Journal of Intelligent Manufacturing (2022) 33:537–554
https://doi.org/10.1007/s10845-021-01853-5

Simulation-based layout optimization for multi-station assembly lines

Daria Leiber[1] · David Eickholt[1] · Anh-Tu Vuong[1] · Gunther Reinhart[1]

Received: 14 May 2021 / Accepted: 22 September 2021 / Published online: 19 November 2021

Abstract
This article presents a novel approach for the automated 3D-layout planning of multi-station assembly lines. The planning method is based on a comprehensive model of the used production resources, including their geometry, kinematic properties, and general characteristics. Different resource types can be included in the planning system. A genetic algorithm generates and optimizes possible layouts for a line. The optimization aims to minimize the line's area and the costs for assembling the line while simultaneously optimizing the resources' positioning to perform their tasks. The line's cycle time is considered as a boundary condition. For the evaluation of different layout alternatives, a multi-body simulation is performed. A parameter study is used to set the algorithm's parameters. Afterward, the algorithm is applied to three increasingly complex examples to validate and evaluate its functionality. The approach is promising for industrial applications as it allows the integration of various resource types and individualization of the optimization function.

Keywords Layout planning · Assembly line design · Layout optimization · Genetic algorithm · Production resource modelling

Introduction

Today automated assembly lines are commonly used to manufacture products. The spatial positioning of production resources has a crucial influence on the planned system's quality and is a significant step in designing a new line (Bullinger, 1986). The ideal layout is highly dependent on the used resources and their characteristics (like kinematics), especially if they need to access the assembled product to perform their task. For example, an assembly station might contain multiple robots that have to access the product sequentially and a camera to visually monitor the performed processes simultaneously. Another challenging factor is the flow of material and the transport of the product from one station to the next end which needs to be ensured. Traditional planning methods rely on the planner's expertise and do not use objective criteria to optimize the layout. Manually arranging the resources is also a time-consuming process, whose automation offers the potential to decrease the necessary manual effort during the planning process (Michniewicz et al., 2017).

There are numerous approaches for the automated planning and optimization of the site layout, i.e., the placement of individual areas and facilities on the factory floor. This problem is known as the Facility Layout Problem (FLP) and the aim is to determine the most efficient layout. Studies addressing the FLP consider the creation of a 2D layout, where the elements to be arranged are represented by rectangles or polygons (Drira et al., 2007, p. 255; Hosseini-Nasab et al., 2018, pp. 957–958). For the detailed design of an assembly line, the level of abstraction used in FLP is too high, as the individual resources within the line's stations must be regarded and positioned. Automating this planning step is one of the biggest challenges in the automation of assembly system planning (Michalos et al., 2015, p. 86). Various researchers have addressed the automated positioning of production resources. Substantial work has been done regarding the positioning of industrial robots and sensors (Pellegrinelli et al., 2017; Zeng et al., 2020). However, automatically generating a viable three-dimensional layout of a production line remains a challenge (Michalos et al., 2015, p. 86).

In this article, we present a method for automatically generating and optimizing three-dimensional layouts for assembly lines. It is based on a resource library with digital models of the available production resources. These models

✉ Daria Leiber
daria.leiber@tum.de

[1] Department of Mechanical Engineering, Institute for Machine Tools and Industrial Management, Technical University of Munich, Boltzmannstr. 15, 85748 Garching b. München, Germany

contain the resources' relevant properties, including economic characteristics like their price and technical features like a description of their workspace.

The article is structured as follows: after this introduction, in Sect. 2, related literature is presented. Then in Sect. 3, the optimization problem is formulated, and the planning system's components are introduced. Section 4 describes the resource library and the modeling of the resources. Section 5 deals with the algorithm to optimize the layout. Next, in Sect. 6, a parameter study is conducted to determine an appropriate configuration of the algorithm's parameters. In Sect. 7, the presented concept is then applied to three examples to validate and evaluate its functionality. Finally, Sect. 8 formulates a conclusion and gives an outlook on further topics of research.

Related works

Different researchers have focused on optimizing the spatial layout of production resources and production areas in a production system in the past. In the research domain of Facility Layout Problem (FLP), various target criteria, such as material flow or expected reconfiguration costs, are optimized. Boundary conditions such as the shape and size of the available production areas or the type of handling system used are also considered. (Drira et al., 2007, p. 255; Hosseini-Nasab et al., 2018, pp. 957–958). Resent contributions to the field include the work of Gülşen et al. (2019), Mohamadi et al., (2019a, 2019b), Liu et al. (2021), and Besbes et al. (2021).

The existing FLP research can be classified according to different aspects of the presented problems. Hosseini-Nasab et al. (2018) present a classification scheme that considers layout evolution (static/dynamic), workshop characteristics, problem formulation and solution approach. Pérez-Gosende et al. (2021) extend this categorization by also including the problem type (greenfield/re-layout) and the planning phase (block layout/detailed layout). They also sub-divide the category of workshop characteristic into facility characteristics and materials handling system configuration.

In the taxonomy of Pérez-Gosende et al. (2021) the problem presented in this article would be static (meaning that the generated layout does not change over the lifetime of the manufacturing system), greenfield, detail layout, single facility, single floor with 3D space consideration. However, these characteristics are highly unusual for a FLP. In fact, the developed approach does not correspond to the classical definition of a FLP as it focuses on individual resources and their 3-dimensional positioning including the simulation of their kinematics.

Early work on a similar problem was done by Woenckhaus (1994) and Woenckhaus and Milberg (1993) who present a system for 3D layout optimization of assembly cells with different types of resources. In these publications, resources like robots, sensors, and humans, are positioned. Passive, structural elements are also considered. The initial layout is created by the user and then automatically evaluated and iteratively optimized. For the layout evaluation, a 3D simulation environment is used.

Roßgoderer et al. (1995) further develop this approach by also generating the initial layout automatically. For this purpose, they define functional surfaces and model the relationships between the cell components in a structural layout. Reinhart and Roßgoderer (1998) integrate automated robot path planning into the planning system and plan human movements interactively in a Virtual Reality (VR) environment. Roßgoderer (2002) further improves the system and presents a three-step planning method for 3D layout planning of hybrid assembly systems. The resources are manually linked to define the structure of an assembly station, and the assembly process to be performed is specified manually. A 3D simulation model and the necessary robot movements are generated automatically. Manual motion sequences are entered interactively using input devices from the field of VR. By modifying the remaining degrees of freedom, the layout is optimized.

Leiber et al. (2019) and Michniewicz et al. (2017) present concepts for comprehensive planning systems. Besides the layout generation, these systems include selecting production resources and balancing the line. However, the systems' implementation and evaluation are not described. Leiber and Reinhart (2020) build on the concept and present, implement, and evaluate a system that integrates assembly line balancing and 3D layout planning into one bi-level optimization problem. They focus on the nested optimization and simplify the problem by representing production resources through cubes.

However, many works dealing with the positioning of resources that focus either on the positioning of industrial robots or the optimal positioning of optical sensors can be found. The following two sections provide an overview of these two areas of research.

Positioning of robots

A constructive approach for the automated design of robot cells is presented by Lueth (1992). The components are placed one after the other in the cell, testing and ensuring the layout's validity at each step. The approach considers collision freedom and the accessibility of the workpiece as constraints. The optimization criterion is the length of the robot trajectories necessary to perform the defined tasks. Barral et al. (2001) also use a constructive approach for the automated design of an assembly cell layout and combine it with a simulated annealing algorithm. Machines are arranged one after another in the vicinity of a fixed robot. The objective of the optimization is the minimization of the cycle time.

Barral (2002) and Barral (2003) also utilize this approach. Another method to position multiple workstations in an automated cell, which are approached by an industrial robot in a given order, is introduced by Zhang et al. (2013). In this work, a segment of the robot workspace is defined as particularly advantageous. The optimization aims to ensure that the manipulated workpieces are in or as close as possible to this area. Tubaileh (2014) optimizes the positioning of machines in a robotic cell as well. The goal is to minimize the time the robot needs to reach them in a given order. The robot's kinematic properties are taken into account, and the planning result is validated using 3D simulation. Sharma et al. (2017) address the problem, by creating point cloud models of robots and other objects in the cell based on CAD data. Subsequently, the objects' positioning and orientation in a robot cell are optimized using a simulated annealing algorithm. The optimization criterion is the minimum movement of the robot axes.

Other works specifically address planning scenarios with multiple robots. Papakostas et al. (2011) deal with the planning of cooperating robots in the automotive industry. Different alternatives regarding positioning and path planning are generated and evaluated based on a manually created initial solution. Papakostas et al. (2014) extend the approach by including the selection of resources based on rules and previously used solutions. Pellegrinelli et al. (2014) focus on the design of multi-robot cells specifically for spot welding applications. They use a multistage process that includes the selection, positioning, and path planning of the robots. First, the individual robots are considered and then interactions between the robots are taken into account. Pellegrinelli et al. (2015) and Pellegrinelli et al. (2017) further develop this approach and validate it using various application examples. Suemitsu et al. (2016) develop another method for planning cells with more than one robot. Their approach is based on the work of Izui et al. (2013) and Lim et al. (2016). Izui et al. (2013) present a multi-criteria optimization problem for the automated layout planning of robot cells. The optimization considers the minimization of the required area of the layout and the cycle time, as well as the maximization of the robot's manipulability to prevent it from approaching a singular axis position. A genetic algorithm is used to solve the problem. Lim et al. (2016) then compare different heuristic algorithms to solve the optimization problem. An additional planning criterion for the layout planning is regarded by Wang et al. (2020) who consider the aspect of safety while optimizing logistical cost and layout area of the planned assembly line.

Positioning of sensors

Many of the existing methods for the positioning of sensors have the determination of a suitable set of sensor poses to fulfill a specific task as their goal (Chen et al., 2008, p. 39). This is necessary because the measurement volume of sensors often only covers part of the target object and because one view is often not enough to create a 3-dimensional model of a physical body or capture all relevant features. In the industrial context, a digital model of the measured item usually already exists and the goal is to inspect specific attributes in the context of quality control (Chen et al., 2008, p. 2). This scenario is addressed by Trucco et al. (1997) who present a planning system to calculate the optimal sensor poses for defined inspection tasks. The optical sensor used is specified beforehand, and a CAD model of the inspected object is also available. Gronle and Osten (2016) integrate the planning of measurement poses into a more comprehensive assistance system for the inspection planning of surfaces that also includes selecting a sensor and defining the sensor parameters. Ellefsen et al. (2017) on the other hand present a multi-objective optimization using an evolutionary algorithm to determine a robot path to inspect a known structure. The example application is an underwater robot for inspecting oil rigs or offshore wind farms. The optimization objectives are to minimize energy consumption, maximize coverage, and minimize collisions. A problem similar to the inspection of an object or feature is addressed by Rosman et al. (2018) who are concerned with the pose planning of sensors to monitor assembly tasks in the context of cooperating robot teams.

Another area of application of sensor pose planning is the digitalization of items when only their approximate dimensions are known. Since no information about the object is available before starting the digitalization process, the poses must be generated online during the process, based on the information already available (Karaszewski et al., 2016, p. 320). Karaszewski et al. (2016) evaluate and compare different algorithms for pose planning for this application. A current example of such an algorithm is the work of Monica and Aleotti (2018). Their approach is based on the representation of the environment by surface elements. The method shows a better performance than systems based on a representation by volume elements, with a comparable quality of the created models. Besides these use cases for inspecting or digitizing objects, Zeng et al. (2020) mention the digitization of a scene that is not limited to a particular volume (in the context of mobile robots) and the recognition, classification, or localization of objects as scenarios in which sensor poses have to be planned.

Need for research

Although the positioning of production resources has been an active field of research for many years, the automated generation of assembly line layouts can still be considered an unsolved challenge. Most authors focus on positioning one type of resource (industrial robot/optical sensor) for a specific task. Some authors consider multiple resource types but are restricted to single assembly cells (Roßgoderer et al., 1995) or simplify the layout planning problem substantially. Therefore, a need for a layout planning system that considers multiple types of resources and assembly lines with multiple stations exists.

Problem definition and system overview

In this article, we present a solution approach for layout planning of multi-station assembly lines. The proposed approach positions given resources in an assembly line based on a multi-criteria optimization with an objective function (1) that is dependent on the resource poses $pose_{res}$, which are determined during the optimization. The goal is to minimize costs over the line's lifetime.

$$minf\left(pose_{res}\right) = \alpha \cdot \sum_{res} c_{mounting,res} + \beta \cdot A_{line} + \gamma \cdot \sum_{res} \left(1 - PQ_{res}\right) \quad (1)$$

The first term of the objective function is the sum of the resources' mounting costs $c_{mounting,res}$ representing the costs of assembling the line itself. The resources' positioning influences these costs because some positioning options (like overhead mounting of a robot) are more complex and might require expensive support structures. The second term includes the line's area A_{line} since the occupied factory area is associated with significant costs and therefore should not be larger than necessary. The last term is based on the resources' pose quality PQ_{res}, which quantifies the quality of the resources' positioning from a technical standpoint. The definition of PQ_{res} depends on the type of resource used (see Sect. 0). The coefficients α, β und γ are included to allow the user of the optimization system to weigh the terms according to their individual preferences. They might depend on the line's location and the associated costs of space or the line's expected lifetime. The latter is important because while the cost of assembling the line only occurs once, the lines' area and the resources pose quality have a continuous effect over the line's lifetime.

During the layout optimization, the following constraints must be respected:

1. The absence of any overlap between the resources is necessary for a feasible layout since several resources cannot occupy the same space.
2. The reachability of the relevant area of the product (region of interest, ROI) by the station's resources is another precondition for the layout's functionality. If a resource cannot reach the ROI, it cannot perform the necessary manufacturing process.
3. Adherence to the required cycle time for the line has to be ensured when designing a layout.

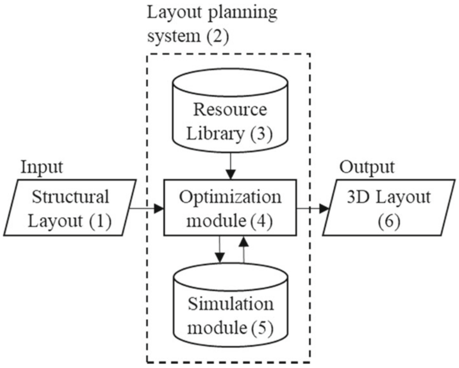


Fig. 1 The proposed system for the layout planning of multi-station assembly lines

To solve the presented problem and automatically create a viable assembly line layout, we propose a planning system (Fig. 1).

Input information (1) is the structural information of the line that contains a list of stations and their resources. For each resource, it is also known where on the product the relevant ROI is located. The layout planning system itself (2) consists of three modules: the resource library (3) that contains models of the used resources, the optimization module (4) that creates and optimizes the layout of the line, and the simulation module (5) that is needed to evaluate the quality of generated layout candidates. Finally, the optimization result is a virtual model of the line's 3D layout (6).

Resource library and modeling of resources

The proposed resource model contains all the characteristics of the resources necessary for the automated layout design. The optimization module then uses the models in the resource library to create different potential layout solutions that are evaluated using the simulation module. Figure 2 gives an overview of the four main aspects of production resources included in the model. First, geometry is a relevant property for any object that is to be placed in space. It is crucial in ensuring that the first constraint is respected, and no collisions occur between resources. Second, for moving resources, a kinematic model is needed for the simulative validation. Third, a model of the resources' workspace is required to ensure that the constraint of reachability is satisfied. Fourth, information about how the resources themselves can be assembled must be included in the model (Leiber et al., 2019).

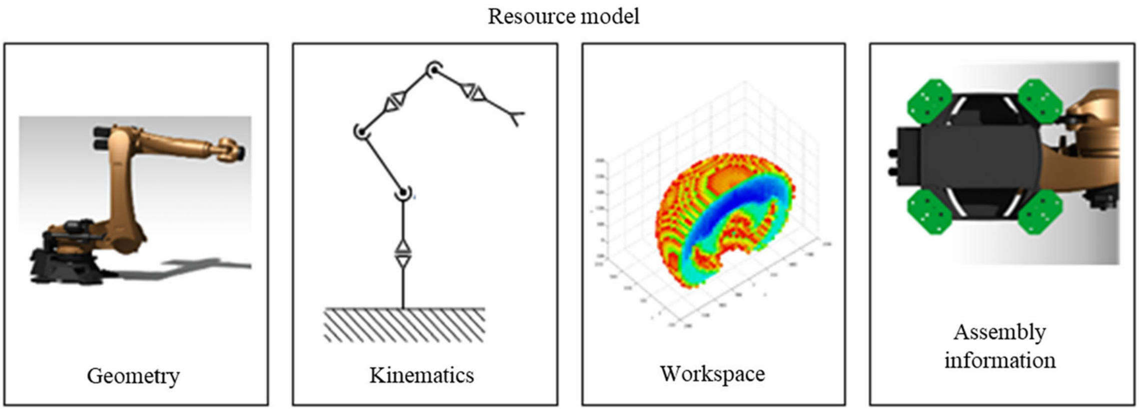


Fig. 2 Structure of the resource model

Geometry and kinematics

A resource's geometry represents its physical form, while the resource's kinematics describes how different parts of its 3D model are connected. Today, CAD tools and geometric 3D models are commonly used in product development and the design of assembly lines (Weidemann & Drath, 2010, p. 30, 2010, pp. 18–19). Many suppliers of production resources provide 3D models of their products to their customers. For moving resources like industrial robots, these models often include information about the resource's kinematics. To ensure these models' usability independent of specific software applications, standardized data formats (e.g., STEP, COLLADA) for geometry or kinematics are used.

In our proposed resource model, the geometry and, if applicable, the resource's kinematics are essential. Without these, it would be impossible to determine the quality or even just the feasibility of a layout. Therefore, every resource model in the resource library includes the geometry of the modeled resource and, for moving resources, also its kinematics.

Modeling of workspaces

The workspace of a resource is the area that the resource can reach and where it can perform the processes it provides. The accessibility of the product and ROI by the respective resources is a requirement for a feasible layout. Therefore, the ROI must be located within the workspace of a resource. However, among the theoretically possible positions, there are better and worse ones. Thus, the relative position between product and process executing resources is a factor that influences the quality of an assembly line layout.

Which poses in the workspace of a resource are good or bad depends on the resource. To consider the quality of positioning in the workspace, the resource model needs to include information about which positioning is advantageous and which is disadvantageous. Therefore, an evaluation function $f(p_{rel})$ that calculates the pose quality PQ for a relative pose p_{rel} between resource and ROI is part of the resource model.

Depending on the type of resource (e.g., robot, camera) and the specific resource variant (e.g., kinematics type of a robot), the evaluation function may consider different factors. The characteristics of the resource determine the structure of the evaluation function. To allow a comparison between different resources, the quality evaluation is normalized to PQ values between 0 and 1. A value of 1 represents an ideal pose, while 0 means that a relative pose is not suitable. A value of 0 can either be because the ROI is not entirely in the workspace or because the orientation between resource and ROI is not practical.

Deciding which criteria are relevant for evaluating the workspace and how they are combined in the evaluation function requires expert knowledge. For each resource, the evaluation function has to be defined when it is added to the resource library. If an evaluation function already exists for a resource class, it can often be adapted for resources of the same type. Figure 3 shows the workflow for determining the workspace evaluation function when modeling a new resource for the resource library.

In the following sections, we present possible workspace evaluation functions for the resource types "camera" and "industrial robot". Depending on the specific requirements, the user is free to adapt or replace the proposed functions.

Workspace evaluation of cameras

The primary function of a camera is to capture images. In an industrial environment, they are used for various tasks such as object recognition or optical quality inspection. The most important consideration for the placement of a camera is the quality of the captured images.

The size and shape of the camera's workspace are mainly determined by the type of optical system (e.g., entocentric,

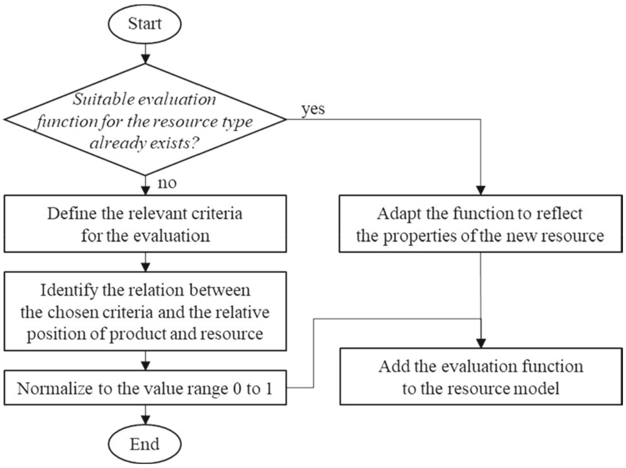

Start
Suitable evaluation function for the resource type already exists?
yes
no
Define the relevant criteria for the evaluation
Identify the relation between the chosen criteria and the relative position of product and resource
Normalize to the value range 0 to 1
End
Adapt the function to reflect the properties of the new resource
Add the evaluation function to the resource model

Fig. 3 Workflow for defining the workspace evaluation function of a new resource

object-space telecentric), the image sensor used, the focal length of the optics, as well as the lens and aperture used (Gronle & Osten, 2016, p. 6).

The camera system's properties result in an ideal distance range within the working area, the so-called depth of field. Outside this zone, the sharpness of the image decreases. Therefore, the distance between the camera and ROI is a significant influencing factor when evaluating a camera's positioning. Another factor influencing the image's quality is the camera's orientation relative to the surface being observed. If the camera is not orthogonal to the work surface, the spatial tilting causes blurring at the recorded image's edges (Chen et al., 2008, p. 68).

The pose quality *PQ* is therefore comprised of a distance-dependent component $PQ_{distance}$ and a rotation-dependent component $PQ_{rotation}$. $PQ_{distance}$ is a function of the distance z between the camera and ROI and is calculated as.

$$PQ_{distance}(z) = \begin{cases} \frac{1}{b-a}(z-a) & \text{for } a \le z \le b \\ 1 & \text{for } b \le z \le c \\ \frac{1}{d-c}(d-z) & \text{for } c \le z \le d \\ 0 & \text{else} \end{cases} \tag{2}$$

with *a* being the minimal distance for the camera to work, *b* and *c* being the near and far end of the ideal interval and *d* being the maximal working distance. This means that if the distance is within the ideal range of the depth of field, $PQ_{distance}$ is 1; outside the range, $PQ_{distance}$ linearly decreases until 0 is reached. Figure 4 shows the course of the function.

The rotation-dependent portion $PQ_{rotation}$ is a function of the angle γ between the visual axis of the camera and the surface normal of the ROI and is calculated as.

$$PQ_{rotation}(\gamma) = 1 - \frac{\gamma}{\gamma_{max}} \tag{3}$$

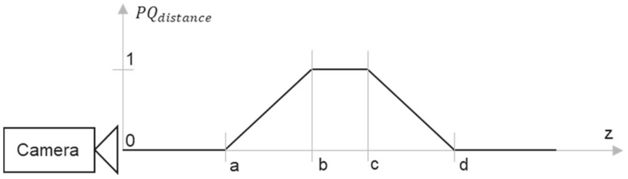


Fig. 4 Development of the position-dependent portion of PQ

with γ_{max} being the maximal tilt permitted. Ideally, the camera is vertical to the ROI, and γ is 0. In this case, $PQ_{rotation}$ takes the value 1. If the ROI is tilted relative to the camera, $PQ_{rotation}$ linearly decreases until 0 is reached. The parameters *a*, *b*, *c*, *d*, and γ_{max} required to calculate $PQ_{distance}$ and $PQ_{rotation}$ depend on each camera's characteristics and must be defined for each type of camera added to the resource library.

To calculate the pose quality *PQ*, the geometric mean value is formed from the distance-dependent and the rotation-dependent part, provided that the ROI is entirely within the workspace and no other object blocks the sight.

$$PQ = \begin{cases} \sqrt{PQ_{\text{distance}} * PQ_{\text{rotation}}} & \text{for ROI in workspace and not blocked} \\ 0 & \text{else} \end{cases} \tag{4}$$

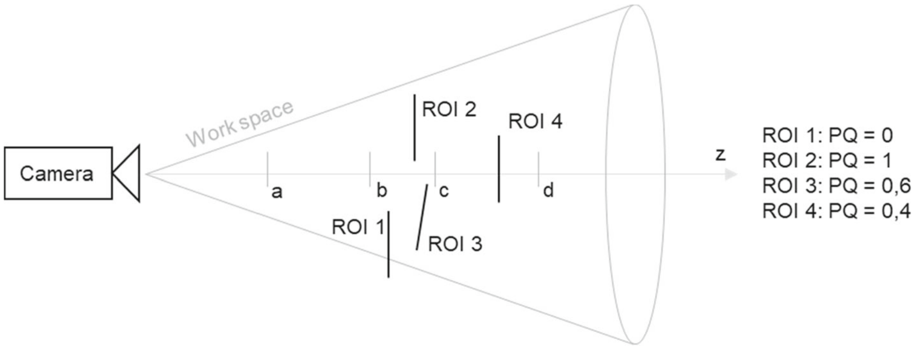
This calculation is more appropriate than the arithmetic mean because the result is closer to the worse portion's value. In extreme cases, if one of the two values is 0, *PQ* will also be 0 overall. This is desirable because if either position or rotation results in poor recording quality, it cannot be compensated by the other aspect.

Figure 5 illustrates various possible positions of the ROI and the resulting pose quality. ROI 1 has a *PQ* of 0 because it is partly outside the workspace. ROI 2 has a *PQ* of 1 because it is located between b and c in the ideal distance range and is vertical to the camera's visual axis. ROI 3 is also in the perfect distance range but is slightly tilted relative to the camera. Its *PQ* is therefore between 0 and 1 at 0.6. While ROI 4 is vertical to the camera, it is not within the ideal distance range but between c and d. It, therefore, has a *PQ* of 0.4, which is also between 0 and 1.

Workspace evaluation of industrial robots

Industrial robots are complex actuator-like resources that can perform a variety of production processes. Consequently, there are various plausible criteria for the evaluation of relative poses between robot and ROI. Examples of such characteristics, which are not solely dependent on the robot type but also on the current axis position, are accuracy, lifting capacity, or robot wear. Depending on the application, different aspects can be focused on (Woenckhaus, 1994, pp. 59–60). Another useful criterion for evaluating robot

Fig. 5 Examples for the evaluation of poses in the workspace of a camera



poses is manipulability. Manipulability describes the robot's ability to freely change the end effector's position and orientation (Izui et al., 2013).

The Jacobian matrix J represents the relationship between a robot's joint velocities $\dot{q}$ and the end effector's movement in a Cartesian reference coordinate system. It depends on the robot's kinematics and the current robot pose, defined by the position q of the robotic joints. The Jacobian consists of a linear part J_P and an angular part J_O. J_P defines the relationship between the joint speeds $\dot{q}$ and the position change $\dot{p}$ of the end effector, while J_O defines the relationship between the joint velocities $\dot{q}$ and the angular velocity ω of the end effector. (Siciliano et al., 2009, p. 106).

$$J(q)\dot{q} = \begin{bmatrix} J_T \\ J_R \end{bmatrix} \dot{q} = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} \tag{5}$$

Depending on the robot's current pose, the robot's capability to change the end effector's position and orientation varies. In extreme cases, so-called singularities occur in which the Jacobian matrix is rank-deficient, and the manipulability of the robot is reduced, meaning that the end-effector cannot be arbitrarily moved. Close to such singular configurations, even low speeds in Cartesian space lead to high axis speeds. (Siciliano et al., 2009, p. 116).

A well-established representation of the mobility of the end effector is the manipulability ellipsoid. It represents the velocities in Cartesian space generated in a given robot pose by given robot joint velocities. A direct metric for the eccentricity of the manipulability ellipsoid and the current pose's distance to a singularity is the ratio σ between the Jacobian matrix's minimum and maximum singular value. In the case of a singularity, the manipulability ellipsoid degenerates, and σ becomes 0. In the case of optimal manipulability, all degrees of freedom of the end effector can be used equally, and σ has the value 1 (Siciliano et al., 2009, pp. 152–153, 2009, p. 155).

Poses close to a singularity are not only disadvantageous for the manipulability but also other properties of the robot, such as its accuracy and stress on the joints (Woenckhaus, 1994, pp. 59–60). Therefore, the proposed pose quality *PQ* is based on the parameter σ that measures the eccentricity of the manipulability ellipsoid and the current pose's distance to a singularity. It is calculated as

$$PQ(\mathrm{q}) = (\sigma(J(q)))^y = \left(\frac{\sigma_{min}(J)}{\sigma_{max}(J)}\right)^y \tag{6}$$

Here $\sigma_{min}(J)$ is the minimum and $\sigma_{max}(J)$ is the maximum singular value of the Jacobian matrix. The value of σ is always between 0 and 1, which corresponds to the position quality's normalized value range. An exponent y is added to scale *PQ* distribution. In our example, y is set to 0.15. Figure 6 shows a histogram of *PQ* for three different robot types.

To calculate the pose quality, the Jacobian is saved for each robot type based on its kinematics as a function of the joint angles. For a specific pose with defined axis positions, the matrix's maximum and minimum singular value and thus *PQ* can be calculated. Figure 7 shows the evaluation of different positions in the workspace of an LBR iiwa 7 industrial robot. For each position, six orientations (along the world coordinate system's axis) were evaluated. The color represents the mean *PQ* value of those evaluations.

Assembly information

Besides the functionality of a line layout, which is ensured if all the resources in a station can reach the ROI, the feasibility of assembling the line itself and the costs associated with the assembly are important criteria for evaluating a layout draft. These costs depend to a large extent on the positioning of the various resources. For example, mounting directly at designated interfaces (e.g., a sensor at the edge of a conveyor belt) is more economical than mounting at an arbitrary position in space, where a special fixture would be needed. The orientation can also be a factor; for example, mounting a robot overhead is more costly than placing the same robot on the floor. To take this into account in the optimization algorithm, the possible mounting locations must be evaluated. Additional information about each resource's assembly conditions is needed to consider these aspects in the automated layout planning. Figure 8 gives an overview of the necessary information.

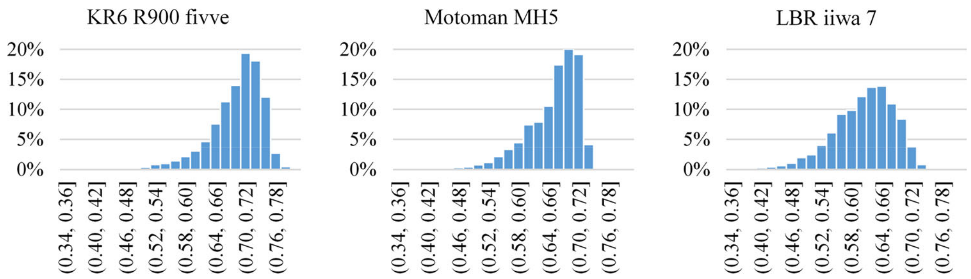


Fig. 6 Distribution of the values of PQ for the achievable poses of a 5-axis articulated arm robot (KR6 R900 fivve), a 6-axis articulated arm robot (Motoman MH5), and a 7-axis articulated arm robot (LBR iiwa 7)

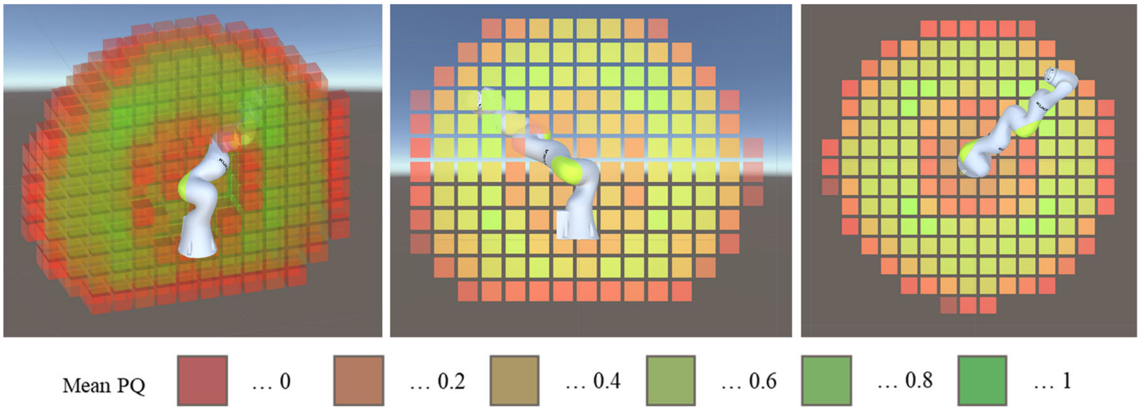


Fig. 7 Examples for the evaluation of positions in the workspace of an LBR iiwa 7 robot, left: 3D view, center: sectional view from left, right: sectional view from above

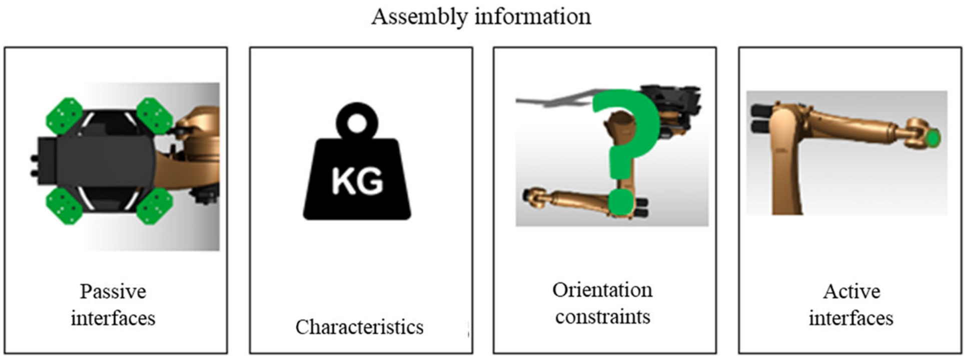


Fig. 8 Structure of the assembly information

The first part of the assembly information is the location of the passive mounting interface of the resource. These interfaces describe where the resource is physically connected to its environment. In the case of an industrial robot, the passive mounting interface would be in the robot base. To automatically generate and optimize 3D layouts, this information is crucial. The second aspect is the resource's characteristics relevant for its placement (like its weight), constraints concerning the ambient conditions (like certain temperatures or air humidity), and the basic costs associated with installing the resource. Thirdly, the resource may contain restrictions on the orientation of its installation. For example, it might be defined that a resource can only be installed upright, and overhead mounting is not permitted. It is also defined how much additional effort would be caused by different mounting options (e.g., on the floor/ceiling/another resource or freely in space with a designated support structure). The fourth part is the active mounting interfaces. They do not describe the requirements for installing the resource itself but rather indicate where other resources can be connected to the resource geometrically. Additionally, they have parameters that provide restrictions for the resources placed on the interface (e.g., a maximum bearing load) and a factor that describes the effort to install another resource.

Solution approach

Generating a feasible layout and optimizing it according to the presented objective function (Eq. 1) is computationally costly. Therefore, a heuristic algorithm, specifically a genetic algorithm, is used (Fig. 9). In a genetic algorithm, a

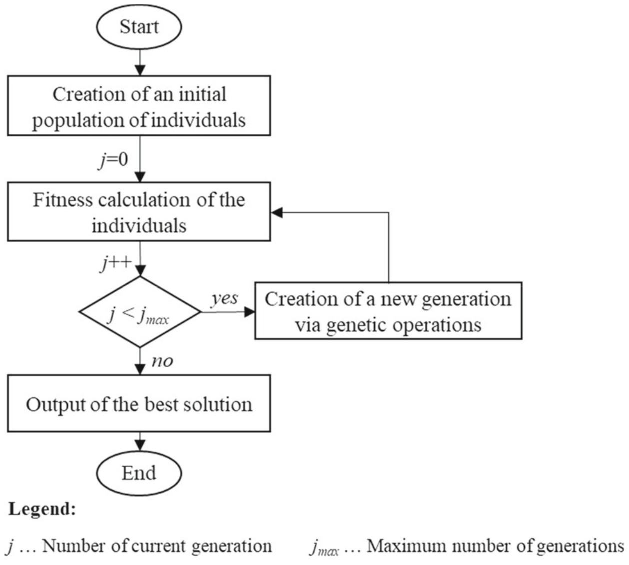

Start
Creation of an initial population of individuals
j=0
Fitness calculation of the individuals
j++
$j < j_{max}$
yes
Creation of a new generation via genetic operations
no
Output of the best solution
End
Legend:
j … Number of current generation
j_{max} … Maximum number of generations

Fig. 9 Basic functionality of a genetic algorithm

population of individuals representing potential solutions is generated. Then the individuals' fitness is calculated, and the next generation is created. By using suitable genetic operators, the individuals' fitness increases from one generation to the other, approximating the optimal solution (Mitchell, 1999).

For the implementation of the genetic algorithm, the open-source library GeneticSharp (Giacomelli, 2019) was used. It provides a genetic algorithm that can be adapted to a specific use case. The encoding of potential solutions into genes and chromosomes, as well as the genetic operations and selection strategies, can be customized. In the proposed layout planning system, the optimization module interacts with the resource library and the simulation module to generate, evaluate, and optimize potential layout solutions. For the necessary simulations (e.g., to ensure that resources do not overlap and can reach the product), the Unity 3D game engine is used because it is very flexible for assembly line simulation (Paul et al., 2019).

Encoding of the potential solutions

For the optimization of the placement of resources, each individual represents a possible layout. It contains the allocation of resources to stations and a chromosome consisting of a list of genes, where each gene is a six-dimensional vector, representing the pose of a resource in space (Fig. 10).The first three entries of the vector represent the resource's position and the last three its orientation. The entries refer to the coordinate system of the respective station.

Creation of a new individual

When a new individual is created (Fig. 11), the regions where each resource can be placed are determined first. On the one hand, these are the active interfaces of the other resources of the same station if they meet the resource's requirements (e.g., maximum bearing load higher than the resource's weight). On the other hand, the resource can be placed in the station independent of the other resources (e.g., on the station's floor or ceiling). Then the resource is randomly assigned a valid pose relative to its station. If placed on another resource, the resource's weight is subtracted from the interface's maximum bearing capacity to avoid an overload through multiple resources. Then, possible collisions of the resource with other objects in its station are evaluated. If there is a collision, the resource is placed anew and checked again for a collision. If no collision-free solution can be found after a certain number of attempts, the station is declared invalid, and the process continues with the next resource. After all the resources are placed in their respective stations, the individual stations are placed one after the other along the line as close as possible, thus creating an overall layout of the line. When evaluating the layout design, the number of invalid stations (with at least one colliding resource) is taken into account, and the individual accordingly is evaluated negatively.

Description of the genetic operations

Genetic operations are used to create the next generation in a genetic algorithm. Mutation operators are applied to existing individuals to create a new one. Recombination operators combine two existing individuals into two new ones. A selection strategy is used to choose suitable individuals for the application of the operators. Those operations have to be adapted to the specific problem and the individuals' encoding to achieve an efficient optimization process. For the layout optimization, the following operators are considered:

- *Uniform Mutation*: Each gene is mutated with a given probability. If the gene is mutated, the respective resource is given a new, random position and orientation.
- *Uniform Crossover*: For each resource position encoded in a gene in the chromosome, it is randomly decided which parent's gene is chosen. This means that some resources are at the positions they had in the one parent, others at the position they had in the other parent.
- *Station Crossover*: All the genes representing resource poses in the same station are taken from the same parent. For each station, it is randomly decided which parent's resource poses are used.

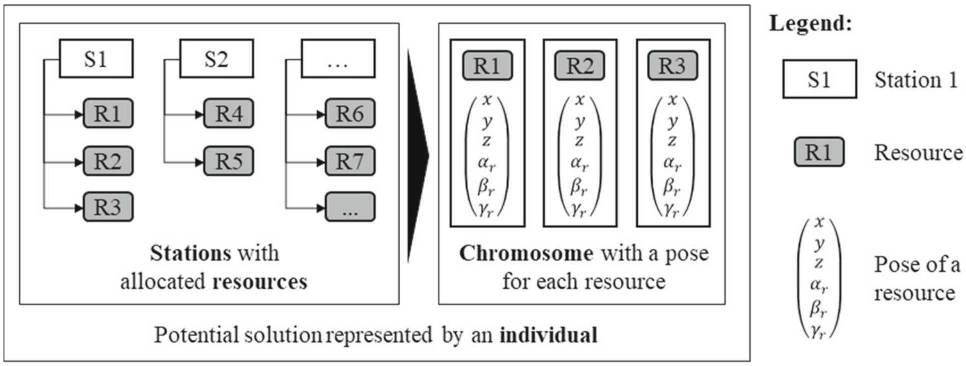

S1
R1
R2
R3
S2
R4
R5
…
R6
R7
…
Stations with allocated resources
R1
R2
R3
x y z α_r β_r γ_r
Chromosome with a pose for each resource
Potential solution represented by an individual
Legend:
S1 Station 1
R1 Resource
Pose of a resource

Fig. 10 Structure of an individual

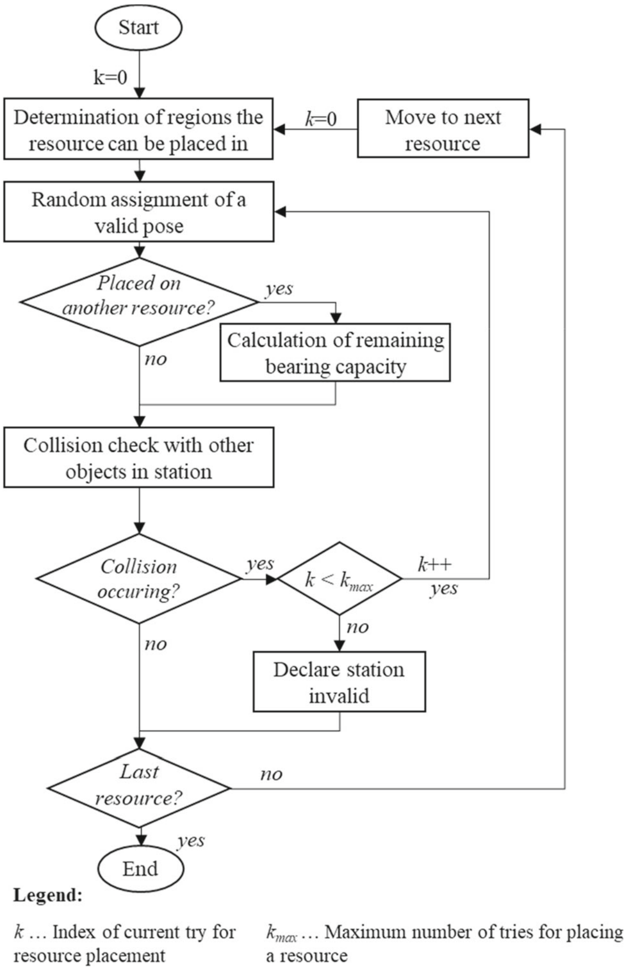

Start
k=0
Determination of regions the resource can be placed in
k=0
Move to next resource
Random assignment of a valid pose
Placed on another resource?
yes
Calculation of remaining bearing capacity
no
Collision check with other objects in station
Collision occuring?
yes
$k < k_{max}$
k++
yes
no
Declare station invalid
no
Last resource?
no
yes
End
Legend:
k … Index of current try for resource placement
k_{max} … Maximum number of tries for placing a resource

Fig. 11 Creation of a new individual

- *Station and uniform crossover:* Crossover is either performed based on resources or stations with a probability of 50%, respectively.
- *Elitist Selection*: The individuals with the best fitness value are chosen among the generated child individuals. If the minimum population size is not reached, parent individuals from the previous generation are added.
- *Tournament Selection*: Two individuals from the current generation are randomly chosen, and the one with the better fitness is selected to be a parent.
- *Roulette Wheel Selection*: Each individual is chosen with a probability corresponding to its fitness value compared to all other chromosomes' fitness in the same generation to be a parent.

Evaluation of the fitness

In a genetic algorithm, the individuals are evaluated with a fitness function. Since a genetic algorithm tries to maximize the fitness, the fitness function to determine a layout's fitness F is defined as the negative of the optimization problem's objective function (Eq. 7).

$$F\,(pose_{Res}) = -\left(\alpha \cdot \sum_{Res} c_{mounting} + \beta \cdot A_{line} + \gamma \cdot \sum_{Res} (1 - PQ)\right) \tag{7}$$

It consists of three terms that contain factors contributing to the line's economic viability:

- $c_{mounting}$, the costs of assembling the line itself
- A_{line}, the line's area that causes costs for the floor space
- PQ, the technical quality of the layout that influences the reliability and the scrap rate of the line

These three aspects are multiplied by the coefficients α, β and γ to reflect the user's preferences in weighting the different elements. In the following, we explain the three terms in detail.

The *mounting costs* $c_{mounting}$ depend on a resource's positioning and the effort necessary to install it there. They are calculated by multiplying the basic mounting costs $c_{basicmounting}$ by several factors representing the resource's specific installation conditions (Eq. 8).

$$c_{mounting} = c_{basicmounting} \cdot \varphi_{mountingtype} \cdot \varphi_{interface} \cdot \varphi_{blocked} \tag{8}$$

The basic mounting costs $c_{basicmounting}$ describe the costs to install a resource in the line. The factor $\varphi_{mountingtype}$ is a property of the resource and is included in the resource model. Its value depends on how the resource is mounted. The factor $\varphi_{interface}$ is only relevant if the resource is installed on the mounting interface of another resource. The resource model that contains the interface then provides its value that describes the effort to use the mounting interface. The last factor $\varphi_{blocked}$ is used to penalize the positioning of resources above another object (e.g., other resources). To reflect the increased effort in this situation, we set it to 1.5 if another object is placed between a resource's passive mounting interface and the floor; otherwise, if there is no blocking object, we set it to 1. The coefficient α in Eq. 7 weights the mounting costs relative to the other terms. It will tend to be inversely proportional to the line's expected lifetime because the longer the line is in use, the less critical the initial costs.

The second aspect influencing the layout fitness is the *line's occupied area* since additional factory space is associated with increased costs. The station areas that are summed up to the line's area A_{line} are determined by drawing the smallest possible rectangle around the resources in a station. The coefficient β that is multiplied by A_{line} can be interpreted to represent the cost per unit of space. It might be proportional to the line's expected lifetime because the space is occupied as long as the line is used.

Lastly, the technical quality of the resources' positioning represented by the pose quality PQ is considered. PQ is a function of the relative pose between the resource and the product, and the specific function is part of the resource's model. An ideal resource pose has the position quality of 1 (Sect. 0), meaning the term $(1 - PQ)$ has the ideal value 0. The coefficient γ shows the weight the user put on the positioning relative to the other factors. Since the positioning has an effect over the whole lifetime of the line, γ might also be proportional to the expected lifetime of the line.

Parameter study

The presented algorithm has a set of parameters and settings that influence the optimization results' quality and the optimization process's speed. To determine a suitable set of parameters for the positioning problem, a parameter study was performed for an example consisting of multiple stations and different resources. The assembly line in the example consists of six stations, each with two or three resources (cameras, light sensors, or robots of various types). The light sensor was modeled similarly to the cameras but with an adapted workspace and a pose quality function independent of orientation.

Table 1 Factors and corresponding levels for the computational experiment

Factor	Levels		
Selection strategy	Elitist	Roulette wheel	Tournament
Crossover strategy	Uniform	Station	Combined
Mutation probability	0%	10%	40%
Crossover probability	25%	75%	100%

Design of experiments

A design of experiments (DoE) approach was used to define which computational experiments to conduct. According to Dean et al. (2017) the first steps in planning an experiment are the definition of the experiment's objective and the identification of the relevant factors and their levels. In the present case the purpose of the experiment is to find the combination of settings of the variable parameters that leads to the best optimization result. The following factors and levels were considered (Table 1):

- The *Selection strategy* determines the criterion by which individuals are chosen for the next generation. Three strategies (see Sect. 0) are evaluated: Elitist selection, Roulette wheel selection, and Tournament selection.
- *Crossover strategy* defines how a crossover of two chromosomes is performed. Three types of strategies (see Sect. 0) are evaluated: uniform crossover, station crossover, and combined station and uniform crossover.
- The *Mutation Probability* sets the probability of a gene being mutated. The mutation introduces randomness into the algorithm. Four values of mutation probability (0%, 10%, 40% and 80%) are evaluated.
- The *Crossover Probability* sets the probability that a crossover happens, leading to new child chromosomes. The higher the crossover probability, the higher the ratio of child chromosomes compared to parent chromosomes in one generation. Three crossover probabilities (25%, 75% and 100%) are evaluated.

The remaining parameters are set as follows:

- $c_{basicmounting} = 100$
- $\varphi_{mountingtype,cameras} = \begin{cases} 1 \text{ for mounting on another resource} \\ 2 \text{ for mounting directly in the station} \end{cases}$
- $\varphi_{mountingtype,robots} = 10$ for mounting on the ground
- $\varphi_{interface,conveyour} = 1$
- $\alpha = 1, \beta = 200, \gamma = 1000$
- Number of generations: 200
- Population size: 25
- Mutation: Uniform

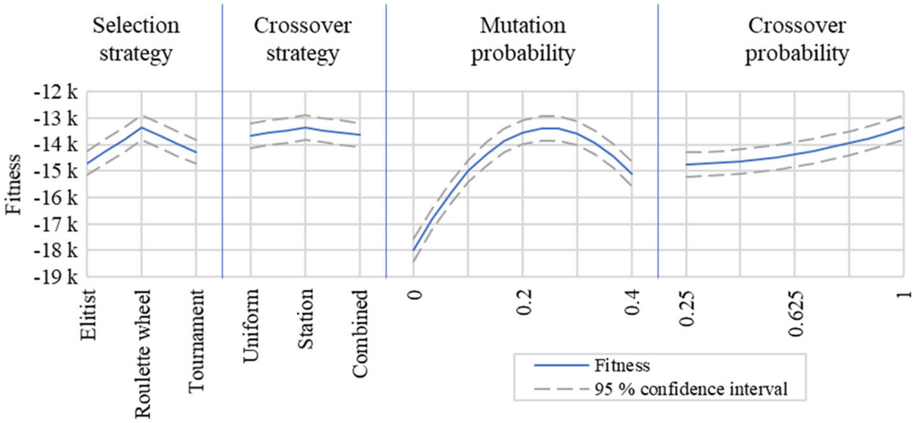


Fig. 12 Optimization result depending on the factors

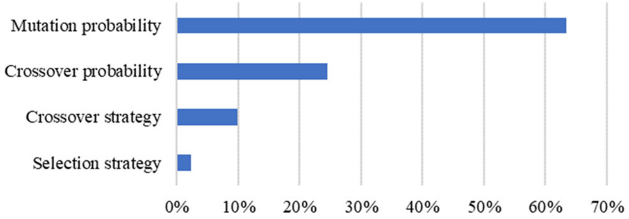


Fig. 13 Influence of individual factors on the optimization result

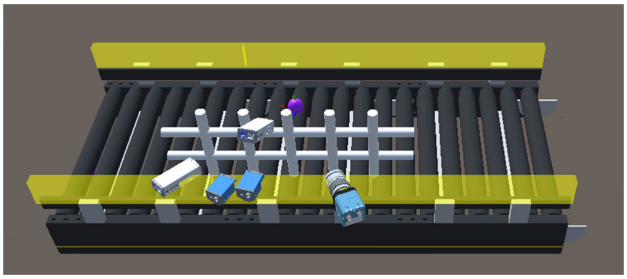


Fig. 14 Setup of the first example

The population size and the number of generations are kept constant, as the dependency of the result on these parameters is clear: the larger the population and the higher the number of generations, the better the result. Of course, this implies higher computational costs and simulation time. The mutation strategy is also unaltered. A mutation strategy based on the resources analogous to the uniform crossover is performed. Even though other strategies are undoubtedly possible, we consider this one the most natural approach.

To study the relationship between the factors and the quality of the optimization process a full factorial experimental design was chosen. This design allows to determine the influence of the individual factors as well as their interdependencies. (Siebertz et al., 2010) For each combination of parameters, three simulation runs were performed. The experimental design as well as the results for each run can be found in Table 4 in the supplementary material.

Analysis of the results

Based on the collected data a quadratic statistical model was created and used to predict optimal values for the factors. The R-squared value of the model is 0.72. A multiple regression analysis suggested the use of roulette wheel selection, station crossover, a crossover probability of 100%, and a mutation probability of 25% (Fig. 12).

Besides identifying the optimal values for the parameters, the statistical model also allows an analysis of the individual factor's influence on the optimization results (Fig. 13). The mutation probability has the highest impact on the quality of the optimization results. This is plausible since a high mutation probability is close to a random search but to little mutation is also not ideal since a certain amount of randomness is necessary. Therefore, striking the right balance is of great importance. The crossover probability is also quite influential. However, here it is clear that a maximal value leads to the best results. This seems reasonable since crossover drives a genetic algorithm by combining the strength of different individuals. The chosen strategies for crossover and selection have a comparatively small influence. This might be because the different strategies are quite similar. In the case of recombination, two individuals are combined, trying to preserve their strengths; in the case of selection, good solutions are favoured, while trying to keep a certain diversity in the population.

Examples

The following section presents three increasingly complex layout planning problems to illustrate and evaluate the proposed optimization algorithm's capabilities.

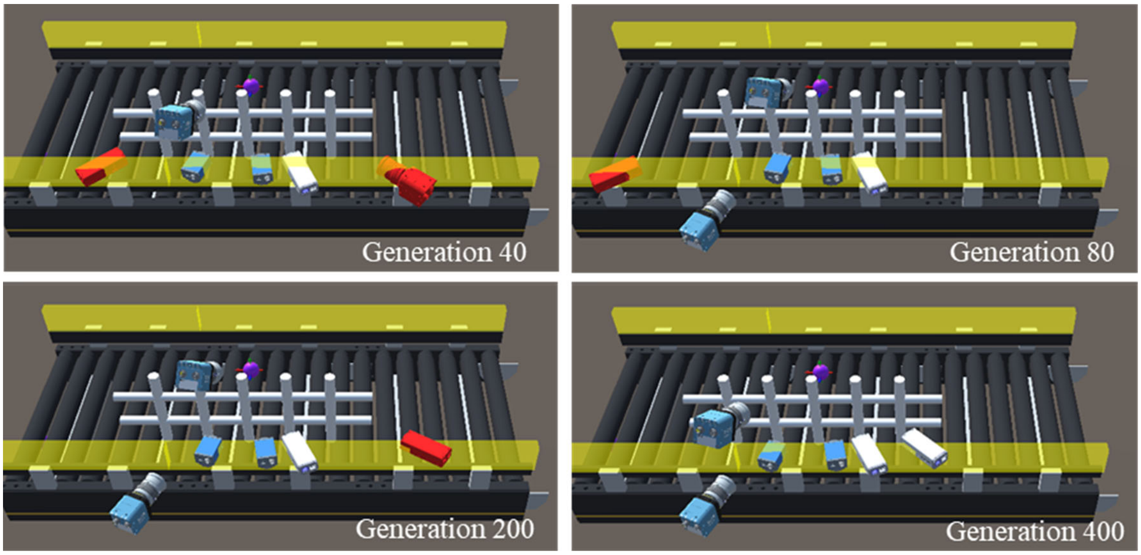


Fig. 15 Best layout in selected generations of the first run

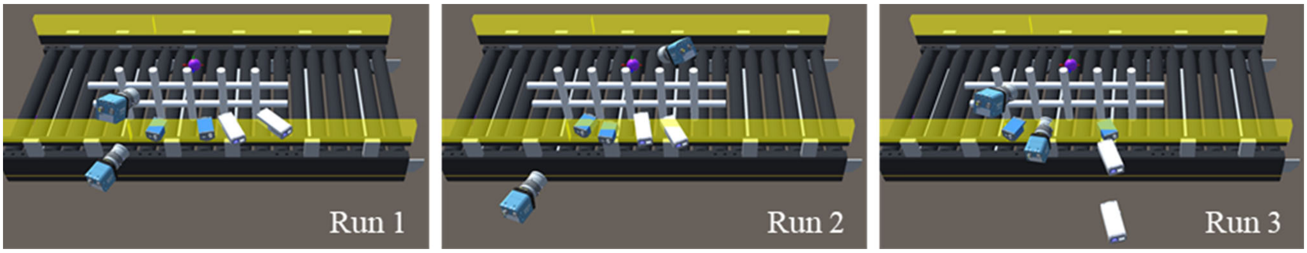


Fig. 16 The final layout of the first three runs of the first example

Camera positioning example

In the first example, the presented algorithm is used to position several cameras with respect to one ROI within one station. The example demonstrates that resources can be placed on the active mounting interfaces of another resource, as well as directly in the station. It also illustrates the system's ability to generate collision-free layouts and to ensure the ROI's visibility for all cameras. The setup is shown in Fig. 14.

The ROI is located on a conveyor. The conveyor model includes active mounting interfaces on the side of the belt (visible as yellow Boxes in Fig. 14). In addition, an obstacle next to the ROI partially blocks the cameras' field of view. Six cameras are positioned. Two instances of three different camera models are used: The Imaging Source DFK 38UX304 (The Imaging Source, 2019), Basler BIP2-1920c (Basler, 2020), and SICK RANGER3—V3DR3-60NE31111 (SICK, 2021).

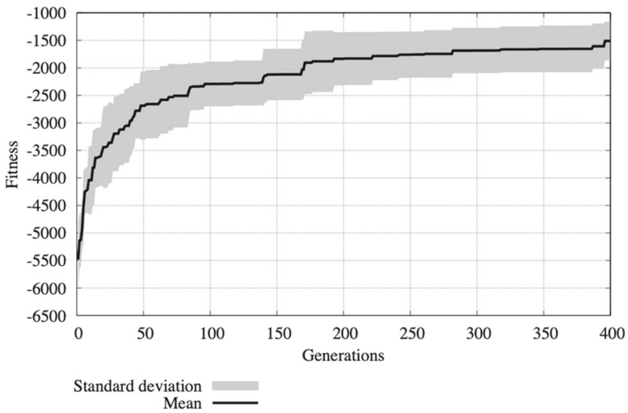


Fig. 17 Development of the fitness of the best individual in a generation in the ten runs

The cameras need to look over or past the obstacle to obtain a good pose quality because the pose quality is only not 0 if the camera can see the whole object (Eq. 4). Unity's ray tracing functionality is used to check if the obstacle blocks the cameras' view. The cameras can be mounted on the conveyor as well as directly in the station. The parameters of the fitness function are set as follows:

- $c_{basicmounting} = 100$
- $\varphi_{mountingtype,cameras} = \begin{cases} 1 \text{ for mounting on another resource} \\ 2 \text{ for mounting directly in the station} \end{cases}$

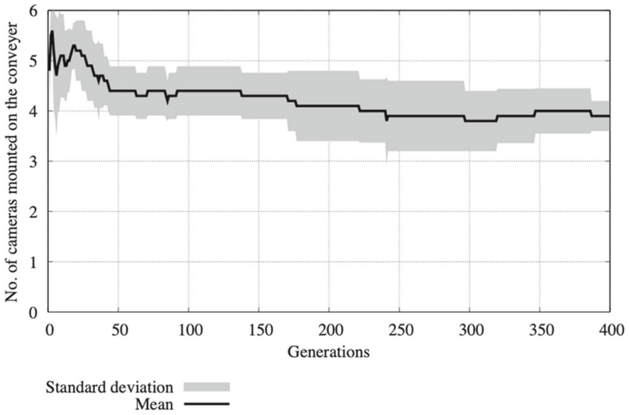


Fig. 18 Number of cameras mounted on the conveyor in the ten runs

- $\varphi_{interface,conveyour} = 1$
- $\alpha = 1, \beta = 0, \gamma = 1000$

There are positions in the mounting space and the free mounting space that result in a perfect pose quality. This means that one camera's optimal position would be on the mounting space looking past the obstacle. Area costs are neglected in this example, leading to $\beta = 0$.

The genetic algorithm itself has the following characteristics:

- Number of generations: 400
- Population size: 10
- Mutation: Uniform mutation
- Selection strategy: Roulette wheel selection
- Crossover strategy: Uniform crossover (station crossover would have no effect as there is only one station)
- Mutation probability: 25%
- Crossover probability: 100%

The optimization was run ten times. Figure 15 shows the best layout from selected generations of the first run. Pictures from further generations and runs can be found in the supplementary data. If a camera has a pose quality of 0, it is marked in red. The pictures give an impression of the evolution of the solutions.

Table 2 Movement time of the robots in the second example

	Run 1	Run 2	Run 3
KUKA KR6 R900 fivve	1.0	1.0	1.0
KUKA LBR iiwa 7 R800	1.2	1.3	1.9
Yaskawa Motoman MH5	1.2	1.2	2.3

Figure 16 shows the final positioning for runs 1 to 3. For all cameras, a suitable position could be found.

Figure 17 depicts the development of the best individual's fitness over the generations in the ten runs. As expected, the fitness gradually increases over the generations.

Figure 18 provides further insight into the optimization process by visualizing the number of cameras placed using the conveyer's active mounting interfaces. A solution with all cameras directly on the conveyer would be feasible and ideal for mounting costs. However, its overall fitness would be relatively low due to some cameras being positioned far from the ROI. This shows that the algorithm successfully strikes a compromise between pose quality and mounting costs.

Robot positioning example

In the second example, robots are positioned within one station of an assembly line. It demonstrates the capabilities of the approach regarding the positioning of moving resources. This includes evaluating the robots' position, simulation of their trajectory, and collision detection. Additionally, we show the planning system's capability to estimate the cycle time for a layout. In the example, three different robot models are positioned around an ROI on a conveyor: KUKA KR6 R900 fivve (KUKA, 2017a), KUKA LBR iiwa 7 R800 (KUKA, 2017b), and Yaskawa Motoman MH5 (YASKAWA, 2019).

The fitness evaluation and the collision detection for moving resources like robots are more complex than for static resources like cameras. Every robot has a starting position for its movement, e.g., the robot's home position or a position at a table (included in the model) where material is supplied.

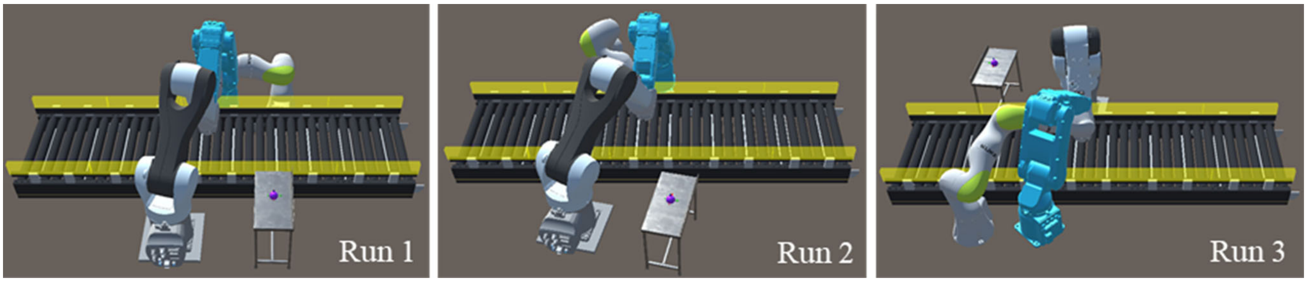


Fig. 19 Final layout of the first three runs of the second example

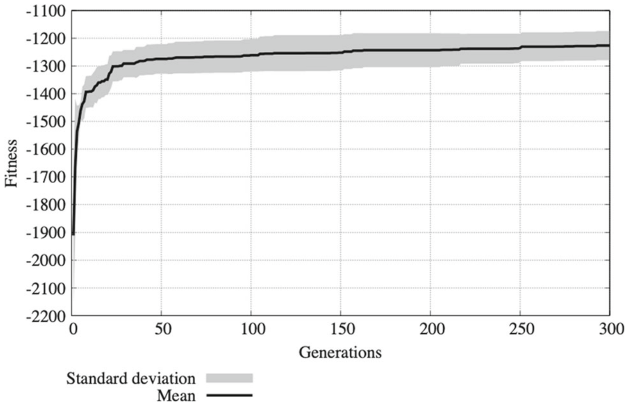


Fig. 20 Development of the fitness of the best individual in a generation in the ten runs

From this position, it moves to the ROI. More target positions (like a gripper change system) could be added. If the robot cannot reach the ROI and possibly other target positions, the pose quality is 0; otherwise, it is calculated with the PQ formula stored in the robot model (Eq. 6). Collision with other physical objects (i.e., other resources) is not allowed in the starting position. At the ROI and other target positions (and during the transition from the reference position to the ROI), no collision with other physical objects is allowed, apart from other moving resources, i.e., robots. This is because they will move to the ROI one after the other, so no actual collision occurs. In the example, no suitable active mounting interfaces for mounting robots exist, and the mounting always takes place on the ground. Consequently, the mounting costs are fixed and can be neglected, reflected in the parameter α being zero. The parameters of the fitness function are set to $\alpha = 0$, $\beta = 200$, $\gamma = 1000$ and the genetic algorithm itself has the following characteristics:

- Number of generations: 300
- Population size: 10
- Mutation: Uniform mutation
- Selection strategy: Roulette wheel selection
- Crossover strategy: Uniform crossover (station crossover would have no effect as there is only one station)
- Mutation probability: 25%
- Crossover probability: 100%

The optimization was run ten times. Figure 19 shows the final layouts of runs 1 to 3.

Run 3 found the best solution in terms of minimizing the area of the station. However, it is noteworthy that the solutions found in runs 1 and 2 have better overall fitness than the one found in run 3 while simultaneously having a larger area. This again demonstrates the algorithm's ability to strike a balance between different objectives. Figure 20 shows the development of the fitness over the generations in the ten runs. As expected, the fitness gradually increases over the generations.

Besides optimizing the layout for defined objectives, the system can also evaluate the time the robot needs to move to its target positions using the BioIK library (Starke et al., 2019). This way, it can be ensured that the required cycle time is met. Table 2 shows the movement time in seconds for the three robots in the station in the first three runs.

Example with ten stations

This example demonstrates the capabilities of the algorithm for a large-scale problem with multiple stations. In the example, the line contains ten stations with two or three resources each. The resources are either a camera (three different models, see the first example), a robot (three different models, see the second example), or a light barrier (one model, Leuze electronic PRK 46B (Leuze electronic, 2011)). The parameters of the fitness function are set as follows:

- $c_{\text{basicmounting}} = 100$
- $\varphi_{mountingtype,cameras} = \begin{cases} 1 \text{ for mounting on another resource} \\ 2 \text{ for mounting directly in the station} \end{cases}$
- $\varphi_{mountingtype,robots} = 10$ for mounting on the ground
- $\varphi_{interface,conveyour} = 1$
- $\alpha = 1$, $\beta = 200$, $\gamma = 1000$

The genetic algorithm itself has the following characteristics:

- Number of generations: 400
- Population size: 25
- Mutation: Uniform mutation
- Selection strategy: Roulette wheel selection
- Crossover strategy: Station crossover
- Mutation probability: 25%
- Crossover probability: 100%

Again, ten runs were performed. The final positioning of three runs is shown in Fig. 21. The algorithm positions all resources successfully (the smaller resources are not clearly visible but also feasibly placed). It is observable that the resources are placed closely together to minimize the area used.

Figure 22 shows the development of the fitness in the ten runs.

Additionally, the robot's movement times from their starting position to the ROI were calculated (Table 3). Combined with the distance between the stations and the speed of the conveyor, these times can be used to calculate the cycle time of the line.

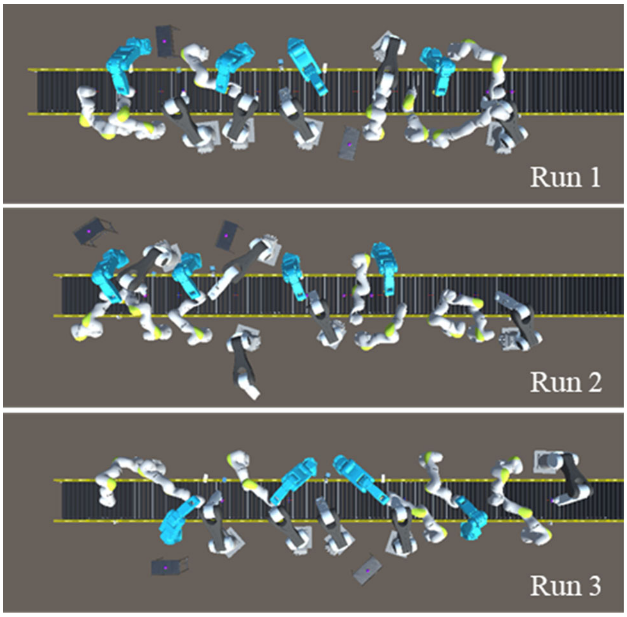


Fig. 21 Final layout of the three runs of the third example

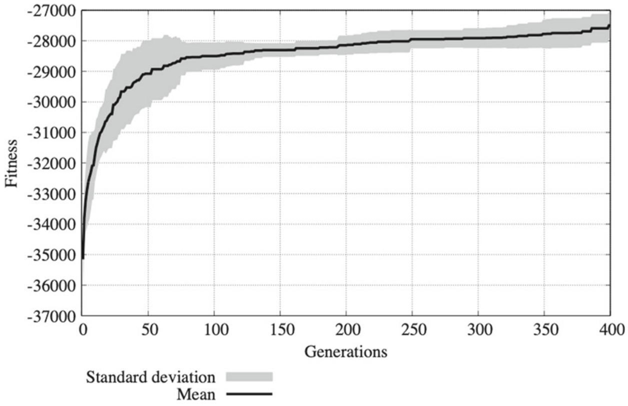


Fig. 22 Development of the fitness of the best individual in a generation in the ten runs

Table 3 Movement time of the robot in the third example

		Run 1	Run 2	Run 3
Station 1	KUKA LBR iiwa 7 R800	1.2	1.2	1.3
Station 2	Yaskawa Motoman MH5 w. table	1.2	2.0	3.0
Station 2	KUKA LBR iiwa 7 R800	1.3	1.2	1.4
Station 3	KUKA KR6 R900 fivve	1.0	1.0	1.5
Station 4	KUKA LBR iiwa 7 R800	1.3	1.3	1.5
Station 4	Yaskawa Motoman MH5	1.9	1.1	1.1
Station 4	KUKA KR6 R900 fivve	1.0	1.0	1.7
Station 5	KUKA KR6 R900 fivve w. table	1.0	1.0	1.9
Station 6	Yaskawa Motoman MH5	1.8	2.4	1.1
Station 7	KUKA KR6 R900 fivve	1.0	1.0	1.4
Station 7	KUKA LBR iiwa 7 R800	1.3	1.2	1.4
Station 8	Yaskawa Motoman MH5	1.5	1.1	1.1
Station 8	KUKA LBR iiwa 7 R800	1.3	1.2	1.4
Station 9	KUKA LBR iiwa 7 R800	1.4	1.3	1.4
Station 9	KUKA LBR iiwa 7 R800	2.3	1.2	1.3
Station 10	KUKA KR6 R900 fivve	1.0	1.0	1.7

Conclusion and outlook

In this article, we presented a novel approach to arrange different production resources into a three-dimensional assembly line layout. The planning system consists of a resource library with digital models of all available resources and an optimization module that generates and optimizes the layouts with a genetic algorithm. A parameter study was performed to determine the optimization algorithm's configuration. For the evaluation of a layout's fitness, a multi-body simulation with Unity 3D is used. The presented system is highly flexible and supports the positioning of multiple resource types. It is also open to integrating more resource variants or customizing the fitness function to reflect the individual user's preferences accurately.

The integration of more resource models and the modeling of more resource types would be the logical next step to improve the system further. In this article, we exemplarily implemented several real resource models to demonstrate the functionality of the approach. Extending the library of available resources would be crucial for industrial usage of the system. Another possible enhancement would be the inclusion of the cycle time in the optimization objective. The presented algorithm allows an evaluation, whether a maximally allowed cycle time is observed. However, it would also be reasonable to aim for a minimization of the cycle time. The calculation of the cycle time of a station could also be refined by considering resources moving in parallel as long as they do not collide. Another interesting degree of freedom that could be included in the planning system is the options for parallel stations. Lastly, it would be desirable to integrate more planning decisions, like balancing the line and the positioning of the resources into one planning/optimization system to consider interdependencies and reach an overall optimum. Besides these possible improvements to the system and the formulated optimization model it would be interesting to compare the chosen genetic algorithm with other solution approaches like ant colony or particle swarm algorithms. Moreover, reinforcement learning could be applied to generate layouts.

Funding Open Access funding enabled and organized by Projekt DEAL. The authors did not receive support from any organization for the submitted work.

Availability of data and material The data that support the findings of this study are openly available in Mendeley Data at http://dx.doi.org/10.17632/cxyxj3fm5g.1

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

Barral, D. (2002). optimization tool for assembly workcell Layout, 22.

Barral, D. (2003). Optimization tool for robot placement, 25.

Barral, D., Perrin, J.-P., Dombre, E., & Liegeois, A. (2001). Simulated annealing combined with a constructive algorithm for optimising assembly workcell layout. *The International Journal of Advanced Manufacturing Technology, 17*, 593–602. https://doi.org/10.1007/s001700170143

Basler. (2020). Basler BIP2–1920c: User's manual for BIP2-XXXXc and BIP2-XXXXc-dn Models. https://www.baslerweb.com/de/produkte/kameras/netzwerkkameras/ip-fixed-box-kameras/bip2-1920c/. Accessed 25 March 2021.

Besbes, M., Zolghadri, M., Costa Affonso, R., Masmoudi, F., & Haddar, M. (2021). 3D facility layout problem. *Journal of Intelligent Manufacturing, 32*, 1065–1090. https://doi.org/10.1007/s10845-020-01603-z

Bullinger, H.-J. (Ed.). (1986). *Systematische Montageplanung: Handbuch für die Praxis*. Hanser.

Chen, S., Li, Y., Zhang, J., & Wang, W. (2008). *Active sensor planning for multiview vision tasks*. s.l.: Springer Berlin Heidelberg.

Dean, A., Voss, D., & Draguljić, D. (2017). *Design and analysis of experiments (Springer texts in statistics)*. Springer.

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control, 31*, 255–267. https://doi.org/10.1016/j.arcontrol.2007.04.001

Ellefsen, K. O., Lepikson, H. A., & Albiez, J. C. (2017). Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures. *Applied Soft Computing, 61*, 264–282. https://doi.org/10.1016/j.asoc.2017.07.051

Giacomelli, D. (2019). *GeneticSharp* .

Gronle, M., & Osten, W. (2016). View and sensor planning for multi-sensor surface inspection. *Surface Topography: Metrology and Properties, 4*, 1–14. https://doi.org/10.1088/2051-672X/4/2/024009

Gülşen, M., Murray, C. C., & Smith, A. E. (2019). Double-row facility layout with replicate machines and split flows. *Computers & Operations Research, 108*, 20–32. https://doi.org/10.1016/j.cor.2019.03.009

Hosseini-Nasab, H., Fereidouni, S., Fatemi Ghomi, S. M. T., & Fakhrzad, M. B. (2018). Classification of facility layout problems: A review study. *The International Journal of Advanced Manufacturing Technology, 94*, 957–977. https://doi.org/10.1007/s00170-017-0895-8

Izui, K., Murakumo, Y., Suemitsu, I., Nishiwaki, S., Noda, A., & Nagatani, T. (2013). Multiobjective layout optimization of robotic cellular manufacturing systems. *Computers & Industrial Engineering, 64*, 537–544. https://doi.org/10.1016/j.cie.2012.12.003

Karaszewski, M., Adamczyk, M., & Sitnik, R. (2016). Assessment of next-best-view algorithms performance with various 3D scanners and manipulator. *ISPRS Journal of Photogrammetry and Remote Sensing, 119*, 320–333. https://doi.org/10.1016/j.isprsjprs.2016.06.015

KUKA. (2017b). Sensitive robotics_LBR iiwa. https://www.kuka.com/de-de/produkte-leistungen/robotersysteme/industrieroboter/lbr-iiwa. Accessed 26 March 2021.

KUKA. (2017a). KR 6 R900 fivve. https://www.kuka.com/de-de/produkte-leistungen/robotersysteme/industrieroboter/kr-agilus. Accessed 26 March 2021.

Leiber, D., Hammerstingl, V., Weiß, F., & Reinhart, G. (2019). Automated design of multi-station assembly lines. *Procedia CIRP, 79*, 137–142. https://doi.org/10.1016/j.procir.2019.02.029

Leiber, D., & Reinhart, G. (2020). A bi-level optimisation approach for assembly line design using a nested genetic algorithm. *International Journal of Production Research, 57*, 1–16. https://doi.org/10.1080/00207543.2020.1845411

Leuze electronic. (2011). PRK 46B: Retro-reflective photoelectric sensors with polarisation filter. https://www.leuze.com/de/deutschland/produkte/schaltende_sensoren/optische_sensoren/lichtschranken_lichttaster_kubisch/baureihe_46/reflexions_lichtschranken_5/prk_46b_7. Accessed 1 April 2021.

Lim, Z. Y., Ponnambalam, S. G., & Izui, K. (2016). Nature inspired algorithms to optimize robot workcell layouts. *Applied Soft Computing, 49*, 570–589. https://doi.org/10.1016/j.asoc.2016.08.048

Liu, S., Zhang, Z., Guan, C., Zhu, L., Zhang, M., & Guo, P. (2021). An improved fireworks algorithm for the constrained single-row facility layout problem. *International Journal of Production Research, 59*, 2309–2327. https://doi.org/10.1080/00207543.2020.1730465

Lueth, T. C. (1992). Automated planning of robot workcell layouts. In *IEEE International Conference on Robotics and Automation 1992, Nice, France, 05/12/1992 - 05/14/1992* (pp. 1103–1108). IEEE. https://doi.org/10.1109/ROBOT.1992.220201.

Michalos, G., Fysikopoulos, A., Makris, S., Mourtzis, D., & Chryssolouris, G. (2015). Multi criteria assembly line design and configuration – An automotive case study. *CIRP Journal of Manufacturing Science and Technology, 9*, 69–87. https://doi.org/10.1016/j.cirpj.2015.01.002

Michniewicz, J., Leiber, D., Riedl, F., Erdogan, H., Hörmann, M., Meyer, T., et al. (2017). Automatisierte digitale Anlagenplanung: Methode für die automatisierte Planung von Montageanlagen auf Basis digitaler Produktdaten. *Wt Werkstattstechnik Online, 107*(9), 582–589.

Mitchell, M. (1999). *An introduction to genetic algorithms* (5th ed., A Bradford book). MIT Press.

Mohamadi, A., Ebrahimnejad, S., Soltani, R., & Khalilzadeh, M. (2019a). A new two-stage approach for a bi-objective facility layout problem considering input/ output points under fuzzy environment. *IEEE Access, 7*, 134083–134103. https://doi.org/10.1109/ACCESS.2019.2941442

Mohamadi, A., Ebrahimnejad, S., Soltani, R., & Khalilzadeh, M. (2019b). An integrated approach based on a bi-level genetic algorithm and a combined zone-LB for the Facility Layout Problem. *South African Journal of Industrial Engineering, 30*(4), 87–101.

Monica, R., & Aleotti, J. (2018). Surfel-based next best view planning. *IEEE Robotics and Automation Letters, 3*, 3324–3331. https://doi.org/10.1109/lra.2018.2852778

Papakostas, N., Alexopoulos, K., & Kopanakis, A. (2011). Integrating digital manufacturing and simulation tools in the assembly design process: A cooperating robots cell case. *CIRP Journal of Manufacturing Science and Technology, 4*, 96–100. https://doi.org/10.1016/j.cirpj.2011.06.016

Papakostas, N., Pintzos, G., Matsas, M., & Chryssolouris, G. (2014). Knowledge-enabled design of cooperating robots assembly cells. *Procedia CIRP, 23*, 165–170. https://doi.org/10.1016/j.procir.2014.10.092

Paul, M., Leiber, D., Pleli, J., & Reinhart, G. (2019). Simulation of reconfigurable assembly cells with Unity3D. In F. Ameri, K. E. Stecke, G. von Cieminski, & D. Kiritsis (Eds.), *Advances in production management systems*: *IFIP WG 5.7 International Conference, APMS 2019, Austin, TX, USA, September 1–5, 2019 : proceedings* (pp. 223–230, IFIP Advances in Information and Communication Technology, Bd. 567). Cham: Springer International Publishing.

Pellegrinelli, S., Pedrocchi, N., Tosatti, L. M., Fischer, A., & Tolio, T. (2014). Design and motion planning of body-in-white assembly cells. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, USA, 09/14/2014 - 09/18/2014* (pp. 4489–4496). IEEE. https://doi.org/10.1109/IROS.2014.6943198.

Pellegrinelli, S., Pedrocchi, N., Tosatti, L. M., Fischer, A., & Tolio, T. (2015). Validation of an extended approach to multi-robot cell design and motion planning. *Procedia CIRP, 36*, 6–11. https://doi.org/10.1016/j.procir.2015.01.062

Pellegrinelli, S., Pedrocchi, N., Tosatti, L. M., Fischer, A., & Tolio, T. (2017). Multi-robot spot-welding cells for car-body assembly: Design and motion planning. *Robotics and Computer-Integrated Manufacturing, 44*, 97–116. https://doi.org/10.1016/j.rcim.2016.08.006

Pérez-Gosende, P., Mula, J., & Díaz-Madroñero, M. (2021). Facility layout planning. An extended literature review. *International Journal of Production Research, 59*, 3777–3816. https://doi.org/10.1080/00207543.2021.1897176

Reinhart, G., & Roßgoderer, U. (1998). A System for three-dimensional layout planning of automated and manual assembly systems. *IFAC Proceedings Volumes, 31*, 29–34. https://doi.org/10.1016/S1474-6670(17)40252-7

Rosman, G., Choi, C., Dogar, M., Fisher IIIl, J. W., & Rus, D. (2018). Task-specific sensor planning for robotic assembly tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 05/21/2018 - 05/25/2018* (pp. 2932–2939). IEEE. doi:https://doi.org/10.1109/ICRA.2018.8460194.

Roßgoderer, U., Woenckhaus, C., Reinhart, G., & Milberg, J. (1995). A concept for automatical layout generation. In *1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 05/21/1995 - 05/27/1995* (pp. 800–805). IEEE. doi:https://doi.org/10.1109/ROBOT.1995.525381.

Roßgoderer, U. (2002). *System zur effizienten Layout- und Prozessplanung von hybriden Montageanlagen*. Zugl.: München, Techn. Univ., Diss., 2002 (Forschungsberichte IWB, Bd. 168). Utz.

Sharma, A., Jha, A. K., & Halder, A. (2017). Layout optimization of a robotic cell for foundry application by CAD based point cloud modeling – a case study. *Industrial Robot: An International Journal, 44*, 788–797. https://doi.org/10.1108/IR-01-2017-0005

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: Modelling, Planning and Control (Advanced Textbooks in Control and Signal Processing)*. Springer.

SICK. (2021). V3DR3–60NE31111 | Ranger3: Product data sheet. https://www.sick.com/de/de/industrielle-bildverarbeitung/3d-machine-vision/ranger3/v3dr3-60ne31111/p/p533947. Accessed 25 March 2021.

Siebertz, K., van Bebber, D., & Hochkirchen, T. (2010). *Statistische Versuchsplanung: Design of Experiments (DoE) (VDI-/Buch])*. Springer.

Starke, S., Hendrich, N., & Zhang, J. (2019). Memetic evolution for generic full-body inverse kinematics in robotics and animation. *IEEE Transactions on Evolutionary Computation, 23*, 406–420. https://doi.org/10.1109/TEVC.2018.2867601

Suemitsu, I., Izui, K., Yamada, T., Nishiwaki, S., Noda, A., & Nagatani, T. (2016). Simultaneous optimization of layout and task schedule for robotic cellular manufacturing systems. *Computers & Industrial Engineering, 102*, 396–407. https://doi.org/10.1016/j.cie.2016.05.036

The Imaging Source. (2019). DFK 38UX304: Technical reference manual. https://www.theimagingsource.de/produkte/industriekameras/usb-3.1-farbe/dfk38ux304/. Accessed 25 March 2021.

Trucco, E., Umasuthan, M., Wallace, A. M., & Roberto, V. (1997). Model-based planning of optimal sensor placements for inspection. *IEEE Transactions on Robotics and Automation, 13*(2), 182–194.

Tubaileh, A. S. (2014). Layout of robot cells based on kinematic constraints. *International Journal of Computer Integrated Manufacturing, 28*, 1–13. https://doi.org/10.1080/0951192X.2014.961552

Wang, S., Guo, R., Wang, H., & Vogel-Heuser, B. (2020). Safe three-dimensional assembly line design for robots based on combined multiobjective approach. *Applied Sciences, 10*, 8844. https://doi.org/10.3390/app10248844

Weidemann, D., & Drath, R. (2010). Einleitung. In R. Drath (Ed.), *Datenaustausch in der anlagenplanung mit AUTOMATIONML*: *Integration von CAEX, PLCopen XML und COLLADA* (pp. 1–44, VDI-Buch). Springer-Verlag Berlin Heidelberg.

Woenckhaus, C., & Milberg, J. (1993). Planning of three-dimensional layouts using numerical optimization methods. In *IEEE International Conference on Robotics and Automation 1993, Atlanta, USA, 05/02/1993 - 05/06/1993* (pp. 44–49). IEEE Computer Society Press. https://doi.org/10.1109/ROBOT.1993.291923.

Woenckhaus, C. (1994). *Rechnergestütztes system zur automatisierten 3D-Layoutoptimierung* (Forschungsberichte IWB, Bd. 65). Springer Berlin Heidelberg.

YASKAWA. (2019). MH5 SII / MH5LS II: Compact and versatile robot. https://www.motoman.com/en-us/products/robots/industrial/assembly-handling/mh-series/mh5ls-ii. Accessed 1 April 2021.

Zeng, R., Wen, Y., Zhao, W., & Liu, Y.-J. (2020). View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media, 6*, 225–245. https://doi.org/10.1007/s41095-020-0179-3

Zhang, E.-D., Qi, L.-L., & Murphy, S. (2013). Method and system for optimizing the layout of a robot work cell.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.