

# Simulation Model for AGVs in Production Environments under Consideration of the Facility Layout Problem

Lea Schmitz<sup>\*</sup>, Jonas Stenzel<sup>#</sup>

Institute of Material Flow and Logistics IML, Dept. Robotics and Cognitive Systems, Joseph-von-Fraunhofer-Str. 2-4, 44227 Dortmund, Germany; <sup>\*</sup>*Lea.Schmitz@iml.fraunhofer.de*, <sup>#</sup>*Jonas.Stenzel@iml.fraunhofer.de*

SNE 34(2), 2024, 91-100, DOI: 10.11128/sne.34.tn.10685  
 Selected ASIM SPL 2023 Postconf. Publication: 2023-09-01  
 Received Rev. Improved: 2024-04-22; Accepted: 2024-05-10  
 SNE - Simulation Notes Europe, ARGESIM Publisher Vienna  
 ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

**Abstract.** In Flexible Manufacturing Systems (FMS), an Automated Guided Vehicle (AGV) is often used for material transport. To simulate this system, a production environment with aisles and I/O points is required. The arrangement of individual areas within a production environment can be done by solving the Facility Layout Problem (FLP). However, the previous methods only consider individual aspects and there are hardly any solution methods that consider the entire production system with regard to the material transport. Due to this, a FLP solving method that also considers aspects regarding the material handling system is useful so that the created production environments can be simulated and/or used in practice. Therefore, both topics - FLP and AGVs - should be considered together.

## Introduction

Flexible Manufacturing Systems (FMS) – a specific type of multi-stage production system [10] – often use an automated guided vehicle system (AGV-System) for material transport [12, 11]. One possibility for optimizing AGV-Systems is the arrangement of various facilities, which can represent, for example, machines, storage locations, workplaces, production locations [13, 15], I/O points (where Automated Guided Vehicles (AGVs) can load and unload material), parking areas for AGVs as well as necessary AGV travel areas within a production environment. The problem of arranging these facilities is known as the Facility Layout Problem (FLP) and aims to minimize the costs of material transport of the resulting production environment

so that they are as low as possible [13, 14, 15]. One problem with the previous solution methods of the FLP is that often aspects for the use of a real AGV-System are not taken into account. These include the driving areas of the AGVs, the connection of the I/O points to the AGV driving area and the consideration of routing, dispatching and scheduling for the AGVs [2] on the basis of the generated production environment.

**Reference to simulation** In this paper, a simulation model is developed which, on the one hand, can be used to the requirements of an AGV-System when creating the layout of the production environment (FLP). On the other hand, the dynamics of the AGV-System are already taken into layout by means of an abstract simulation of the AGV routing. For example, it is checked whether the AGV-System is capable of handling all necessary transport orders within a certain period of time.

## 1 State of the Art

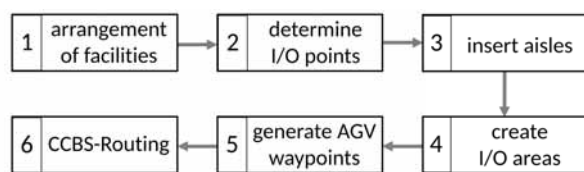
In this section the basics are presented. The FLP is explained first, followed by the AGV-System and all related aspects.

### 1.1 Facility Layout Problem

The FLP positions a fixed number of facilities within a limited area so that the cost of material transport is minimized [13, 15, 14]. The FLP belongs to the NP-hard combination problems and has been researched for decades [15]. Due to this, there is a multitude of solution methods with different approaches.

Many of these only consider selected aspects, such as the determination of I/O points. Steps one to three from Figure 1 are known from the literature.

In [13] a FLP approach for the arrangement of facilities based on a special data structure – Slicing Tree (ST) – was developed. This offers the possibility of inserting an aisle structure in the layout with a simple procedure. In [9] the FLP solution procedure extended by the determination of the I/O points. Thereby, an I/O point could be positioned within one of the four corners of the facility. This is useful for the subsequent connection to the transport aisles – the insertion of the transfer areas – makes sense. In [15] they added the transport aisles depending on the transport relationships and intensities between the facilities within the production environment for material transport. However, this method does not take into account the aspects of AGV routing.



**Figure 1:** Illustration of the steps to generate a FLP layout for the creation of a production environment. Steps one to three can be found in the literature. Steps four to six should be considered with regard to aspects of a material handling system and should also be taken into account during the creation.

To evaluate a layout, the material transport costs  $C$  are determined [13, 14, 15]:

$$C = \sum_{i=1, j=1}^M (t_{ij} \cdot d_{ij}) \quad (1)$$

This sum is defined by the distance  $d_{ij}$  from facility  $i$  to facility  $j$  weighted by the transport intensity  $t_{ij}$ . The transport intensity can be taken from the transport matrix and corresponds to the material units that are transported between the facility pairs.  $M$  is the number of facilities to be placed in the layout. The distance  $d_{ij}$  is calculated differently depending on the FLP approach. The FLP approaches in [13, 7, 8, 6] calculate the distance between the centers of gravity of the facility. With regard to AGVs, this assumption of the travel distance between the centers of gravity is unrealistic. In the method of [9], it is assumed that the AGVs travel along the boundaries of the facility. I/O points are determined which lie at the edge and the distance is calculated between them along the boundaries of the facilities.

The physical space to travel in is missing. In the approach of [14], it is assumed that the travel paths are always in the middle of the corridor. The path-based distance is calculated between the I/O points taking the aisles into account.

## 1.2 AGV-System

An AGV-System consists of one or more AGVs. It can now be found in almost all industries and production areas and is therefore an important part of intralogistics [1]. According to the VDI 2510, AGVs are defined as indoor, floor-based systems with automatically controlled vehicles for the handling of material transport [1]. In order to coordinate the AGVs, the AGV needs to know about the possible routes. One possibility is to create these in the form of topological graphs [19].

**Topological graphs** A topological graph  $G = (V, E)$  is a planar graph with a finite set of nodes  $N$  and edges  $E$  [18]. Each edge connects two nodes. In the context of AGVs, the edges of the topological graph represent paths or roads for the AGV. Intersections and important stops, e.g. stations for loading and unloading material at the facility, are represented by graph nodes. To create the topological graph, an approach further developed at the Fraunhofer IML called Roadmap Graph Creator (RGC), based on [19] is used.

Readers are referred to the literature [20]. The topological graph is used to calculate the distance between the I/O points (path-based distance metric). The simulation model for routing requires the topological graph to determine the paths for the AGVs to execute the transport orders.

**CCBS-Routing** An abstract simulation model is created to take into account the dynamic behavior of the AGVs in the created production environment. For this purpose Continuous-time conflict-based search (CCBS) is used. CCBS is an algorithm for searching paths for multiple AGVs in a defined environment, so that each AGV reaches its destination without causing conflicts with other AGVs [17, 16]. In doing so, a set of transport orders is defined where the start and destination are defined by nodes of the topological graph. For each transport order one AGV is assumed. The paths of the AGVs are selected via the topological graph.

As a result, CCBS returns certain characteristic metrics for the routing of the set of transport orders, including the makespan  $m$ .

$$m = M(\Pi) \quad (2)$$

The makespan corresponds to the time it takes the last AGV to reach its destination. CCBS plans in batches, i.e. no new transport orders can be issued to an AGV until all AGVs have reached their destination [17, 16].

## 2 Requirements regarding AGV-System

Before a solution can be developed, the requirements with regard to layout and topological graph for use within an AGV-System must be defined. Based on these criteria, the solution procedure is further developed so that a production environment model that is well matched to the requirements of the AGV-System can be generated. This model can be used for simulation and/or implemented directly in practice if required.

### 2.1 Layout Requirements

In order for the layout created on the basis of the FLP to be suitable for use within an AGV-System, the following aspects should be fulfilled: Transport aisles for the AGVs between the individual facilities should be defined [14, 15]. Additional several I/O points for each facility should be created, instead of only one I/O point for each facility like in [9]. With regard to application in the AGV-System, only one I/O point can lead to a problem. For example several AGVs arrive at the same facility and at the same time. First of all it must be clarified which AGV is allowed to drive to the I/O point to load or unload the materials. This means waiting for the other AGVs. The waiting AGVs could obstruct other AGVs that are in the process of fulfilling other transport orders. This leads to congestion and delay in the entire FMS. Therefore, it may be necessary if multiple I/O points can be identified. In our approach we use an I/O area for this purpose (see later section 3.1).

### 2.2 Graph Requirements

The topological graph and thus the waypoints for the AGVs are to be generated on the basis of the aisle structure. The topological graph must fulfill two requirements for use within an AGV-System.

First, it must be ensured that from one node every other node in the graph can be reached via the edges. This property can be tested via the graph coherence, if this is one, the requirement is fulfilled [3]. Furthermore, it must be checked whether all facilities also have a connection to the aisle structure via graph nodes and edges. For this purpose, it is checked whether at least one graph node is located within the I/O area.

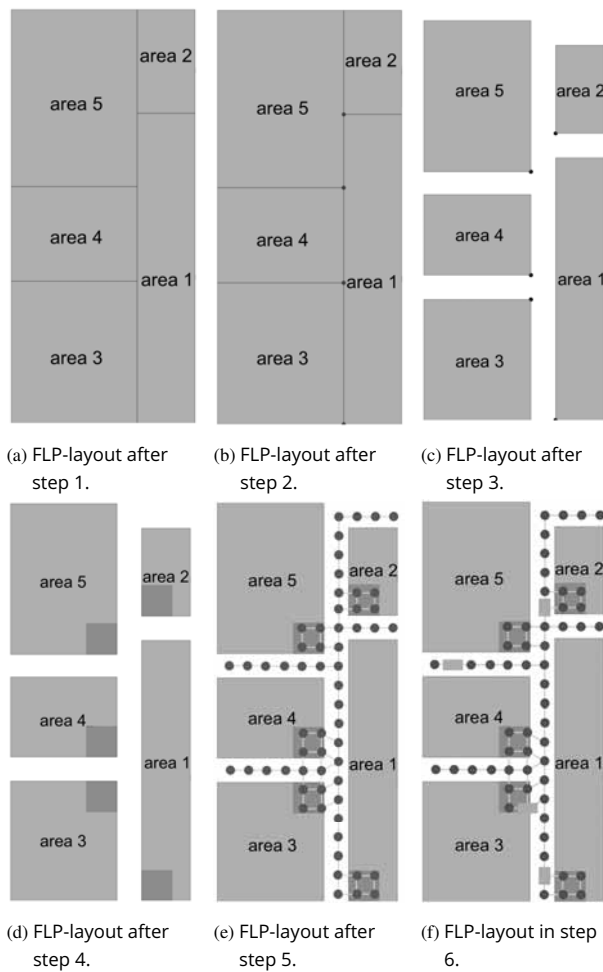
## 3 FLP - Solver

Due to the FLP solution procedures known to us, which do not consider the requirements with regard to the application in an AGV-System, we have developed our own approach. This means that the production environments created can be used in practice. Steps one, two and three (see Figure 2) are known from the literature. Our further development of the FLP solution method includes steps four to six from Figure 2, which take into account the requirements with regard to the AGV-System. For this purpose, the FLP solution procedure of [13] is extended. The individual steps are explained below.

### 3.1 Arrangement of the Facilities and Determination of the I/O Points

In order to solve the FLP, a number of input data are generally required [13]. These include the dimensions of the location support, the number of facilities to be placed, the required area for each facility, the dimensions of each facility and the transport matrix. The transport matrix indicates how much material needs to be transported between the individual facilities [13]. Based on this required random STs are generated in order to determine the final positions of the facilities and thus their arrangement (see Figure 2a).

To determine the I/O points (= stations for AGVs to load and unload material at the facility), the procedure of [9] is applied. First, candidates of I/O points are identified based on the dominant region. For each corner of the facility a node is added to a graph. The boundaries of the facilities are represented by edges. The dominant region consists of one or more contiguous facilities that have at least one vertex or graph node for each facility. These nodes in the dominant region represent the candidates of I/O points. We have resolved the restriction of [9] that a dominant region must be rectangular.



**Figure 2:** Representation of the generated production environment according to the steps from Figure 1.

In our approach, the dominant region can also be a polygon. After the candidate transfer points have been determined, the facility pairs are sorted in descending order according to their transport intensity. For each facility pair, the two I/O points with the smallest distance to each other are selected (see Figure 2b) [9].

### 3.2 Creation of the AGV Driving Areas

The structure of the STs is used to create the driving areas. The idea is based on [13] and was adapted for our purpose. Each cut node in the ST is considered. The right child of the intersection node and all nodes are shifted along the x- or y-axis in a positive direction by a predefined path width. This creates the driving areas for the AGVs (see Figure 2c).

### 3.3 Inserting the I/O Areas

On the basis of the I/O points determined in step 2 (see Figure 2b), I/O areas of a predefined size are inserted. All I/O areas are of the same size (see Figure 2d). The idea is to treat these transfer areas also as driving areas for the AGVs. This automatically generates a set of I/O points for each facility related to the next step: generation of the AGV waypoints. All graph nodes of the topological graph within the I/O areas can be identified as I/O points.

### 3.4 Generation of the AGV Waypoints

The generation of the topological graph is carried out over the open space, consisting of the driving and I/O areas using the RGC software tool (see Figure 2e). It is important that the graph fulfills the two requirements mentioned (see section 2.2), otherwise the created layout cannot be evaluated.

### 3.5 Simulation of the AGVs with CCBS Routing

To evaluate the layout, the CCBS makespan is used in our approach (see eq. 2) and replaces the cost function from the literature (see eq. 1). This leads to the consideration of the dynamic driving behavior of the AGVs and the parallelism of the transport orders to be processed in the layout, which is not considered in other evaluation criteria for the FLPs layout (e.g. in eq. 1). For this purpose a set of transport orders must first be generated. Then, taking into account a time window (see section 3.6), CCBS is executed with the generated transport orders and the makespan is assumed to be the cost of the layout. If CCBS does not find a solution within the time window, twice the longest distance in the layout is calculated for evaluation:

$$c = 2 \cdot \max(d_{i,j}) \quad (3)$$

This can be done, because in CCBS the assumption is made that the AGV has a velocity of 1 m/s [17, 16]. The longest path therefore also corresponds to the makespan. In order to evaluate this solution more negatively, since CCBS has not found a solution, the longest path is considered twice.

### 3.6 Parameters

There are a number of parameters for the FLP-solver that influence the result. In principle, the FLP-solver distinguishes between the necessary input data and the

parameters to be set for the best possible result. The following input parameters are required for each data instance: the number of facilities to place, the dimensions of the entire layout or maximum allowed size for the FLP-layout to be generated e.g. the production hall, the minimal and maximal width and length of each facility, the area for each facility and the transport matrix containing the transport intensities for each pair of facilities.

The parameters to be set for the best possible result can be further differentiated. The first group of parameters influences the layout itself, e.g. the size of the I/O area (see section 3.3). The width of transport aisles (see section 3.2) can also be defined and thus the number of parallel roads within a transport aisle. A time limit is set at two points when creating a layout. When creating the topological graph (see section 3.4), the time limit is determined depending on the size of the layout and the width of the transport aisle. To determine the costs via CCBS (see section 3.5), a time limit of 10 seconds is set, which is based on previous studies [20].

The second group of parameters influences the search for the best FLP-layout, because the number of STs considered is limited so that the runtime remains within a certain period. These parameters are already known from the literature e.g. [13]. An example is the parameter *iterations*. The parameter defines how often a random ST is created for searching a best FLP-layout.

One problem in creating a production environment is finding the optimal parameters so that the runtime remains within reasonable limits and the result is still as satisfactory as possible. To find the right parameters, we focused on the runtime. So that production environments can be created in a reasonable time for us (see Table 2, 3, 4, 5).

## 4 Evaluation

For the evaluation, we have compared several data instances (DI) from different publications and compared their results with our approach. For this we took into account different numbers of steps from the literature to generate a layout (see Figure 1). The evaluations were performed on an AWS server instance EC2 C5A.XLarge [4]. Table 1 shows the four different versions. The specified minimum and maximum side lengths of the individual facilities could not always be adhered due to the selected cuts in the ST.

The limitations for the side lengths are not taken into account in the evaluation. Tables 2 - 5 compare our results with the best results from the literature. The numerical value in the columns "best result" and "our result" represents the material transport cost *C* for the respective DI. The value "nr. of STs" corresponds to the number of valid ST found for the listed run of the FLP-solver.

| FLP-solver | literature    | distance metric                           | considered steps for generating a layout (see Figure 1) |
|------------|---------------|---|---|
| V1         | [13, 7, 8, 6] | center of gravity distance                | 1   |
| V2         | [9]           | contour-based distance                    | 1 and 2   |
| V3         | [14]          | path-based distance                       | 1, 2 and 3  |
| V4         | our approach  | path-based distance via topological graph | all steps seen in Figure 1                              |

**Table 1:** Overview of evaluation applications V1 - V4.

### 4.1 FLP-solver: Arrangement of the Facilities (V1)

Table 2 shows that the results from the literature for the FLP procedure with center of gravity distance. With the exception of data instance AB20, all results are better by an average of 7.53%. It is likely that with further elaboration for data instance AB20 a similar or better result could be obtained.

### 4.2 FLP-solver: Determination of I/O Points (V2)

The results in [9] could not be achieved (see Table 3). One reason for this is the adherence to the dimensions of the layout. While [9] allows layouts with larger dimensions, our approach only allows layouts that lie within the specified dimensions. This is more practical, since the necessary space, e.g. a production hall, is limited. To enlarge the hall afterwards is unrealistic. As before, there is also the chance to achieve better results with further designs.

| data instance | best result ↓ | our approach  |           |                    | deviation ↓ |
|---------------|---------------|---------------|-----------|--------------------|-------------|
|               |               | best result ↓ | nr of STs | runtime [HH:MM:SS] |             |
| OE7           | 131.57 [7]    | 108.78        | 5         | 00:00:06           | -17.32%     |
| OE8           | 242.73 [8]    | 220.94        | 7         | 00:00:16           | -8.98%      |
| OE9           | 235.84 [7]    | 207.51        | 9         | 00:00:25           | -12.01%     |
| VC10          | 19994.10 [6]  | 19183.98      | 6         | 00:02:58           | -4.05%      |
| BA12          | 8021 [13]     | 8071.28       | 34        | 00:02:21           | +0.63%      |
| BA14          | 4628.84 [13]  | 4469.25       | 19        | 00:04:27           | -3.45%      |
| AB20          | 5073.82 [6]   | 67725.68      | 1         | 00:10:46           | +92.51%     |

**Table 2:** Results of FLP-solver V1: arrangement of the facilities.

| data instance | best result in [9] ↓ | our approach  |          |                    | deviation ↓ |
|---------------|----------------------|---------------|----------|--------------------|-------------|
|               |                      | best result ↓ | nr of ST | runtime [HH:MM:SS] |             |
| OE7           | 21.64                | 36.85         | 7        | 00:00:12           | +41.28%     |
| OE8           | 52.09                | 66.64         | 6        | 00:00:25           | +21.83%     |
| OE9           | 53.29                | 76.32         | 7        | 00:00:32           | +30.18%     |
| VC10          | 3097.91              | 18134.46      | 6        | 00:04:46           | +82.92%     |
| BA12          | 3089.91              | 4747.06       | 27       | 00:08:25           | +34.91%     |
| BA14          | 2188.33              | 3318.85       | 21       | 00:15:19           | +34.06%     |
| AB20          | 1185.99              | 89280.45      | 2        | 04:02:16           | +98.67%     |

**Table 3:** Results of FLP-solver V2: arrangement of the facilities and determination of the I/O points.

### 4.3 FLP-solver with Aisles (V3)

The comparability between [14] and our approach is difficult due to some differences, e.g. the procedure for inserting waypoints and the individual parameters, which additionally influence the result [13].

In comparison with the four published data instances in [14], data instance BA14 performs best. Here, too, there is a chance to achieve better results with further versions.

### 4.4 FLP-solver: I/O areas and CCBS-Routing (V4)

Table 5 presents the results for our own metric, which takes into account CCBS routing and thus parallelism in the AGV-System and FMS.

Just as before there is a chance for better results with further executions, since with each execution random data instances are generated. Based on this random ST the optimization is carried out [13].

## 5 Discussion

In the following we will pick out some aspects and look at them more closely. To do this, we will first examine how much time it takes to create a layout (see section 5.1). Then we will look at the question of why STs are created randomly (see section 5.2). Finally, we will briefly show that the approach cannot create standard layouts exclusively (see section 5.3).

### 5.1 A Runtime Analysis - Data Instance BA12

This section presents the results of a runtime analysis of data instance BA12. For this purpose, the runtime analysis is split into two parts. The first part aims to analyze which steps for creating a layout (see Figure 1) take the longest in terms of runtime. In the second part, some experiments are carried out to show the effects of the optimization and some input parameters.

For the first part, the optimization itself plays no role. Only a layout is created using a ST. To do this, a ST is first generated randomly. This is then passed to all four FLP-solver versions (V1- V4).

| data instance | best result in [14] ↓ | our approach  |           |                    | deviation ↓ |
|---------------|-----------------------|---------------|-----------|--------------------|-------------|
|               |                       | best result ↓ | nr of STs | runtime [HH:MM:SS] |             |
| VC10          | 7116.14               | 27315.12      | 15        | 04:43:27           | +73.95%     |
| BA12          | 5561.12               | 8105.33       | 1         | 00:15:28           | +31.39%     |
| BA14          | 3784.47               | 3964.14       | 1         | 00:20:27           | +4.53%      |
| AB20          | 1529.63               | 89516.26      | 3         | 08:26:22           | +98.29%     |

**Table 4:** Results of FLP-solver V3 with aisles.

| data instance | our approach  |           |                    |
|---------------|---------------|-----------|--------------------|
|               | best result ↓ | nr of STs | runtime [HH:MM:SS] |
| VC10          | 59,31         | 2         | 04:21:18           |
| BA12          | 14,85         | 1         | 00:34:03           |
| BA14          | 15,15         | 1         | 00:41:37           |
| AB20          | 62,15         | 1         | 08:17:12           |

**Table 5:** Results of FLP-solver V4 with aisles and CCBS-Routing.

This increases the comparability with regard to the runtimes during the creation of the layouts.

The total runtime is recorded as well as the runtime for certain steps (see Figure 1) during layout creation. Step 4 is not considered because the final position is already given by the I/O point and thus this step consists only of cutting rectangular areas. As the previous time display [HH:MM:SS] is not suitable at this point, all running times determined are recorded in seconds. For this evaluation, a timeout of 30 seconds is selected for the creation of the topological graph. The time to start and end the ROS-nodes to create the topological graph is also measured, but the pure creation is limited to 30 seconds. The CCBS has a time limit of 10 seconds.

Table 6 shows an example of an evaluation for the scenario described. As expected, the time required to create the random ST is the same, as the same ST is used for all FLP-solver versions (see column initialization in Table 7). No time is required for steps 2, 3, and 5 with V1, as this FLP-solver version does not carry out these steps. The same applies to V2 with steps 3 and 5. Each step takes approximately the same amount of time to complete. Obviously, the most time-consuming step is the creation of the topological graph. The more steps are carried out to create the layout, the more time is required overall (see column total time in Table 7).

To find out whether the runtimes from Table 6 are average values or exceptions, the scenario described is

repeated 100 more times. The average value is determined from these 100 evaluations in Table 7. When looking at the results, it was noticed that some of the random STs are not suitable for creating a layout. These invalid STs were filtered out in order to recalculate the average values for the 82 remaining STs (see Table 7 at the bottom). This gives a clearer picture of the runtime to create a layout. It is striking that CCBS was not even used in all 100 results. The time to calculate the costs in V4 is lower than 10 seconds (see Table 7). This phenomenon should be analyzed in detail in the future. Just like the fact that some of the randomly generated STs for the FLP-solvers V2, V3 and V4 are invalid. Perhaps the solution space can be narrowed down even more beforehand. This would lead to a reduction in the total runtime in connection with the optimization, as invalid STs are no longer considered from the outset. To summarize the first part of the runtime analysis: the values in Table 6 correspond to the general average from Table 7 (lower part) when considering the valid STs from the initialization. In the case of data instance BA12 in combination with the selected parameters, a layout can be created in all FLP-solver versions within 60 seconds. The longest part in terms of runtime is the creation of the topological graph. The aforementioned phenomena should be investigated further in the future.

The second part of the runtime analysis is about the optimization time. Again for each FLP-solver (V1, V2, V3, V4) the same ST is used. In principle, there are no major changes, but (in contrast to the previous analysis) the optimization of the created layout is carried out after each step. The time required for the optimization and how many different layouts (= STs) are created and checked during the optimization are examined.

In the first experiment, the parameter `iterations` is set to one. Table 8 shows the results. Compared to the previous results (see Table 2, 3, 4, 5), Table 8 shows a significantly longer total runtime for data instance BA12. This highlights a problem with the heuristic procedure: depending on which ST is randomly created at

| FLP-solver | time in seconds |                         |                        |                    |                        |                   |                           |
|------------|-----------------|-------------------------|------------------------|--------------------|------------------------|-------------------|---------------------------|
|            | initial         | arrangement<br>(step 1) | I/O points<br>(step 2) | aisles<br>(step 3) | topo graph<br>(step 5) | costs<br>(step 6) | total time<br>(steps 1-6) |
| V1         | 0.02362         | 0.0017                  | 0                      | 0                  | 0                      | 0.0015            | 0.00321                   |
| V2         | 0.02362         | 0.00132                 | 0.18767                | 0                  | 0                      | 0.00578           | 0.19479                   |
| V3         | 0.02362         | 0.00173                 | 0.18379                | 0.00814            | 47.89754               | 0.02436           | 48.11566                  |
| V4         | 0.02362         | 0.00144                 | 0.18852                | 0.00832            | 49.2727                | 0.08869           | 49.55979                  |

**Table 6:** An example of an evaluation for the first part of the runtime analysis of data instance BA12. For each FLP-solver version (V1 - V4), the same ST – randomly generated ST in the initialization – is used to create a layout. The total runtime and selected steps for creating the layout (see Figure 1) are recorded. The time is displayed in seconds.

| FLP-solver | nr of evaluations | averaged time in seconds |                         |                        |                    |                        |                   |                           |
|------------|-------------------|--------------------------|-------------------------|------------------------|--------------------|------------------------|-------------------|---------------------------|
|            |                   | initial                  | arrangement<br>(step 1) | I/O points<br>(step 2) | aisles<br>(step 3) | topo graph<br>(step 5) | costs<br>(step 6) | total time<br>(steps 1-6) |
| V1         | 100               | 0.01822                  | 0.00154                 | 0                      | 0                  | 0                      | 0.00149           | 0.00304                   |
| V2         | 100               | 0.01822                  | 0.00142                 | 0.14851                | 0                  | 0                      | 0.00845           | 0.18018                   |
| V3         | 100               | 0.01822                  | 0.00144                 | 0.14793                | 0.00635            | 39.46056               | 0.02364           | 39.66755                  |
| V4         | 100               | 0.01822                  | 0.00166                 | 0.1525                 | 0.00663            | 40.71342               | 0.07399           | 40.96945                  |
| V2         | 82                | 0.01846                  | 0.00141                 | 0.18111                | 0                  | 0                      | 0.0103            | 0.19284                   |
| V3         | 82                | 0.01846                  | 0.00143                 | 0.1804                 | 0.00774            | 48.12264               | 0.02883           | 48.34702                  |
| V4         | 82                | 0.01846                  | 0.00168                 | 0.18598                | 0.00808            | 49.65051               | 0.09023           | 49.9366                   |

**Table 7:** Result of the first part of the runtime analysis for data instance BA12. The average times (in seconds) are shown on the basis of 100 evaluations as in Table 6.

the beginning, many STs may be invalid and therefore not considered. Furthermore, the search for the local minimum for BA12 converged relatively late compared to other results. As result, significantly more layouts were viewed than in the previous search (see Table 8).

| FLP-solver | layout     |              |           |
|------------|------------|--------------|-----------|
|            | creation   | optimization | nr of STs |
| V1         | [00:00:01] | [00:00:01]   | 215       |
| V2         | [00:00:01] | [00:01:02]   | 215       |
| V3         | [00:00:48] | [07:34:20]   | 644       |
| V4         | [00:00:50] | [02:59:44]   | 236       |

**Table 8:** Runtime analysis of data instance BA12 with layout optimization (iterations=1). The time required to create the first layout is also listed (for comparison with the values from Table 7) as well as the time required to optimize this created layout. A number of layouts are created during the optimization process (see column "nr of STs").

In the second experiment the parameter *iterations* is set to five. This will be used to analyze the impact of the number of iterations in the runtime. Table 9 shows the results. As expected, the number of STs (=layouts) seen has increased: However, only for V1 and V2. For the other two FLP-solver versions, the number of STs has actually fallen. This is also accompanied by the lower runtime. It is likely that the 5 randomly created STs have led to faster convergence to the local minimum than with the previous 3 randomly created STs. The random creation of STs at the start of the approach is always disruptive in this case, as the influencing factors – in this case parameter *iterations* – cannot be analyzed in a meaningful way. This raises the question of how useful the random generation of STs is (see section 5.2).

## 5.2 The Complexity of Creating Layouts

In this section we discuss the number of possible STs for a given data instance. Figure 3 shows the number of possible STs (y-axis) for a data instance with differ-

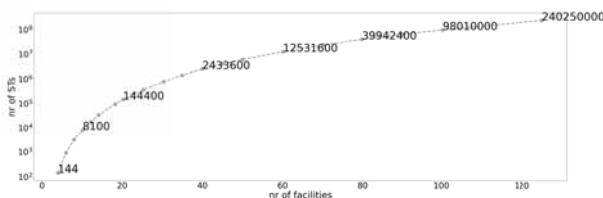


| FLP-solver | layout     |              |           |
|------------|------------|--------------|-----------|
|            | creation   | optimization | nr of STs |
| V1         | [00:00:01] | [00:00:01]   | 338       |
| V2         | [00:00:01] | [00:01:34]   | 324       |
| V3         | [00:00:48] | [03:29:36]   | 635       |
| V4         | [00:00:50] | [02:32:53]   | 209       |

**Table 9:** Runtime analysis of data instance BA12 with layout optimization (iterations=5). The time required to create the first layout is also listed (for comparison with the values from Table 7) as well as the time required to optimize this created layout. A number of layouts are created during the optimization process (see column "nr of STs").

ent facilities (x-axis). The number of possible STs is growing exponentially.

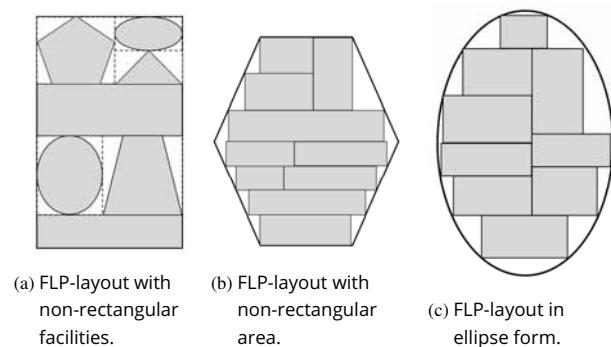
It can be observed that the number of STs increases with the number of facilities. For example there are 8100 possible STs for a layout with 10 facilities. For a data instance with 100 facilities 98010000 possible STs. The number of possible STs is growing exponentially. As soon as the number of facilities increases, the ST is getting bigger and thus the runtime increases. Therefore we decided to follow Scholz's approach [13] with the creation of random STs and test them to find a local minimum FLP-layout. An alternative would be to calculate the costs for all STs. Based on data instance AB20 with 20 facilities and a runtime of approx. 160 seconds per ST, a runtime of 267.4 days is calculated from the 144400 possible STs. First of all, it was important to create a layout that took material transport aspects (e.g. I/O points, transport aisles and routing) into account. In future, research can be carried out into how the topological graph can be created more quickly so that more layouts can be considered in less time. This would mean that it would no longer be a decision of benefit vs. runtime.



**Figure 3:** The number of STs in addition to the number of facilities due to a data instance.

### 5.3 How to Create Non-rectangular Layouts?

With small changes, layouts with non-rectangular facilities and also non-rectangular layouts can be created. To create layouts with non-rectangular plants, a list of nodes of the actual shape must also be stored for each leaf node of the ST. When reading in the input data, a rectangle is placed around each non-rectangular shape. To create layouts with non-rectangular shapes, a list of points (or a function) describing the desired shape is passed instead of the previous two points (min, max) of the area. Figure 4 shows some examples. The vertical and horizontal alignment due to the ST is maintained.



**Figure 4:** Representation of possible FLP-layout without the limitation of rectangular shapes.

## 6 Conclusion

In summary, it can be said that in this publication a practical FLP approach for the generation of layouts for production environments taking into account the requirements for the use in an AGV-System is presented. Each resulting layout of our approach can be used in an AGV-System.

For future research work, on the one hand, optimization should be carried out with regard to the runtime, so that more layouts can be calculated in less time. This increases the probability of a good result. Furthermore, some aspects should be optimized further, such as the question whether transfer areas should be chosen that are adapted to the transport intensities of the facilities. Thus, facilities with high transport intensities could be approached by more AGVs at the same time. Also the procedure for determining the I/O points and thus the positioning of the I/O areas can be adapted, as it is based on the procedure of [9], it is also possible to drive along the outer edges (=boundaries of the facilities).

In reality, however, after insertion of the aisle structure, this is hardly possible, as additional space would be required. Furthermore, the routing itself can be exchanged, for example by using a more practical anytime routing method.

In contrast to the stacked planning CCBS [17, 16], with anytime routing, e.g. CARP [5], an AGV can directly receive a new transport order as soon as the previous one has been completed, without having to wait for the completion of transport orders from the other AGVs.

## References

- [1] Association of German Engineers / Verein Deutscher Ingenieure: VDI 2510, Fahrerlose Transportsysteme (FTS). Berlin: Beuth 2005.
- [2] Vivaldini, K. C.; Rocha, L. F.; Becker, M.; Moreira, A. P.: Comprehensive review of the dispatching, scheduling and routing of AGVs. In CONTROLO'2014 – proceedings of the 11th Portuguese conference on automatic control (2015). pp. 505-514. Springer International Publishing.
- [3] Diestel, R.: Graphentheorie. Bd. 5. Springer Spektrum, 2017. ISBN: 978-3-662-53633-9. url: <https://www.springer.com/de/book/9783662536339>
- [4] Amazon Web Services, Inc. bzw. Tochtergesellschaften des Unternehmens: Amazon EC2 C5-Instances (2023). url: <https://aws.amazon.com/de/ec2/instance-types/c5/>
- [5] Mors, A. ter.; Zutt, J., Witteveen, C.: Context-Aware Logistic Routing and Scheduling. In ICAPS (2007), pp. 328 – 335.
- [6] Wong, K.; Komarudin, K.: Solving facility layout problems using Flexible Bay Structure representation and Ant System algorithm. In Expert Syst. Appl. 37 (2010) pp. 5523-5527.
- [7] Castillo, I; Westerlund, J.; Emet, S.; Westerlund, T.: Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods. In Computers and Chemical Engineering 30.1 (2005), pp. 54-69.
- [8] Castillo, I.; Westerlund, T.: An  $\epsilon$ -accurate model for optimal unequal-area block layout design. In Computers and Operations Research. 32 (2005). pp. 429-447.
- [9] Friedrich, C.; Klausnitzer, A.; Lasch, R. Integrated slicing tree approach for solving the facility layout problem with input and output locations based on contour distance. In European Journal of Operational Research, 270(3) (2018). pp. 837-851.
- [10] Hehenberger, P.: Computergestützte Fertigung. Eine kompakte Einführung. Berlin, Heidelberg: Springer 2011.
- [11] Arifin, R.; Egbelu, P.J.. Determination of vehicle requirements in automated guided vehicle systems: A statistical approach. In: Production Planning and Control 11.3 (2000), S. 258–270.
- [12] Rajotia, S.; Shanker K.; Batra, J.L. Determination of optimal AGV fleet size for an FMS. In International Journal of Production Research 36.5 (1998), S. 1177–1198.
- [13] Scholz, D. *Innerbetriebliche Standortplanung. Das Konzept der Slicing Trees bei der Optimierung von Layoutstrukturen*. Wiesbaden: Gabler Verlag 2010.
- [14] A. Klausnitzer. A Scatter Search Approach for the Facility Layout Problem with Aisle Design. In: Operations Research Proceedings 2018. S. 435–441.
- [15] Klausnitzer, A.; Lasch, R.. Extended Model Formulation of the Facility Layout Problem with Aisle Structure. In Logistics Management. Lecture Notes in Logistics. Springer, Cham 2016. S. 89-101.
- [16] Andreychuk, A.; Yakovlev, K.; Surynek, P.; Atzmon, D.; Stern, R. Multi-agent pathfinding with continuous time. In Artificial Intelligence, 305 (2022). S. 103662.
- [17] Andreychuk, A.; Yakovlev, K.; Boyarski, E.; and Stern, R.: Improving continuous-time conflict based search. In: Proceedings of the AAAI Conference on Artificial Intelligence Vol. 35 (2021). S. 11220-11227.
- [18] Archdeacon, D. Topological graph theory. In A survey. Congressus Numerantium 115.5-54 (1996), p. 18.
- [19] V. Digani, L. Sabattini, C. Secchi and C. Fantuzzi. An automatic approach for the generation of the roadmap for multi-AGV systems in an industrial environment. In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, Chicago, IL, USA, 2014, pp. 1736-1741, doi: 10.1109/IROS.2014.6942789.
- [20] J. Stenzel and L. Schmitz. Automated Roadmap Graph Creation and MAPF Benchmarking for Large AGV Fleets. In 8th International Conference on Automation, Robotics and Applications (ICARA), 2022, Prague, Czech Republic, 2022, pp. 146-153, doi: 10.1109/ICARA55094.2022.9738574.