Multi-agent Environment for Decision-Support in Production Systems Using Machine Learning Methods

Jarosław Koźlak[1](✉), Bartlomiej Sniezynski[1], Dorota Wilk-Kołodziejczyk[1,2], Albert Leśniak[1], and Krzysztof Jaśkowiec[2]

[1] AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
{kozlak,sniezyn}@agh.edu.pl, albert@vp.pl

[2] Foundry Research Institute in Krakow, Al. Zakopiańska 73, Kraków, Poland
{dorota.wilk,krzysztof.jaskowiec}@iod.krakow.pl

Abstract. This paper presents a model and implementation of a multi-agent system to support decisions to optimize a production process in companies. Our goal is to choose the most desirable parameters of the technological process using computer simulation, which will help to avoid or reduce the number of much more expensive trial production processes, using physical production lines. These identified values of production process parameters will be applied later in a real mass production. Decision-making strategies are selected using different machine learning techniques that assist in obtaining products with the required parameters, taking into account sets of historical data. The focus was primarily on the analysis of the quality of prediction of the obtained product parameters for the different algorithms used and different sizes of historical data sets, and therefore different details of information, and secondly on the examination of the times necessary for building decision models for individual algorithms and data–sets. The following algorithms were used: Multilayer Perceptron, Bagging, RandomForest, M5P and Voting. The experiments presented were carried out using data obtained for foundry processes. The JADE platform and the Weka environment were used to implement the multi–agent system.

Keywords: Multi–agent systems · Production planning · Machine learning · Casting production

1 Introduction

Effective decision making in the management of the production process in an enterprise requires a proper selection of activities, some of which may present a significant complexity. Due to the high competitiveness within the industry markets, it is important to reduce necessary costs when offering the final product but to guarantee its required quality. During the production process, numerous

J. M. F. Rodrigues et al. (Eds.): ICCS 2019, LNCS 11537, pp. 517–529, 2019.
https://doi.org/10.1007/978-3-030-22741-8_37

decisions are made at various levels of detail, which in addition can be carried out either according to clearly justified procedures or using practical knowledge gained by previous experience and decision making taking into account the results of decisions which were made previously.

Decision support systems are developed since 1970s. Their goal is to help decision makers with use of models [13], data [12] and knowledge, based on AI approach [3]. Classically, it is implemented as a part of an enterprise-wide application like Enterprise Resource Planning (ERP). Along time, new technologies that can be used for decision support appear, such as genetic algorithms, machine learning, data mining and big data. It is now possible and advantageous to apply approaches such as Industry 4.0 [2,19] or Internet of Things [10] in the production process. Centralized approach becomes inefficient in integrating heterogeneous data sources, many data processing techniques and decision making methods. An approach especially designed for such a case is agent technology [20]. It allows to cooperate with various elements, and to enrich the developed systems with decentralized decision-making methods provided by multi-agent systems approach, and decision-making techniques based on knowledge learned by machine learning algorithms.

In this work, we deal with specific production assumptions in which not all of the campaigns are fully automated. This often happens due to the earlier purchase of components of production lines with limited functional ranges, whose full automatic integration in the future would require very high costs. Therefore, it is beneficial to use a multi–agent approach and data mining methods for a selection of basic decisions in certain selected fragments of the company production process, assuming that certain periods of the process cannot be precisely described and then analyze this information in order to draw specific conclusions that after can be aggregated to develop appropriate policy strategies.

Our scientific contribution is to propose and verify a multi–agent model which supports decision making during the production process and selection of different parameters crucial to the process. We focus on attempts to optimize activities related to the organization and control of the production process in the foundry. A simulator of such a system was prepared, which can predict the results of decisions made during the casting preparation process. Its initial version and preliminary results have been presented in [8]. The system has been expanded and much more experimental research has been carried out based on real data collected during the observation of the actual production process.

The paper is organized as follows. Section 2 contains a description of related research. In section 3 the model of our system is presented. In section 4 we describe main agents, their responsibilities and how they might be represented using the presented model. In section four we present our data-sets and results of our analyses. Section 5 concludes.

2 Related Research

In this section we tackle following research fields: application of multi–agent approaches to production systems, machine learning in production optimization and decision support systems in production systems, and in foundries particularly.

2.1 Multi-agent Approach in Production Systems

The multi-agent approach in production control processes is related to the assurance of decentralization of the decision-making process and the possibilities of using algorithms of collective decision making. At the same time, methodologies specific to the multi-agent approach are being developed that can be used to describe such systems. In [10] there is an analyses of various types of cyber–physical systems, composed of closely connected physical and software components. A classification of such systems was proposed while focusing on the advantages offered by a multi-agent approach in such systems: the autonomy, the use of artificial intelligence techniques or the search for operation strategies using agent simulations. The paper [19] is focused on the use of such important mechanisms used in agent systems which are the decision-making processes that are being developed and negotiations.

The paper [7] is focused on ensuring the fulfillment of requirements related to real-time systems and the use of low-cost technologies. In [23] an expert system based on agents using fuzzy calculation techniques is presented.

2.2 Machine Learning in Production Optimization

One of the important areas in production is machine scheduling problem. In [6] rule learning framework with creation of optimal solution is introduced. Support vector machines (SVM) algorithm is applied in [5]. The goal is to solve resource constraint scheduling problem. The main problem is labelling cost. It needs to be done by experts which is time-consuming. Therefore approach more often being used is based on reinforcement learning. In [14], reinforcement learning approach is proposed for scheduling in online flow-shop problem. Similar problem is solved in [15] using supervised learning approach.

2.3 Decision–Support Systems in Production Systems in Foundries

Taking into account the complexity of the decision-making process of production control, a number of computer decision support systems have been created to optimize these activities. In particular, various description models for such systems have been developed.

In [26] management issues of foundry enterprises and especially production management, are discussed. The authors propose several models such as: Single-piece management model (based on casting life-cycle), Process management model (based on task-driven technology), Duration monitoring model (based on surplus period), and Business intelligence data analysis model (based on data mining).

In [11] is presented a mathematical model based on the actual production process of a foundry flow shop. An improved genetic algorithm is proposed to solve the problem.

In [9], another meta-heuristic optimization method, a particle swarm optimization, is applied to find Pareto optimal solutions that minimize fuel consumption.

The paper [4] contains a model for ranking the suppliers for the industry using the multi-criteria decision making tool, where two hierarchies are distinguished: main criteria (price, quality, delivery and service are ranked based on the experts' opinions) and sub criteria (identified and ranked with respect to their associated main criteria).

In [16], one can find a model for production planning for dynamic cellular manufacturing system. The robust optimization approach is developed with deterministic non-linear mathematical model, which comprises cell formation, inter-cell layout design, production planning, operator assignment, machine reliability and alternative process routings, with the aim to minimize several parameters (machine breakdown cost, inter-intra cell part trip costs, machines relocation cost, inventory holding and backorder costs, operator's training and hiring costs).

Different particular production problems are solved using decision support systems. The paper [17] describes an automatic assist to estimate a production capacity in a casting company, In [25] there is a proposition of heat treatment batch plan, which is very important to ensure the quality of casting products. A charge plan for heat treatment is not a simple combination of different castings, castings with great difference in heat treatment process cannot be put into the same furnace, otherwise, the quality of castings can be adversely affected. On the other hand, furnace capacity and the delivery deadline of castings must be considered to maximize the use of resources and guarantee delivery on time. The charge plan for heat treatment is considered as a complex combinatorial optimization problem.

In [24], the key technology of intelligent factory is reviewed, and a new "Data + Prediction + Decision Support" mode of operation analysis and decision system based on data driven is applied in a die casting intelligent factory. Three layers of a cyber-physical system are designed. In order to form effective decision support, pre-processes the manufacturing data and uses the data mining technology to predict the key performance.

3 System Model

Let us define a multi-agent system for decision support with machine learning and data mining as the following pair:

$$System = (E, A), \tag{1}$$

where E is environment and A is a set of agents.

$$E = (P, C, I), \tag{2}$$

where P is set of resources representing process parameters that are observed by agents, C is a set controlled resources representing decisions about the process that may be taken by the agents, I is a set resources representing intermediate results calculated by agents. We define R as the set of all resources: $R = P \cup C \cup I$.

Every resource $r \in R$ has corresponding domain D_r that contains a special empty value ε. We assume that in the time stamp t the environment has specific values assigned to the resources. It is represented by a value function $f_t : R \ni r \rightarrow v_r \in D_r$. This function may be extended to assign values to set of resources. To represent a set of possible values of resources $X \subset R$ we write $D(X)$.

Agent definition is a simplified version of a learning agent proposed in [18]. The agent is the following tuple:

$$A_i = (X_i, Y_i, S_i, s_i, s_i^0, Act_i, \pi_i, L_i, E_i, K_i, Col_i). \tag{3}$$

X_i represents observations of agent i, $X_i \subset R$, Y_i represents variables calculated by the agent i, $Y_i \subset C \cup I$, S_i is a set of states, $s_i \in S_i$ is a current state and $s_i^0 \in S_i$ is the initialization state, Act_i is a set of actions, π_i is a strategy, $\pi_i : D(X_i) \times S_i, \rightarrow Act_i$, L_i is a possibly empty learning algorithm, that may modify π_i, E_i represents experience collected by the agent that can be used by L_i to learn the knowledge K_i. This knowledge can be used by π_i to choose appropriate action or during action execution. $Col_i = \{(r, A_j) : r \in I, A_j \in A\}$ represents a set of agents that A_i can ask to calculate given resource.

The type of the learning algorithm L_i depends on the application. For example, reinforcement learning directly creates the strategy. The agent may also apply supervised learning algorithm to learn a model that is used to choose the best action (see [18]). In complex environments the agent could apply several learning algorithms to create more than one model; however, we did not need it in our experiments yet.

Relation between the agents is defined by a common use of resources. We say that agent A_i depends on A_j if $Y_j \cap X_i \neq \emptyset$. It means that A_i observes resources calculated by A_j.

The behaviour of the agents is defined in Algorithm 1. After the initialization (lines 1–3) there is a main loop in which the agent observes the environment, processes communicates sent by other agents (see below) and updates its state (taking into account these communicates). Next (lines 8–11), it decides if results of the previous action or observed situation should be stored in the experience. If yes, appropriate example is added to the E_i. Next, an action a_i is chosen according to the current strategy π_i and it is executed (lines 12–13) using current knowledge K_i. Finally, the learning algorithm is executed (lines 14–16). In case of reinforcement learning, it may happen every turn. It results in update of the K_i representing the Q table used by the π_i to chose the action. In case of supervised or unsupervised learning, a new model (or models), which is applied in π_i is relearned or the old one is incrementally updated. This is more time consuming; therefore, it may be performed only occasionally (e.g. every 100 steps).

```
$t = 0$;
$E_i = K_i = \emptyset$;
$s_i = s_i^0$;
while agent $A_i$ lives do
    collect values of observations $f_t(X_i)$;
    process communicates received;
    update state $s_i$;
    if previous action have given results that should be learned then
        generate an example $e$;
        store the example $e$ in the experience $E_i$;
    end
    chose the action using current strategy $a_i = \pi_i(f_t(X_i), s_i)$;
    execute action $a_i$ using $K_i$;
    if it is time for learning then
        update the knowledge $K_i$ using $L_i(E_i)$;
    end
end
```

Algorithm 1. Control algorithm of agent A_i

Two types of actions should be distinguished: ones that calculate resource values represented by the set Act^c and ones that are related to a social behaviour Act^s. If $a_i \in Act^c$ then its execution calculates values of resources that are subset of Y_i from values of X_i. Knowledge K_i may be used during this calculation.

There should be at least three social actions: $\mathrm{ASKFOR}(r, j), \mathrm{YES}, \mathrm{NO} \subset Act^s$. The action $\mathrm{ASKFOR}(r, j)$ represents a request sent to the agent A_j, on which A_i depends, to provide value of resource r. We assume that $(r, A_j) \in Col_i$. It is sent if the agent needs this value. The receiver may respond with answer YES and calculate this value or with answer NO.

4 System Realization

The implemented system uses multi–agent platform JADE [1] and machine learning and data mining environment Weka [22]. The developed system uses following external entities:

- Process – production process controlled by agents. Sensors of the Process send and receive data to/from Process Management Agents and controls the variables process parameters.
- User – obtains data regarding the processes and proposed recommendations as well as sends queries to the systems.
- Data base – stores historical data with parameters describing previous production processes.

The following agents are implemented in the system:

- *Production Management Agent* (PMA) - responsible for the management of the Process Management Agents and management of the data gathered from them, Its life cycle equals the life cycle of the whole system.

- *Process Controlling Agent* (PCA) – responsible for communication with the given production process, gathers information from sensor of the production process and sends them to the Production Management Agent, it is also responsible for setting the parameters of the production process. Its life cycle is limited to the duration of the given production process. There may be several PCA associated with various production stages.
- *Product Parameter Prediction Agent* (PPPA) – agent associated with Process Controlling Agent, it is possible to have multiple instances of such agent for various parameters and using different learning algorithms or different datasets of historical data. It functions during the whole life cycle of the system.
- *User Interface Agent* (UIA) – responsible for the interactions with user and subsequent communications with other agents. Its life cycle equals the life cycle of the whole system.

The first three agents are essential for the system. The last provide auxiliary functions. In our implementation process parameters, P represents composition of the cast and casting parameters that are observed by agents, C represents casting parameters that can be controlled, I represent parameters of the product created in the casting process. X_{PCA} correspond to the production process stage. Strategy π_{PCA} is constant (L_{PCA} is empty) and it executes actions Act_{PCA} calculating high level parameters $Y_{PCA} \subset I$ of this stage. X_{PPPA} is a subset of R that have influence on a predicted cast parameters Y_i. This type of agent stores examples consisting of the production process description (independent variables) and achieved cast parameters (dependent variables) in E_{PPPA}. A learning algorithm L_{PPPA} is used to create a knowledge K_{PPPA} predicting cast parameters. We use regression algorithms for this purpose. Col_{PPPA} consists of PCAs, because PPPA may ask PCA to calculate high level parameters. PMA is responsible for supervising the whole production process. X_{PMA} is a subset of R that are important. Col_{PMA} consists of PCAs and PPPAs, because PMA make ask these agents for current values to recommend changes in in the process. Recommendations are presented to the user through (UIA) and if they are accepted, appropriate Y_{PMA} values are set.

5 Experiments

5.1 Description of Scenarios

Casting production is connected with the completion of a number of simulations and test casts in advance, which aim at gathering data allowing for the subsequent launch of production. The tests were carried out on samples whose chemical composition was as presented in Table 1.

Table 1. Chemical composition of cast iron melts

Melt	C	Si	Mn	P	S	Mg	Cu	Ni
W1	3.65	2.59	0.18	0.052	0.014	0.060	-	-
W2	3.41	2.62	0.30	0.046	0.016	0.056	0.48	-
W3	3.39	2.62	0.29	0.042	0.010	0.036	0.51	0.72

The tests were carried out according to two different variants of heat treatment and data gatherer during real world experiments described in [21]. Heat treatment variants influence the final cast parameters. Samples prepared in this way have been tested in the field of breaking energy parameters for these casts. The research aims to determine the parameters of the production process. The tests are carried out during the preparation of the production of a cast element from a new material. Looking at the whole process, this can be placed in the scope of performing a trial series. The results of this work are supported by the designer and technologist at the stage of preparing the element for production (planning process parameters, the shape of the mold and casting), as well as at the stage of performing the test series and process simulation. In the absence of data in this area, there is a need for many technological trials. The developed tool allows you to limit their number.

The evaluation carried out for the purposes of the current article focuses on the following elements:

(i) determination of the correlation, we investigate correlations between parameters such as metal chemical composition and heat treatment of the casting. These processes have an effect on obtaining the parameters of the finished product, in our case of impact strength. The impact test was carried out on a machine called the Charpy hammer, type PSW 300, with a maximum impact energy of 300 J. During the examination of the predicted forces necessary to distort the considered alloys and the actual acquired information on the distorting forces of the considered sample we calculate also the prediction error of these quantities using different types of classifiers and different sizes of the considered set of samples.

(ii) analysis of the time costs of building the model for the larger artificially generated data sets describing samples, taking into account different classifiers and different sizes of the analyzed sets.

We decided to use for the prediction algorithms with a high level of complexity, of which a majority contain basic algorithms, which should increase the chances of obtaining high-quality results. We used the Weka [22] environment and following algorithms: *Bagging*, *RandomForest*, *M5P*, *MultilayerPerceptron* and *Vote*. The *Bagging* algorithm provides averaging of values returned by underlying REPTree algorithms using learning decision trees with reduced-pruning, which improves the stability and accuracy of results. The *RandomForest* algorithm calculates the results as an average obtained from several trees and it is considered that to significantly reduces the risk of over-fitting and limits the influence of a tree that does not achieve good results. The *M5P* algorithm generates 5 model trees. *MultilayerPerceptron* uses a supervised learning method called back propagation of error to set appropriate values of weights associated with neurons building the model. The *Vote* algorithm is a complex algorithm using Bagging, RandomForest, M5P and MultilayerPerceptron algorithms, from which the average value is calculated.

The considered data-sets are as follows:

- *Full* – 432 samples, for each considered set of parameter values it contains 3 records with different calculated results of sample breaking energy,
- *Sample1* – 144 samples, it contains the average value of sample breaking energy for each considered set of parameter values,
- *Sample2* – the smallest considered sample, it contains 14 randomly selected samples that were extracted from the reference pool containing 144 samples with mean values of breaking energy.

5.2 Results for Different Classifiers and Different Selection of Sample Data

Figure 1 shows that for different algorithms and for different sizes of data-sets it was possible to obtain a similar, high correlation level. It can be seen that usually for averaged values (*Sample1*) good results were obtained compared to other sets, especially in the case of MultilayerPerceptron. A very limited, randomly selected *Sample2* set also showed a fairly high correlation of predicted values.

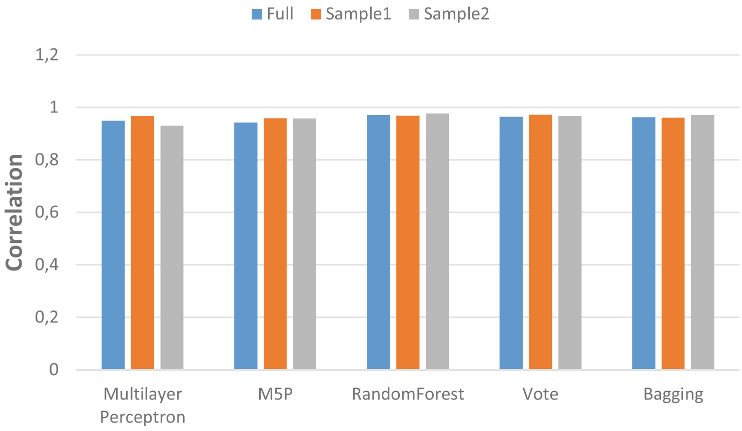

Fig. 1. Quality of prediction for selected classifiers and considered data-sets

In Fig. 2 the smallest relative absolute error for the Full set was obtained for the Vote algorithm, for the *Sample1* set - for the Vote algorithm as well, and for the *Sample2* set - for the RandomForest algorithm. It can be seen that for the majority of cases the smallest error is obtained for the data-sets with the most detailed data (and the largest set), only in the case of MultilayerPerceptron the best quality of prediction was obtained for *Sample1*, and for Random Forest for *Sample2*, which is definitely the smallest set. The smallest set (*Sample2*) is characterized by the highest error of the relative absolute error prediction, only in the case of Random Forest the analysis of data-set with the averaged values (*Sample1*) give the highest error.

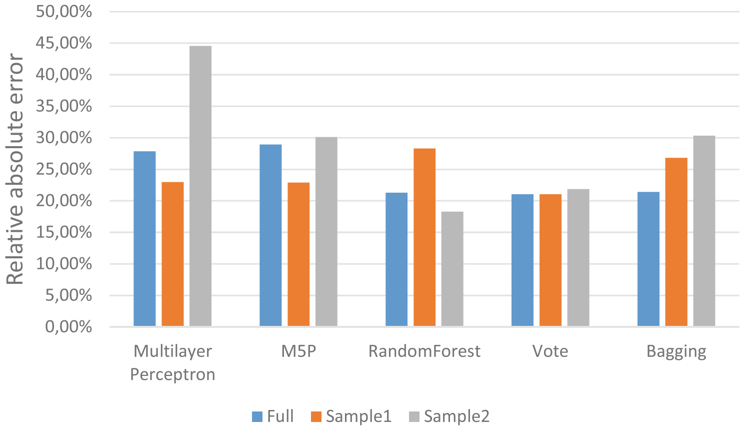

Fig. 2. Relative errors for selected classifiers and considered data-sets

5.3 Discussion of Costs of Model Construction

Subsequent tests concerned different times of building models for prediction using different sizes of data sets and different algorithms. The larger data sets than in the previous tests were used (90, 900 and 3600 samples), which, however, were fully randomly generated, and not based on the acquired data describing the technological process.

For two larger data-sets (900 and 3600 samples), the correlation was clearly higher than the value obtained for the smallest data-set (90 samples) (Fig. 3). The quality of results for larger sets was similar in case all used classifiers.

The considered models differ quite significantly from the point of view of the model's construction time, these differences occur for all sizes of data sets (Fig. 4).

It can be noted that the longest time of model building for the largest sets (3600 and 900 samples) has the Vote algorithm. This is in line with expectations because this algorithm not only uses several base algorithms, but they are of different kinds, which increases complications and resource consumption. For the smallest data-sets (90 samples) the RandomForest algorithm is the slowest. For a large set (3600) the RandomForest algorithm is the fastest, which may be

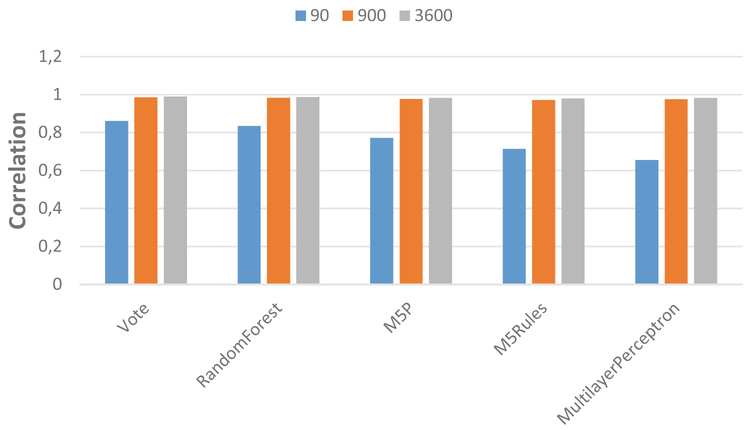

Fig. 3. Quality of prediction obtained for larger generated data-sets

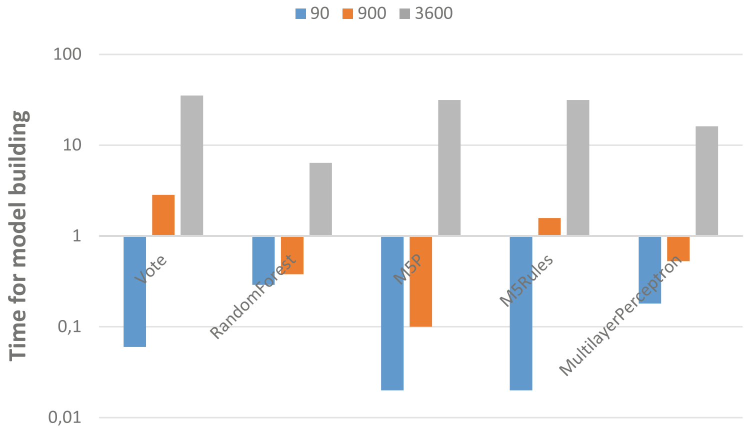

Fig. 4. Time for model building for different classifiers and data-set sizes [s]

due to the similarity of the underlying algorithms, for the average data-set (900 samples) the M5P wins, and for the smallest data-set (90 samples) M5P and M5Rules are the fastest.

6 Conclusions

In the paper a multi-agent system model for the optimization of the production process for the foundry companies and prediction of the production parameters using data-sets with different characteristics are presented.

The work carried out is aimed at obtaining a reduction in costs by performing a large number of virtual experiments and limiting the number of physical experiments, which is associated with the saving of materials and devices.

Experimental results are very promising. Therefore, these new methods may give good estimations of configuration of the production process with limited costs when considering materials, use of machines, avoiding the need to make changes in the configuration of production, which is cumbersome from the point of view of the work organization and even impossible due to the lack of free time, when it would be a possibility of carrying out such work.

Further planned works include more detailed analyses of the data-sets describing production process, the use of other existing data-sets describing the actual physical processes and the development of the agent system with more advanced interaction mechanisms (negotiations, setting constraints and norms, which particular agents building models should respect). We are also planning to apply machine learning algorithms in these interaction mechanisms.

Acknowledgment. This work is partially supported by the TECHMAT-STRATEG1/348072/2/NCBR/2017 Project.

References

1. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-agent Systems with JADE. Wiley, Hoboken (2007)
2. Brettel, M., Friederichsen, N., Keller, M., Rosenberg, M.: How virtualization, decentralization and network building change the manufacturing landscape: an industry 4.0 perspective. Int. J. Mech. Aerosp. Ind. Mechatron. Manuf. Eng. **8**, 37–44 (2014)
3. Dhar, V., Stein, R.: Intelligent Decision Support Methods: The Science of Knowledge Work. Prentice-Hall Inc., Upper Saddle River (1997)
4. Dweiri, F., Kumar, S., Khan, S.A., Jain, V.: Designing an integrated AHP based decision support system for supplier selection in automotive industry. Expert Syst. Appl. **62**, 273–283 (2016)
5. Gersmann, K., Hammer, B.: Improving iterative repair strategies for scheduling with the SVM. Neurocomputing **63**, 271–292 (2005). New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks
6. Ingimundardottir, H., Runarsson, T.P.: Supervised learning linear priority dispatch rules for job-shop scheduling. In: Coello, C.A.C. (ed.) LION 2011. LNCS, vol. 6683, pp. 263–277. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25566-3_20
7. Kantamneni, A., Brown, L.E., Parker, G., Weaver, W.W.: Survey of multi-agent systems for microgrid control. Eng. Appl. Artif. Intell. **45**, 192–203 (2015)
8. Koźlak, J., Śnieżyński, B., Wilk-Kołodziejczyk, D., Kluska-Nawarecka, S., Jaśkowiec, K., Żabińska, M.: Agent-based decision-information system supporting effective resource management of companies. In: Nguyen, N.T., Pimenidis, E., Khan, Z., Trawiński, B. (eds.) ICCCI 2018. LNCS (LNAI), vol. 11055, pp. 309–318. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98443-8_28
9. Lee, H., Aydin, N., Choi, Y., Lekhavat, S., Irani, Z.: A decision support system for vessel speed decision in maritime logistics using weather archive big data. Comput. Oper. Res. **98**, 330–342 (2018)
10. Leitão, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., Colombo, A.W.: Smart agents in industrial cyber-physical systems. Proc. IEEE **104**(5), 1086–1101 (2016)

11. Li, X., Guo, S., Liu, Y., Du, B., Wang, L.: A production planning model for make-to-order foundry flow shop with capacity constraint. Math. Probl. Eng. **2017**, 1–15 (2017)
12. Power, D.J.: Understanding data-driven decision support systems. Inf. Syst. Manag. **25**(2), 149–154 (2008)
13. Power, D.J., Sharda, R.: Model-driven decision support systems: concepts and research directions. Decis. Support Syst. **43**(3), 1044–1061 (2007)
14. Qu, S., Chu, T., Wang, J., Leckie, J., Jian, W.: A centralized reinforcement learning approach for proactive scheduling in manufacturing. In: ETFA, pp. 1–8. IEEE (2015)
15. Sadel, B., Sniezynski, B.: Online supervised learning approach for machine scheduling. Schedae Inf. **25**, 165–176 (2017)
16. Sakhaii, M., Tavakkoli-Moghaddam, R., Bagheri, M., Vatani, B.: A robust optimization approach for an integrated dynamic cellular manufacturing system and production planning with unreliable machines. Appl. Math. Model. **40**, 169–191 (2016)
17. Sika, R., et al.: Trends and Advances in InformationSystems and Technologies, vol. 747. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-77700-9
18. Sniezynski, B.: A strategy learning model for autonomous agents based on classification. Int. J. Appl. Math. Comput. Sci. **25**(3), 471–482 (2015)
19. Wang, S., Wan, J., Zhang, D., Li, D., Zhang, C.: Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. Comput. Netw. **101**, 158–168 (2016). Industrial Technologies and Applications for the Internet of Things
20. Weiss, G. (ed.): Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, 2nd edn. MIT Press, Cambridge (2013)
21. Wilk-Kołodziejczyk, D., Regulski, K., Giętka, T., Gumienny, G., Kluska-Nawarecka, S., Jaśkowiec, K.: The selection of heat treatment parameters to obtain austempered ductile iron with the required impact strength. J. Mater. Eng. Perform. **27**, 5865–5878 (2018)
22. Witten, I.H., Frank, E., Hell, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Elsevier, Amsterdam (2011)
23. Zarandi, M.H.F., Tarimoradi, M., Shirazi, M.A., Turksan, I.B.: Fuzzy intelligent agent-based expert system to keep information systems aligned with the strategy plans: a novel approach toward SISP. In: 2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) Held Jointly with 2015 5th World Conference on Soft Computing (WConSC), pp. 1–5, August 2015
24. Zhao, Y., Qian, F., Gao, Y.: Data driven die casting smart factory solution. In: Wang, S., Price, M., Lim, M.K., Jin, Y., Luo, Y., Chen, R. (eds.) ICSEE/IMIOT -2018. CCIS, vol. 923, pp. 13–21. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-2396-6_2
25. Zhou, J., Ye, H., Ji, X., Deng, W.: An improved backtracking search algorithm for casting heat treatment charge plan problem. J. Intell. Manuf. **20**(3), 1335–1350 (2019)
26. Zhou, J., et al.: Research and application of enterprise resource planning system for foundry enterprises. Appl. Energy **10**, 7–17 (2013)