Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

AGENT-BASED SELF-ORGANIZATION VERSUS CENTRAL PRODUCTION PLANNING

Prof. Dr. rer. pol. Torsten Munkelt

Dresden University of Applied Sciences
Faculty of Computer Science
Friedrich-List-Platz 1
Dresden, 01069, GERMANY

M.Sc. Martin Krockert

Dresden University of Applied Sciences
Faculty of Computer Science
Friedrich-List-Platz 1
Dresden, 01069, GERMANY

ABSTRACT

Industry 4.0, a part of the German high tech strategy, prefers self-organization in production over central production planning for the sake of greater flexibility, faster response to disruptions and to deviations, and less effort. Current planning systems usually plan centrally. We developed a universal self-organizing production and empirically compared its performance to a centrally planned production. The self-organizing production is based upon agents. For better comprehensibility, we additionally implemented central planning. The results of self-organization in production are promising in relation to central planning; especially when disruptions and deviations occur.

1 INTRODUCTION

Current ERP, APS, and ME systems plan, control, and optimize production rather centrally. The systems provide „naive“ resources with the following targets: when, what, where, how much, and by what means to purchase, to produce, to store, etc. Thereupon, the resources give feedback regarding time, quantity, and completion of production (Figure 1 a). According to the feedback, the above systems plan, control, and optimize anew after every shift or just every night.

Industry 4.0 turns away from central planning, control, and optimization and promotes self-organization in production (Figure 1 b) (Lasi 2014) which is expected to result in greater flexibility, in faster response to disruptions of the production and deviations in processing times, and in less computational effort. A combination of rough central planning and decentral self-organization may lead to even better results.

In this contribution, we just provide empirical evidence that pure self-organization in production is already possible. We compare a centrally planned and a self-organizing production empirically, which means we gain statistical results from experiments with a stochastic production. Our experiments document that a self-organizing production performs as good as a centrally planned production in most cases and often even better; especially when disruptions or deviations of processing times occur.

After the introduction, we explain self-organization as an ability of a system, list features of a self-organizing system, define a self-organizing production, and propose agents to achieve self-organization in production. Those agents we deduce from "intelligent" resources and "intelligent" machines proposed by Industry 4.0. The next section describes the data model and the applied algorithms of central production planning. In order to compare a centrally planned and a self-organizing production comprehensibly and fairly, we apply central production planning as well as concepts of self-organization to the very same production, which we describe in detail next. Afterwards, we design and implement a multi-agent-based self-organizing production, and we explain how we simulate the production. Next, we compare the self-organizing production and the centrally planned production empirically with regard to different configurations and objectives. Finally, we summarize this contribution and give an outlook on the combination of central production planning and a self-organizing production.

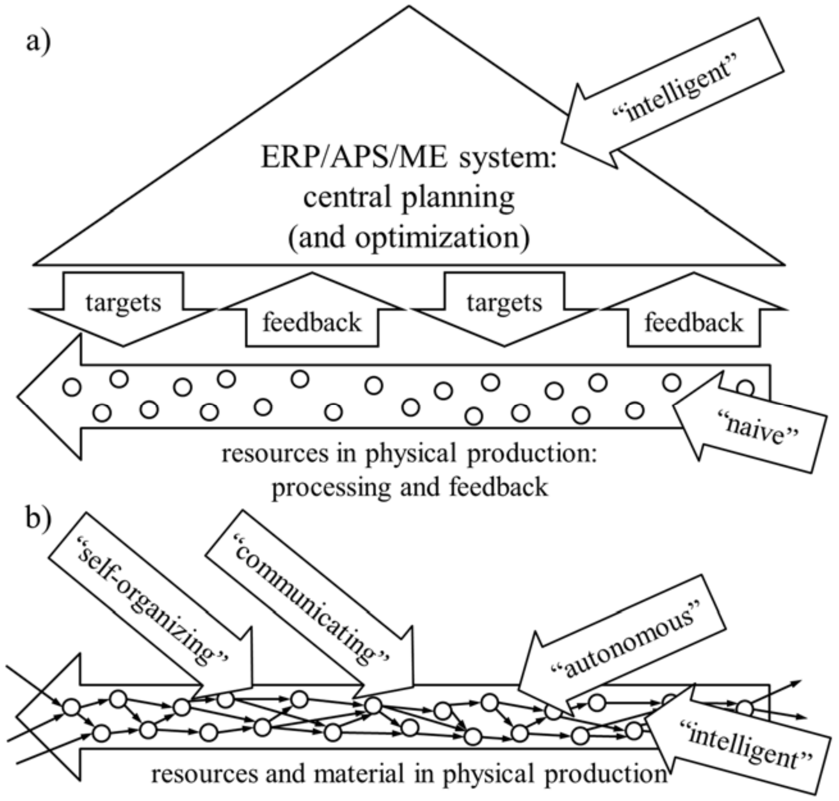

a)
"intelligent"
ERP/APS/ME system: central planning (and optimization)
targets
feedback
targets
feedback
"naive"
resources in physical production: processing and feedback
b)
"self-organizing"
"communicating"
"autonomous"
"intelligent"
resources and material in physical production

Figure 1: a) central production planning and b) self-organizing production.

2 SELF-ORGANIZATION IN OUR PRODUCTION

Since self-organization is always an ability of a system, we initially define a system: A system is a set of components which interact with each other and differ from the environment of the system (Buckley 1998). There are many different definitions of self-organizing systems, e.g., "A self-organizing system is a system that changes its basic structure as a function of its experience and environment" (Clark and Farley 1954). and "[A self-organizing system is a system in which] pattern formation occurs through interactions internal to the system, without intervention by external directing influences" (Camazine et al. 2003). In the literature, the following properties are attributed to self-organizing systems (Köhler-Bußmeier 2010; Camazine et al. 2003; Kauffman 1996). Those properties may not be mutually exclusive and collectively exhaustive:

1. Openness: The system exchanges matter, energy, or information with its environment, and the exchange is the cause of the development of the system.
2. Adaptiveness: The systems adapts to changes in its environment.
3. Autonomy: The system and its components process all (external) stimuli from the environment according to internal mechanisms.
4. Nonlinearity: The output of the system does not always change proportionally to its input.
5. Indeterminism: The system develops in a way which is hard to predict because it depends on random events, or almost equal inputs lead to completely different paths of development.
6. Attractedness: Over time, the system converges asymptotically to a region (a so-called attractor) in the state space of the system, and if the environment disturbs the system only slightly, the system will stay near the region.
7. Path dependence/operational closedness: The history (path of development) of the system influences the further development of the system much more than the environment of the system does.

8. Emergence: The system develops new spatio-temporal characteristics or structures as result of the interaction of its components. Those new characteristics and structures cannot be explained by the rather simple behavior of the single components.
9. Autopoiesis: The system or its components can autonomously create the components of the system. Hence the systems preserves itself.

According to the above definitions, properties 1, 2, 3, 7 and 8 are essential for a self-organizing system. Thus, if a system exhibits these properties, we will call it a self-organizing system. A production is a system which produces goods (Schuh 2006). If the production is also a self-organizing system, then it is a self-organizing production (system). Productions often also exhibit property 5. Because of property 5, analytical methods often fail to analyze productions successfully. Hence, we analyze our production by means of discrete-event simulation.

Self-organization in production is a central topic of Industry 4.0. The German government introduced Industry 4.0 as part of its high-tech strategy in 2011 (Kagermann et al. 2013). Industry 4.0 describes the tight interlinkage between industrial production and modern information and communication technology. Industry 4.0 is supposed to enable self-organizing production. Industry 4.0 comprises "intelligent" resources and "intelligent" materials. Resources are humans, machines, means of transportation, and storage systems. Materials are raw materials, purchase parts, assembly groups as well as final products. Resources and materials communicate and cooperate directly and locally and should thereby self-organize the whole value-added network. Communication and cooperation span all phases of the product lifecycle; from the idea and the design of the product, to its production and its application, and up to its maintenance and its recycling. Concepts, programs, or platforms similar to Industry 4.0 are US-American "Smart Manufacturing", "Advanced Manufacturing", or "Industrial Value-Chain Initiative" (Anderl et al. 2016), or the Chinese "Made in China 2015".

Production planning, control, and optimization already successfully applied self-organizing systems like self-organizing maps, artificial ant colonies (Klüver et al. 2012), artificial neural networks (Hammami et al. 2015), and genetic algorithms. However, the combination of self-organizing production and Industry 4.0 leads straight to multi-agent systems (MASs). There are three possible relationships between the agents on one side and the "intelligent" resources and materials on the other: Resources and materials *are* (two types of) agents, or agents connect to resources and materials physically, or agents are digital twins (Grieves 2014) of resources and materials. In the latter two cases, the agents make resources and materials "intelligent". In either case, the agents communicate and collaborate. Thus, the production organizes itself. Additionally, our MAS exhibits property 9: The MAS and its agents can autonomously create their agents.

3 OUR CENTRAL PRODUCTION PLANNING IMPEMENTED FOR THE PURPOSE OF COMPARISON

We implemented our own central production planning in order to compare a self-organizing production and a centrally planned production fairly and comprehensibly. We adopted and stripped data structures and algorithms from current ERP systems, like SAP, as far as possible. Figure 2 shows the classes, the interfaces and the associations, aggregations, and generalizations/specializations connecting the classes.

We omitted the attributes, the methods, and the cardinalities in the diagram for the sake of clarity and simplicity. An arrow head or a diamond at one end of a relationship indicates the "one" side of a one-to-many relationship, and the other side is always the "many" side. A relational database persists the data according to a schema similar to the UML class diagram in figure 2. Although agents of our self-organizing production could store their own data, the same database also persistently holds the data of our self-organizing production for the purpose of reporting and analysis.

Our algorithm for central planning adapts scheduling and planning algorithms known from SAP and other ERP and PPC systems. The algorithm covers the tactical midterm production and capacity planning of the manufacturing resource planning (MRP II). It works as follows: In the first phase, the algorithm creates the network of requirements and satisfiers. It starts with requirements which are not yet satisfied.

During the same phase, the algorithm merges and splits lots according to fixed lot sizes and it schedules the network backwards, starting at their due time. If at least one satisfier starts in the past, the algorithm will execute phase two and phase three: In the second phase, the algorithm schedules the network forwards. It begins with the satisfiers which start in the past and have no predecessors. In phase three, the algorithm schedules the network backwards again. It starts at the maximum of two points in time: the due time or the end time calculated by forwards scheduling. Phase three of the algorithm guarantees that the satisfiers start as late as possible. Phase four of the central planning executes finite capacity planning. It adapts the algorithm of Giffler and Thompson (Zäpfel and Braune 2005), takes earliest start times into account additionally, and schedules first the operation of that production order with shortest remaining slack time. Our central planning deliberately omits the phases of infinite capacity demand planning and capacity levelling because existing planning systems also do not automate these steps and we take care of capacity utilization ourselves. Our central production planning operates periodically. When it starts anew, it changes remaining processing times and quantities according to the feedback from the production. Moreover, all "Required and Satisfied" objects (see Figure 2) are discarded and created anew with respect to the current situation of the production.

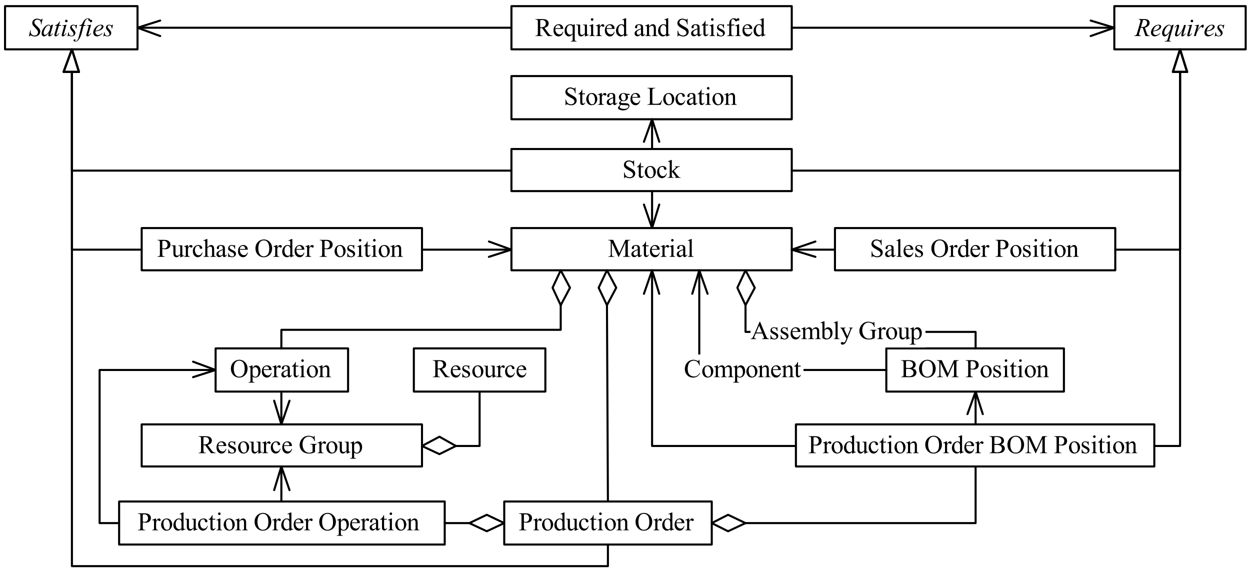


Figure 2: Simplified UML class diagram of our central production planning.

4 OUR PRODUCTION CREATED FOR THE COMPARISON OF SELF-ORGANIZATION AND CENTRAL PLANNING

In order to compare self-organization and central planning, we created a production (Figure 3). It produces two products. The products are wooden toys. The bill of material of each product is three levels deep and contains 30 materials. The routings of each product contain 20 operations. During the simulation of the production, new customer orders are injected into the production. The inter-arrival time of the orders is exponentially distributed. We choose an inter-arrival time which leads to a well-utilized production but does not cause an overload. The production runs for two weeks, 24 hours a day. The production will produce approximately 35 products per day if the processing times do not deviate. To examine the flexibility of the production, we vary the processing times of the operations. The processing times are distributed log-normally. The inter-arrival times, the processing times, the capacity of the machines as well as the duration of the simulation can be configured separately for each simulation.

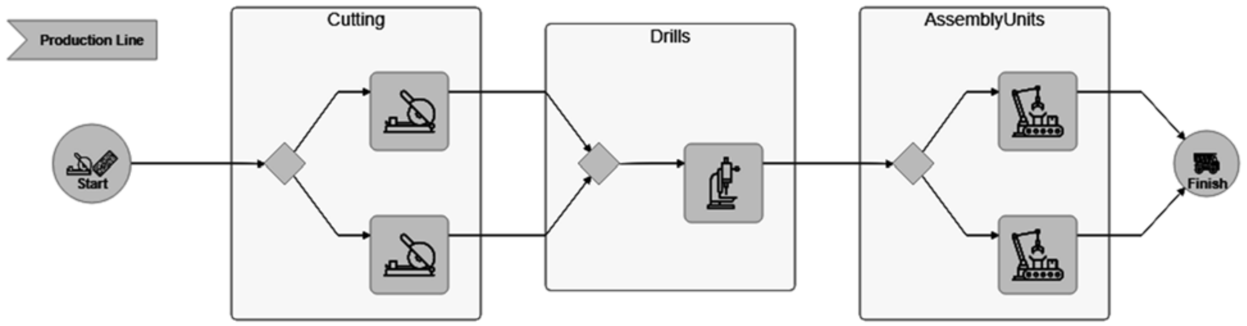


Figure 3: A production with two cutting machines, one drill, and two assembly units. The production allows different routings for one product. The drill is the bottleneck.

5 MULTI-AGENT-BASED SELF-ORGANIZATION

5.1 Concept of a Self-organizing Production

As defined in Section 2, a self-organizing production adapts to environmental changes and its components interact as equals. Simulations of self-organizing systems commonly apply multi-agent systems (Prokopenko 2008; Hewitt 2010). The "intelligent" resources and material of Industry 4.0 also suggest a multi-agent-based approach to a self-organizing production. Hence, we also applied a multi-agent-based approach.

We defined eight different types of agents (Figure 4) which can be further divided into transient and persistent agents. The lifetime of transient agents starts and ends with their task. The lifetime of persistent agents starts and ends with the lifetime of the production. Figure 4 shows that all agents only know their direct neighbors. They are not aware of any other agent within the production.

Next, we explain the functionality of the multi-agent-based self-organizing production as well as the interactions between the agents. Therefore, we follow the processing of a single sales order throughout the production step by step. For the sake of shortness, the multi-agent-based self-organizing production is abbreviated as MAS.

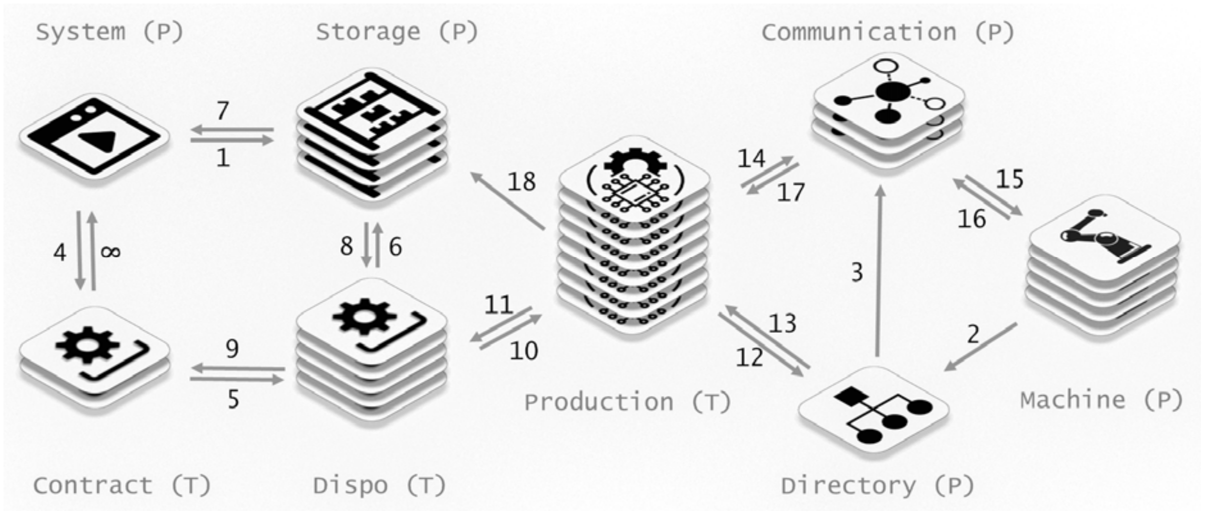


Figure 4: Concept of a multi-agent-based self-organizing production. Each symbol represents one type of agent. Each arrow is a communication path between two types of agents. The communications paths are referred to by their numbers (#number). The letters in brackets describe the type of the agents according to their lifetime: (P) .. persistent and (T) .. transient.

5.1.1 System Initialization

The MAS initializes the *system*, *machine*, and *directory* agents during startup and provides all required data, i.e., stock quantities to the *storage* agents (#1), machine capabilities to the *machine* agents, etc. The agent initialized first during startup is the *system* agent, followed by the *storage* agents and the *directory* agent. The *system* agent is responsible for all in- and outbound communication of the MAS with its environment. Second, the MAS creates all *machine* agents. The *machine* agents register at the *directory* agent and inform the *directory* agent about their capabilities (#2). The *directory* agent creates and assigns a *communication* agent to each machine group (#3). Afterwards the production is ready to take orders and to start.

5.1.2 Order Processing

If a new sales order arrives, the *system* agent will create a *contract* agent for this sales order (#4). The *contract* agent supervises the order fulfillment. Therefore, the *contract* agent creates one *dispo* agent (#5) for each product required to fulfill the order. The *dispo* agent represents a material in the real world: a digital twin. The *dispo* agent will ask the *storage* agent (#6) if there is an equivalent product in stock at the time the sales order requires it. The *storage* agent will check if there is a product in stock or needs to be purchased by the *system* agent (#7). If the *storage* agent's response (#8) to the request is positive, the *dispo* agent's task will be done and it will signal its parent agent, in this case the contract agent, that the task is finished (#9). The *contract* agent signals the *system* agent that the contract is fulfilled (#∞). If the *storage* agent's response is negative, the *dispo* agent will create a *production* agent for the required material (#10). Then, the *production* agent instantiates new *dispo* agents for each of its components (#11). The cycle will start again until the bill of material of the ordered product is fully processed.

5.1.3 Self-organizing Scheduling with Restricted Horizon

The task of each *production* agent is not only to order its components but also to organize all required operations to assemble its material. All *production* agents are competing for the earliest production timeslot on a machine. Therefore, the *production* agent requests one *communication* agent for each of its operations from the *directory* agent (#12). The *directory* agent returns the *communication* agent which is responsible for the operation specified at the request (#13).

Afterwards, the *production* agent sends a "request for proposal" to the *communication* agent assigned to the operation (#14). The *communication* agent forwards each incoming request to all of its *machine* agents (#15). Each *machine* agent returns a proposal containing a specific timeslot to the *communication agent* (#16). The calculation of the time slot is based on the slack time of the production order of the operation in the "request for proposal". After each machine has returned a proposal to the *communication* agent, the *communication* agent decides for the proposal with the earliest start time and sends an acknowledgement to the assigned *machine* agent (#15). Furthermore, the *communication* agent sends the assigned scheduling information to the requesting *production* agent (#17).

The *machine* agent organizes the acknowledged proposals as operations in its own queue. Operations that would not be processed for more than 60 minutes are rejected with the prompt to try again after 45 minutes. If the *machine* agent receives an acknowledgement of an operation (#15), this operation will be enqueued. Operations with a longer slack time than the previously enqueued operation are dropped from the queue, and the *machine* agent informs their *communication* agents to request new proposals for them (#16). This way, another *machine* agent could respond with a better proposal.

After the machine processes a material, its *machine* agent sends a completion message to the *communication* agent (#16). The *communication* agent forwards the message to the related *production* agent (#17). The *production* agent receives the message and sets the status of the subsequent operation to ready, to machines are only allowed to process ready operations. If there is no subsequent operation, the *production* agent will assume that the material is fully processed and forward the completion message to the related *dispo* agent (#11), which terminates itself because its task is done. The *production* agent also sends a message to the *storage* agent (#18), and terminates itself because its task is also done. The *storage* agent provides the currently produced material to the requesting *dispo* agent (#6) with the shortest slack time. After the *dispo* agent receives the required material, the production cycle starts over until the product is fully assembled.

5.2 Principle of Communication Between the Agents

As explained in Section 5.1, communication is one of the major issues of our self-organizing production. The implementation of the agents is based on the ACTOR pattern of Carl Hewitt (1973). All agents have exactly one message queue. This message queue is the only input to an agent. All messages are immutable and processed one by one by the target agent, first come, first served. The agent's behavior depends on what kind of message it receives and in what state the agent is. The message is typically defined by a quadruple:

1. Source agent
2. Target agent
3. Object to be processed (could be anything)
4. Method discriminator (indicates target agent's method to handle the object to be processed)

The ACTOR pattern makes the implementation of the agents very slim and readable. Another benefit of the ACTOR pattern is that one does not have to deal with concurrency. The simulation and all its components were implemented in C# .Net Core.

6 RESULTS OF THE EMPIRICAL COMPARISON OF A SELF-ORGANIZING AND A CENTRALLY PLANNED PRODUCTION

6.1 Parameters to Vary

The simulation applied the parameters specified in Table 1. For each combination of the parameters, we ran one simulation. The production became steady state after approximately 24 hours (Figure 5). We added another 24 hours before we started the measurement of the KPIs. A single simulation run lasted about 20 minutes. First, we choose a set of parameters that both productions behaved well with: deviation of processing times 20 %, lot size 1, planning scope 32 h, planning horizon 24 h, planning reset after 24 h, and average time to delivery 72 h. Applying these parameters, both productions had an average machine workload of 80 % and delivered 100 % of the sales orders in time. A major difference between the centrally planned and the self-organizing production was a significant deviation of lead times for both products. The centrally planned production had an average lead time of 4 h in contrast to the self-organizing production with an average lead time of 8 h. This results in longer lay time between the operations and higher stock for all components (Figure 6). We define the lay time as the time which elapses between the end of the production of a material and the start of its usage either in a production order or in a sales order.

Table 1: Parameters, their descriptions, and their values for the initial simulation run, [C]entrally planned and [S]elf-organizing production.

Parameter	Description	Type	Values	
Deviation	Time duration deviation for operations	C/S	0/20/40	%
Lot size	Lot size for assembly groups	C	1/5/10	Pcs.
Planning scope	Time limit up to which orders are taken into account	C	8/24/32	h
Planning reset	Time for rescheduling of all remaining and new orders	C	8/24	h
Order due	Average time from order placement to delivery	C/S	48/72	h

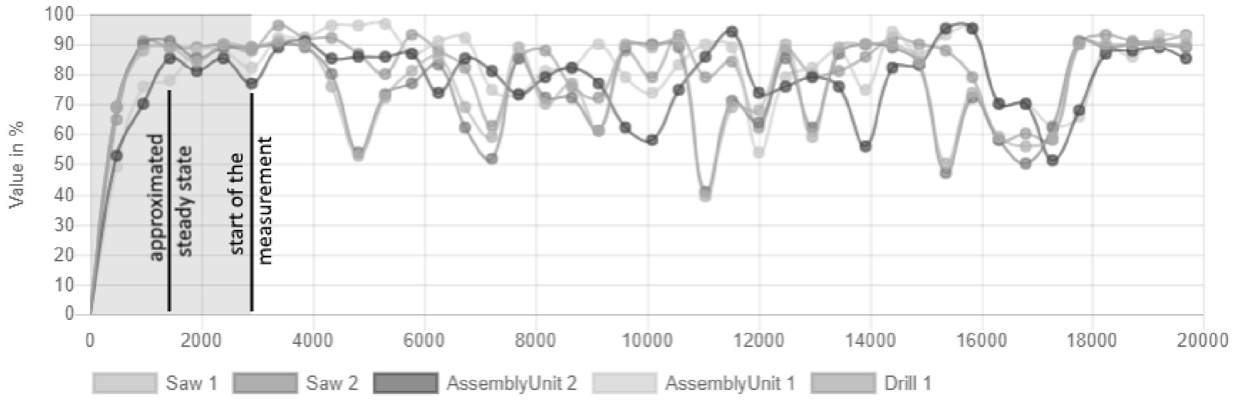


Figure 5: Exemplary workload of the five machines over time. The figure also shows when the production approximately reached its steady state and when we started the measurement of the KPIs.

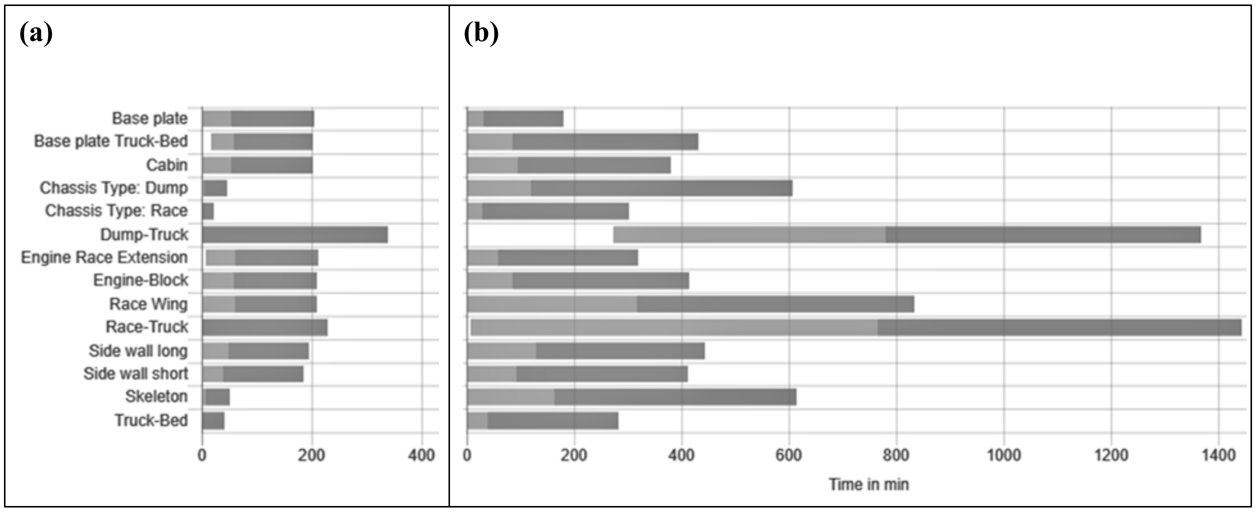
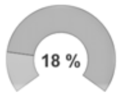
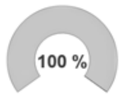


Figure 6: Lay time range for each component at 20 % deviation. Light grey equals the lower half and dark grey equals the upper half of the average lay time. The median lies at the transition from light to dark grey (a) centrally planned production; (b) self-organizing production.

6.2 Shortening the Time to Delivery

We shortened the time to delivery, to examine the performance of both productions under stress. For this purpose, we ran the simulation with 20 % deviation, lot size 1, planning scope and reset after 24 h, and in average 48 h to delivery. Again, both productions reached a similar average machine workload of 80 %. The lead time for the products increased moderately by about 20 %. But, the centrally planned production hit only 14 % of the product due time. In contrast, the self-organizing production was able to adapt and delivered 100 % of the products in time. Table 2 shows the results of the simulation and visualizes that the self-organizing production was not only able to produce everything in time, but also produced 41 more end products during the analyzed period of time. Timeliness is the ratio of the number of end products produced in time and the number of all end products produced.

Table 2: Product timeliness with 20 % deviation and average 48 hours to delivery for all orders.

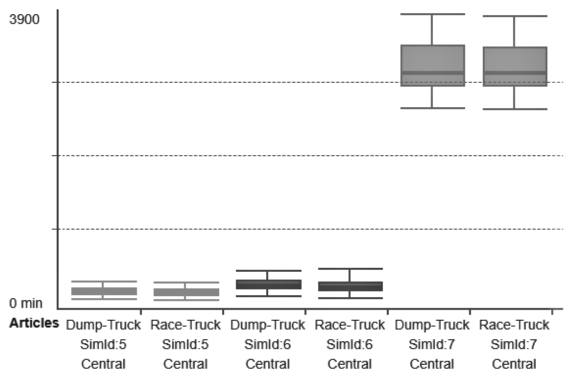
Parameter	centrally planned production		self-organizing production	
total timeliness in [%]				
Product	Dump-Truck	Race-Truck	Dump-Truck	Race-Truck
quantity in pieces	187	198	222	201
minimum/maximum deviation from delivery time in [min]	-320/385	-263/375	-1431/-274	-1444/-7

We tried to increase the timeliness of the centrally planned production. For this purpose, we tested fixed lot sizes with 5 and 10 pieces; but the timeliness increased only marginally. In further tests, we allowed the centrally planned production to plan anew every 8 hours but the timeliness did not improve either. We explain the rather poor performance of the centrally planned production as follows: New customer orders arrive while the production takes place. The self-organizing production reacts immediately to the new orders. In contrast, the centrally planned production does not consider the new orders until the next planning. This leads us to the conclusion that this centrally planned production is not able to cope with shorter times to delivery.

6.3 Varying the Deviation of Processing Times of Operations

Many disruptions and deviations may occur during the manufacturing of a product. Machines may break down, products may be rejected because of quality issues, personnel may drop out, operations may take more or less time than originally planned, etc. All these disruptions and deviations lead to deviating processing times of operations. We varied the duration of processing times of operations randomly by 20 % and 40 % to examine the influence of the variations on the KPIs. The self-organizing production was able to handle both. It handled 20 % deviation with ease, and even at 40 % deviation, it delivered 94 % of all incoming sales orders in time. The centrally planned production performed even worse than expected: At 20 % deviation, the centrally planned production reached 14 % timeliness only, and at 40 % deviation, it was not able to finish any product in time. Due to the high deviation of processing times and the rather long time between two consecutive central planning runs, the average lead times grew from 200 min (at 20 % deviation) up to 3000 min (at 40 % deviation). In contrast, the self-organizing production has a maximum of average lead time of 1100 min (Figure 7).

(a)



(b)

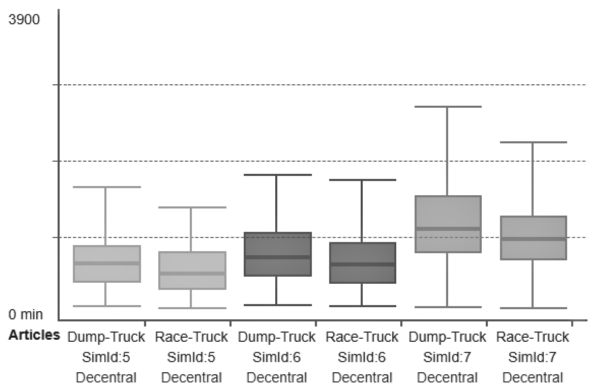


Figure 7: Lead time comparison from left to right with 0 %, 20 %, and 40 % deviation of processing times; (a) centrally planned production; (b) self-organizing production.

7 CONCLUSION AND OUTLOOK

The target of our research was to develop a concept for a self-organizing production and prove the concept empirically. We developed the concept and a prototype of a self-organizing production. We let the prototype compete with a widely used and well-established algorithm for central production planning. We simulated a self-organizing and a centrally planned production with the same presets of random data and were able to prove the viability of the self-organizing production under the given circumstances. The results show that a self-organizing production is extremely robust against disruptions and deviations during physical production. While the centrally planned production stuck with its originally created plan, the self-organizing production timely adapted to deviations in processing times. The tradeoff for this flexibility seems to be higher lay and lead times of the material. To cope with this tradeoff, we will investigate different priority rules, and how load-dependent or delayed production start influences lay and lead times. In order to achieve even better results, we will combine rough central planning and our self-organizing production. Furthermore, we will experiment with production data from different companies to gain more general results.

REFERENCES

Anderl, R., R. Dumitrescu, M. Eigner, C. Ganz, A. Huber, J. Michels, and P. Zhi. 2016. *Industrie 4.0 grenzenlos*. Heidelberg: Springer Vieweg.

Buckley, W. F. 1998. *Society – A Complex Adaptive System: Essays in Social Theory*. Amsterdam: Gordon and Breach.

Camazine, S., J.-L. Deneubourg, N. Franks, J. Sneyd, T. Guy, and B. Eric. 2003. *Self-Organization in Biological Systems*. Woodstock: Princton University Press.

Clark, W., and B. Farley. 1954. "Simulation of Self-organizing Systems by Digital Computer". *Transactions of the IRE Professional Group on Information Theory* 4(4):76-84.

Grieves, D. M. 2014. Digital Twin: Manufacturing Excellence through Virtual Factory Replication. Florida Institute of Technology – Center for Lifecycle and Innovation Management. Florida. http://innovate.fit.edu/plm/documents/doc_mgr/912/1411.0_Digital_Twin_White_Paper_Dr_Grieves.pdf, accessed 15.05.2018.

Hammami, Z., W. Mouelhi, and L. Said. 2015. "A Self-adaptive Neural-agent-based Decision Support System for Solving Dynamic Real Time Scheduling Problems". *10th International Conference on Intelligent Systems and Knowledge Engineering*. Nov. 24th – 27th, Tunis.

Hewitt, C. 1973. "A Universal Modular ACTOR Formalism for Artificial Intelligence". *IJCAI'73 Proceedings of the 3rd International Joint Conference on Artificial Intelligence*. Stanford: Morgan Kaufmann Publishers, 235-245.

Hewitt, C. 2010. *Actor Model of Computation: Scalable Robust Information Systems*. Cornell University Library, Ithaca. https://arxiv.org/ftp/arxiv/papers/1008/1008.1459.pdf , accessed 03.06.2018.

Kagermann, P. D., P. D. Wahlster, and D. J. Helbig. 2013. *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0*. Frankfurt/Main: Deutsche Akademie der Technikwissenschaften.

Kauffman, S. 1996. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford: Oxford University Press.

Klüver, C., J. Klüver, and S. Jörn. 2012. *Modellierung komplexer Prozesse durch naturanaloge Verfahren*. 2nd ed. Wiesbaden: Springer.

Köhler-Bußmeier, M. 2010. *Koordinierte Selbstorganisation und selbstorganisierte Koordination*. Hamburg: Universität Hamburg.

Lasi, H. 2014. „Industrie 4.0“. *Wirtschaftsinformatik* 4(4):261-264.

Prokopenko, M. 2008. *Advances in Applied Self-organizing Systems*. London: Springer.

Schuh, G. 2006. *Produktionsplanung und -steuerung*. Aachen: Springer.

Zäpfel, G. and R. Braune. 2005. *Moderne Heuristiken der Produktionsplanung*. München: Vahlen.

AUTHOR BIOGRAPHIES

TORSTEN MUNKELT is a full professor for business information systems and database systems at the Dresden University of Applied Sciences. He teaches business information systems, e-commerce, as well as basics of business informatics. His research focuses on the development and application of software to support and improve logistic processes. Torsten Munkelt introduced an ERP system worldwide. He was responisble for the long-term development of the ERP system and still serves companies as an IT consultant. In addition, he was head of development and product management for planning and monitoring systems in logistics. He studied Business Informatics at the Ilmenau Technical University and wrote his PhD thesis on Bayesian networks in production planning and -control. He is author of many papers in the context of business information systems. His interest in discret-event simulation in logistic environments leads back until the early 1990th. His e-mail address is Torsten.Munkelt@HTW-Dresden.De.

MARTIN KROCKERT is a research associate at the Dresden University of Applied Sciences. His research focuses on self-organizing systems. Before starting his career in research, he accumulated industrial experience from being a business process developer at Howden Group, where he was responsible for the introduction of the ERP-System APPlus and its adaption to the requirements of the company. In addition, he led several software projects concerning system integration and automating information workflows. He received his master's degree of Applied Information Systems at the Dresden University of Applied Sciences. His email address is Martin.Krockert@HTW-Dresden.De.