement of the weekly target. In addition, considering once more Fig. 8, maintenance regarding machine G8 was advanced in relation to Fig. 7 however, this advancement translated into a scheduled maintenance instead of an emergency maintenance which is also consistent with the fitness function as emergency maintenances, F_{em}, are more penalizing than scheduled maintenances, F_{sm}.

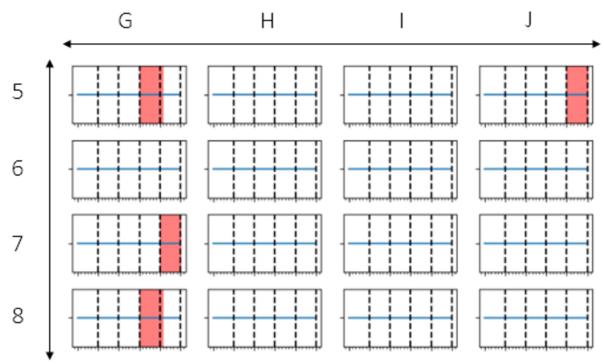

Fig. 7. Part of layout of configuration 10x10. Simulation correspondent to Run1 of test4b where no optimization took place. The red vertical bands represent the time that a machine is under maintenance and the blue horizontal lines are the throughput rates in place during certain day.

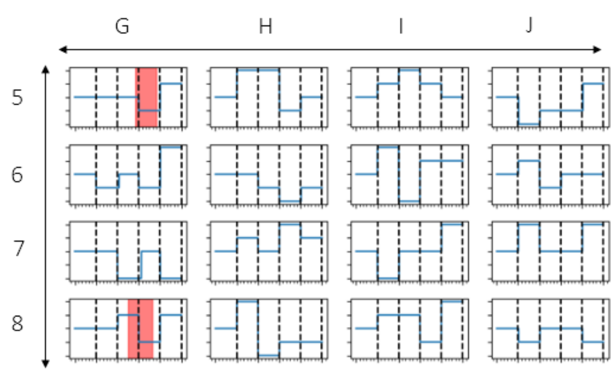

Fig. 8. Part of layout of configuration 10x10. Simulation correspondent to Run1 of test4b where the measures recommended by the Prescriptive System were adopted.The red vertical bands represent the time that a machine is under maintenance and the blue horizontal lines are the throughput rates in place during certain day.

TABLE VI
RESULTS OF RUN1 OF TEST4B

Target	Pieces Produced	Differential	Availability
2436	2441 (+5)	0,205 %	98,8 %

To evaluate how the results vary from configuration to configuration in order to draw some conclusions, the averages of the differential were plotted and the graphs are presented in Fig. 9 and Fig. 10, tests type A and tests type B, respectively.

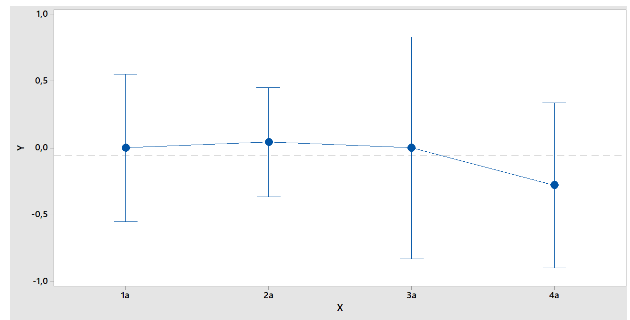

Fig. 9. Differential Averages per Configuration in tests type A

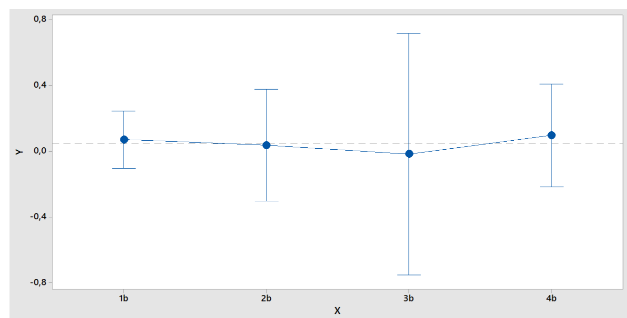

Fig. 10. Differential Averages per Configuration in tests type B

IV. DISCUSSION

The results show large improvements in the pieces differential and, in some instances, a slight increase in availability. Despite the decrease in differential, in some instances, the target value was not fully met, presenting low deficits ($<1\%$), but always by far better than the results without optimization.

The parameters of the GA are problem dependent. In the GA implementation employed in this system, both generations and population size are fixed. However the size of each chromosome is not. Remembering previous sections, the chromosome size is equal to $N \times d$ where N is the total number of machines and d, the days from the point the optimizer was triggered until the end of the week. So, not only between different configurations but also within configurations, the chromosome size varies but the parameters are not recalculated. This could led to believe that the algorithm when applied to bigger configurations would generate worse solutions.

When comparing the averages of each configuration, the desired results are that they gravitate towards zero with low deviations. The solutions seem to follow this behaviour, however, there is a visible increase in deviation from configuration 3 to configuration 4, Fig. 9, in tests of type A but it did not go beyond 1%. In fact, this corresponds to an average deviation of 0,279% as can be observed in Table IV. Therefore, this increase does not seem enough to jeopardize the results regarding the tested configurations, and can be attributed to the search strategy and convergence of the GA. However, more testing should be conducted. Despite the increase in complexity of the system, the GA model was always able to find solutions with 1% margin. As a matter of fact, the biggest differential was a deficit of 0,542% that occurred during Run 1 of Test4a. Additionally, note that the Test 4 refers to a configuration 10×10 meaning that 100 machines are operating which is already a considerable amount of equipment.

V. Conclusion

A Prescriptive System capable of adapting machines' throughput depending on variable demand and taking into consideration pending machine failures was presented. The factory is modelled based on graphs theory which allows a quick response in layout changes and the throughput sequences are managed by a simulation-based GA. The proposed system was evaluated to different layouts and showed consistent results among them – the Differential decreased and had a positive influence in the availability of the system as previously stated.

There is no denying that Prescriptive Systems rely heavily on each company's goals and specifications, which leads to one of the main reasons for the lack of prescriptive systems in the current literature. In this respect, this paper tries to tackle this gap by implementing both manufacturing simulation environment and optimization module in a considerably generic manner. Thus, regarding the optimization module, despite the objectives being already established they were chosen in order to be suitable to any manufacturing industry.

The proposed system was developed with a future integration with a Predictive Maintenance Module in mind and that would be one of the immediate improvements that could be done to this system in order to offer a whole cohesive framework that assists in the process of making decisions based on constant monitorization of the machines health status. Also, further research should be conducted not only by increasing the number of runs per tests but also explore how the system performs with real-data. In addition, other limitation that needs to be addressed are the long processing times needed which are a huge restriction when applied to real-life scenarios. In this case, the exploration of distributed or parallel GA approaches can help to overcame this constraint.

Acknowledgment

This paper is integrated in the project INDTECH 4.0 – New technologies for intelligent manufacturing. Support on behalf of IS for Technological Research and Development (SI à Investigação e Desenvolvimento Tecnológico). POCI-01-0247-FEDER-026653.

References

[1] Z. M. Bi, S. Y. T. Lang, W. Shen, and L. Wang, "Reconfigurable manufacturing systems: The state of the art," Int. J. Prod. Res., vol. 46, no. 4, pp. 967–992, 2008.

[2] Y. Koren, X. Gu, and W. Guo, "Reconfigurable manufacturing systems: Principles, design, and future trends," Front. Mech. Eng., vol. 13, no. 2, pp. 121–136, 2018.

[3] G. W. Vogl, B. A. Weiss, and M. Helu, "A review of diagnostic and prognostic capabilities and best practices for manufacturing," J. Intell. Manuf., vol. 30, no. 1, pp. 79–95, 2019.

[4] L. Zhang, J. Lin, B. Liu, Z. Zhang, X. Yan, and M. Wei, "A Review on Deep Learning Applications in Prognostics and Health Management," IEEE Access, vol. 7, pp. 162415–162438, 2019.

[5] G. Xu et al., "Data-driven fault diagnostics and prognostics for predictive maintenance: A brief overview," IEEE Int. Conf. Autom. Sci. Eng., vol. 2019-Augus, no. 1, pp. 103–108, 2019.

[6] F. Ansari, R. Glawar, and T. Nemeth, "PriMa: a prescriptive maintenance model for cyber-physical production systems," Int. J. Comput. Integr. Manuf., vol. 32, no. 4–5, pp. 482–503, 2019.

[7] K. Lepenioti, A. Bousdekis, D. Apostolou, and G. Mentzas, "Prescriptive analytics: Literature review and research challenges," Int. J. Inf. Manage., vol. 50, no. April 2019, pp. 57–70, 2020.

[8] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0," Inf. Fusion, vol. 50, pp. 92–111, 2019.

[9] F. Zhao, X. Liu, R. Peng, and J. Kang, "Joint optimization of inspection and spare ordering policy with multi-level defect information," Comput. Ind. Eng., vol. 139, no. 3, p. 106205, 2020.

[10] K. T. P. Nguyen and K. Medjaher, "A new dynamic predictive maintenance framework using deep learning for failure prognostics," Reliab. Eng. Syst. Saf., vol. 188, pp. 251–262, 2019.

[11] Z. (Max) Yang, D. Djurdjanovic, and J. Ni, "Maintenance scheduling in manufacturing systems based on predicted machine degradation," vol. 19, no. 1, pp. 87–98, 2008.

[12] Z. Yang, D. Djurdjanovic, and J. Ni, "Maintenance scheduling for a manufacturing system of machines with adjustable throughput," IIE Trans. (Institute Ind. Eng., vol. 39, no. 12, pp. 1111–1125, 2007.

[13] L. Antao, J. Reis, and G. Goncalves, "Continuous Maintenance System for Optimal Scheduling Based on Real-Time Machine Monitoring," in IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2018, vol. 2018-Septe, pp. 410–417.

[14] B. Çaliş and S. Bulkan, "A research survey: review of AI solution strategies of job shop scheduling problem," J. Intell. Manuf., vol. 26, no. 5, pp. 961–973, 2015.

[15] Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," J. Manuf. Syst., vol. 29, no. 4, pp. 130–141, 2010.

[16] S. Aguiar, R. Pinto, and G. Gonc, "Life-cycle Approach to Extend Equipment Re-use in Flexible Manufacturing," INTELLI 2016 Fifth Int. Conf. Intell. Syst. Appl. (includes InManEnt 2016) Life-cycle, no. November, pp. 148–153, 2016.

[17] C. Renzi, F. Leali, M. Cavazzuti, and A. O. Andrisano, "A review on artificial intelligence applications to the optimal design of dedicated and reconfigurable manufacturing systems," Int. J. Adv. Manuf. Technol., vol. 72, no. 1–4, pp. 403–418, 2014.

[18] S. N. Mirabedini and H. Iranmanesh, "A scheduling model for serial jobs on parallel machines with different preventive maintenance (PM)," Int. J. Adv. Manuf. Technol., vol. 70, no. 9–12, pp. 1579–1589, 2014.

[19] K. Jebari and M. Madiafi, "Selection Methods for Genetic Algorithms," Int. J. Emerg. Sci., vol. 3, no. 4, pp. 333–344, 2013.

[20] S. H. Huang et al., "Manufacturing productivity improvement using effectiveness metrics and simulation analysis," Int. J. Prod. Res., vol. 41, no. 3, pp. 513–527, 2003.