1st Conference on Production Systems and Logistics

Method to Automatically Create an Initial Layout for a Production System

Matthias Bartelt[1], Fabian Katzwinkel[1], Bernd Kuhlenkötter[1]
[1]*Ruhr University Bochum, Chair of Production Systems, Bochum, Germany*

Abstract

A well-proven means to automate processes are industrial robots. Nevertheless, there are still many processes that are not automated, especially in small and medium sized companies. A main reason is a missing automatism to create suitable solutions. To meet this challenge, the ROBOTOP research project tries to find a list of appropriate components and arrange them in a suitable layout. This paper addresses the second step and an algorithm is described that generates a suitable layout. Thereby, a main aim is to generate a layout that is plausible to the user. The problem relates to the facility layout problem and the proposed algorithm is also applicable to non-robotic related tasks. However, current methods do not yield to an appropriate, because due to the simplifications of the used models a manual effort is required when transforming the model to a 3D scene. The algorithms accepts a description of the process and identifies different kind of patterns. For each kind of pattern, layout rules are defined. With this, the model can be transformed to a plausible 3D setup.

Keywords

Facility layout problem; Design pattern; 3D layout

1. Introduction

The aim of the ROBOTOP research project is to develop a platform that enables non-experts to plan robot-based production systems. The user of the platform must not have any robotic knowledge. Instead, an online assistant acquires certain information about the user's needs. Using a case-based reasoning method, the assistant derives a process description as well as a list of production units. The method also provides a layout hint, but the user may add, change, or remove units. Additionally, the user may start without using the assistant. Hence, the layout starts with a process definition and a list of production units in general. For this, the problem of how to find an initial layout automatically must be solved.

This problem of arranging production units with respect to each other is referred in literature as *facility layout problem*. A survey of different approaches are given in [1–3]. Because most layout problems are complex and generally NP-hard [4], heuristic methods are used to find a solution. An important sub-problem in facility layout planning is the adjacency problem: which unit should be placed next to which other unit?

A common approach to solve this problem is the method according to Schmigalla [5]. The method uses a triangular mesh to arrange units and yields to an initial solution that is not necessarily an optimum. Other methods try to improve the resulting solution, e.g. by utilizing genetic algorithms [6]. Graph theory is used by another popular approach: the adjacency problem can be solved by constructing a maximum planar weighted graph [7]. Thereby, a unit is described as vertex, material handling from one unit to another is described as edge between the corresponding vertices, and the handling effort defines the weight. With this, the production can be formulated as a complete, weighted graph $G = \{V, E, w\}$ with vertex set V and edge set E. w assigns a weight to each edge. With this, a planar sub-graph with maximum weight has to be found. Further approaches are the quadratic assignment problem (cf. e.g. [8]), or slicing-tree-based methods (e.g. [9]).

These methods have usually a simplified view of the units: units are of the same size and the distance of the units is only treated partially within the handling effort. Additionally, transportation to and from a unit can be done in any direction. As a result, a transformation to the real physical environment is difficult. In [10], a software tool for layout modelling and optimization is used to take real sizes into account. Starting with a layout received by the Schmigalla method, the units are rearranged by using a material flow visualization and a distance-intensity graph. However, this rearrangement must be done manually and is not applicable to find a solution automatically.

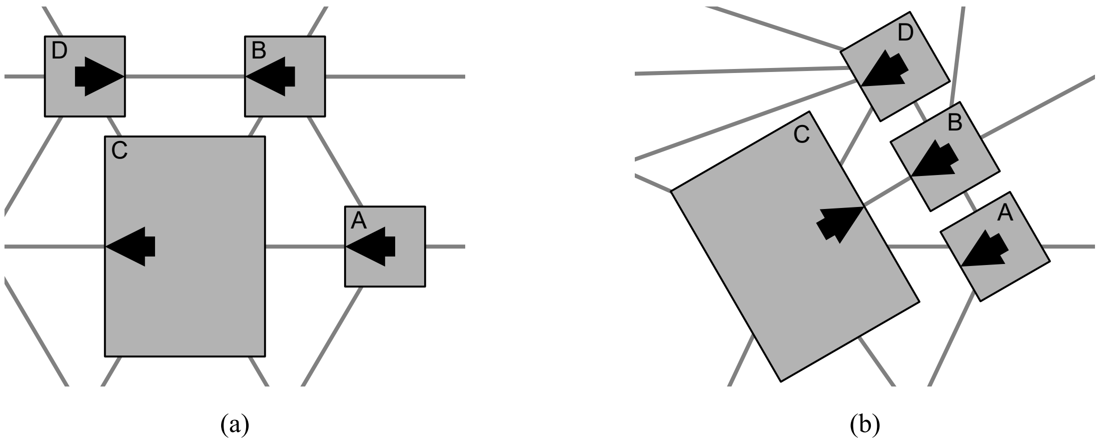

Figure 1: Exemplary layout with four units *A*, *B*, *C*, and *D* received by Schmigalla's method (a). The arrows drawn in the units indicate the interconnection points, where material must be handled through. Manually changing the layout with respect to the units' sizes and the interconnection points results in the optimized layout (b).

Furthermore, a main drawback is the simplification of interconnection points, where material is passed to and from the unit. The most units – especially units to automate processes – have certain interconnection points or areas. An industrial robot, for instance, can often not move to its back. Hence, there is an interconnection area in front of and at the side of the robot. Conveyor belts commonly have interconnection points at the beginning and at the end of the belt. CNC machines have a small interconnection area in front of the machine, and so on. Consider an example of four units as depicted in Figure 1a. It shows a layout as received by using Schmigalla's method. Obviously, the manual layout shown in Figure 1b is a better one, although the adjacencies are the same. Merely, the units are moved and rotated with respect to the interconnection points.

2. Manufacturing and Handling Units

In order to find an appropriate layout, it is essential to create an appropriate model of the units. Like most methods (cf. e.g. [11]), manufacturing units and handling units are handled differently. Consider a manufacturing unit m as $m = \{s, I, O\}$ with size s, ingoing interconnection set I, and outgoing interconnection set O. Any interconnection i can be both, part of set I and part of set O. Figure 2 depicts a simple example. The unit's area (8) is most important for the layout process, because other manufacturing units should not cover this area. The machine's required space (5) is completely embedded within the unit's area. However, it may be allowed for handling units to overlap with the unit's area but not with the machine's area. Besides these areas, the algorithm must consider more regions. For instance safety regions, regions to place human machine interfaces, or regions for maintenance. Some of these regions may overlap, some may not. An overview on different regions and how they may overlap is given in [12]. For now, the algorithm uses a simplified model that contains the unit's area and the interconnection points, as depicted in Figure 3.

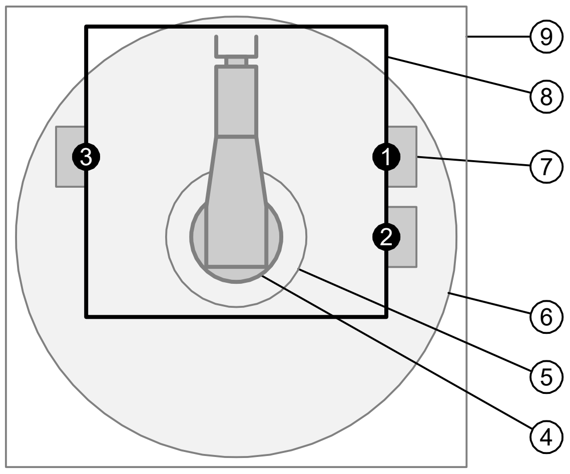

Figure 2: Sketch of the required space for a production unit for the example of an assembly unit. The unit's area (8) has interconnection points to receive material at (1) and (2) as well as a point to provide material at (3). Each point requires area to place the materials (7). In addition, the machine (4) inside the unit requires a certain area (5) and has a safety region (6). (9) visualizes the overall size.

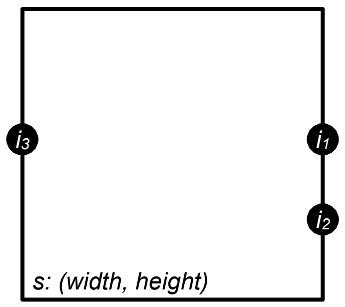

Figure 3: Manufacturing unit's model *m* with $I = \{i_1, i_2\}$ and $O = \{i_3\}$ of the example given in Figure 2.

A handling unit h connects two manufacturing units m_1 and m_2. In this context, a material buffer is also a manufacturing unit, albeit it manufactures nothing. With this, h can be defined as $h = \{m_j \rightarrow m_k\}$. Handling units do not have a specific size at this point of planning, because their size highly depends on the position of the connecting manufacturing systems.

3. Algorithm

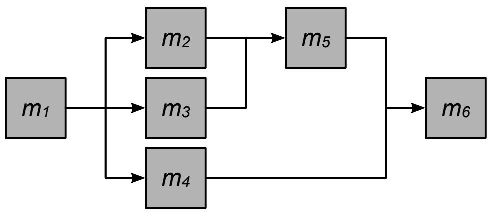

Figure 4: Example process with six manufacturing units.

Consider a manufacturing process with n manufacturing units m_i. An adjacency matrix A describes the connections by handling units. Figure 4 shows a simple example. The corresponding adjacency matrix is

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{1}$$

A one in a row describes an outgoing connection to the corresponding unit. A one in a column describes an incoming connection to the corresponding unit. With row j and column k, a handling unit $h = \{m_j \rightarrow m_k\}$ yields to $A_{j,k} = 1$. In order to find a suitable solution, a divide-and-conquer algorithm [13] is utilized. The algorithm operates on A and will search for units that define serial or parallel structures. Similar to an equivalent circuit (cf. e.g. [14]), a found structure replaces the corresponding units in A. The result is a new adjacency matrix A' with a reduced complexity.

3.1 Auxiliary Units

In order to identify groups, two auxiliary units are introduced: *distributor* and *merger* units. A distributor is a virtual unit with one incoming and two outgoing connection. Thus, material can be distributed from one to two other units. The merger unit works vice versa. It has two incoming connections, but only a single outgoing one. Even though both type of units are used for handling, they are treated as (virtual) manufacturing units. The main purpose of the auxiliary units is to reduce the number of incoming and outgoing connections, which simplifies the structure identification.

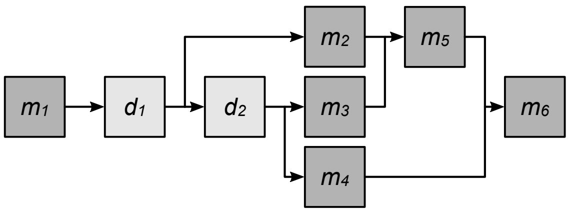

Figure 5: Example process with two added distributor units *d*.

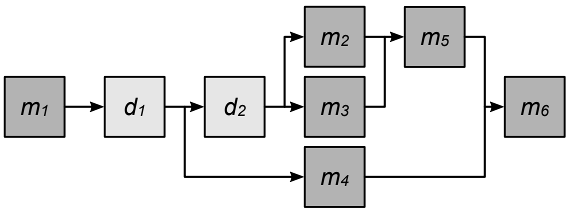

Figure 6: Alternative option to add the two distributor units.

With this auxiliary units, the example given in Figure 4 can be modified by adding distributor units between m_1 and m_2 to m_4. As shown in Figure 5, the number of connections of m_1 reduces from three to one. However, there is not only a single way to add the auxiliary units. Figure 6 depicts another option to add the two distributor units. Obviously, the latter one yields to a structure that can be simplified easily.

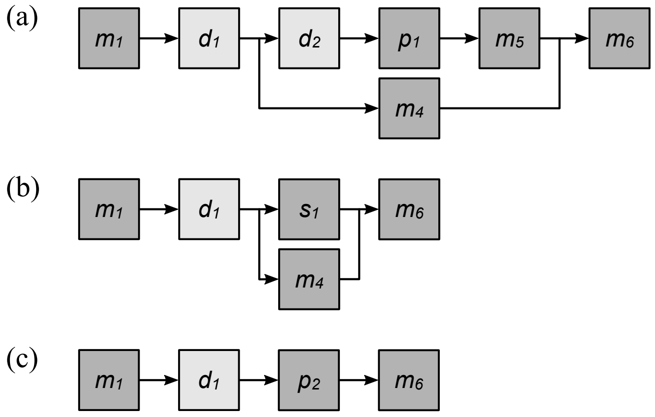

Figure 7: Simplification of the structure by identifying serial structures *s* or parallel structures *p*.

As shown in Figure 7, m_2 and m_3 build a parallel structure and can be replaced by $p_1 = \{m_2, m_3\}$. Next, there is a serial structure with d_2, p_1, and m_5 that is replaced by $s_1 = \{d_2, p_1, m_5\}$. s_1 and m_4 are now parallel and are replaced by $p_2 = \{s_1, m_4\}$. Finally, there is only a serial structure left.

After adding auxiliary units, the algorithm operates on the adjacency matrix to find serial and parallel structures. In order to find a serial structure, the algorithm search for two units m_j and m_k that are linked with a single connection. Such a pair is found, if the conditions (2) to (5) are true:

$$j \neq k, \tag{2}$$

$$A_{j,k} = 1, \tag{3}$$

$A_{j,x} = 0 \text{ } with \text{ } x = 1 \dots n, x \neq k$, and (4)

$A_{x,k} = 0 \text{ } with \text{ } x = 1 \dots n, x \neq j.$ (5)

Next the algorithm searches for a unit m_l that builds a serial structure with m_j or m_k. If found, there is serial structure with three units and the algorithm searches for another unit that corresponds to the structure, and so on. A set of units that have the same single origin for an incoming connection and the same single target for an outgoing connection defines a parallel structure. Hence, there is a parallel structure between m_j and m_k if

$A_{j,x} = A_{x,k} \text{ } with \text{ } x = 1 \dots n,$ (6)

and furthermore $\forall \text{ l } \in \{\text{x} \mid \text{A}_{\text{j,x}} = 1\}$ it is

$A_{l,x} = 0 \text{ } with \text{ } x = 1 \dots n, x \neq k$, and (7)

$A_{x,l} = 0 \text{ } with \text{ } x = 1 \dots n, x \neq j.$ (8)

3.2 Loops

There are two types of loops that may occur in a structure: bypassing and rework of materials. A bypassing structure is sketched in Figure 8a. Material is partially handled from one unit m_j directly to another unit m_k, while other parts are treated by additional units between. The solution is to add a virtual unit into the bypassing, as depicted in Figure 8b. With this, the algorithm can treat the bypassing structure similar to the parallel structure as described above.

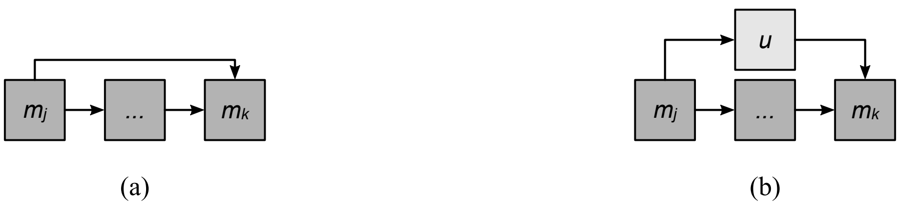

Figure 8: Bypassing structure (a) and the transformation to a parallel structure (b) by adding a virtual unit *u*.

The other kind of loop is a rework of materials. If, for instance, a manufacturing unit polishes a product and a subsequent inspection unit verifies the process, the inspection unit may identify the need for a rework. Another example is the use of a carrier system, because at some point in the manufacturing process, the carriers return to the beginning. Figure 9a visualizes this structure, where a material is handled backwards from m_k to m_j. The solution to resolve this loop is a substitution of the units involved from m_j to m_k by a single one, as depicted in Figure 9b. If there are non-serial connection between m_j and m_k, they have to be reduced first.

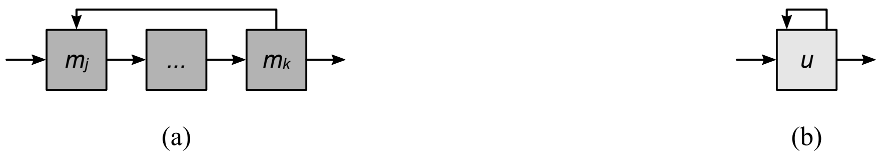

Figure 9: Rework structure (a) and the transformation to a single unit (b).

3.3 Cross-Connections

Consider a structure as sketched in Figure 10. For such kind of structures, the algorithm can identify neither a serial nor a parallel structure. The algorithm must handle such structures in a special way. For this, a further substitution unit is defined that substitutes the complete cross-connection structure consisting of m_1, m_2, m_3, and m_4. The algorithm treats the remaining structure again as described above.

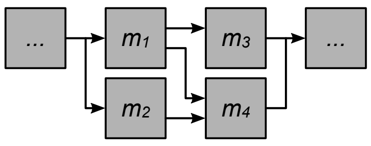

Figure 10: Example of a cross-connection.

4. Deriving a Layout

The previous section describes how to identify different kind of structures, e.g. serial or parallel ones. In order to find an appropriate layout, a design patterns is defined for each possible structure. By applying these patterns, the algorithm creates the final layout.

As discussed above, a set of units linked by a single connection defines a serial structure. The interconnection points of the units may be located on different sides of the unit, i.e. on opposite sides, on adjacent sides, or on the same side. Figure 11 depicts design patterns to arrange these types.

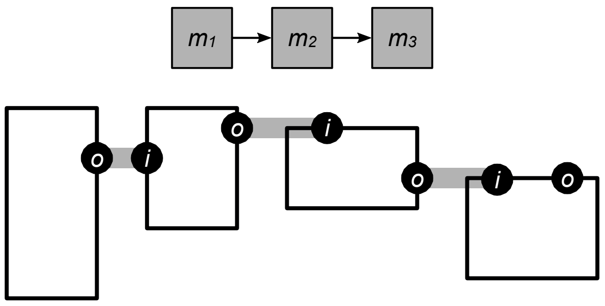

Figure 11: Design pattern for serial structures. The grey regions indicates handling between outgoing connections *o* and incoming connections *i*.

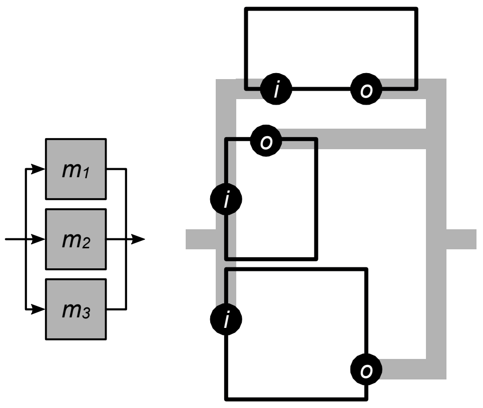

Figure 12: Design pattern for parallel structures. The grey regions indicates handling between outgoing connections *o* and incoming connections *i*.

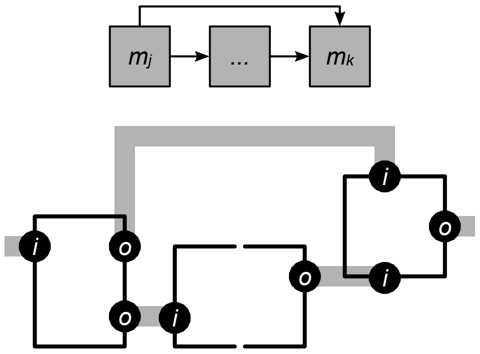

Figure 13: Design pattern for bypassing structures. The grey regions indicates handling between outgoing connections *o* and incoming connections *i*.

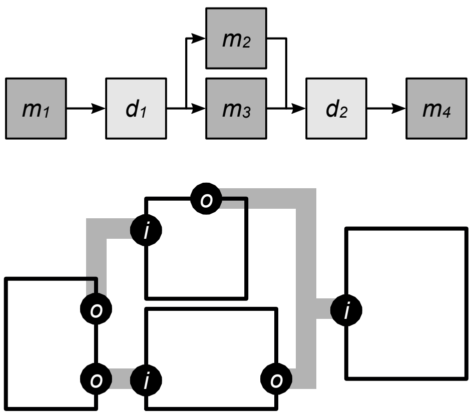

Figure 14: Handling of auxiliary units. The grey regions indicates handling between outgoing connections *o* and incoming connections *i*.

Similar to serial structure, a patterns for parallel structures is defined. Figure 12 sketches the required patterns for units with connections on opposite sides, adjacent sides, and the same side. Obviously, the layout is not optimal, as the complexity for the handling-system can be reduced. However, as stated above, the main aim is to find a plausible layout, not an optimal.

Both bypassing and rework structures can be treated similar. Figure 13 shows the design pattern for a bypassing structure. The pattern for a rework structure looks similar, except that incoming and outgoing connections are swapped.

The main purpose of the auxiliary units is to identify serial and parallel structures. When designing the layout, they can be ignored usually. However, it may occur that the described pattern yield to an overlap of units. In such cases, the algorithm moves the units until the overlap is eliminated, as depicted in Figure 14.

5. Conclusion

A main problem with existing layout methods is the idealization of material handling. As stated, it is important to generate a layout that is plausible for the user automatically. The presented approach introduces auxiliary units that help to identify especially serial and parallel structures.

For the different structures, design patterns are defined. Each design pattern can be used to replace an identified structure with a suitable layout. After the top-down process of identifying structures, the described algorithm constructs the complete layout by combining the different patterns.

Although the presented method is able to create layouts, there are further improvements possible. By now, the interconnections are considered as a single point. Nevertheless, units usually have an interconnection area. Furthermore, the connections are unweighted. With an appropriate weight, the resulting layout may change. Next, the space between the units must be optimized with respect to e.g. safety and maintenance regions. Finally, the handling paths must be optimized by allowing them to overlap the units' areas.

The main aim was to derive a plausible layout. Thus, the received layout is not optimized. In future work, the algorithm must be improved in order to optimize the current results. Nevertheless, the method yields to a suitable layout that considers interconnection points of the manufacturing units.

Acknowledgements

The research project ROBOTOP is funded by the Federal Ministry of Economic Affairs and Energy (BMWi). It is part of the technology program “PAICE Digitale Technologien für die Wirtschaft” and is guided be the DLR Project Management Agency “Information Technologies / Electromobility”, Cologne.

References

[1] Drira, A., Pierreval, H., Hajri-Gabouj, S., 2007. Facility layout problems: A survey. Annual Reviews in Control 31 (2), 255–267.

[2] Hosseini-Nasab, H., Fereidouni, S., Fatemi Ghomi, S.M.T., Fakhrzad, M.B., 2018. Classification of facility layout problems: A review study. Int J Adv Manuf Technol 94 (1-4), 957–977.

[3] Singh, S.P., Sharma, R.R.K., 2006. A review of different approaches to the facility layout problems. Int J Adv Manuf Technol 30 (5-6), 425–433.

[4] Garey, M.R., Johnson, D.S., 2007. Computers and intractability: A guide to the theory of NP-completeness, 26th ed. Freeman, New York u.a, 338 pp.

[5] Schmigalla, H., 1970. Methoden zur optimalen Maschinenanordnung. Verlag Technik, VEB, Berlin.

[6] Ficko, M., Palcic, I., 2013. Designing a Layout Using the Modified Triangle Method, and Genetic Algorithms. Int. j. simul. model. 12 (4), 237–251.

[7] Wäscher, G., Merker, J., 1997. A comparative evaluation of heuristics for the adjacency problem in facility layout planning. International Journal of Production Research 35 (2), 447–466.

[8] Burkard, R.E., Çela, E., Pardalos, P.M., Pitsoulis, L.S., 1998. The Quadratic Assignment Problem, in: Du, D.-Z., Pardalos, P.M. (Eds.), Handbook of Combinatorial Optimization: Volume1-3. Springer US, Boston, MA, pp. 1713–1809.

[9] Scholz, D., Petrick, A., Domschke, W., 2009. STaTS: A Slicing Tree and Tabu Search based heuristic for the unequal area facility layout problem. European Journal of Operational Research 197 (1), 166–178.

[10] Banduka, N., Mladineo, M., Eric, M., 2017. Designing a Layout Using Schmigalla Method Combined with Software Tool visTABLE. Int. j. simul. model. 16 (3), 375–385.

[11] Li, J., Meerkov, S.M., 2009. Production Systems Engineering. Springer US, Boston, MA.

[12] Schenk, M., Wirth, S., Müller, E., 2014. Fabrikplanung und Fabrikbetrieb. Springer Berlin Heidelberg, Berlin, Heidelberg.

[13] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. Introduction to algorithms, 3rd ed. MIT Press, Cambridge, Mass., 1292 pp.

[14] Johnson, D.H., 2003. Origins of the equivalent circuit concept: The voltage-source equivalent. Proc. IEEE 91 (4), 636–640.

Biography

Matthias Bartelt (*1978) obtained his PhD degree from Ruhr University Bochum in 2017. Currently, he is post-doctoral fellow at the Chair of Production Systems. His specific focus of research interest is on the application of cyber-physical systems, virtual commissioning as well as simulation of production systems.

Fabian Katzwinkel (*1990) worked at the Chair of Production Systems as a student assistant. While working at the Chair, he started his Master thesis with the title *Automatic arrangement of automating components*. In 2018, he received his Master degree.

Bernd Kuhlenkötter (*1971) was responsible for the product management and technology at ABB Robotics Germany until 2009. In 2009, he took over the Professorship for "Industrial Robotics and Production Automation" at the Technical University of Dortmund. Since 2015, he holds the professorship of "Production Systems" at Ruhr University Bochum. In parallel, he is the vice president of the Academic Society for Assembly, Handling and Industrial Robotics (MHI e.V.).