Journal of Machine Engineering, 2023, Vol. 23, No. 3, 69–85
ISSN 1895-7595 (Print) ISSN 2391-8071 (Online)

Received: *27 July 2023* / Accepted: *31 August 2023* / Published online: *06 September 2023*

scheduling, robotic cell, algorithm, makespan optimization

Eddi MILLER[1*], Bastian ENGELMANN[1],
Tobias KAUPP[1], Jan SCHMITT[1]

ADVANCED CASCADED SCHEDULING FOR HIGHLY AUTONOMOUS PRODUCTION CELLS WITH MATERIAL FLOW AND TOOL LIFETIME CONSIDERATION USING AGVS

In today's manufacturing systems, especially in Industry 4.0, highly autonomous production cells play an important role. To reach this goal of autonomy, different technologies like industrial robots, machine tools, and automated guided vehicles (AGV) are deployed simultaneously which creates numerous challenges on various automation levels. One of those challenges regards the scheduling of all applied resources and their corresponding tasks. Combining data from a real production environment and Constraint Programming (CP-SAT), we provide a cascaded scheduling approach that plans production orders for machine tools to minimize makespan and tool changeover time while enabling the corresponding robot for robot-collaborated processes. Simultaneously, AGVs provide all production cells with the necessary material and tools. Hereby, magazine capacity for raw material as well as finished parts and tool service life are taken into account.

1. INTRODUCTION

The continuous and immediate development in science and industry drives companies and their manufacturing systems to be more autonomous to improve their efficiency, quality, flexibility, and reduce costs. Therefore, they are relying on different technologies such as robots for automated material handling and manufacturing tasks with light force, as well as AGVs for surrounding logistic processes, e.g., for materials or tools. Machine tools working in parallel with robots are called robotic cells [1] and are a core element of today's smart manufacturing systems [2]. There are multiple setups in which machine tools and robots can operate, i.e., one robot handles material for multiple machine tools [3] or each machine tool works in parallel with one robot. Additionally, each robot in a production cell is used for material handling, like loading or unloading the machine which induces long idle times between the two tasks. Complementary, AGVs are used to supply robotic cells with raw materials and tools, while also carting off finished parts. To efficiently manage and utilize these different technologies, over the last two decades, researchers developed varying

[1] Institute of Digital Engineering, Technical University of Applied Sciences Würzburg-Schweinfurt, Germany
[*] E-mail: eddi.miller@thws.de
https//doi.org/10.36897/jme/171749

scheduling approaches. This optimization problem is broadly known as the Job Shop Problem (JSP) and occurs when several jobs and tasks need to be planned on multiple machines or resources [4]. Generally speaking, companies that try to achieve a higher degree of autonomy, will find themselves in a situation, where comprehensive scheduling algorithms are needed to efficiently deploy cost-intensive machine tools, robots, or AGVs, see also [5]. All these technologies together with scheduling algorithms are fundamental to realize a flexible manufacturing system [6].

During the last 15 years, many attempts and approaches have been made to solve the JSP using different algorithms and metaheuristics, see Table 1.

Table 1. Job shop scheduling approaches over the last 15 years

Authors	Algorithm	Target variable	Use of AGV	Tool life time	RCP
Vallada and Ruiz (2011)	Genetic algorithm	Minimize makespan	✗	✗	✗
Abdulkader et al. (2013)	Genetic algorithm	Minimizing cycle time	✗	✗	✗
Lacomme, Larabi, and Tchernev (2013)	Memetic algorithm	Minimize makespan	✓	✗	✗
Zarandi, Mosadegh and Fattahi (2013)	Gilmore and Gomory algorithm	Minimizing cycle time	✗	✗	✗
Gundogdu and Gultekin (2016)	Itertive algorithm	Throughput maximization	✗	✗	✗
Nouri, Driss, and Ghedira (2016)	Clustered holonic multiagent model	Minimize makespan	✗	✗	✗
Zabihzadeh and Rezaeian (2016)	Mixed integer linear programming	Optimal number of robots	✗	✗	✗
Zhou and Li (2017)	Tabu circulatory time point	Minimizing cycle time	✗	✗	✓
Yan et al. (2018)	Mixed integer programming	Optimal solution for job orders	✗	✗	✗
Ghadiri Nejad et al. (2018)	Universal scheduling model	Minimizing cycle time	✗	✗	✗
Li et al. (2020)	Mixed integer linear programming	Minimize makespan	✗	✗	✗
Reddy et al. (2021)	Mixed integer linear programming	Minimize makespan	✓	✓	✗

Vallada and Ruiz [7] proposed a genetic algorithm that uses fast local search and enhanced local search crossover operators to optimize the makespan of an unrelated parallel machine scheduling problem. Here, the total amount of jobs have to be scheduled on a given amount of machines. In the end, the authors compared their proposed algorithm to existing methods, with their genetic algorithm showing the best results.

An alternative genetic algorithm for scheduling and sequencing robotic cells was presented by Abdulkader et al. [8]. The setup for their scheduling problem was a four-machine blocking robotic cell where the robot supplies parts to all machines. The author’s goal was to optimize the job sequencing while minimizing the robot cycle time. The proposed algorithm was tested against a full enumeration solution, and it was quickly evident that the genetic algorithm outperforms the enumeration solution when it comes to higher number of production jobs. Building on genetic algorithms, Lacomme et al. [9] relied on memetic algorithms to solve their scheduling problem. Memetic algorithms are an extension of the previously mentioned genetic algorithms [10] and are in this case comprised of a powerful local search procedure [9]. The purpose of the algorithm was to plan multiple machines and AGVs to minimize the makespan and find the best sequences for jobs and transport operations.

Deviating from genetic algorithms into a two-machine robotic cell scheduling problem, Zarandi et al. [11] applied the Gilmore and Gomory algorithm to solve this problem. In this setup, one robot supplies two machines with material while getting material from the input station and storing parts in the output station. The goal of the algorithm is to minimize the cycle time by determining the number of robot moves and the sequence of parts. The same arrangement of two machines and one robot was investigated by Gundogdu and Gultekin [12]. The robot handles material for both machines, again using two buffers for input and output material. Besides these buffers, there is an additional buffer that moves with the robot and can also hold a set number of parts. All parts produced on the machines are identical. The authors differentiated between different capacity buffers to determine the cyclic schedule in which the robot can move to maximize the throughput time. Depending on the buffer capacity, the proposed iterative algorithm could find solutions, more so if the capacity was larger.

It is also possible to combine algorithms, as seen in [13]. The scheduling approach is a hybrid matheuristic between a neighborhood-based genetic algorithm and a set of cluster agents that uses a tabu search technique. The hybrid algorithm is used to plan a set amount of jobs on multiple machines. Said jobs are hereby conveyed between machines by transport robots, similar to AGVs. The authors compared their algorithm against other genetic algorithms and tabu search procedures. The results show that the proposed algorithm outperforms existing solutions while creating new optimal solutions.

A comparable scheduling setup was explored by Zabihzadeh and Rezaeian [14]. Robots are used to transport and load parts to a set amount of machines. The goal is to find the optimal sequence of processing parts and robot movements. This enables minimizing the makespan and finding the optimal number of robots needed. The chosen algorithm is a mixed integer linear programming model which is widely used in the context of the JSP [3]. The authors strengthen the mixed integer programming model by adding constraints that take advantage of specific relations regarding their scheduling problem. They investigate dynamic job-shop scheduling of robotic cells where jobs enter with unanticipated arriving rates. With a robot handling transportation, the goal is to find the optimal solution for job orders.

A similar approach was taken by Ghadiri et al. [15] and Li et al. [16], where a flexible robotic cell with multiple machines and one robot was studied. Ghadiri et al. [15] try to determine the order of activities the robot should partake in. Establishing a universal scheduling model to minimize the cycle time, the results show that the modified version is ultimately more efficient than the model of literature. Alternatively, Li et al. [16] go one step further by simultaneously calculating the assignment on each machine, the transport order as well as the operations assignment for the robot. The computational results of the teaching learning based optimization algorithm could effectively minimize the makespan.

In robotic cells, where one robot supplies one or multiple machines with material or parts, there is always the risk of idle time for the robot, especially in one-to-one setups. Therefore, it would be possible to incorporate the robot to perform collaborative tasks in addition to transportation. Zhou and Li [17] explored the possibility of so-called robot-collaborated processes (RCP) within a robotic cell, where one robot supplied five machines with material. To reduce downtime between material handling jobs, the robot performs measurements or quality control tasks. As in previous works, the goal is to optimize the sequences of the robot and the manufacturing cycle time. This is achieved through a tabu circulatory time point searching algorithm. Depending on the production setup, if a robot has to supply more than one machine, this can lead to a bottleneck where the robot is too slow, and the total makespan increases. In a one-to-one production cell, this is definitely not the case, on the contrary, here lies the opportunity to reduce the makespan by incorporating the robot.

In a recent study by Reddy et al. [18], an approach to create a scheduling method that combines machines, AGVs, and tool transporter was demonstrated. To further enhance flexible manufacturing systems, the authors looked into a circulatory setup of four CNC machines. A total of two AGVs and one tool transporter supply all machines with tools, with all tools being stored in a central tool magazine. With their scheduling approach, they came to the conclusion that tool switching and tool waiting time pose a considerable influence on the makespan. Reddy et al. [18] also conclude that it would be unfeasible to leave out tool switching and job/part transfer times. Based on this, the existing research by the Authors [19] will be extended with the following additions to create a cascaded scheduling approach for robotic cells, to schedule production jobs on multiple machines with AGV supply:

1. Transport orders for raw material and finished parts.
2. Transport orders for tool switching.
3. Warning for tool lifetime transgression.

These additions further enhance the provided scheduling approach which leads to a closer representation of a real production environment. Furthermore, it provides a novel method to make robotic cells more autonomous by integrating AGVs for material and tool allocation tasks. Additionally, idle time of the material management robots is reduced by incorporating robot-collaborated processes to reduce makespan. This results in a cascaded approach which schedules production orders, contemplating necessary material stocks and tool changes and tool lifetime while also outsourcing tasks from machine to its corresponding robot.

The following work is structured as follows: Section 2 emphasizes the importance of material flow, tool changeover and tool lifetime consideration. In Section 3 we provide insight into problems that occur when jobs or task are being scheduled. Building on this, the derived constraints and the scheduling algorithm are presented in Section 4.1. At last, Section 5 displays the results of the proposed scheduling approach.

2. ADDITION OF MATERIAL FLOW AND TOOL LIFETIME

When looking at past literature, Table 1 shows that the combination of robotic cells, AGVs, and robot-collaborated processes has not been explored much. Most authors focus on the material handling part between machines and robots. The external processes on how to get material or parts to the workstations in the context of JSP are as important. As mentioned by Ferenczi et al. [20], the combination of scheduling and material handling is one of the biggest challenges that a production system can face. Furthermore, the authors add that the transportation of materials from one production element to another is elemental for modernized manufacturing systems. Optimizing the flow of materials helps reduce production time, minimize work-in-progress inventory, and avoid bottlenecks that can slow down production and cause idle time. By planning and monitoring the material flow, it is possible to ensure the efficient use of materials. To realize this, AGVs have found an extensive utilization in flexible and autonomous production systems [21].

Furthermore, the provision of tools for machining tools is also a vital part of flexible manufacturing systems, as these can also be supplied with the help of AGVs. To enable the possibility of efficient tool changeover, there must be a tool lifetime consideration. As tools degrade when cutting metal or other rigid materials, eventually those tools must be swapped to ensure the high quality of parts and avoid machine downtime. Especially for milling processes, which are a big part of the underlying manufacturing tasks, surface finishes are key factors, which can only be obtained with high-performing tools. Worn-out tools also produce poor-quality products which can lead to customer complaints or product recalls. Moreover, in the field of metal cutting, where manufactured parts are big in size and need longer hours of manufacturing, parts cannot be discarded. This would not only result in wasted materials but also wasted time, money, and energy. If the part can still be saved, it would only be possible through touch-ups or reworks which also cost time and money.

3. PROBLEM DEFINITION

The following section provides insight into problems that arise using the previously mentioned technologies and how such a manufacturing setup can look like. This paper focuses on four main problems that emerge in robotic cells and how they can be overcome using scheduling algorithms. Therefore, the chosen scheduling approach will be explained.

As mentioned in Section 1, there are multiple ways in which machine tools and robots can collaborate. In this contribution, the underlying setup is a machine park where each machine tool is assigned with one robot, as seen in Fig1. For this use case, there are a total of three machine tools, three robots and two AGVs, one for materials and one for tools. Each robotic cell has a shelf for raw material and finished parts which are supplied or transported away by the material AGV. On the other hand, all machine tools have their own tool magazine, where the tool AGV can deliver all needed tools for planned jobs.

When looking at this kind of manufacturing setup, there are four main problems that have to be addressed:

Problem 1: The first problem regards tool availability. Some tools are only available once and have to be distributed between machines, depending on which production order is assigned to what machine. Tools have to be scheduled efficiently to reduce changeover time and impede idle time.

Problem 2: Problem number two addresses tool lifetime and coincides with problem one. Tools are only capable of processing a limited amount of parts, before the tools have to be replaced. Therefore, a warning or trigger before a tool reaches its final lifetime is needed.

Problem 3: Moving on from tools to materials, problem three is the supply of raw material and work pieces, as well as carting away finished parts. Again, to impede idle time of all machines, there has to be sufficient raw material and enough space for finished parts to not interrupt the manufacturing process of the machine tool.

Problem 4: The last problem is related to the previously mentioned RCP. To avoid idle time of the robot, after handling materials, they have to take over tasks of the machine tool. To enable RCP, tasks between robot and machine tool have to be scheduled as well.

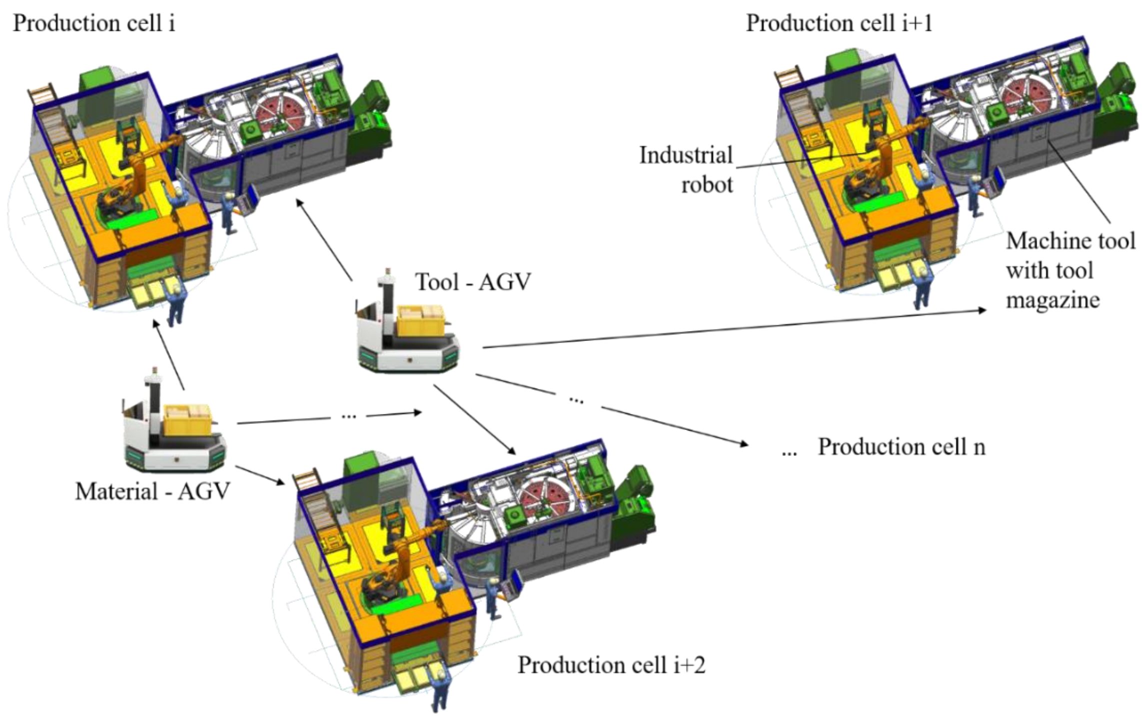

Fig. 1. Manufacturing setup

Addressing distribution, production and transportation planning simultaneously enhances the efficiency of all these processes and operations significantly. It results in a very complex system with a multitude of activities within the production planning and shop floor management. These vast systems are subject to a large quantity of constraints, restrictions, and problems which makes it easy to overengineer or over constrain them. Therefore, it is important to concentrate on problems that are solvable and are important for the actual production environment which were selected in this case study. The same goes for the selection of optimization algorithms. As mentioned in Section 1, there are several optimization algorithms, like genetic algorithms or mixed integer programming models. Considering the existing problems of job/task allocation and constraints like tool availability or manufacturing task order, an established method stood out. A combination of constraint programming (CP) and boolean satisfiability (SAT), combined CP-SAT, was chosen for the presented scheduling problem. Constraint programming has already been proven to work for different scheduling problems [22–25]. Here, CP finds a variety of applications and is mostly used in combination with other techniques, such as mixed integer linear programming to solve variations of the JSP. In order to solve combinatorial optimization problems, constraint programming enables to set different constraints that help narrow down the possible solution [26].

The issue when looking at problems 1–4 is to plan all resources accordingly, to not create deadlock situations. A deadlock creates a scenario where two or more parts are in circular wait, which means parts are held back by other processes in the chain [27]. Coffman et al. [28] presented four conditions that need to arise simultaneously for a deadlock to arise:

1. Mutual exclusion: processes claim exclusive control over the resources they require.
2. Hold and wait: allocation of additional resources to already holding processes.
3. No pre-emption: resources can only be removed by using them for completion.
4. Circular wait: each process in the chain requires one or more resources that are requested by the following process.

In order to avoid deadlocks, it is sufficient to ensure that at least one of the four conditions is not met. As elaborated by Banaszak and Krogh [29], in manufacturing systems conditions 1–3 are always met and deadlocks are only created by circular wait and therefore needs to be avoided. All AGV transport order are calculated with enough buffer to ensure that the circular wait condition does not arise.

4. CONSTRAINT PROGRAMMING AND SCHEDULING ALGORITHM

The following section dives deeper into the above-mentioned restrictions and constraints and how they help to delimit the manufacturing setup and its scheduling problem. These constraints will be split into two sections. After the constraints have been set, the next paragraph will describe the algorithm, especially on how to minimize the target variables' makespan and tool changeover time.

4.1. CONSTRAINTS

The first set of constraints was developed for production order planning on multiple machines. Here, a set number of production jobs will be planned onto all available machines to minimize makespan and tool changeover time. The second section displays constraints that are necessary to plan the tasks between every robot and machine combination, to again, minimize the makespan. All constraints are derived from a real engineering and production environment and facilitate the importance of this topic.

The following assumptions have been made for all production cells to schedule production jobs respecting material and tool availability:

(1) All machines M_i $(i = 1, 2, ..., M)$ can manufacture all parts J_n $(n = 1, 2, ..., N)$,
(2) Every production order can be assigned to any machine,
(3) Each machine can manufacture only one part at a time,
(4) All production intervals on one machine cannot overlap,

(5) All parts in a lot are identical and are processed consecutively,
(6) Specific tools T_i *(i = 1, 2, ..., T)* are only available once,
(7) These specific tools have to be shared between all machines,
(8) All tools are supplied via tool AGV,
(9) Tool lifetime has to be considered,
(10) Raw material RM_i *(i = 1, 2, ..., RM)* is supplied by material AGV,
(11) Final goods FG_i *(i = 1, 2, ..., FG)* are carted away by material AGV.

With *M* consisting of all machines and *N* being the total amount of parts to be produced.

The variable *T* contains all tools that are needed for the selected jobs. Lastly, the variables *RM* and *FG* describe raw material and final goods respectively. The provided data contains all necessary information beginning from process ID, over tasks for each job and their duration, to every single tool, their cutting time, and lifetime. With the number of machines and the data for all jobs, the algorithm schedules all jobs while minimizing the makespan. After scheduling all jobs, the next step is to plan all tasks inside each job. To allow for robot-collaborated processes, the algorithm decides if there are tasks that the robot can undertake, to reduce cycle time. New assumptions had to be made to enable RCP:

(1) The first and last task P_i *(i = 1, 2, ..., P)* are loading and unloading the machine,
(2) The robot R_i *(i = 1, 2, ..., R)* can take over processes from the machine (RCP),
(3) The duration of RCP cannot be longer than the main process of the machine M,
(4) Technological sequences must be followed,

with *P* consisting of all tasks inside a production order and R containing the total number or robots.

Considering all constraints, the developed method optimizes for two objectives: makespan and tool change over time (see equations 1 and 2).

$$min_{Makespan} = \sum_{i=1}^{n} J_i \tag{1}$$

with J_i being the processing times of each production order, *n* the total amount of production orders. This processing time can be shortened by allocation tasks from machine to robot. Similarly, equation 2 describes the tool change objective where TC_i describes one tool changeover time and m represents the total amount of tool changeover that need to be carried out.

$$min_{Tool\_changeover} = \sum_{i=1}^{m} TC_i \tag{2}$$

Both objective functions are always calculated and considered simultaneously to achieve the shortest makespan with the least tool changeover possible. With all mentioned constraints taken into consideration, an algorithm based on CP-SAT was developed and will be explained further.

4.2. ALGORITHM PRINCIPLE

Based on CP-SAT and the derived constraints, an algorithm was created that solves all problems mentioned in Section 3, by considering all relations and restrictions mentioned in Section 4.1.

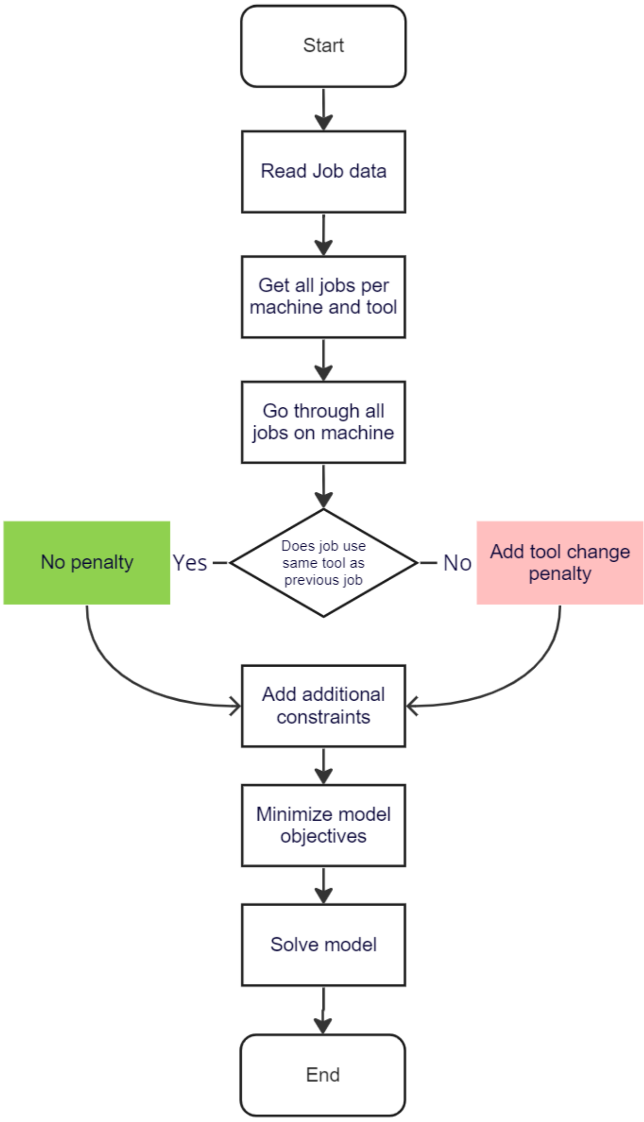

Fig. 2. Flow Diagram of the job allocation algorithm and tool change penalty

Depending on the complexity of the constraints, it is possible to use existing constraints like "AddNoOverlapp" [30] which coincides with the fourth constraint of job scheduling, that the job intervals cannot intersect. Constraint programming in general has the advantage that it already provides a lot of constraints that can just be applied to the problems. This makes it easier to adjust to new problems and new scenarios. Nonetheless, these instruments are not capable of solving all constraints. More complex problems like the scheduling depending on tools and their availability have to be programmed themselves. For this case, a tool change penalty was introduced. The algorithm first lays out every job on all machines available, scans for all possible job sequences on each machine, and then compares the tools of all job combinations. If two jobs have different tools, the algorithm impedes a tool change penalty and writes it down into the tool change variable. Accordingly, the algorithm learns which job sequences provide the shortest manufacturing time. The flow chart (Fig. 2) provides insight into the structure of the main part of the algorithm which allocates jobs to machines by considering tool change penalties.

The second target variable, besides the tool change penalty, is the makespan. While solving the optimization problem, the algorithm tries to minimize both variables while considering all constraints. In parallel, the algorithm monitors raw materials, finished goods, and tools available on each machine. Each machine is equipped with a separate magazine for raw materials and finished parts. The program oversees the material flow and creates transport orders for the material AGV depending on if the raw material magazine is running low or finished goods need to be carted away. Similarly, the tool AGV transports tools from one machine to another. In this case, tool lifetime is also considered and gives a warning for said tools that the lifetime is ending, and it needs to be changed soon.

To follow up on the cascaded scheduling model, the algorithm has found a plan for all production jobs and is now able to schedule tasks inside of each job. First, the script looks for all tasks inside a job that can be done by the robot. After clarifying which tasks can be transferred to the robot, the program looks for the allocation of tasks with the smallest makespan and puts them into an efficient order while considering technological sequences. The scheduling results for all applications will be displayed in the following chapter.

5. APPLICATION ON REAL LIFE DATA

This section addresses the results of the provided scheduling approach using data from a real manufacturing setup that uses numerous robotic cells as mentioned in Section 3. The following Fig. 3 summarizes all data relations and all schedule interrelations. First, the scheduling results for the job allocation will be displayed to decide which job will be planned on which machine and when regarding the time.

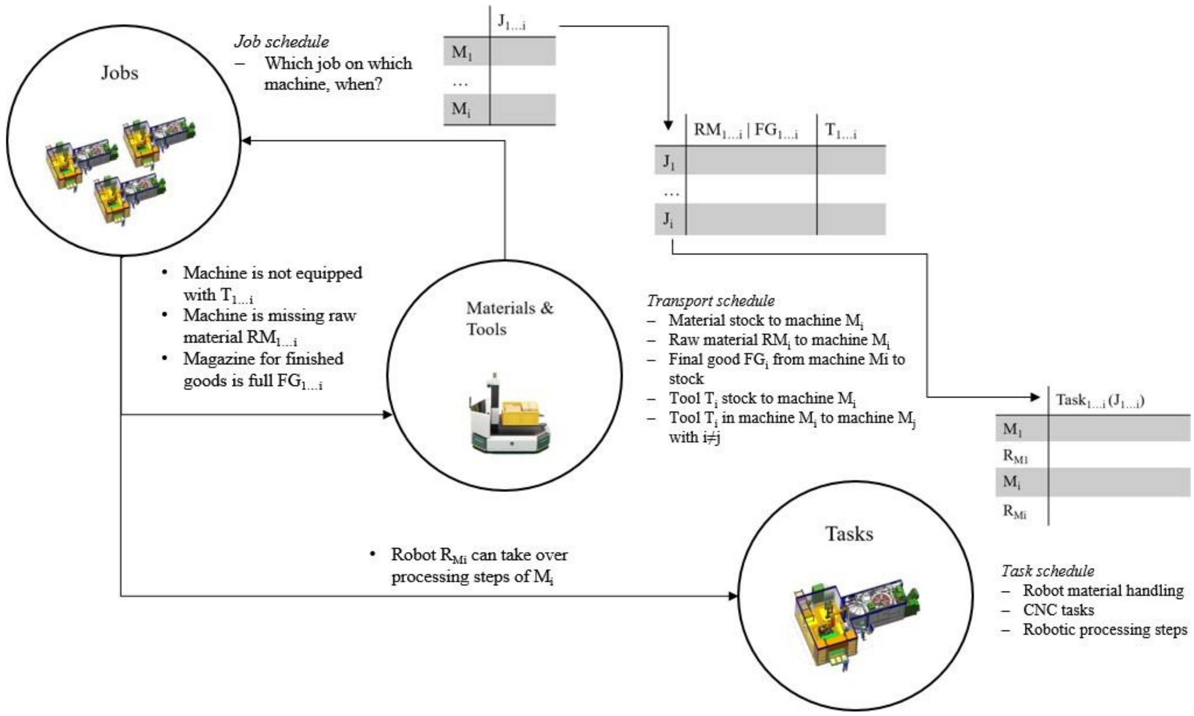

Fig. 3. Cascaded scheduling approach and data relations

Afterwards, data on missing tools, raw material and finished goods is gathered to create transport orders to fill the transport schedule. These transport sequences include orders for raw material to specific machines or tool transports from one machine to another. Simultaneously, tool lifetime is monitored to devise warnings for impending tool failure. At last, the task scheduling between single robot and machine, which incorporate robot-collaborated processes, will be shown. Tasks hereby include material handling, CNC tasks and robotic processing steps.

5.1. JOB SCHEDULING ON MULTIPLE MACHINES

The data basis for this scheduling method are production orders and their corresponding data. This includes the duration all tasks needed to finish a work piece, all tools needed with their specific cutting and lifetime and the total amount of parts to be manufactured in this lot. Furthermore, the magazine capacities for raw material and finished goods are considered. The following example, Fig. 4, shows 12 jobs scheduled on three machines and addresses Problem 1 mentioned in Section 3.

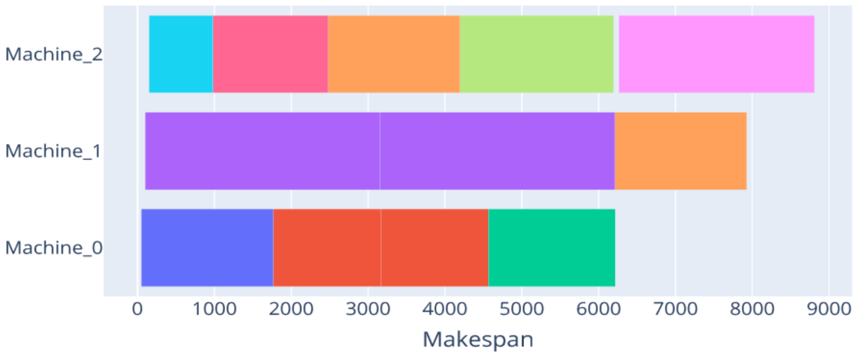

Fig. 4. Job scheduling on three machines indicating different tool sets with unique colours

Each block on every machine (y-axis) shows one production order which includes up to 13 parts. Production orders can vary widely in manufacturing time (x-axis), depending on the work piece and their complementary numerical control (NC) program. Identical colours of jobs indicate that these jobs also use completely identical toolsets for their corresponding tasks, which usually indicates that the same part will be manufactured for different production order. Therefore, the algorithm tries to schedule these jobs onto the same machine to reduce tool changeover between machines. This allocation of jobs is heavily influenced by the number of jobs and their duration. Nonetheless, it is still possible for some individual tools to be shared between jobs which is not indicated here but will be considered for tool transport orders. The number of jobs and machines can be adjusted, but has a direct impact on computation time, since the number of possible solution changes. Depending on the amount of production orders to be scheduled and especially their length, heavily influences the outcome. The shortest production duration on all machines shown in Fig. 4 is nearly 7000 minutes which results in approximately 117 hours. This is due to the fact, that the production data consists of orders with semi big lot sizes but long manufacturing processes.

Despite that, it is still possible to increase the number of jobs, as seen in Fig. 5, with a total number of 17 jobs which increases the total makespan on each machine to over 9000 minutes, approximately 150 hours. The method stays the same, but with different jobs and especially more production orders, the algorithm takes longer to calculate, but still finds one or more possible solutions to minimize makespan and tool changeover time. The colours in this plot were changed to a single colour for each machine from colour matching different tool sets to simplify the following addition of AGV transport orders. One more thing to note is the slight delay on every machine before the first job, also seen in Fig. 4. This is due to raw material supply from the material AGV. This will be further explained in the next paragraph explaining transport orders.

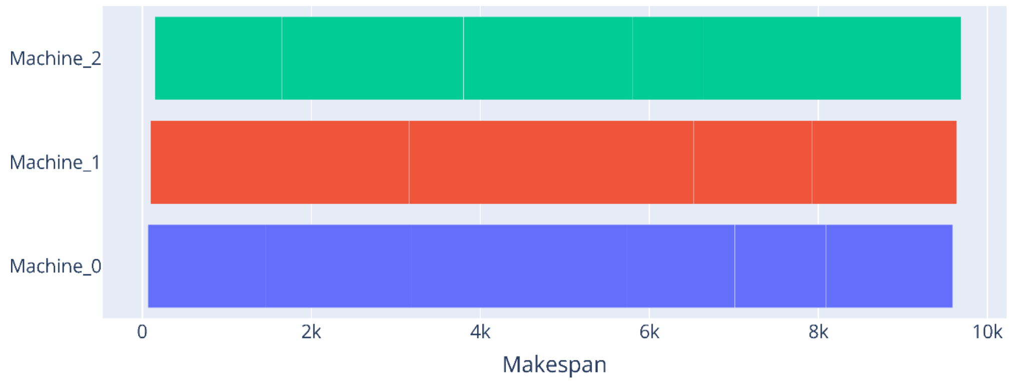

Fig. 5. 17 jobs scheduled on three machines

5.2. AGV TRANSPORT ORDERS AND TOOL LIFETIME

In this section, the schedule further expands with the addition of transport orders and tool lifetime to address Problems 2 and 3. Material availability is calculated via magazine capacity and order quantity. The algorithm tracks both values and generates transport orders before the machine reaches an impasse and must stop production. The calculation is therefore based on a predictive logic where the material AGV pre-emptively supplies each machine to not run out of material while still considering the other transport orders. The same applies for finished work pieces. Finished goods are not as critical as raw material since the machine can still manufacture. Nonetheless, the algorithm avoids overfilling the magazine and creates transport orders for the AGV to cart away finished work pieces. As mentioned in the previous section, before each machine can start manufacturing, the material AGV provides all machines with raw goods consecutively which results in a small delay before the first job on every machine. All transport orders can be seen in Fig. 6, displayed in the row of “Material AGV”. Each small column in this section correlates to one transport order for the AGV. This can either mean raw material is being brought to the machine, or finished goods are carted away from the robotic cell. The colours of the material AGV correspond to the colour of the machine that is supplied in that moment. Since there is only one AGV for material handling, there can’t be any transport orders that overlap. The time for every transport order is heavily buffered to enable to AGV to fulfil all the necessary tasks in time.

For the tool transport orders, the windows of transport are split into two colours. Both colours match machines and indicate where the tool AGV takes tools out of a machine (left colour) and then moves them to the target machine (right colour). Once all tools necessary for the current production job are freed, they are then moved to the next machine. The number of shipped tools depends on the job and on the capacity of the tool AGV. Based on the machine tools used in this manufacturing environment, it is possible to change tools in the tool magazine even during production, which makes tool interchange even more flexible.

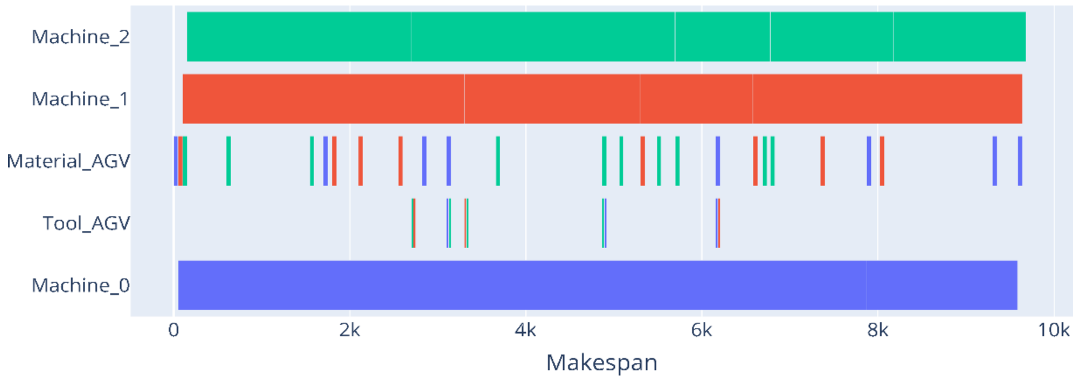

Fig. 6. Job scheduling with the addition of transport orders for materials and tools

The last addition of tool lifetime warnings can be seen in Fig. 7. The plot corresponds to the jobs that are planned and displayed in Fig. 4. The x-axis represents the manufacturing time or rather the makespan of the production orders. The y-axis on the other hand shows the three machines used in this kind of setup. Each data point in Fig. 7 belongs to a job and shows that during production one or more tools for this process will reach the end of their lifetime. Based on this data set, it means that there must be changeover of tools for every job, to ensure steady quality of parts.

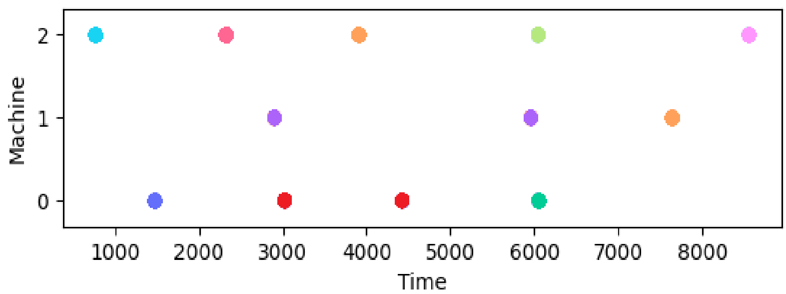

Fig. 7. Tool lifetime consideration and changeover time

5.3. TASK SCHEDULING BETWEEN ROBOT AND MACHINE

The last step of the proposed cascaded scheduling method is the task allocation between robot and machine of every robotic cell. Specific manufacturing steps, that would usually be done by the machine, can now be handled by the robot. These tasks could include processing steps like deburring, washing, or measuring. The idea is not to just simply transfer all possible tasks from machine to robot but consider which allocation of tasks results in the shortest makespan. The proposed algorithm looks for possible solutions and decides which tasks fit what resource. Generally speaking, the robot takes over tasks of the machine for the previously manufactured part, while the machine produces the current part, to finish the last steps of the preceding part. The duration of the RCP has to be shorter than the main processing time of the machine, in order to avoid idle time of the machine tool.

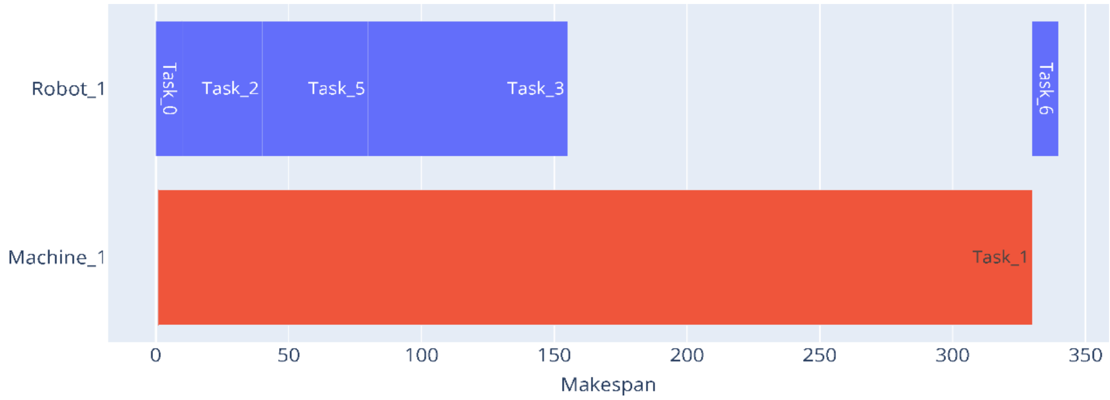

Fig. 8. Provided schedule of task allocation between robot and machine tool (RCP)

Figure 8 shows the created schedule for a production order on the first machine. The red bar for Machine 1 contains all manufacturing tasks that could or should not be transferred to the robot and results in the main processing portion of the work piece. During this step, the robot avoids idle time by executing robot-collaborated processes. Additionally, material handling tasks are still present in the form of loading (Task 0) and unloading of the finished work piece (Task 6). This shows that the robot is capable of taking over tasks while considering the constraints set in Section 4.1 like taking into account the manufacturing time and RCP time to avoid idle time of both robot and machine.

Summarizing, the cascaded approach combines the scheduling for three main aspects: Production orders on the planning level, AGV transport orders for materials and tools on the shop floor and lastly task allocation between robot and machine. The algorithm provides a basis that can help make manufacturing environments, especially robotic cells in combination with AGVs, fully autonomous. With the possibility to add more constraint and to challenge new problems it has the flexibility to address new scenarios and create new outcomes. By including real production variables like inventory or tool life time it can help improve robotic cells by improving work piece quality and saving costs.

6. CONCLUSION AND FURTHER RESEARCH

In this paper we provide a cascaded scheduling approach that incorporates technologies like machine tools, robots and AGVs, where robots and AGVs act as supplier for material and tools. Nevertheless, robots are not only used for material handling but also for so called robot-collaborated processes, which are also present in the scheduling process. In contrast to previous works on the topic of dynamic scheduling for robotic cells which rely on genetic algorithms, mixed integer linear programming, or tabu circulatory programming, the proposed scheduling method uses a combination of constraint programming and boolean satisfiability (CP-SAT). With the addition of material and tool flow, as well as the consideration of tool lifetime, different problems in a robotic cell arise. The unavailability or uniqueness of tools requires the relocation of said tools between machines, while still regarding their service life to provide an interference-free manufacturing process. Besides tool management, to ensure a stable process, there needs to be a constant flow of materials, which can also be enabled using AGVs. To tackle these problems, different constraints derived from an actual manufacturing environment have been set. These constraints are split into job scheduling on multiple machines and task allocation between single robot and machine. The proposed algorithm has two distinct target variables: Minimization of makespan and tool changeover time. The latter is solved be creating a tool change penalty. The algorithm impedes penalty for every tool changeover that happens between two jobs and saves the penalty into a variable. Ultimately, the proposed method creates a schedule for a set amount of production orders and minimizes makespan while considering tool status. Furthermore, it monitors material magazines for raw material and finished goods to create transport orders for the material AGV to supply or cart away parts. At last, each robot is enabled for robot-collaborated processes, where another makespan optimization takes place. This is achieved by relocating tasks from machine tool to robot. All of this is presented using real-life production data which speaks for this method. Values, constraints, and problems are directly derived from experts in the field of robotic cells and manufacturing systems. The proposed method is tailored to the needs of an autonomous manufacturing environment. Furthermore, more constraints or even resources like machines, robots or AGVs can be added at any time to adjust for a changing layout. Nonetheless, there are also negative points like computation time and no real time monitoring. The computation time rises fast with higher amounts of production order because of more potential tool changeovers. This is not a big of a problem in this kind of manufacturing setup, because of long manufacturing durations per job, but could be different for shorter manufacturing times with more orders. Real life monitoring as of now is not possible since the scheduling approach only creates a plan. It cannot adjust for any unexpected situations that would change the workflow.

Further research will include the transition to a real-life setup that demonstrates the manufacturing setup where the proposed algorithm can be tested and validated. The setup includes multiple robots and an AGV to indicate material and tool flow. This proposal requires further understanding of machine controls and how to apply the scheduling algorithm to the different resources to supply machines and robots with their respective codes. Another addition should be consideration of real time monitoring, where the algorithm recalculates the schedule depending on external factors.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the funding support of the ABBA project by the state of Bavaria (Bayerisches Staatsministeriums fur Wirtschaft, Landesentwicklung und Energie, Grant no. DIK0248/03).

REFERENCES

[1] YI J., DING S., SONG D., ZHANG M.T., 2008, *Steady-State Throughput and Scheduling Analysis of Multicluster Tools: A Decomposition Approach*, IEEE Transactions on Automation Science and Engineering, 5/2, 321–336.

[2] WU X., YUAN Q., WANG L., 2020, *Multiobjective Differential Evolution Algorithm for Solving Robotic Cell Scheduling Problem with Batch-Processing Machines*, IEEE Transactions on Automation Science and Engineering, 18/2, 757–775.

[3] YAN P., LIU S.Q., SUN T., MA K., 2018, *A Dynamic Scheduling Approach for Optimizing the Material Handling Operations in a Robotic Cell*, Computers & Operations Research, 99, 166–177.

[4] SONMEZ A., BAYKASOGLU A., 1998, *A new dynamic programming formulation of (nxm) flowshop sequencing problems with due dates*, International Journal of Production Research, 36/8, 2269–2283.

[5] MICHAEL L.P., 2018, *Scheduling: Theory, Algorithms, and Systems*, Springer.

[6] BLAZEWICZ J., EISELT H.A., FINKE G., LAPORTE G., WEGLARZ J., 1991, *Scheduling Tasks and Vehicles in a Flexible Manufacturing System*, International Journal of Flexible Manufacturing Systems, 4/1, 5–16.

[7] VALLADA E., RUIZ R., 2011, *A Genetic Algorithm for the Unrelated Parallel Machine Scheduling Problem with Sequence Dependent Setup Times*, European Journal of Operational Research, 211/3, 612–622.

[8] ABDULKADER M., ELBEHEIRY M., AFIA N., EL-KHARBOTLY A., 2013, *Scheduling and Sequencing in Four Machines Robotic Cell: Application of Genetic Algorithm and Enumeration Techniques,* Ain Shams Engineering Journal, 4/3, 465–474.

[9] LACOMME P., LARABI M., TCHERNEV N., 2013, *Job-Shop Based Framework for Simultaneous Scheduling of Machines and Automated Guided Vehicles*, International Journal of Production Economics, 143/1, 24–34.

[10] WILSON A.J., PALLAVI D., RAMACHANDRAN M., CHINNASAMY S., SOWMIYA S., 2022, *A Review on Memetic Algorithms and its Developments*, Electrical and Automation Engineering, 1/1, 7–12.

[11] ZARANDI M.F., MOSADEGH H., FATTAHI M., 2013, *Two-Machine Robotic Cell Scheduling Problem with Sequence-Dependent Setup Times,* Computers & Operations Research, 40/5, 1420–1434.

[12] GUNDOGDU E., GULTEKIN H., 2016, *Scheduling in Two-Machine Robotic Cells with a Self-Buffered Robot,* IIE Transactions, 48/2, 170–191.

[13] NOURI H.E., DRISS O.B., GHE´DIRA K., 2016, *Simultaneous Scheduling of Machines and Transport Robots in Flexible Job Shop Environment Using Hybrid Metaheuristics Based on Clustered Holonic Multiagent Model,* Computers & Industrial Engineering, 102, 488–501.

[14] ZABIHZADEH S.S., REZAEIAN J., 2016, *Two Meta-Heuristic Algorithms for Flexible Flow Shop Scheduling Problem with Robotic Transportation and Release Time,* Applied Soft Computing, 40, 319–330.

[15] GHADIRI NEJAD M., KOVACS G., VIZVARI B., BARENJI R.V., 2018, *An Optimization Model for Cyclic Scheduling Problem in Flexible Robotic Cells,* The International Journal of Advanced Manufacturing Technology, 95/9, 3863–3873.

[16] LI X., YANG X., ZHAO Y., TENG Y., DONG Y., 2020, *Metaheuristic for Solving Multi-Objective Job Shop Scheduling Problem in a Robotic Cell*, IEEE Access, 8, 147015–147028.

[17] ZHOU B., LI M., 2017, *Scheduling Method of Robotic Cells with Robot-Collaborated Process and Residency Constraints,* International Journal of Computer Integrated Manufacturing, 30/11, 1164–1178.

[18] REDDY N.S., RAMAMURTHY D., RAO K.P., LALITHA M.P., 2021, *Practical Simultaneous Scheduling of Machines, AGVs, Tool Transporter and Tools in a Multi Machine FMS Using Symbiotic Organisms Search Algorithm,* International Journal of Computer Integrated Manufacturing, 34/2, 153–174.

[19] MILLER E., KAUPP T., SCHMITT J., 2022, *Cascaded Scheduling for Highly Autonomous Production Cells with AGVs*, Global Conference on Sustainable Manufacturing, 383–390.

[20] FERENCZI B.T., KOCZY L., LILIK F., 2022, *Fuzzy Signature Based Model in Material Handling Management,* Computational Intelligence and Mathematics for Tackling Complex Problems, 4, 169–179.

[21] CHAWLA V., CHANDA A., ANGRA S., 2018, *Multi-Load AGVs Scheduling by Application of Modified Memetic Particle Swarm Optimization Algorithm,* Journal of the Brazilian Society of Mechanical Sciences and Engineering, 40, 1–13.

[22] ZHANG S., WANG S., 2018, *Flexible Assembly Job-Shop Scheduling with Sequence-Dependent Setup Times and Part Sharing in a Dynamic Environment: Constraint Programming Model, Mixed-Integer Programming Model, and Dispatching Rules,* IEEE Transactions on Engineering Management, 65/3, 487–504.

[23] NOVAS J.M., 2019, *Production Scheduling and lot Streaming at Flexible Job-Shops Environments Using Constraint Programming,* Computers & Industrial Engineering, 136, 252–264.

[24] MENG L., ZHANG C., REN Y., ZHANG B., LV C., 2020, *Mixed-Integer Linear Programming and Constraint Programming Formulations for Solving Distributed Flexible Job Shop Scheduling Problem*, Computers & Industrial Engineering, 142, 106347.
[25] LUNARDI W.T., BIRGIN E.G., LABORIE P., RONCONI D.P., VOOS H., 2020, *Mixed Integer Linear Programming and Constraint Programming Models for the Online Printing Shop Scheduling Problem,* Computers & Operations Research, 123, 105020.
[26] ROSSI F., VAN BEEK P., WALSH T., 2006, *Handbook of Constraint Programming*, Elsevier.
[27] YOON H.J., 2010, *Scheduling for Deadlock Avoidance Operation in Robotic Manufacturing Cells,* Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 224/2, 329–340.
[28] COFFMAN J.R.E.G., ELPHIK M.J., SHOSHANI A., 1971, *System Deadlocks*, Computing Surveys, 3/2, 67–78.
[29] BANASZAK Z.A., KROGH B.H., 1990, *Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows*, IEEE Transactions on robotics and automation, 6/6, 724–734.
[30] GOOGLE ORTOOLS, https://developers.google.com/optimization/reference/python/sat/python/cp_model#addnooverlap