A framework for adaptive offline task scheduling and resources assignment in industrial environments

S. Zikos, M. Tsourma, A. Drosou and D. Tzovaras

Information Technologies Institute,

Centre for Research and Technology Hellas (CERTH), Thermi-Thessaloniki, Greece

szikos@iti.gr, mtsourma@iti.gr, drosou@iti.gr, dimitrios.tzovaras@iti.gr

May 15, 2018

ABSTRACT

The proliferation of IoT (Internet of Things) devices and networks along with the use of interactive user interfaces in the modern production environments, provide new opportunities for more effective assignment of tasks to resources. This paper proposes a novel integrated framework for offline task scheduling and resources assignment, which can be used in manufacturing industries. The proposed framework can be a part of an intelligent production system in an Industry 4.0 shop floor. For the creation of the work schedule referring to a time interval given a set of tasks and resources, such as workers and machines, multiple criteria are considered: execution restrictions, capabilities of resources, preferences, and past performance data. The main advantage of the scheduling framework is its potential to support adaptive and evolving decision-making based on feedback information received from various sources of the environment. Continuously updated data models that include the characteristics, requirements and performance metrics for each resource in a dynamic environment, as well as configuration settings and feedback provided by the manager, are taken into account in order to generate work schedules which can be efficient in terms of task-to-resource mapping and in accordance with manager's preferences and policies. The implementation requirements of the proposed framework along with an initial system towards the implementation of the framework are also presented in this paper.

Keywords: scheduling, resource assignment, Industry 4.0, smart manufacturing, optimization, evolving intelligent system.

INTRODUCTION

According to the definition provided by [1], scheduling is a decision-making process, which deals with the allocation of resources to tasks over defined time-periods, towards optimizing one or more objectives. The generation of the work schedule of a given shift in an industrial environment is a challenging task for managers or supervisors for various reasons. Different types of resources are usually involved, such as machinery and human resources. Heterogeneity is common in complex manufacturing environments, where each resource has specific capabilities that define the tasks that can be executed as well as task execution performance. Moreover, usually a large number of tasks of different priorities and requirements have to be scheduled and assigned to resources. This process has a dynamic nature, as policies and preferences of stakeholders may change over time, and thus decisions about task scheduling and assignment have to be adapted accordingly. From the computational point of view, tasks have to be assigned to resources in such a way that one or more objectives are optimized. Search for optimal solutions can only be applied in cases of problems of limited size, as required computation time increases exponentially. This is due to the fact that job-shop scheduling problems are generally NP-hard [2].

Due to the availability of data regarding task execution and the use of intelligent automation systems in modern manufacturing industries, the opportunity to automatically improve the quality of the produced work schedule by exploiting feedback data, arises. This paper presents a novel integrated framework for offline task scheduling and resources assignment, which handles those procedures as an adaptive and evolving process. The main objective is to create work schedules that are very close to the expected optimal ones, based on preferences and performance data analysis. In particular, the task execution order must respect the preferences and characteristics of tasks, and on the other hand, tasks have to be assigned to the most appropriate resources, that can be either workers or machines. The main contribution of this paper is the presentation of a concept, which exploits feedback data acquired from operators and machines, in order to update internal information and models. Its aim is to improve the output of a Decision Support System (DSS) responsible for the offline task scheduling and assignment operations.

The proposed scheduling framework has several advantages. Firstly, it utilizes generic data models for the definition of tasks, workers, and machinery resources. Additionally, adaptation according to feedback received from the user on performance related issues is supported. Furthermore, the modular architecture allows easy configuration of the framework’s functionalities. For instance, distinct steps are being followed, allowing the possibility to alter certain parts of the algorithmic procedure, in order to use the most suitable algorithms based on needs and computational resources.

The rest of the paper is organized as follows: Next section presents related work. In section 'PROPOSED FRAMEWORK', the overall concept and architecture of the proposed integrated scheduling framework is presented. A prototype web-based application developed is presented in the 'CASE STUDY' section. Lastly, the final section summarizes the main conclusions and includes topics for future work.

RELATED WORKS

This section presents related work on scheduling and adaptive scheduling fields, including state of the art approaches. Scheduling in industrial production environments has attracted much attention in the last years. Furthermore, opportunities for new scheduling approaches in production environments arise due to the growth of data-driven smart manufacturing [3].

A recent literature review on machine scheduling problems in production can be found in [4]. The authors of [2] have proposed a reactive/online scheduling system, where each resource is assigned to a decision-making agent. Neural reinforcement learning is utilized to enable the agents to learn a dispatching policy and adapt their behavior to the environment. Study [5] investigates the use of genetic programming in automated synthesis of heuristics for the parallel-unrelated machines environment with arbitrary performance criteria. Experiments showed that the proposed algorithm was able to find acceptable and qualitative solutions in a small amount of time. In [6], the authors present an adaptive algorithm with a learning phase for solving the parallel machines job-shop problem, where there are identical machines of the same type. The job-shop problem is modelled via a weighted mixed graph, and the learning phase produces knowledge about a benchmark of priority dispatching rules allowing a scheduler to improve the quality of a schedule. In [7], a mathematical model is proposed to simultaneously solve the process planning and the flexible job shop scheduling problems considering precedence constraints. A genetic algorithm is used in the solving procedure and the objective function considers the makespan, critical machine workload, and total workload of machines. However, dynamic characteristics and possible heterogeneity of resources were not considered in that work.

As far as worker assignment is concerned, recent studies take into account the workers' skills, such as [8], wherein a method with objectives of improving the inter-seru and iner-worker workload balance is proposed in order to solve the worker assignment problem in the context of seru production systems. Alongside, in paper [9] the authors propose a method to solve the personnel assignment problem using

three objective functions, which require simultaneous optimization and computer modelling based on Petri nets.

In our previous work [10], we have presented a web-based task distribution engine and its simulation capabilities with regard to online task distribution to resources. In that work, the problem is formulated as a constrained optimization problem, which considers priorities for each criterion, set as preferences by the user.

The majority of researches found in the literature do not take into consideration multiple types of resources simultaneously, such as human and machinery resources. Furthermore, it is usually assumed that all the resources involved have the same or similar capabilities. Lastly, the occurrence of different types of tasks of different priorities that cannot be static is an important factor in real-life use cases, which is rarely considered.

PROPOSED FRAMEWORK

A. Concept and objectives

The proposed framework for offline task scheduling employs dynamic models of resources (workers, machines) and tasks, which include variables that define several aspects of the modeled entities. The main concept of the framework proposed is displayed in Figure 1. Characteristics for tasks and resources are provided as input to the DSS responsible to generate the work schedule and allocate the tasks to resources based on multiple criteria. The framework provides to users the ability to interact with the work schedule and give feedback, which is then forwarded to the DSS. Additionally, the DSS is able to utilize performance-related information deriving from the system after a task execution. In the former case, the actions of the user are logged and affect the preferences metrics that are taken into account when defining the priority of tasks and the most suitable resources that can be assigned to the tasks. In the latter case, historical data about the performance of each resource on the execution of a particular task are acquired and thus the suitability indexes of each resource are updated. The proposed framework allows to continuously exploit information received from the user and the task execution monitoring system, in order to adapt and produce an output of higher quality, resulting in the development of an evolving intelligent system [11]. In this context, machine learning algorithms can be explored and utilized, in order

to assess the suitability or the expected performance of work schedules by taking into consideration users' preferences and historical task execution performance data as input.

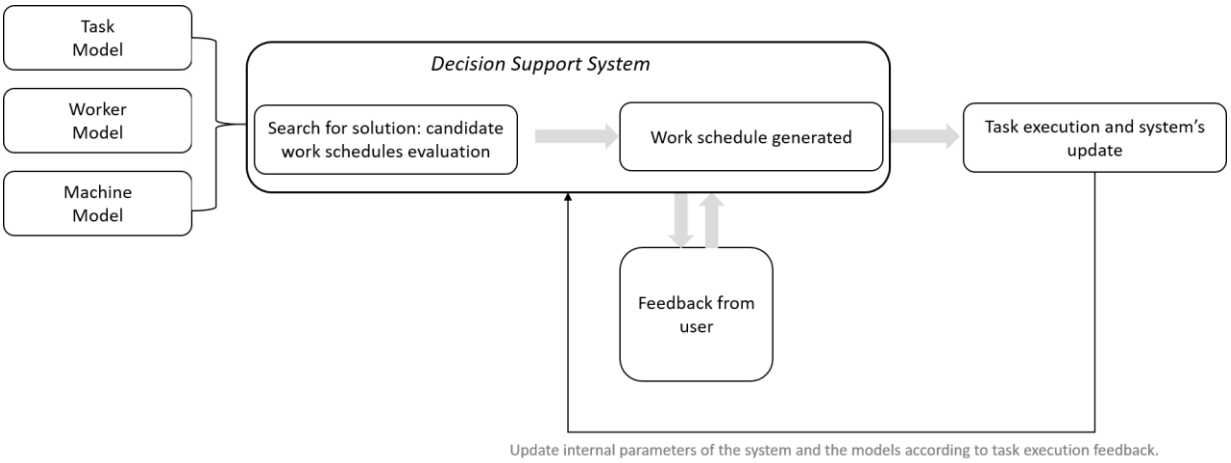

Figure 1: Information flow according to the concept of the proposed framework.

B. Models definition

The DSS will use information concerning the tasks and the resources needed in order to generate a work schedule. This information is included in three data models, the Worker Model, the Task Model, and the Machine Model. Each model includes several static and dynamic parameters for the description of each entity, aiming at better matching with the potentiality to improve the work schedule generation process. Those models are interrelated via the use of several parameters for the realization of the assignment of proper resources to tasks (Figure 2).

Worker Model defines a wide variety of characteristics that describe the abilities of a worker in multiple dimensions (physical, sensorial, cognitive). The characteristics included in the Worker Model refer to the well-being of the worker, including information concerning the worker's physical and emotional status, skills, special abilities and performance, preferences and other general properties. Well-being related characteristics can be extracted automatically after processing variables (e.g. step count) received from IoT wearable devices such as smartwatches or smart bands.

Task Model includes task characteristics, describing certain predefined or possible to be measured features of the tasks that will take place on a shop floor. The model supports tasks that can be comprised of multiple sub-tasks, where precedence constraints can be applied. Precedence constraints define the execution order of sub-tasks and affect their starting times.

In order to define the characteristics of a machine and to ensure a better match to the abilities of a worker, similar features have been selected for the Machine Model. Therefore, Machine Model includes generic characteristics, concerning the general parameters of a machine (e.g. dimensions), variables concerning the required personnel capabilities, equipment and the performance of the machine.

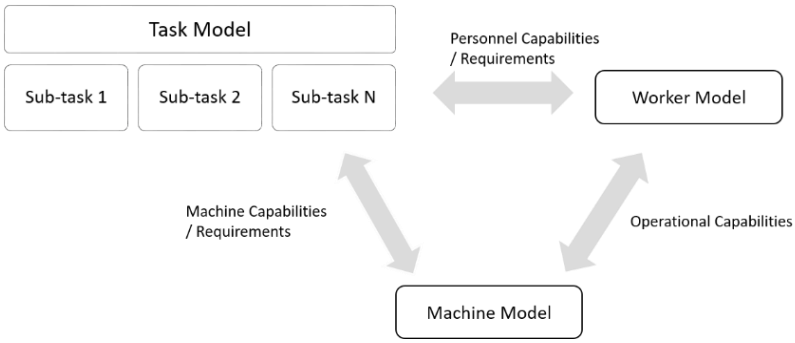

Figure 2: Task/sub-tasks relation with workers and machines.

C. Scheduling and resource assignment

The core functionality of the scheduler, which is a part of the DSS in the proposed framework, is the generation of the work schedule referring to a specific day or shift, given a set of tasks and a set of available heterogeneous resources. The scheduler's objectives are (a) to set the scheduled starting/ending time for each sub-task of all tasks, and (b) to assign each sub-task to specific resource(s) (workers, machines). The priorities of tasks affect their execution order, and consequently it is important for the creation of the work schedule, as constraints may be applied due to specific task requirements such as a due date time.

Due to the large search space, a heuristic method or a method such as a genetic algorithm has to be applied, in order to produce a near to optimal solution in feasible time in the case of a large number of tasks and resources. Generally, several types of constraints usually exist in the context of scheduling, such as temporal, procedural, and resource constraints. Furthermore, for the evaluation of the quality of the work schedule produced, more than one objectives are usually set. As a result, it seems that the formulation of the problem as a constrained optimization problem is a well-suited option. Both hard and soft constraints can be defined. In order to evaluate a candidate solution (work schedule), a utility function is defined and computed per each possible solution. Unfeasible solutions, which do not satisfy the hard constraints, are discarded.

As far as the objectives for the evaluation of the produced solutions is concerned, these are related to user's preferences and the estimated performance of the resources, as well as workload balancing among

resources and task's completion time, based on the data models and the scheduling framework. Regarding the task to resources mapping, each resource has pre-defined capabilities and thus can perform certain types of tasks. In addition, a sub-task may require more than one resources of the same type (e.g. two workers) or more than one resource types (e.g. a worker and a machine), or both. Two different resources can perform a specific task with different efficiency, as it can be determined via the analysis of historical task execution data.

A User Interface (UI), which can be part of the DSS, is required to present metrics that assess the quality of each work schedule produced. Some of them may refer to overall metrics, such as the total makespan, while others may refer to each entity, such as workload per resource. A produced work schedule can be accepted as it is or may be altered by the user. In particular, a sub-task can be re-scheduled in another time, or can be assigned to another human or machinery resource. In case the altered work schedule is valid, the preferences for task priority and competence of resources must be updated accordingly. Any modifications or updates to the work schedule, means the immediate update on the performance metrics, too.

D. Framework implementation requirements

In order to implement the framework for use in a real life system, specific hardware or software may be required, based on the characteristics that will be used. The DSS must provide a user-friendly interface allowing data input, settings configuration, visualization of the DSS's output and of performance metrics. A central database is also required for data storage and data exchange among different modules. Furthermore, a system able to collect task execution performance-related data needs to be available. Sensors will be used for the monitoring of the states of machines and the fluency of production (machine status), the observation of the properties of the working environment (context) and the monitoring of the task execution procedure. In case the well-being status of each worker is of interest, a sensor network comprised of smartwatches for retrieving workers' well-being related data can be used. For example, sleep quality can be taken into account when assigning tasks to workers. In its simplest form, the architecture of the system would be similar to the one shown in Figure 3 below. The DSS component, a central database and a task execution monitoring component are mandatory.

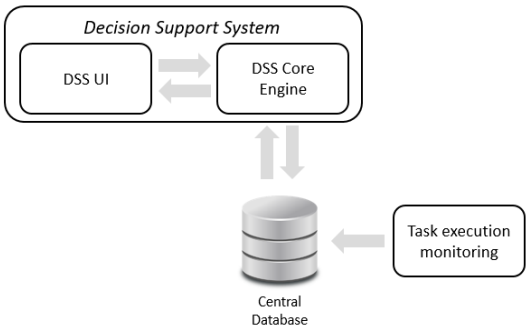

Figure 3: Overall architecture of the proposed framework.

CASE STUDY

The case study that is presented in this section is an initial effort towards the implementation of the proposed framework, and demonstrates work schedule generation when tasks and machines are involved, as well as the visualization of the output result and interaction with the user. Using the basic principles of the framework presented, a prototype web-based application has been developed. In particular, 12 tasks that include 76 sub-tasks in total, have to be scheduled in a shift where 9 machines of different types are available. It should be noted that the data utilized were not gathered from an actual production environment. For this case study, the task and machine models include variables that denote the type, the capabilities, and requirements, as shown in Table 1.

Task/Sub-task characteristics	**Machine characteristics**
ID	ID
Class ID	Class ID
Definition ID	Status
Due Date Time	Capabilities List
Estimated Duration	
Start Time	
End Time	
Priority Level	
Execution Dependencies	
List of Physical Asset Requirement (Machine class and capabilities required)	

Table 1: Characteristics of each model utilized for the case study.

In this study, a two-stage method has been applied in order to solve the offline task-scheduling problem. At the first stage, where the starting times of tasks are defined, tasks are prioritized based on due date time, ascending duration, or pre-determined priority level. When the number of tasks that have to be

scheduled is relatively short, the user can set the order of tasks according to his preferences. Subsequently, an earliest start time approach for each sub-task is followed for determining the scheduled start times. This process takes into account possible precedence constraints among sub-tasks and ensures that there will be at least a set of resources able to perform the scheduled sub-task. According to the precedence constraints set, the first sub-task of each task must be executed before every other sub-task of the task takes place. At the second stage, where, the assignment to resources is performed, different candidate assignment solutions are evaluated, in terms of workload balancing among the machines, in order to select the one of the maximum workload balance efficiency (minimum workload deviation). In this example, it is assumed that all the machines, that are able to execute a sub-task, can execute it with the same efficiency. A mathematical formulation of the problem and the approach is as follows.

M	set of available machines
T	set of tasks
$SubT_j$	set of sub-tasks in task j
$subT_{js}$	the s^{th} sub-task of task j
p_j	priority of task j
d_j	estimated duration of task j
$dtDue_j$	due date time of task j
st_{js}	start time of sub-task s in task j
et_{js}	end time of sub-task s in task j
d_{js}	estimated duration of sub-task s in task j
Pr_subT_{js}	set of sub-tasks that must have been completed before starting $subT_{js}$
$CapM_m$	set of capabilities of machine m
Cap_subT_{js}	capabilities required by sub-task $subT_{js}$
M_{js}	set of candidate machines for sub-task $subT_{js}$, $M_{js} \in M$, $Cap\_subT_{js} \in CapM_m$

Task ordering and definition of sub-tasks at earliest possible start times

Define $A=(t_1, t_2, ..., t_j, \ldots, t_N)$ according to p, d, dtDue. For each $subT \in A$, set EST_{js}: maximum et_s of Pr_subT_{js}. If $Pr\_subT_{js} = \emptyset$, set start shift time as the EST_{js}.

Definition of start and end times for each $subT_{js}$ of a t_j

1. Extract M_{js}
2. Set initial candidate start time and end time, where $et_{js} - st_{js} = d_{js}$, $st_{js} = EST_{js}$
3. Define $SubT_{overlap}$ of overlapping sub-tasks requiring machines of the same capabilities.
4. Calculate the min. number of machines $Machines_{min}$ capable to execute the sub-tasks in $SubT_{overlap}$.
5. If $|M_{js}| \leq Machines_{min}$, advance the starting time (st_{js}) by 1 minute, and start over from step 2. In the opposite case, proceed with the next sub-task.

Assignment of sub-tasks to machines

Let S_f denote a feasible assignment solution, where a machine is assigned to each sub-task. Then the workload balancing index of S_f is defined as,

$$WBI_f = \sum_{i=1}^{m} |WL_i - WL_{AVG}| \quad (1)$$

where WL_i is the total workload of machine i and WL_{AVG} is the average workload of machines. The solution of $\min(WBI_f)$ is finally selected.

Figure 4 presents the generated work schedule and the sub-tasks to machines assignment, based on the most efficient combinations. Each task including its sub-tasks is displayed in a different color. Machines 1 and 2 have the proper capabilities in order to handle all the first sub-tasks of each task. When the first sub-task of a task finishes, the rest of the sub-tasks of a task can be executed. Figure 5 presents the performance metrics produced. In this case, according to the output results, there are no tasks that exceed their due date time on the work schedule produced. Regarding the makespan of the derived work schedule, its value corresponds to 7.95 hours, which is the time needed for all the tasks included in the input dataset to be completed.

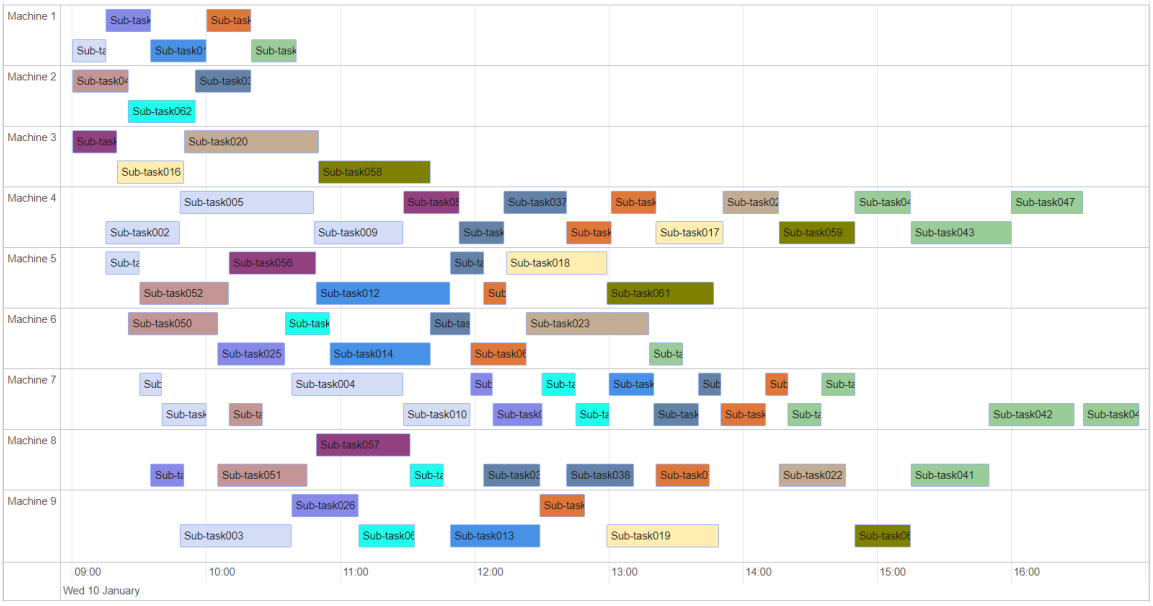

Figure 4: Work schedule generation – assignment of sub-tasks to machines.

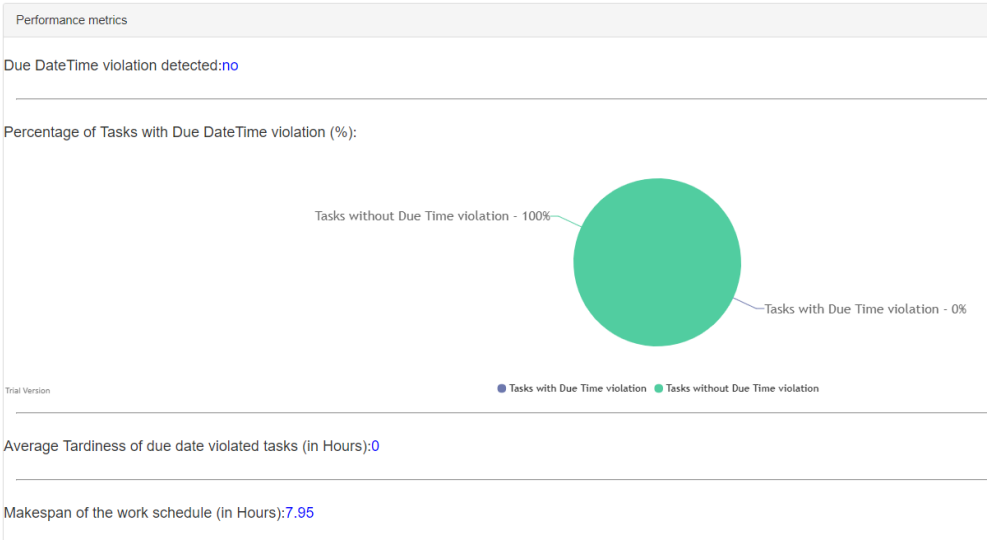

Figure 5: Performance metrics in the automated work schedule creation.

CONCLUSION AND FUTURE WORK

In this paper, a novel integrated framework for offline task scheduling and resources (human, machines) allocation has been proposed. The framework exploits dynamic information available from the user and from a task execution monitoring system (e.g. manufacturing execution system), in order to set task ordering and assign the tasks to the most appropriate resources. As a result, it has the potential to generate more efficient work schedules automatically. The framework offers some important advantages by design, such as the ability to configure the aspects that affect the decisions and the ability to adapt to user's preferences and production needs. An initial implementation of the framework has been developed and demonstrated in this paper. As a future work, we plan to evaluate all features of the proposed framework under long-term use. Furthermore, the possibility to use the framework for real-time online scheduling can be explored.

ACKNOWLEDGEMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723277 (project Factory2Fit). This paper reflects only the authors' view and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1]. Pinedo, M.L. (2008). Scheduling: Theory, Algorithms, and Systems. Third Edition. Springer Science+Business Media. 2008.

[2]. Gabel, T., & Riedmiller, M. (2008). Adaptive reactive job-shop scheduling with reinforcement learning agents. *International Journal of Information Technology and Intelligent Computing*, 24 (4).

[3]. Tao, F., Qi, Q., Liu, A., Kusiak, A., Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D., Zhang, G. & Zhang, Y. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*.

[4]. Abedinnia, H., Glock, C. H., Grosse, E. H., & Schneider, M. (2017). Machine scheduling problems in production: A tertiary study. Computers & Industrial Engineering, 111, 403-416.

[5]. Đurasević, M., Jakobović, D., & Knežević, K. (2016). Adaptive scheduling on unrelated machines with genetic programming. *Applied Soft Computing*, *48*, 419-430.

[6]. Gholami, O., & Sotskov, Y. N. (2014). Solving parallel machines job-shop scheduling problems by an adaptive algorithm. *International Journal of Production Research*, *52* (13), 3888-3904.

[7]. Shokouhi, E. (2018). Integrated multi-objective process planning and flexible job shop scheduling considering precedence constraints. *Production & Manufacturing Research*, *6* (1), 61-89.

[8]. Lian, J., Liu, C., Li, W., & Yin, Y. (2018). Multi-skilled worker assignment in seru production systems considering worker heterogeneity. *Computers & Industrial Engineering*.

[9]. Klosowski, G., Gola, A., & Swic, A. (2015). Human Resource Selection for Manufacturing System Using Petri Nets. *Applied Mechanics and Materials*, *791*, 132.

[10]. Tsourma, M., Zikos, S., Drosou, A., & Tzovaras, D. (2018). Online task distribution simulation in smart factories. in Proc. of the 2nd International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS2018), accepted for publication.

[11]. Lemos, A., Caminhas, W., & Gomide, F. (2013). Evolving intelligent systems: Methods, algorithms and applications. In Emerging Paradigms in Machine Learning (pp. 117-159). Springer, Berlin, Heidelberg.