Filipe Alves
Research Centre in Digitalization and Intelligent Robotics (CeDRI), IPB, Bragança
Algoritmi R&D Centre, University of Minho
Braga
Portugal

Maria Leonilde R. Varela
Department of Production and Systems
Algoritmi Research Centre, University of Minho, Braga
Portugal

Ana Maria A. C. Rocha
Department of Production and Systems
Algoritmi Research Centre, University of Minho, Braga
Portugal

Ana I. Pereira
Research Centre in Digitalization and Intelligent Robotics (CeDRI), IPB, Bragança
Algoritmi R&D Centre, University of Minho
Braga
Portugal

Paulo Leitão
Research Centre in Digitalization and Intelligent Robotics (CeDRI), IPB, Bragança
Portugal

A Human Centred Hybrid MAS and Meta-Heuristics Based System for Simultaneously Supporting Scheduling and Plant Layout Adjustment

Manufacturing activities and production control are constantly growing. Despite this, it is necessary to improve the increasing variety of scheduling and layout adjustments for dynamic and flexible responses in volatile environments with disruptions or failures. Faced with the lack of realistic and practical manufacturing scenarios, this approach allows simulating and solving the problem of job shop scheduling on a production system by taking advantage of genetic algorithm and particle swarm optimization algorithm combined with the flexibility and robustness of a multi-agent system and dynamic rescheduling alternatives. Therefore, this hybrid decision support system intends to obtain optimized solutions and enable humans to interact with the system to properly adjust priorities or refine setups or solutions, in an interactive and user-friendly way. The system allows to evaluate the optimization performance of each one of the algorithms proposed, as well as to obtain decentralization in responsiveness and dynamic decisions for rescheduling due to the occurance of unexpected events.

Keyword: *Human-centred, Industry 4.0, multi-agent system, metaheuristics, scheduling, optimization.*

1. INTRODUCTION

Manufacturing acivities have evolved over the years through revolutions brought about by the impact, especially, on intelligence [1].

Typically, intelligent platforms and systems intend to be autonomous, for instance based on multi-Agent negotiation processes to reach automatic decisions [2, 3].

Nowadays, with the forthcoming increasing need to contextualize manufacturing companies within the scope of Industry 4.0 (I4.0), along with Cyber Physical Systems (CPS), and full digitalization of manufacturing processes, there is a growing need in both directions, sometimes to include the intervention of humans in manufacturing decision making or reverse, to enable automated solutions for manufacturing decision problems to be reached, in supporting management functions.

Besides, the insertion of I4.0 into current-day manufacturing practices will result in a move from centralized realization to a distributed realization of data-generation, optimization, decision-making, and supervisory control [1].

Moreover, this currently growing concern about human inclusion in manufacturing management is not just the focus on more traditional and local factories, but also, and even more intensely, in the context of Networked Manufacturing Environments [4, 5].

Additionally, it is also of upmost importance to provide platforms and systems that enable manufacturing and management processes and functions integration, such as scheduling and layout readjustments.

Generally, scheduling problem consists in the identification of methods to organize the operations set execution under certain time constraints (e.g. total time of execution, operations precedence constraints, etc.), as well as capacity constraints satisfaction in resources [1]. However, real world scheduling problems are generally much more complex than the ones being solved in theory. Job shop scheduling problem (JSSP) is one of the scheduling problem classes that belongs to the well-known combinatorial optimization problem domain. In this sense, production scheduling optimization technology can introduce significant improvements in the efficiency of manufacturing facilities, eliminating or reducing scheduling conflicts, reducing flowtime and work-in-process, improving the production resources utilization and adapting to irregular shop floor disturbances [6].

It is assumed that there is no flexibility of the resources scheduling (including machines and tools) for each operation of every job. It may meet the requirements of manufacturing system, such as optimization and intelligent support to bridge the lack of control and flexibility in dynamic systems with unexpected events, which may include uncertain data to be treated [7-12].

Received: April 2019, Accepted: June 2019
Correspondence to: Prof. Maria Leonilde Rocha Varela
Department of Production and Systems Engineering.
University of Minho, Guimarães. Portugal
E-mail: leonilde@dps.uminho.pt
doi:10.5937/fmet1904699A

Regarding the state of the art in this field, the literature review leads to surveys related to different strategies and techniques of artificial intelligence and optimization in the JSSP in manufacturing systems [13, 14]. Moreover some authors address, for example, flexible job shop scheduling using meta-heuristics based on multi-agents [15, 16]. It is also important to note that few studies present the job shop scheduling problem for efficient and simultaneous layouts ajustment [17]. Although, some authors integrate already, in a collaborative perspective, the job shop scheduling with layout planning, that is, presenting a hybrid system capable of optimizing multiple solutions [18, 19].

The main idea of this work is also to provide a contribution for simultaneous (re)scheduling and layout adjustment based on a hybrid approach based on a multi-agent system combined with the genetic algorithm and the particle swarm optimization algorithm, in a dynamic manufacturing and management environment. In this sense, the proposed approach and underlying system, regarding an update to existing works, is to consider and enable an integration between optimization techniques and agents negotiation by providing dynamic and autonomous simulations that involve and overcome disruptions or unexpected events.

Therefore, in this paper a human centred hybrid decision support system is proposed to simultaneously integrate scheduling and plant layout adjustments. This proposed system is based on multi agents and meta-heuristics, although it can also be used to obtain automatically generated solutions to integrated problems in an interactive and user-friendly way.

This paper is an extension of the paper in [20] on which the following three main issues were incorporated:

(1) a new <u>population-based meta-heuristic, the PSO</u>, was included, for the difference in relation to the GA algorithm in [20], which turns out to be more feasible to solve simultaneous scheduling and layout adjustment problems with increased complexity, and a more complex problem extracted from a real case study is used in this paper for illustrating this improved approach, with a higher quantity and variety of jobs to be processed on a set of machines, within the context of FMS;

(2) <u>decision factor tests</u> are included in this extended approach, as an alternative and improved method to compare the possibility of trying to put the remaining tasks to be rescheduled in the middle or final time range on a given available machine through the MAS negotiation process, after a breakdown or unexpected event occurs in the manufacturing system; and

(3) <u>dynamic rescheduling alternatives</u> were implemented in this improved approach for more easily overcome disruptions problems affecting the rescheduling and layout rearrangements, based on further products' and jobs' information, such as priorities.

The paper is organized as follows. In Section 2 we provide the background for the problem under study and its formulation, both for an optimized approach and for a dynamic approach. Section 3 presents the system architecture. In Section 4 the optimization methods of centralized scheduling will be briefly described. The model based on agents, their behaviors and interactions will be shown in Section 5. Therefore, the experimental simulation protocol is illustrated. The case study is presented in Section 6 and the discussion of the results is provided in Section 7. Finally, Section 8 rounds up the paper with conclusions and future work.

2. PROBLEM DESCRIPTION

The present work aims to formulate a dynamic and flexible manufacturing system, with the possibility of creating different n products (jobs): J_1, J_2, … , J_n. In order to achieve the production of the mentioned products, different types of processes (operations) must be manufactured, denoted by O_{ij} ($i = 1,\ldots,n$, $j = 1,\ldots n$), that represents the operation i of the job j and n_j is the number of operations of the job j. This set of operations can be performed on a set of positions/machines: M_1, M_2, …, M_k, that are responsible for completing the process manufacturing operations, given a processing time to perform the O_{ij} operation. Regarding the set of machines, they may be able to complete the same manufacturing operation, while some operations will have to be specific to a single machine (dedicated and more flexible - multi purpose machines). In addition, all machines are continuously available at the beginning of the system, with their static availability at the beginning of scheduling (dynamic problem with arrival time = 0).

In this work the following constraints are considered, which are rules that limit the possible assignments of the scheduling:

- The products are independent, with the possibility of having priorities assigned to some type of work, that is, the user can identify and assume the priorities according to the needs.
- Each machine processes only one operation at a time.
- All jobs are available simultaneously at zero time.
- There are no precedence constraints between the operations of different jobs.
- After a job is processed on a machine, it is immediately transported to the next machine.
- Failures and disruptions will be considered during processing, thus simulating a real environment.

In this way, it is possible to verify the instance of the problem illustrated of Figure 1.

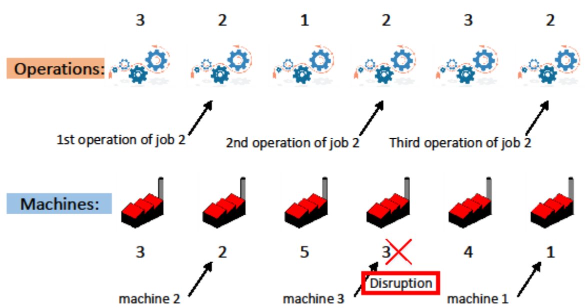

Figure 1. Encoding Scheme.

2.1 Layout Adjustment

The layout is a feature inherent in any operation. In this way, it is necessary to continuously optimize the layout arrangement, not only for the scheduling of inputs but also to improve the profitability of the outputs (Figure 2).

Through the form and scheduling, appearance and the way materials, information and customers flow through the layout, the fullness of functions will be triggered [21]. The layout optimization problems are quite complex and in general are NP-hard, which requires a high computational complexity.

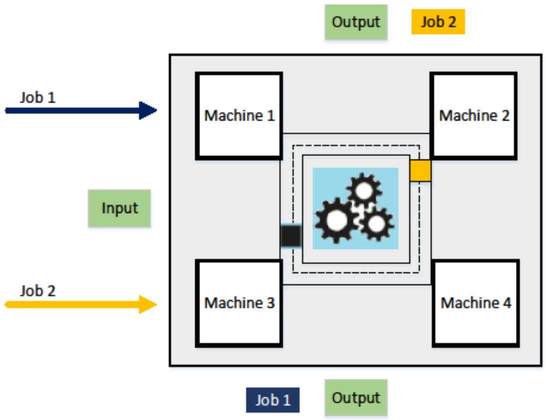

Figure 2. Design of layout problems and variants (adapted [20]).

The layout problem focuses on the flexible scheduling, reconfigurable, and agile manufacturing environments where the demand is affected (by machine disruption for example) [21]. With dynamics layout models embedded in optimization models and combined with multi-agent systems, it will be possible to minimize the sum of material handling costs. On the other hand, it will be possible guarantee the best disposition of human and material resources, allowing efficiency, but also dynamic and autonomous re-arranging from the agent-based model.

2.2 Problem formulation

The problem consists in finding the jobs/operations schedule on the machines, taking into account the precedence constraints minimizing the batch makespan, i.e., the finish time of the last operation completed in the schedule.

Therefore, consider O_{ij} the operation i of the job j and O_j represents the set of all operations of the job j. Define $l = \max\{n_j, j = 1,\ldots,n\}$ as the maximum number of the operations in a given job and n is the jobs total number. It is necessary also to define the matrix MS, with size $l \times n$, representing the machines schedule. MS_{ij} represents the machine where the operation i of the job j will be done. The matrix t, with size $l \times n$, represents the time schedule. On the other hand, t_{ij} represents the final time of the operation i of the job j.

Consider the vector $x = (x_1,\ldots,x_{nx},y_1,\ldots,y_{nx})$ where x_q represents a given operation, y_q represents the machine where the operation x_q will be perform and nx represents the total operations number to execute the n jobs. Thus, it is necessary to solve the following optimization problem defined as:

$$C_{\max} = \min\left(t_{sp}\right) \tag{1}$$

where s is the last operation of the last job p.

The main question was if the jobs order affect the optimal time. For that it is needed to solve this problem several times, where the difference is the jobs order. To solve the minimization problem presented previously, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) Algorithm were prepared and used to solve the problem and are presented in the following sections.

In a first approach the above formulation will be used in conjunction with the optimization methods to optimally process and solve the problem, where each job can be processed on each machine and the processing times of jobs are controllable. Machines are non-identical in the sense that each job has different technical specifications and manufacturing properties on different machines.

In a second approach, a rescheduling problem will be taken into account, where, assuming the original scheduling, it will be submitted to an disruption.

A breakdown occurs on one of the machines. Disrupted machine will be unavailable for a certain period of time for maintenance activities. During this period, the machine will be down and will not process any operations. In the original schedule, each job has a planned completion time. One of the objectives of our rescheduling problem is to reschedule jobs so that they will be completed before their original completion times, if possible. We assume that once a disruption occurs, the decision-maker can move each operation from its original machine to another machine at a very short time, based on negotiation protocols (explained later). Given this scheduling environment, the problem is to reallocate the operations to the machines, resequence them and replan their processing times to minimise the total manufacturing cost and the number of disrupted jobs objectives. As we generate a new schedule, some of the jobs may complete later than their original completion times. By making appropriate reallocation, resequencing and compression decisions one may decrease the number of disrupted jobs. However, one would expect that this will increase the total manufacturing time of the schedule. Therefore, the idea will be to solve the problem dynamically, centered on the minimum human interaction but, using the interaction of the agents in terms of communication and negotiation of the scheduling.

3. SYSTEM ARCHITECTURE

The information trade-off between the timing of the arrival of works, the creation of setups and scheduling, the assignment of priorities, the testing of disruptions and the obtaining of the solutions for simulation of a real and dynamic job shop environment, will be according to an architecture that encompasses all these parameters in a distributed system.

In this context, the system architecture for the manufacturing industry scheduling system integrates off-line and on-line modules, presented in Figure 3, which can deal sequentially with the two sub-processes: optimized planning using optimization methods and real-time responsiveness solutions using MAS. The two modules are able to exchange information, balancing the decision-making according to the needs.

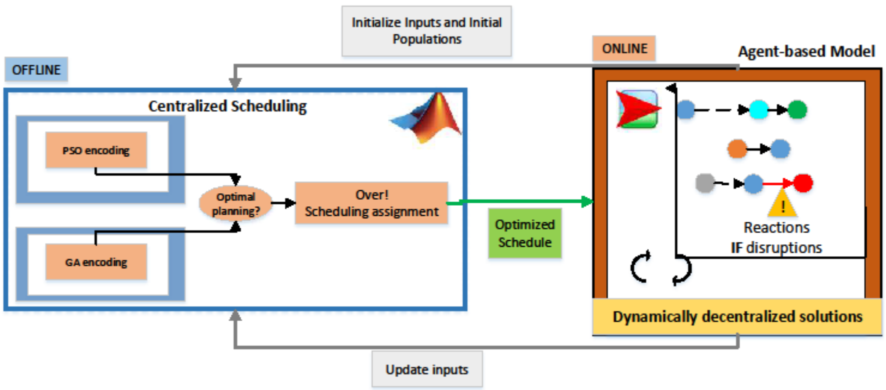

Figure 3. Developed system architecture.

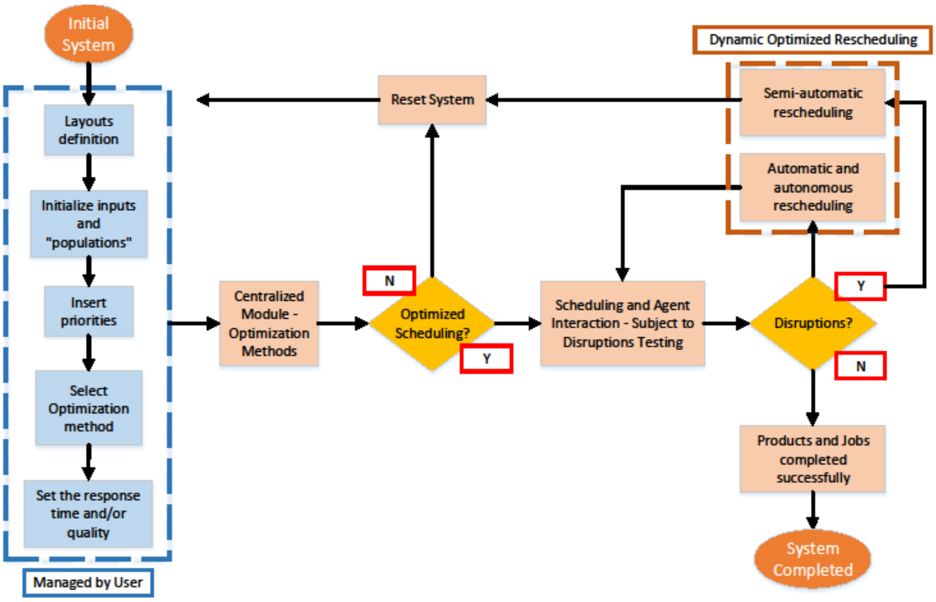

Figure 4. Flowchart of system architecture and decision-making.

The left module performs the optimized scheduling for the jobs, running off-line, for a given situation under study.

The right module concerns to the dynamic re-scheduling, is response to disruptions or condition changes, e.g. a broken process in some situation or a failure in some position. In this module, the re-scheduling is obtained by the interaction of individual entities, each one reasoning about its own schedule. The individual entities (jobs and machines) can share information and have the ability to negotiate schedules in order to overcome failures or interruptions, that is, they can obtain autonomous and dynamic re-scheduling solutions without human intervention. On the other hand, the user will also be able to perform information updates and get new re-schedulings from the optimization methods.

Regarding the system architecture presented above, it is possible to analyse the different interactions of the system with the user and the possible decision-making in the flowchart of the Figure 4

The user will be a key element in the system. However, the scheduling in case of disruptions can be updated autonomously and automatically, using protocols of interaction and negotiation of agents, without the need for human intervention. This will allow fast response in decision-making, ensuring the quality and success of the scheduling task.

4. OPTIMIZATION METHODS FOR CENTRALIZED SCHEDULING

It is essential to develop efficient decision support methods to solve job shop scheduling problems, because the operators need to test several scenarios, which makes time requirements crucial for this applications. This section describes the approach embedded in the off-line module to perform the optimized scheduling by Genetic Algorithm and Particle Swarm Optimization method.

4.1. Genetic Algorithm

Initially proposed by John Holland [27], GA inspired by the natural biological evolution, uses a population of individuals to apply genetic procedures: crossover between two different individuals or/and mutation in one individual. The values of the control parameters used in GA were adjusted to a suitable experience of the problem, i.e. it was considered a population size (Ps) and concerning the probability of the procedures (crossover and mutation), 50% rate was selected. Is expected that the following population (next generation) of individuals has a better capability. The algorithm repeats the crossover and mutation procedures in new populations until the desired diversity of solutions is performed [28, 29].

The GA is summarized in the following algorithm (Figure 5).

Algorithm 1. Genetic Algorithm

Generates a randomly population of individuals, $\mathcal{P}^0$, with dimension N_{pop}. Set $k = 0$.
while stopping criterion is not met **do**
  Set $k = k + 1$.
  $\mathcal{P}'$ = Apply crossover procedure in population $\mathcal{P}^k$.
  $\mathcal{P}''$ = Apply mutation procedure in population $\mathcal{P}^k$.
  $\mathcal{P}^{k+1} = N_{pop}$ best individuals of $\{\mathcal{P}^k \cup \mathcal{P}' \cup \mathcal{P}''\}$.
end while

Figure 5. GA algorithm.

The iterative procedure terminates after a maximum number of iterations (*NI*) or after a maximum number of function evaluations (*NFE*).

4.2. Particle Swarm Optimization Algorithm

The Particle Swarm Optimization Algorithm was developed by Kennedy and Eberhart and it is based on natural social intelligent behaviors [22].

PSO method is a computational method that optimizes a given problem by iteratively measuring the quality of the various solutions. This method consists of optimizing an objective function through the exchange of information between individuals (particles) of a population (swarm). The PSO algorithm idea is to perform a set of operations and move each particle to promising regions in the search space. The Particle Swarm Optimization method also works with a population of solutions and stops when the stopping criteria are met [23, 24]. At each iteration the velocity of each individual is adjusted. The velocity calculation is based on the best position found by the neighborhood of the individual, the best position found by the particle itself - *xbest* and the best position found by the whole population, taking into account all individual - *gbest* or the best position overall [25]. The steps of PSO are summarized in the following algorithm (Figure 6).

Algorithm 2 : Particle Swarm Optimization Algorithm

Generates a randomly population of individuals, $\mathcal{P}^0$, with dimension N_{pop}.
Set the values of w, c_1, r_1. Define c_2, r_2 random numbers in $[0, 1]$. Set $v_i = 1$, for $i = 1, ..., N_{pop}$, and $k = 0$.
while stopping criterion is not met **do**
  Set $k = k + 1$.
  Update the value of $xbest_i$ for the individual with index i, for $i = 1, ..., N_{pop}$.
  Update the value of $gbest$ for all population $\mathcal{P}^j$, for $j = 1, ..., k$.
  Update the individual velocity according to:

$$v_i^{k+1} = wv_i^k + c_1r_1(xbest_i - x_i^k) + \lfloor c_2r_2 \rceil (gbest - x_i^k).$$

  Update the individual position according to: $x_i^{k+1} = x_i^k + v_i^{k+1}$.
  If necessary, adapt x_i^{k+1} to a feasible schedule.

Figure 6. PSO algorithm.

During the iterative process if x_i^{k+1} is not a feasible solution, the coordinate that is not feasible will be projected to the feasible region [26]. The iterative procedure terminates after a maximum number of iterations or after a maximum number of function evaluations.

5. AGENT-BASED MODEL

The on-line module considers the use of MAS to implement the dynamic and responsive re-scheduling in case of disruption. In addition, it will be possible with this module to interact, simulate and visualize the scheduling performed by the off-line module. For this purpose, the agent-based model will be described in this section.

5.1 Behaviour of agents

The agent-based model considers three types of agents:

- **Machine agent**: These agents represent the work machines, which will allow the set of manufacturing operations in order to obtain the products concerned. They are immobile, totally passive, and do not take the initiative to start the decision process.
- **Operation agent:** These agents represent the "operations" (a job with a set of operations) that move around to provide the manufacturing of the products according to the scheduling and location of the machines agents. Operations agent only interacts with their own machines.
- **Product agent:** These agents represent the final product to be obtained. After the complete set of operations they require, they send a warning message informing the user of their conclusion.

In this paper, only the global behaviour of process agents will be described, since they are the ones that move and interact with the other agents. Figure 7 presents the two main categories of global process behaviour: the passive and autonomous behaviours.

In the passive behavior, the processes follow and move carefully the planned optimized schedule provided by the off-line module, i.e. using the GA or PSO methods codified in MatLab, without taking into account the disruptions or external problems.

In the autonomous behavior, the process agent follows the planned route but is able to dynamically adapt the schedule in case of disruptions through the interaction with other processes, which may be available and with operations in common, to re-route the allocation that was previously allocated to the broken process/machine.

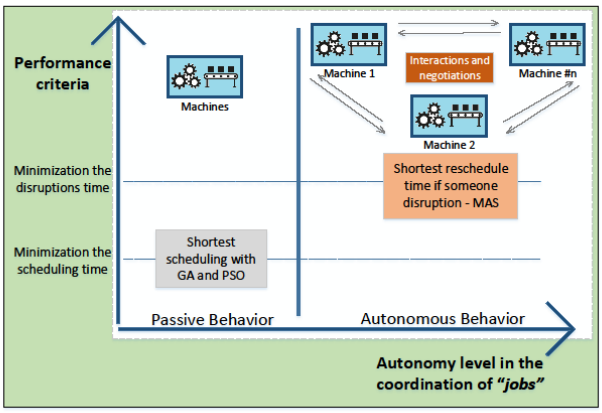

Figure 7. Performance criteria and autonomy level in schedule (adapted [15]).

In the passive behavior, the processes follow and move carefully the planned optimized schedule provided by the off-line module, i.e. using the GA or PSO methods codified in MatLab, without taking into account the disruptions or external problems.

In the autonomous behavior, the process agent follows the planned route but is able to dynamically adapt the schedule in case of disruptions through the interaction with other processes, which may be available and with operations in common, to re-route the allocation that was previously allocated to the broken process /machine.

5.2 Interaction patterns in autonomous behaviour

Initially, this dynamic involves passive behaviour, since it follows the planned schedule, it is subject to disruption tests or processes failures to evaluate interaction and autonomous cooperation between agents.

This dynamic procedure involves decisions that are considered to support a machine in the disruption event (which becomes unavailable) when choosing your support, in other words, the choice of another available machine that can take the remaining schedule of operations to allocate. Each decision attempts to optimize the following criteria:

- The products waiting time when a machine is interrupted on the schedule.
- Optimize schedule time and its cost.

In this case, the set of operations of the processes follow the planned route defined by GA or PSO methods. When occurs a disruption in the schedule (it can be an instantaneous interruption or caused by the user), the procedure can opportunistically select other available process in an on-line reactive way (by comparing joint operations and their availability). Thus, the waiting time for a solution of the problem is minimized, taking into account the final time of the scheduling.

As depicted in Figure 8, the "Initiator", the disruption machine, asks for an available machine in the set of "Participants". These interactions are based on contract net protocol [30], that allows the communication and control in the problem-solving, by a negotiation process.

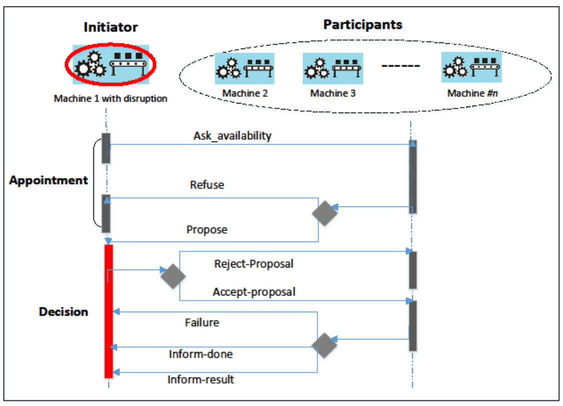

Figure 8. Interactions and negotiations between "machines".

The initiator calls the processes until its finds one or more available, in other words, the process must be still on schedule moving and be able until the operation task is completed. As shown in the Figure 8, the initiator scans the agents list and checks if there are machines available to take over the remaining schedule. Finally, when the best proposal with the most optimized time is found, the initiator decides and a participant will assume and update its schedule with the remaining operations.

As previously mentioned, the autonomous behaviour includes the interaction with the machines and uses as a decision factor the comparison of the shortest time between the remaining schedule and the next or final operation of the available machine schedule. It should be noted, that the remaining schedule is already optimized by the GA or PSO methods and the precedence of the operations given by them is mantained along the process. It is important to note that for interacting with an available machine, this machine will need to be able to perform the operations left. The machines are not identical, so it is necessary to take into account if the specifications are adequate to be able to carry out the rescheduling process. Therefore, this decision factor switches between the manufacturing goal (minimize the time and reduction costs of production) and the machines goal (minimize the time with disruptions).

For decision factor tests, an alternative method to compare the possibility of trying to put the remaining schedule in the middle or final time range on the available machine was used, as illustrated in Figure 9.

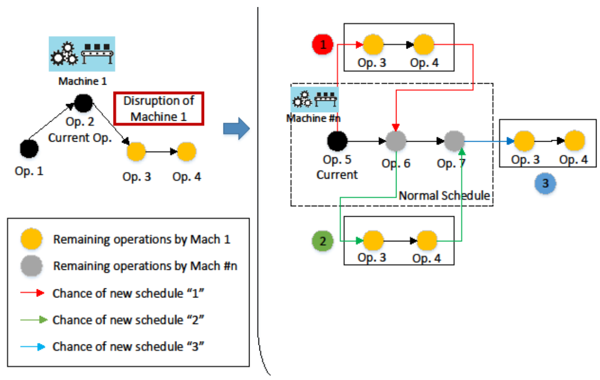

Figure 9. Decision factor for rescheduling.

Therefore, the agent model will have the decision, depending on its needs, where the remaining schedule will be tested in the various intervals in the schedule of available machine (trying to fit the remaining schedule in the range with the shortest distance and time).

The final solution always looks for the total schedule (job schedule available "+" remaining schedule) with the most optimized time.

5.3 Simulation experiment protocol

The described agent-based model was implemented in NetLogo [31], which is an agent-based modelling and simulation platform that allows to rapidly instantiate models to observe the behaviour agent-based systems. It provides an intuitive user interface where one can add buttons and control widgets to easily manipulate a model to view different scenarios [32]. As previously described, the agent-based model developed in NetLogo is connected with MatLab to allow the exchange of the optimized scheduling solution. It is often advantageous to implement separate portions of a model in the most appropriate language and to combine the results dynamically. The MatLab - NetLogo extension (MatNet) provides new functions within NetLogo that allow data passing between NetLogo and MatLab, and the calling of any valid, online MatLab commands from within NetLogo. The new tool presented herein, as does occur in other typical integrated use of tools, facilitates future dynamic integration of software platforms [33, 34]. Figure 10 presents the NetLogo interface for the implemented agent-based model. In each simulation, some parameters are used to vary the populations of the agents according to the database. For the case study, the simulation protocol involved three crucial steps, it can be applied to many other simulations within the database, thus:

- Step 1: The user will have at his disposal the various specifications of the problem. It will be possible to select the different jobs to carry out, being able to assign priorities (for example, taking into account deadlines) and also to define the quantities of operations from the selection already existing in the database. On the other hand, the set of machines available for the operations will also be assigned on the layout and their coordinates remained unchanged during the experiments. It is possible also to select simulation with disruption (tested after certain ticks or selected by the user), the optimization method and also the number of iteractions.

- Step 2: The scheduling is performed by the user, which can be set the desired number of iterations depending on need, the time and quality of the solutions and also the optimization method to perform the scheduling. The information will be transmitted and the overall schedule of the optimization method for the jobs planning (output viewer) will be received.

- Step 3: It is displayed the jobs schedule sequence and if there is disruption, we can update the dynamic reschedule by agents in their autonomous behaviour.

In conclusion, the trade-off between centralized decisions and dynamic interface provides easy visualization and handling for the independent user.

This data was created in an attempt to generate different products (jobs), which may reflect a generic case of a job shop scheduling. The operation type list ($\bar{O}_i$ defines the type of process/operation and the processing times. In this case study, there are four jobs (Job 1, Job 2, Job 3, Job 4), eight different manufacturing operations types ($\bar{O}_1$, and an order set of operations (#1, #2,…, #10), as shown in the Table 1.

In some jobs, for a given sequence of operations, it is possible to have repetition of a given process type. Thus, for a given job j, the operation type list will define the order set of operations O_{ij}, that means the order #i (i = 1,...,n_j) by the Job j. For example, the Job 1 needs a sequence of seven operations, where two of them are equal ($O_{21} = \bar{O}_2 = O_{31}$). So, $O_{ij} \in \{\bar{O}_1, \bar{O}_2, ..., \bar{O}_8\}$ for i = 1,..., n and j = 1,..., n.

Table 1. Production sequence for each type of job.

	Job 1	Job 2	Job 3	Job 4
#1	$\bar{O}_1$	$\bar{O}_1$	$\bar{O}_1$	$\bar{O}_1$
#2	$\bar{O}_2$	$\bar{O}_2$	$\bar{O}_2$	$\bar{O}_2$
#3	$\bar{O}_2$	$\bar{O}_2$	$\bar{O}_2$	$\bar{O}_2$
#4	$\bar{O}_4$	$\bar{O}_2$	$\bar{O}_3$	$\bar{O}_3$
#5	$\bar{O}_6$	$\bar{O}_3$	$\bar{O}_4$	$\bar{O}_5$
#6	$\bar{O}_7$	$\bar{O}_5$	$\bar{O}_4$	$\bar{O}_4$
#7	$\bar{O}_8$	$\bar{O}_4$	$\bar{O}_6$	$\bar{O}_6$
#8		$\bar{O}_6$	$\bar{O}_6$	$\bar{O}_7$
#9		$\bar{O}_7$	$\bar{O}_7$	$\bar{O}_8$
#10		$\bar{O}_8$	$\bar{O}_8$	

Besides this, the layout of the manufacturing system is composed by eight machines, each one being able to perform a set of operations. The machines are responsible for the completion of manufacturing operations to do the jobs. Some machines are able to complete the same manufacturing operation, while some operations can only be completed on a single machine. Additionally, each machine is continuously available as the system start and each machine can process only one operation at a time. The cell is composed with eight machines and the processing times of the eight manufacturing operations are described on Table 2.

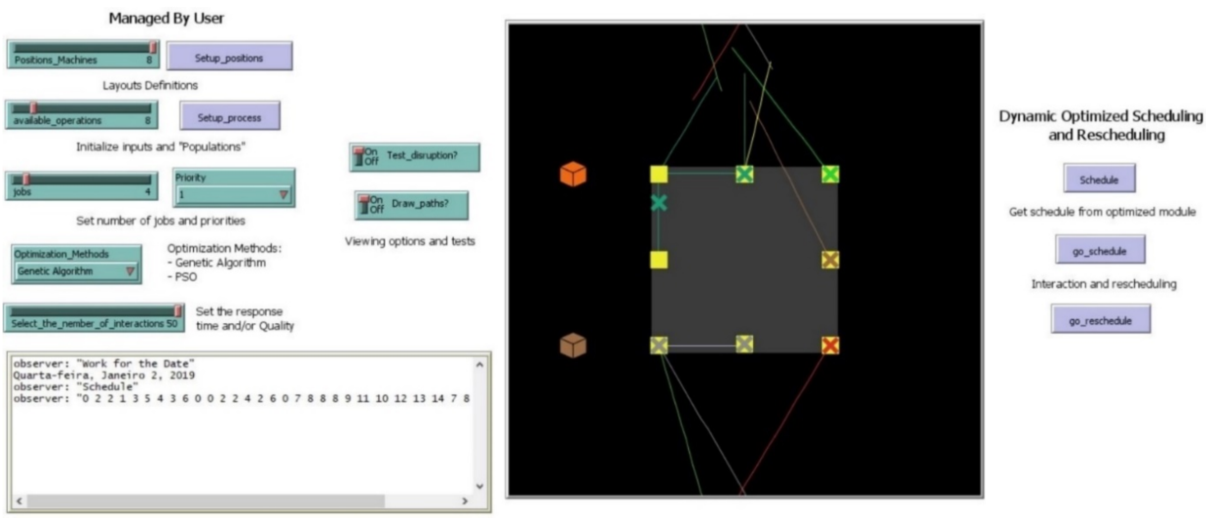

Figure 10. NetLogo interface managed by the user.

Table 2. Processing time of each manufacturing operation (in seconds)

Operations type	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
$\bar{O}_1$	10							
$\bar{O}_2$		20	20					
$\bar{O}_3$		20		20				
$\bar{O}_4$					20			
$\bar{O}_5$				30		30		
$\bar{O}_6$			20		20			
$\bar{O}_7$							30	
$\bar{O}_8$	10							10

6. RESULTS DISCUSSION

For case study, simulations were carried out on a PC Intel(R) Core(TM) i7 CPU 2.2 GHz with 6.0GB of RAM using the MatLab software.

In an attempt to plan the schedules automatically, two computational algorithms were used — GA and PSO. The parameters used by the two methods were $Npop$ = 30, w = 1 and $c_1 = r_1 = 2$. Since they are stochastic methods, each implementation was tested with 10 executions in order to evaluate the results obtained and compare them between both algorithms with random initial points.

The values of the control parameters used in the algorithms were adjusted to a suitable experience of the problem, i.e. both methods used the same stop criteria, limit the number of function evaluations to NFE = 5000 or after 100 iterations (NI). It is important to also note that, for example, regarding the probability of the procedures crossover and mutation of GA, 50% rate was selected. In the present case study, four products are created.

The objective of this round of experiments (i.e., optimized scheduling with passive behavior and disruptions tests with autonomous/dynamic behavior) was to highlight the efficiency of the proposed approach defined by the ability of the latter to minimize the time and automatically optimize disruptions or failures in machines.

The scheduling tool, that represent the schedule obtained by both optimization methods, had 100% of successful rate since they found a feasible solution in all runs. In turn, the scheduling is obtained extremely fast by GA and PSO methods.

On the other hand, allowed to obtain the final solution that characterizes the optimal time of execution of the jobs in question, tested for unprecedented scheduling. Table 3 presents the summary of both methods, such as: the best solution obtained in all runs (C_{max}) in seconds, and, the average time to solve the optimization problem ($Time_{avg}$) also in seconds. It is important to note that the scheduling obtained by the GA and PSO methods has always ensured a feasible solution as well as it allows to identify the sets of operations to their respective machines.

Figure 11 presents the optimized solution by GA to the problem under study as well as the subject dynamic rescheduling in tests of machine disruptions. At the top of the Figure 11 it is possible to verify the set of operations, identifying their respective machines and allocating themselves properly in the layout for the production of the jobs.

Table 3. Results obtained by GA and PSO methods.

	GA	PSO
$Time_{avg}$	29,86	21,7
C_{max}	300	270

Consequently, it is also possible to analyze that the time required to create Job 1, Job 2, Job 3 and Job 4 is respectively 130, 240, 280 and 300 seconds (optimized scheduling at the top of the image). After this, using the interface in NetLogo, the same scheduling of the case under study was subjected to a random disruption of a machine after a certain time (in this case machine 3, represented by red “X”), leaving four operations, one of each job without conclusion, as shown at the bottom of the Figure 11.

This solution enabled not only the minimal human interaction but also allowed to finish the jobs in optimized time, affecting only substantially the end of job 4 (now needing 320 seconds), keeping all other jobs in accordance with the optimized solution and/or priorities imposed.

On the other hand, the Figure 12 presents the optimized solution by PSO algorithm to the problem under study as well as the subject dynamic rescheduling in tests of machine disruptions.

Similar to previous analysis, it is also possible to analyze that the time required to create Job 1, Job 2, Job 3 and Job 4 is respectively 130, 240, 210 and 270 seconds (optimized scheduling). After this, using again the interface in NetLogo, the same scheduling of PSO algorithm was subjected to a random disruption of a machine after a certain time (machine 3, represented also by red “X”), leaving four operations, one of each job without conclusion, as shown at the bottom of the Figure 12.

However, through the agent-based model, the dynamic and autonomous rescheduling protocol was started to allocate the remaining operations in the layout (again represented by red circles). They were allocated to the machine 5 and 2, with some disturbance on the global schedule. This solution, one mor time, enabled not only the minimal human interaction but also allowed to finish the jobs in optimized time, affecting only substantially the end of Job 2 and Job 4 (now needing 300 and 260 seconds, respectively), keeping all other jobs in accordance with the optimized solution and/or priorities imposed.

It is important to note that, despite a little worsening of global scheduling through Job 2, it has consequently improved the conclusion of Job 4.

Despite this, the PSO method proved robustness and efficiency when achieving the optimal solution quickly, and, in case of disruptions, the agents interact in a more sophisticated way and achieve better results than with the GA method.

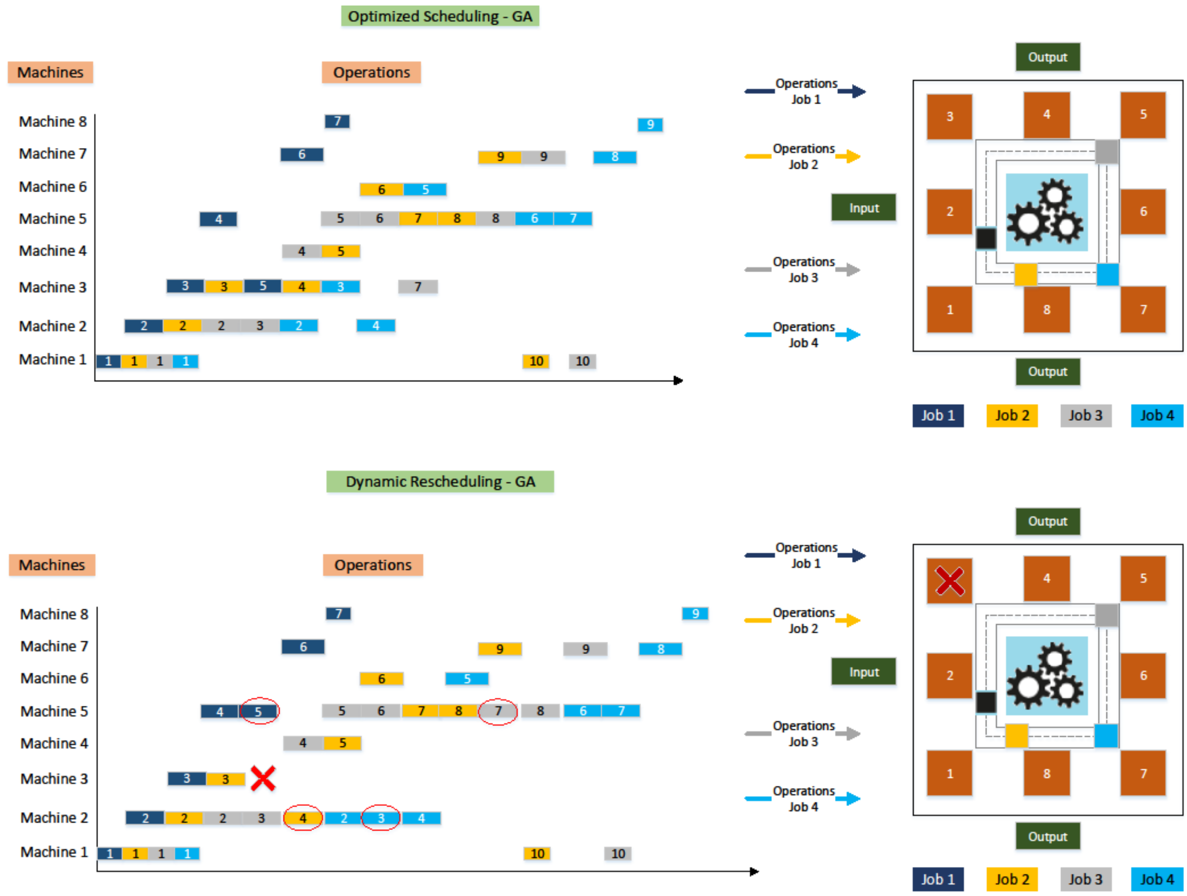
Figure 11. Results of optimized scheduling (top) and rescheduling (bottom) by GA algorithm and agents

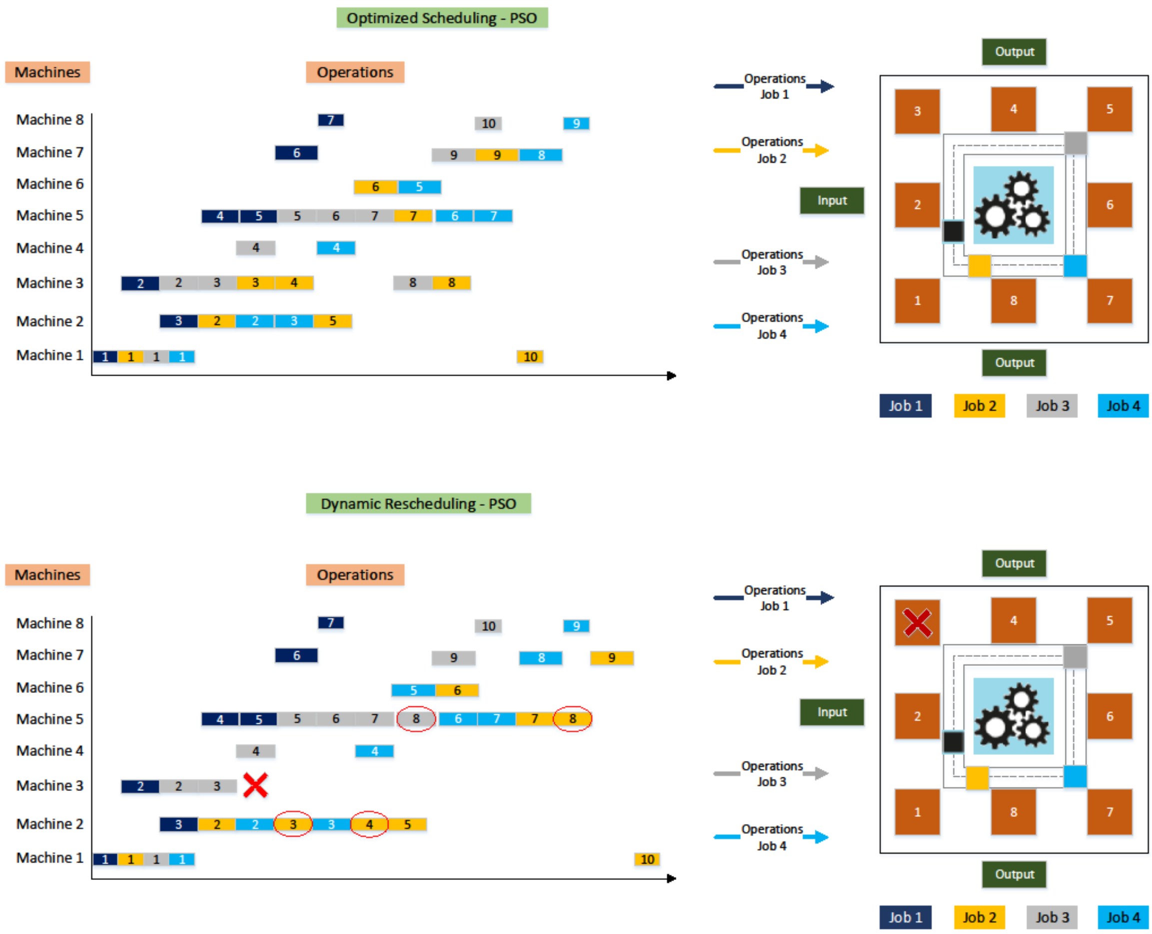
Figure 12. Results of optimized scheduling (top) and rescheduling (bottom) by PSO algorithm and agents.

7. CONCLUSION

In this paper we discussed a hybrid MAS and meta-heuristics based system for simultaneously support scheduling and plant layout adjustment in a dynamic context with several constraints, uncertainties and random events, for example, disruptions in machines.

The proposed approach is optimized and supports dynamics, allowing optimized scheduling solutions from centralized tool, presenting also forecasting information under uncertain contexts.

The main contribution of this paper as to provide a new way of solving a job shop scheduling problems in simultaneous with layout adjustments, using the machines ability to dynamically adapt and design his own schedule in case of external disruptions. Thus, we can improve both flexibility and efficiency in today's competitive manufacturing environments, such as flexible manufacturing systems and just-in-time production, as well as add strategies and dynamics to improve quality, time and increase profits.

An experimental case study was proposed based on job shop scheduling. Thus, through the results presented, it was possible to obtain a fast and optimized scheduling by the centralized module and at the same time interact with the NetLogo platform. In this sense, the same scheduling was subject to random interruption of a machine, which autonomously and with the use of a negotiation protocol between agents, allowed the dynamic rescheduling of the remaining operations. This approach did not only enable to optimize solutions but also to ensure minimal human intervention in real-time disruption testing for layout re-arranging and self-rescheduling.

In order to validate the performance and robustness of the proposed approach, the analysis was presented based on two different algorithms – each with different solutions, times and criteria data quality and capabilities. However, the optimized results presented by the PSO algorithm as well as the rescheduling presented better results than the GA method applied in [20], not only in the obtained solution, but also in the scheduling itself subject to interruptions or failures. Therefore, the PSO algorithm presents itself as a good method to optimize the scheduling and rescheduling in a job shop scheduling problem in an optimized, fast and flexible way. The developed interface allows the user a way of communication and interaction, with two modules that combined, allow the improvement of flexibility and efficiency in manufacturing environments, with dynamics and strategies to improve the quality of solutions and thus increase profits.

In summary, it is worth concluding that this improved approach is also highly customizable and flexible. Further, the capability to deal with business problems in uncertain environments provides a good solution for decision support scenarios where environment quality may be compromised.

For future work, it is possible to reformulate and increase the complexity of the problem, for instance, for solving a problem with big data. Another approach to be further developped could be to use a decentralized approach that nevertheless combines optimized solutions, but without the need for a centralized tool.

ACKNOWLEDGMENT

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019.

REFERENCES

[1] Upasani, K., Bakshi, M., Pandhare, V., and Lad, B.K.: Distributed maintenance planning in manufacturing industries, Computers & Industrial Engineering, Vol. 108, No., pp. 1-14, 2017.

[2] Ha, S.H.: Agent-Based Decision Making in the Electronic Marketplace: Interactive Negotiation, in: Editor (Ed.)^(Eds.): Book Agent-Based Decision Making in the Electronic Marketplace: Interactive Negotiation (Springer Berlin Heidelberg, 2010, edn.), pp. 1-10.

[3] Khedri Liraviasl, K., ElMaraghy, H., Hanafy, M., and Samy, S.N.: A Framework for Modelling Reconfigurable Manufacturing Systems Using Hybridized Discrete-Event and Agent-based Simulation, IFAC-PapersOnLine, Vol. 48, No. 3, pp. 1490-1495, 2015.

[4] Guo, L., Wang, S., Kang, L., and Cao, Y.: Agent-based manufacturing service discovery method for cloud manufacturing, The International Journal of Advanced Manufacturing Technology, Vol. 81, No. 9, pp. 2167-2181, 2015.

[5] Nagorny, K., Colombo, A.W., and Schmidtmann, U.: A service- and multi-agent-oriented manufacturing automation architecture: An IEC 62264 level 2 compliant implementation, Computers in Industry, Vol. 63, No. 8, pp. 813-823, 2012.

[6] Li, X., and Gao, L.: An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem, International Journal of Production Economics, Vol. 174, No., pp. 93-110, 2016.

[7] Arrais-Castro, A., Varela, M. L. R., Putnik, G. D., Ribeiro, R., & Dargam, F. C. (2015). Collaborative negotiation platform using a dynamic multi-criteria decision model. *International Journal of Decision Support System Technology* (IJDSST), 7(1), 1-14.

[8] Varela, M. L. R., & Ribeiro, R. A. (2014). Distributed manufacturing scheduling based on a dynamic multi-criteria decision model. In *Recent Developments and New Directions in Soft Computing* (pp. 81-93). Springer.

[9] Vieira, G., Varela, M. L. R., & Putnik, G. D. (2012). Technologies integration for distributed manufacturing scheduling in a virtual enterprise. In *Virtual and Networked Organizations, Emergent Technologies and Tools* (pp. 337-347). Springer, Berlin, Heidelberg.

[10] Varela, M. L. R., Aparício, J. N., & Silva, S. C. (2003). A Web-based application for manufacturing

scheduling. In *IASTED International Conference on Intelligent Systems and Control* (pp. 400-405).

[11] Varela, L. R., & Ribeiro, R. A. (2003). Evaluation of simulated annealing to solve fuzzy optimization problems. Journal of Intelligent & Fuzzy Systems, 14(2), 59-71.

[12] Arrais-Castro, A., Varela, M. L. R., Ribeiro, R. A., & Putnik, G. D. (2015). Spatial-temporal business partnership selection in uncertain environments. FME Transactions, 43(4), 353-361.

[13] Çaliş, B., and Bulkan, S.: A research survey: review of AI solution strategies of job shop scheduling problem, Journal of Intelligent Manufacturing, Vol. 26, No. 5, pp. 961-973, 2015.

[14] Gen, M. et al.: Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art-survey, Computers & Industrial Engineering, Vol. 56, No. 3, pp. 779-808, 2009.

[15] Nouiri, M., Bekrar, A., Jemai, A., Niar, S., and Ammari, A.C.: An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem, Journal of Intelligent Manufacturing, Vol. 29, No. 3, pp. 603-615, 2018.

[16] Nouri, H.E., Belkahla Driss, O., and Ghédira, K.: Simultaneous scheduling of machines and transport robots in flexible job shop environment using hybrid metaheuristics based on clustered holonic multiagent model, Computers & Industrial Engineering, Vol. 102, No., pp. 488-501, 2016.

[17] Sudo, Y., Matsuda, M.: Agent based Manufacturing Simulation for Efficient Assembly Operations, Procedia CIRP, Vol. 7, No., pp. 437-442, 2013.

[18] Ripon, K.S.N., Torresen, J.: Integrated job shop scheduling and layout planning: a hybrid evolutionary method for optimizing multiple objectives, Evolving Systems, Vol. 5, No. 2, pp. 121-132, 2014.

[19] Sahin, C., Demirtas, M., Erol, R., Baykasoğlu, A., and Kaplanoğlu, V.: A multi-agent based approach to dynamic scheduling with flexible processing capabilities, Journal of Intelligent Manufacturing, Vol. 28, No. 8, pp. 1827-1845, 2017.

[20] Alves, F., Varela, M. L. R., Rocha, A. M. A. C., Pereira, A. I., Barbosa, J., Leitão, P: Hybrid System for Simultaneous Job Shop Scheduling and Layout Optimization based on Multi-agents and Genetic Algorithm. *Proc. 18th International Conference on Hybrid Intelligent Systems (HIS 2018)*, Porto, Portugal 2018.

[21] Tompkins, J.A. et al.: Facilities planning (John Wiley & Sons. 2010).

[22] Kennedy, J.: Particle swarm optimization: Encyclopedia of machine learning (Springer), pp. 760-766, 2011.

[23] Poli, R., Kennedy, J., and Blackwell, T.: Particle swarm optimization, Swarm intelligence, Vol. 1, No. 1, pp. 33-57, 2007.

[24] Simões, G., and Ebecken, N.F.: Algoritmo genético e enxame de partículas para a otimização de suportes laterais de fornos, Revista Internacional de Métodos Numéricos para Calculo y Diseño en Ingeniería, Vol. 32, No. 1, pp. 7-12, 2016.

[25] Bratton, D., and Kennedy, J.: Defining a standard for particle swarm optimization, in: (Eds.) Book Defining a standard for particle swarm optimization (IEEE, 2007, edn.), pp. 120-127.

[26] Alves, F. et al.: Optimal Schedule of Home Care Visits for a Health Care Center, in: (Eds.): Book Optimal Schedule of Home Care Visits for a Health Care Center (Springer International Publishing, 2017, edn.), pp. 135-147.

[27] Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence (MIT press. 1992).

[28] Ghaheri, A., Shoar, S., Naderan, M., Hoseini, S.S.: The applications of genetic algorithms in medicine, Oman medical journal, Vol. 30, No. 6, pp. 406, 2015.

[29] Kumar, M., Husian, M., Upreti, N., and Gupta, D.: Genetic algorithm: Review and application, International Journal of Information Technology and Knowledge Management, Vol. 2, No. 2, pp. 451-454, 2010.

[30] Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, IEEE Trans. Comput., Vol. 29, No. 12, pp. 1104-1113, 1980.

[31] Wilensky, U.: NetLogo: Center for connected learning and computer-based modeling, Northwestern University. Evanston, IL, 1999, in: Editor (Ed.) ^(Eds.): Book NetLogo: Center for connected learning and computer-based modeling, Northwestern University. Evanston, IL, 1999 (2014, edn.), pp.

[32] Chiacchio, F., Pennisi, M., Russo, G., Motta, S., Pappalardo, F.: Agent-based modeling of the immune system: NetLogo, a promising framework, BioMed research international, Vol. 2014, No. 2014.

[33] Biggs, M.B., Papin, J.A.: Novel multiscale modeling tool applied to Pseudomonas aeruginosa biofilm formation, PLoS One, Vol. 8, No. 10, pp. e78011, 2013.

[34] Abdollahzadeh, H., Varela, L., Atashgar, K., & Putnik, G. D. (2015). Condition based maintenance optimization for multi-state wind power generation systems under periodic inspection. FME Transactions, 43(4), 319-327.

ХУМАНО-ЦЕНТРИЧНИ ХИБРИДНИ МУЛТИ-АГЕНТ-СИСТЕМ И СИСТЕМ БАЗИРАН НА МЕТА ХЕУРИСТИКАМА ЗА СИМУЛТАНО ПОДЕШАВАЊЕ ПРОГРАМИРАЊА ПРОИЗВОДЊЕ И РАСПОРЕДА МАШИНА У ПОГОНУ

Ф. Алвеш, М.Л.Р. Варела, А.М.А. К. Роша, А.И. Переира, П.А. Леитао

Производне активности и контрола производње стално расту. Упркос томе, неопходно је побољшати

све већу разноликост подешавања програмирања производње и распореда машина у погону, за динамичке и флексибилне одговоре у нестабилним окружењима са поремећајима или кваровима. Суочен са недостатком реалних и практичних сценарија производње, предложен приступ омогућава симулацију и решавање проблема појединачне производње у производном систему користећи предности генетских оптомозационих алгоритама и оптимизационих алгоритама на бази ројева, у комбинацији са флексибилношћу и робусношћу мулти-агент система. и алтернатива динамичког репрограмирања. Због тога, овај хибридни систем за подршку одлучивању намерава да се добију оптимизирана решења и да омогући људима да комуницирају са системом како би правилно прилагодили приоритете или побољшали подешавања или решења, на интерактиван и корисничко-пријатљски начин. Систем омогућава процену оптимизационих перформанси сваког од предложених алгоритама као и за децентрализацију одзива и динамичких одлука за репрограмирање услед појава неочекиваних догађаја.