eISSN 2783-6851

VILNIUS TECH
Vilnius Gediminas Technical University

NEW TRENDS in COMPUTER SCIENCES

2024
Volume 2
Issue 2
Pages 101–116
https://doi.org/10.3846/ntcs.2024.22160

PREDICTION OF SIMULATED FACTORY LAYOUT THROUGHPUT USING ARTIFICIAL INTELLIGENCE

Patrick ESCHEMANN ✉, Astrid NIEẞE, Jürgen SAUER

Department of Computer Science Oldenburg, Universität Oldenburg, Oldenburg, Germany

Article History:
- received 22 October 2024
- accepted 13 December 2024

Abstract. The use of artificial neural networks for the optimisation of factory layouts is not a common practice, primarily due to the challenge of collecting sufficient layout data to form datasets for artificial intelligence (AI) model training. This paper presents a supervised learning method derived from a PhD thesis that employs neural networks to assess factory layouts. The training data is generated using a random layout algorithm, which is capable of producing numerous layouts. These layouts are then labeled through a discrete event simulation. The combination of layouts and simulation metrics serves as the training basis for the neural network. The AI framework integrates a convolutional neural network with a multilayer perceptron, which is capable of handling both tabular and image data. Ultimately, this allows us to calculate of the simulated throughput.

Keywords: discrete simulation, AI-supported simulation, neural network, transportation, decision support systems.

✉Corresponding author. E-mail: *patrick.eschemann@googlemail.com*

1. Introduction

This paper is an extension of work initially presented at the conference ISC2024[1]. It presents the interim result of an ongoing PhD thesis, which will be completed by the end of 2024 (Eschemann et al., 2024). This paper introduces an artificial intelligence (AI) framework designed to replace traditional simulation methods for evaluating factory layouts. The objective of the current research is to incorporate a trained artificial neural network (ANN) into metaheuristic algorithms, where simulations are typically employed used to evaluate factory layouts. This research also builds on a previous study, which focused on optimising a single factory layout (Eschemann et al., 2021). To extend the approach, the present study trains an ANN across layouts with different configurations in terms of number of included units to be located. The selection of an appropriate learning method is of paramount importance for the delivery of precise and dependable evaluations. The three main categories of learning, namely supervised, unsupervised, and reinforcement learning, offer distinctive advantages contingent on the characteristics of the dataset and objectives. Supervised learning is identified as the most suitable due to its effectiveness in handling regression problems and its compatibility with the structured dataset, which is derived from a layout generator and discrete event simulation, resulting in a substantial amount of data.

[1] Proceedings are expected to be published here: https://www.eurosis.org/cms/?q=taxonomy/term/26

For this study, a random layout generator was developed to produce a wide variety of layouts automatically. These layouts include a number of variables, such as the number of machines, types of machines, buffer sizes, manufacturing times, layout dimensions, and loading and unloading times and capacities. In contrast to the generation of optimal layouts, which involves the strategic placement of facilities to enhance throughput and reduce material handling costs, this generator produces random configurations by placing operational entities (OE) that represent machines, storage areas, and logistics elements as squared black boxes. The challenge of determining the optimal composition of the OE within the boundaries of a layout is known as the Facility Layout Problem (FLP). As more constraints are added, the NP-complete problem becomes increasingly complex, leading to exponential growth in computational time.

Given the NP-complete nature of the FLP, it is common practice to employ metaheuristic solution methods in order to optimise layouts. In the current research landscape, the majority of approaches rely on genetic algorithms (GA), with simulated annealing and tabu search methods also being popular. These conventional metaheuristic techniques have been pivotal in addressing complex optimisation challenges in factory layout design, as noted in a review study (Hosseini-Nasab et al., 2018).

However, as we move away from these established methods, it is a noticeable gap in the use of advanced AI-based techniques in factory layout optimisation. Despite the proven success of metaheuristic algorithms, there is still limited exploration and application of advanced AI methods like deep learning. This gap is particularly evident when we consider the potential and demonstrated benefits of AI in other research areas. Unfortunately, AI-driven methods are not yet extensively explored in the context of the FLP. This presents a significant opportunity to utilize AI's strengths in generalising and tackling complex problems, such as those encountered in the FLP (Burggraef et al., 2021).

Several factors might explain the limited use of AI in this area, including insufficient training data, a lack of interdisciplinary expertise to integrate AI with traditional manufacturing methods, and the complexity of AI algorithms, which poses challenges in implementation and optimization for specific industrial applications (Burggraef et al., 2021, p. 15). Furthermore, the high costs associated with developing and implementing AI solutions, along with concerns about their reliability and predictability in critical production environments, may further impede widespread adoption in factory layout optimization.

In order to overcome the difficulties presented by the FLP and the limited use of AI in this field, this paper proposes a concept focused on developing an ANN to evaluate layouts, which would replace the traditional event-driven simulation approach. In the following chapter (Section 2) reviews related work, after which the FLP is introduced (Section 3). The following Section 4 explains concept, which includes three main components: a factory layout generator capable of creating numerous layouts, a simulator for evaluating these layouts, and a supervised learning-based neural network trained to mimic the simulation process and estimate throughput. The fourth section (Section 5) presents the results of experiments conducted to evaluate the effectiveness of the network win learning and predicting the efficiency of various layouts for five different factory configurations, ranging from four to eight machines, to effectively learn and predict the efficiency of various layouts without relying on conventional simulation techniques. Section 6 concludes the paper.

2. Related work

This section provides a review of the literature on factory layout optimization, focusing on the application of AI to solve the FLP. A comprehensive literature search was conducted in *ResearchGate* and *Google Scholar* using a variety of search terms in both English and German. The search strategy combined specific and broad keywords, covering various terms and translations related to the aforementioned field, and focused on engineering and computer science literature following Ball and Tunger (2005) and Brocke et al. (2009) guidelines (Ball & Tunger, 2005; Brocke et al., 2009).

The literature search was further cross-referenced with the findings of a recent study by (Burggraef et al., 2021), which specifically investigated the use of AI in addressing the FLP. Burggraef et al. examined 1,290 articles, meticulously selected from an initial pool of 11,851 articles across nine databases, and supplemented this with 134 articles identified through snowball sampling. Their analysis revealed 22 articles that specifically discussed the application of AI techniques to solve FLP, meeting the inclusion criteria for relevance to this review. Among these 22 publications, nine employed supervised learning, eleven utilized unsupervised learning, and two focused on reinforcement learning as the primary techniques for tackling FLP challenges.

2.1. Machine learning for solving the FLP

Jaber et al. (2007) pointed out a potential pitfall of GAs: their tendency to get stuck in evolutionary dead ends, much like a species can evolve into an unfavourable niche in nature (Jaber et al., 2007). As they put it:

> "While the great advantage of GA is the fact that they find a solution through evolution, this is also the biggest disadvantage. Evolution is inductive; in nature life does not evolve towards a good solution but it evolves away from bad circumstances. This can cause a species to evolve into an evolutionary dead end (Jaber et al., 2007)."

To overcome this limitation, they augmented their GA with a learning module called KEP (Keeping Efficient Population). This module acts like a guide, comparing past and future generations of solutions to steer the GA towards more promising areas of the solution space, enabling a more effective and balanced exploration.

In a different vein, Rummukainen et al. (2018) proposed a novel approach to the FLP that departs from traditional mathematical modelling in favour of ML (Rummukainen et al., 2018). Their algorithm learns from expert-designed layouts of similar factories, using a "similarity model" to assess how closely a proposed layout resembles these proven examples. This approach frames the target layout as a Multi-Floor Layout Problem (MFLP), drawing on the wisdom embedded in expert layouts. While their approach proved successful for small-scale datasets, the authors emphasized the need for more extensive data to train more accurate and robust models.

2.2. Deep learning for solving the FLP

Tsuchiya et al. presented a pioneering effort using an artificial ANN to solve the Quadratic Assignment Problem (QAP), a key challenge in optimizing facility distributions on a grid (Tsuchiya et al., 1996). Their approach involved minimizing a defined “energy” function through gradient descent, where each network node, representing a grid position, is assigned weights based on the Manhattan distance to facilitate lower transportation costs for closer facilities.

Tam and Tong introduced a hybrid approach, combining a ANN with a GA to optimize the positioning of tower cranes and supply points by predicting lift times to minimize transport durations (Tam & Tong, 2003). Their method demonstrated how AI could enhance traditional optimization techniques in construction logistics.

Garcia-Hernandez et al. (2018) explored a GA for generating factory layout solutions, which were then evaluated by experts (Garcia-Hernandez et al., 2018). The results from this human-expert evaluation were used to train an artificial neural network, effectively digitizing expert knowledge to assess new layout instances. This approach emphasized the integration of human expertise into the AI-driven optimization process, showcasing a blend of human intuition and machine efficiency.

Fast forward to 2022 publication “A Study of Throughput Prediction using CNN over Factory Environment”, Hou et al. (2022) introduced an approach to predict factory throughput using Convolutional Neural Networks (CNNs) (Hou et al., 2022). Focusing on overcoming the challenges associated with the centralized distribution of data, which can hinder prediction accuracy, they propose a target vectorization technique within the CNN framework. This methodology significantly enhances prediction accuracy, providing valuable insights for the integration of ML in smart manufacturing and IoT applications, especially in improving the reliability of wireless communication for factory productivity.

Also in 2022, Ikeda et al. published “Towards Automatic Facility Layout Design Using Reinforcement Learning”, which introduces a mechanism for optimizing the arrangement of OEs by accurately representing their physical characteristics (Ikeda et al., 2022). Their Reinforcement Learning (RL) algorithm exhibited a preference for placing larger units before smaller ones during experiments. Furthermore, they observed that the algorithm learned more effectively with a continuous influx of new information, rather than being repeatedly fed the same data.

The reviewed literature aligns with a 2021 analysis by Burggraef et al. (2021), which highlights the underrepresentation of ML in FLP research. Their study found only nine studies employing supervised learning, and none utilizing it as a direct solution for FLPs. This shortage may be attributed to the difficulty in obtaining appropriate labeled data, as FLPs often involve unstructured and incomplete information. Additionally, the inherent complexity and unstructured nature of FLPs, coupled with the subjective nature of layouts determined by expert judgment, make suitable training data challenging to identify. The primary challenge lies in the NP-hardness of FLPs, which restricts optimal solutions for layouts with more than 15 units. However, this limit may be pushed further with future advances in computational power, as noted in Burggraef’s et al. analysis.

The relevance of the aforementioned research for this study highlights the importance of combining metaheuristics and ML approaches for approaching FLPs. GA’s ability to generate layouts, paired with ANN’s predictive capacity, opens opportunities for supervised learning approaches. A major challenge in this context is the availability of labeled data, which can be addressed through event-driven simulation to generate training datasets, including metrics like throughput. Despite factory layouts being inherently represented as images CNN have not been widely applied. They could offer further analytical methods by identifying patterns such as material flows or bottlenecks, by detecting local and spatial patterns. Furthermore, CNNs' translational invariance enables them to recognize identical patterns regardless of their position in the layout. Unlike multilayer perceptron (MLP) architectures, CNNs are not restricted to fixed input vectors, making them adaptable to various layout configurations.

3. Facility layout problem

The Facility Layout Problem (FLP) involves determining the most efficient arrangement of operational units within a designated space. FLP is a complex process that is divided in different levels of planning (Bochmann, 2018):

1. *General Layout Planning*: At this high-level stage, the overall concept of the facility's spatial structure is developed. It includes defining the major production areas and support zones, with a focus on the strategic distribution of space without diving into specific details.
2. *Macro Layout Planning* (also known as *Block Layout Planning*): Here, the facility's general areas are divided into distinct blocks, organizing major functional units such as departments or production sections. The aim is to optimize the proximity of units to ensure efficient material flow between them.
3. *Detailed Layout Planning*: At this stage, precise decisions are made regarding the placement of individual machines, workstations, and transportation pathways within the defined blocks.

Facilities can be classified as *uniform* or *non-uniform* in shape. Non-uniform layouts introduce added complexity, as they must adhere to specific geometric constraints–namely, at least one corner must form an angle of 270 degrees or more (Drira et al., 2007). Beyond the shape of the layout, the complexity increases further when considering different *material flow types*. These flows can vary based on the spatial arrangement and transport systems. Common material flows include linear, looped, or grid-based movement patterns, which influence how efficiently goods and materials are transported across the facility used (Hosseini-Nasab et al., 2018). This study focuses on a specific layout category known as the Open Field Layout Problem (OFLP), which is characterized by an absence of predefined material flows such as circular or linear patterns, see Figure 1.

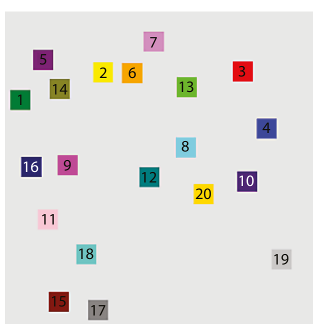

Figure 1. Exemplary layout with 20 factory units

4. Conceptual formulation

Figure 2 shows the concept overview regarding the AI supported layout optimization.

The end-to-end overview consists of three main stages which are explained in more detail in the following subsections. The first stage is the data creation, utilizing a layout generator that can produce arbitrary layout variations. Further components of this stage are a transport matrix that reflects the interconnections of the factory units and an event-driven simulation, designed to estimate the theoretical throughput of each generated layout. The second stage involves training the ANN, which is structured as a combination of a CNN for processing image data and an MLP for handling tabular data. The final stage integrates the trained network into a metaheuristic optimization algorithm (genetic algorithm in this case). A related approach utilizing reinforced learning was recently published by Klar et al. in 2023 (Klar et al., 2024).

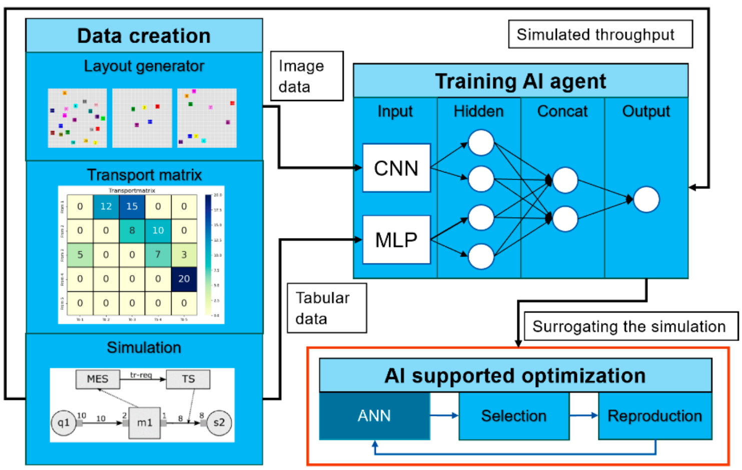

Figure 2. Conceptual overview for the framework end-to-end

4.1. Random layout creation

In previous research GA were used to generate layout data (Azimi & Soofi, 2017; García-Hernández et al., 2014). While GAs explore a broad search space, they tend to focus on areas with previously successful solutions, neglecting less efficient layouts that are required for a full comprehensive problem representation. The underrepresented "bad" layouts are important for training ANNs, as they enhance the model's generalization ability. The datasets used in related studies were small – Garcia-Hernandez et al. used only 365 samples, and Azimi and Soofi used 24 samples – leading to a high risk of overfitting (Azimi & Soofi, 2017; Garcia-Hernandez et al., 2018). For this study, access to a larger dataset of layout variations is prioritized. A study by Sun et al. (2017) demonstrated that ANN accuracy improves with the size of the training dataset, even when model architecture and optimization techniques remain constant (Sun et al., 2017). With this in mind, a Monte Carlo-based layout generator was developed to create representative data by randomly sampling from a probability distribution, see Figure 3.

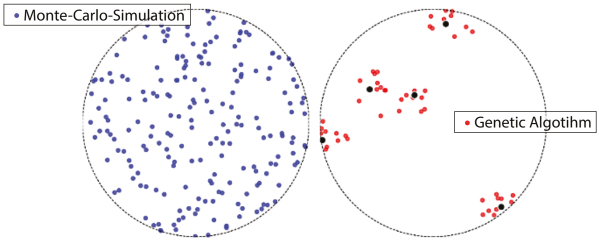

Figure 3. Monte Carlo simulation versus genetic algorithm

Unlike heuristic approaches like GA, Monte Carlo simulation covers a broader range of the solution space, generating independent layouts without relying on previous iterations. Factory layouts are typically proprietary and time-intensive to digitize for simulation or digital processing. Since supervised learning requires a large volume of data, a random layout generator was implemented. Each generated layout includes at least three types of facilities: an input stock, output stock, and one machine. For layouts with more than three units, the generator can create various combinations of machine types and numbers. Dependencies between machines are established using a random transport matrix. In addition to generating layouts, the layout generator produces supplementary data for each layout, including:

- Number of OE X and Y positions of OE centres.
- Number of transport units and transport matrix.
- Number of OEs.

4.2. Discrete event simulation

The generated factory layouts must be labeled for use in training the ANN. This process is accomplished through the use of an event-driven simulation. Figure 4 illustrates the simulation model.

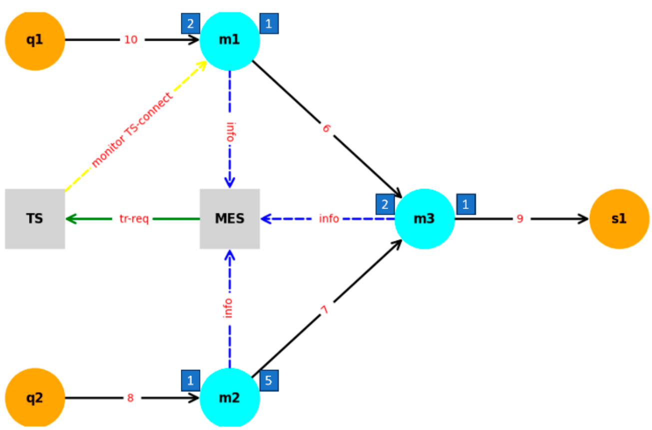

Figure 4. Simulation graph

The simulation graph models a factory layout focused on the material flow between sources (q1, q2), machines (m1, m2, m3), and sinks (s1). The sources and sinks model the input and output storage areas of a factory. In this layout, each machine serves as a primary processing unit, material flows from the sources to the machines and finally to the sinks. The nodes in the graph are color-coded: orange for sources and sinks, cyan for machines, and light gray for the Manufacturing Execution System (MES) and Transport System (TS). In the simplest example, there is one source (q1) providing material to machine m1. After processing, the material is transported to a sink (s1). The flow of material between the components is represented by directed edges indicating the flow rate. The flow is governed by predefined recipes and the buffer capacities. Each machine processes material from its respective input buffer to its output buffer. The MES coordinates the process by monitoring the buffer levels and generating transport requests (tr-req) when full output buffers need to be emptied. These transport requests are directed to the Transport System (TS), which executes the transport job. For example, in the current setup, machine m1 processes an input of quantity 2 coming from q1 and produces an output of quantity 1, for example through a welding process. The material flow is illustrated by the arrows, with the input buffers receiving material from sources and the output buffers supplying material to subsequent processes or sinks. The system dynamically adjusts to different layouts by scaling the number of machines, stocks, and buffers, as well as the recipes.

The simulation tracks key performance metrics such as throughput, which represents the total material processed, the driven distance by the transport system, and simulation duration.

4.3. Neural network architecture

The second stage of the conceptual formulation, as depicted in Figure 2, integrates two distinct components: a convolutional neural network (CNN) for processing image data and a multi-layer perceptron (MLP) for handling tabular data. This structure enhances the network's ability to generalize across various layout configurations. The CNN is designed to accept image data of flexible sizes, meaning the input layer adjusts based on the image format, allowing it to process layouts with differing numbers of machines. As long as the input image adheres to the correct format, it can be effectively processed by the CNN, making the model adaptable to diverse layout scenarios.

In contrast, the MLP portion has a fixed-dimensional input layer by design, which requires a structured approach to handle varying amounts of tabular data. To address this, placeholders are introduced in the MLP's input layer, enabling the model to account for configurations with different machine quantities. For instance, in a layout with eight machines, all placeholders are used, while in a layout with fewer machines, such as five, the remaining placeholders are left empty. This setup ensures that the ANN maintains its capacity to generalize effectively, even when the number of machines varies.

By leveraging this architecture, the network can replace traditional simulation methods to perform layout evaluations. Figure 5 provides a visualization of the training process.

The inner red circle, starting with layout generation followed by simulation and preprocessing create a subset of labeled training data. As part of preprocessing, the data is checked for integrity, duplicates, outliers, distribution and then split into training, validation,

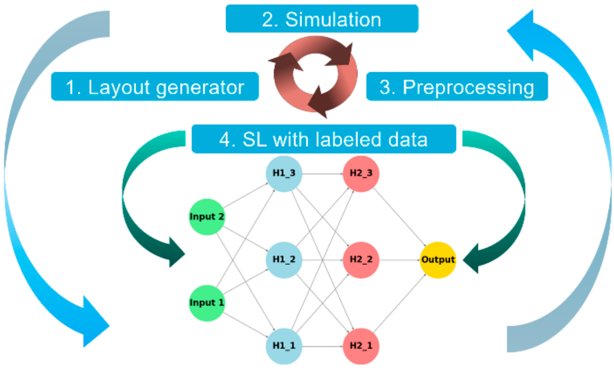

Figure 5. Training Process of the neural network

and evaluation datasets. For example, layouts designed in a way that they could not generate any throughput were removed. The resulting data is used as both input and output in the ANN's supervised learning framework, indicated by the green arrows. The outer blue arrows represent an additional loop, signifying the use of transfer learning, where the aforementioned process is repeated with different factory configurations. Initially, the ANN is trained on a dataset containing layouts with four machines. Afterward, the pre-trained network is progressively refined using additional generated datasets, gradually increasing the complexity. For this study, this process was repeated five times, culminating in a dataset that includes layouts with up to eight machines over a total of 150000 layouts.

To support this methodology, a custom dataset class was developed within the PyTorch framework. This class consolidates input images, supplementary tabular data, and labels into a unified dataset, simplifying the process of feeding information into the ANN during training. Additionally, Bayesian hyperparameter optimization was used to find an optimized ANN configuration. An early stopping mechanism was integrated to control the number of training epochs and prevent overfitting. It is triggered once the validation loss does not improve for five consecutive epochs; a threshold known as "patience". This value was determined experimentally.

5. Evaluation and validation

In the study, traditional validation methods such as dataset splitting, ensuring dataset distribution and integrity, duplicate checks, and K-Fold cross-validation were rigorously applied to guarantee the robustness and reliability of the ANN's performance in evaluating layouts. Additionally, the model was compared with other regression techniques, including ensemble learning methods, to benchmark its effectiveness. These assessments were successful, forming the foundation for the hypothesis that guided this research:

> "The developed neural network is capable of replicating the simulation and is more performant in doing so."

To assess the validity of this hypothesis, the following sub questions were formulated:

- Is the ANN suitable for use in layout optimization, specifically concerning the research questions?
- Can permutation analysis, baseline comparisons, and feature analysis support the network's reliability compared to other AI methods?
- How does the model scale when applied to realistic factory sizes?

It was also investigated whether the architecture of the ANN offers inherent advantages over existing simulation-based methods for layout evaluation. One major benefit associated with this model, particularly in the context of NP-completeness and limited resources, is its performance. If proven, this performance advantage could significantly contribute to the broader research on the FLP by enabling the consideration of more constraints while reducing computation times.

5.1. Evaluation

After transfer learning with five different layout configurations across 150000 layouts, the model achieved an R^2-score of ~0.9 on test data, see Figure 6 (Eschemann et al., 2024).

The scatter plot presents a comparison between the actual values, normalized within the range [0, 1] on the x-axis, and the predicted values by the ANN on the y-axis. The results demonstrate that the network is capable of accurately predicting the simulation outcomes across the entire value spectrum, with no significant outliers. The blue line serves as a reference, representing the baseline results of a mean value estimator. Figure 7 further visualizes how effectively the network predicts the simulated throughput based on the different layout configurations within the dataset (Eschemann et al., 2024).

The results demonstrate that the ANN consistently performs well across the entire range of layout complexities. However, as the complexity of the layouts increases, there is a noticeable decline in the R^2-score. This indicates that more training data and prolonged training sessions may be required to maintain performance at higher levels of complexity. To further assess the importance of different input types for the model, a permutation analysis was conducted. The datasets were deliberately modified in two separate runs: in one, only the

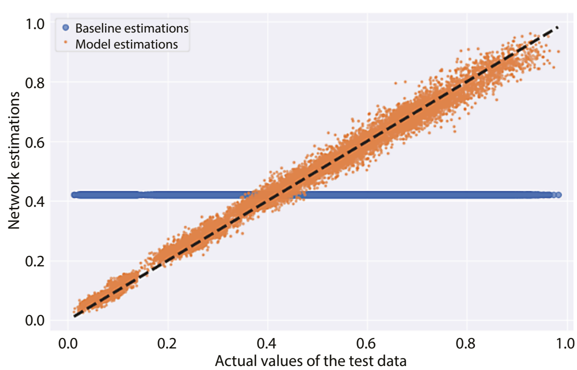

Figure 6. Actual simulations values vs. network and baseline estimations (Eschemann et al., 2024)

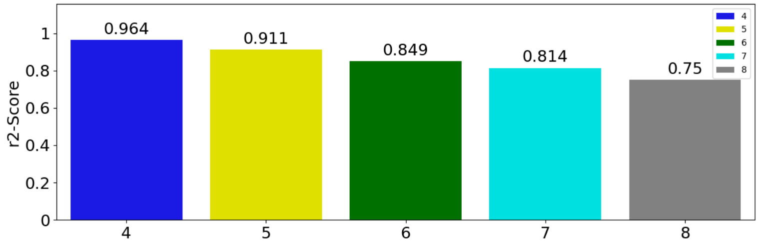

Figure 7. Prediction accuracy over all layout configurations (Eschemann et al., 2024)

image data was permuted, and in the other, the additional tabular data was permuted. For the images, permutation was performed at the channel level. For each image in the batch, one channel was randomly selected, and the pixel positions within that channel were randomly shuffled both vertically and horizontally. This disrupts the spatial structure within the selected channel, while the other channels remain unchanged. Such a permutation distorts visual patterns, which negatively impacts the model's performance, especially when it has learned to recognize specific visual features or structures. For the additional data, the permutation was applied at the dataset level. Each feature in the additional data for every entry in the batch was randomly shuffled, breaking the connection between the data and the corresponding images and labels. This results in a loss of meaningful alignment between the additional data and the images. Consequently, the ANN is left to rely solely on the unchanged image data to make accurate predictions. The result is shown in Figure 8 (Eschemann et al., 2024).

A more significant drop in the R^2-score following the permutation of a specific data type indicates that this data type plays a more critical role in the model's performance. With non-permuted data, the model achieves an R^2-score of ~0.91 (blue bar), highlighting its strong ability to account for the variability in the test data. When the image data is permuted, the R^2-score decreases to ~0.69 (orange bar), demonstrating the importance of images for the network's accuracy in making predictions. However, the fact that the score remains relatively high suggests that the model still retains considerable predictive capability due to the unaltered additional data. Conversely, when the additional data is permuted, the R^2-score drops more sharply to ~0.36 (green bar). This more substantial decline, compared to the permutation of image data, reveals the greater significance of the additional data in determining the model's performance.

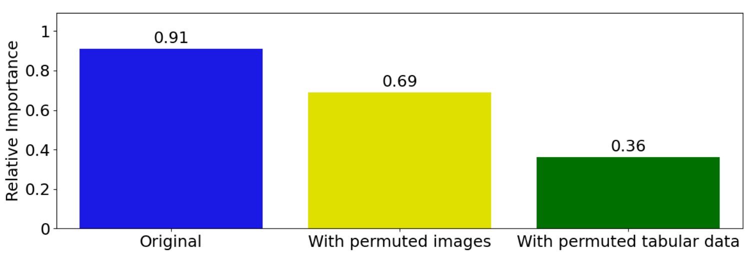

Figure 8. Permutation analysis of the input types (Eschemann et al., 2024)

Overall, these findings indicate that while both image and tabular data are important for the model's predictive success, the additional data exerts a stronger influence. This justifies the decision to use an ANN architecture that combines a CNN (to process image data) and an MLP (for tabular data), affirming that this design choice enhances the model's accuracy and generalization.

To further demonstrate the specific influence of each feature within the additional data, a relative importance analysis was conducted, differing from the prior approach. In this context, relative importance refers to how much the model's performance declines when a particular feature is permuted. It is measured by the difference in the R^2-score between the original model and the one with the permuted feature. A greater difference indicates higher importance of that feature. This method helps quantify the specific contribution of each feature and allows for an assessment of how permuting the feature affects the ANN's prediction accuracy, as illustrated in Figure 9.

The analysis revealed that the most critical feature is the number of operational elements (machines) in the layout. This parameter appears to be a key factor for the model in accurately calculating throughput. Notably, the model assigns greater importance to the coordinates of machines in the later stages of the production process (machines 5–8) compared to those in the earlier stages (machines 1–4). This is not a random observation, as the machines seem to form pairs that are either important or less relevant for the model. It is also noteworthy that the coordinates appear in pairs in terms of their importance.

Several factors could explain this pattern:

1. *Complexity of the Production Process*: The latter stages of production may involve more complex or critical processes, which have a stronger impact on the target variable. As a result, the model might be more sensitive to changes in these areas.
2. *Relation to the Transport System*: The positions of the final machines could have a greater effect on throughput compared to the positions of the initial machines. An optimized arrangement of these later machines might positively influence throughput more than the arrangement of the earlier machines.
3. *Data Structure*: There may be particular characteristics in the data causing variations in the coordinates of the later machines to correlate more strongly with the target variable. This could be due to specific patterns or relationships in the training data.

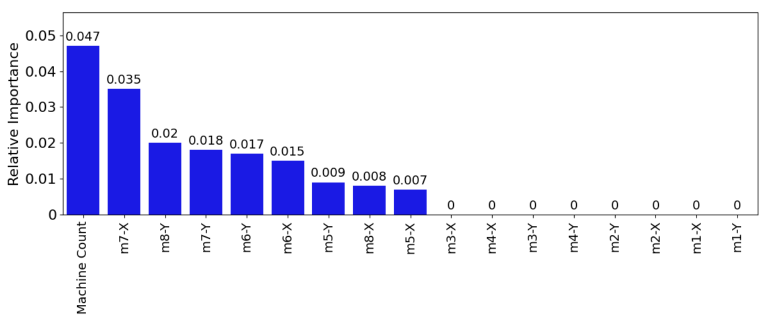

Figure 9. Relative importance of individual features in the additional data

4. *Model Characteristics*: The model's architecture or training process may favor certain features. For example, the later coordinate features might be considered more relevant due to their position in the dataset or their relationship to other variables.

Since the positions of the machines can also be considered as a single collective feature, the individual coordinate features were aggregated to provide an overall evaluation of the importance of the spatial arrangement compared to the other features. This was done to assess the general significance of the machine coordinates, beyond the analysis of each individual position's influence, see Figure 10.

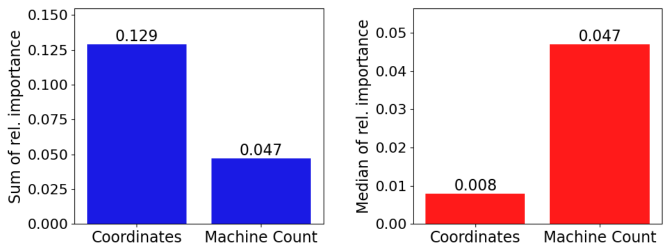

Figure 10. Cumulated importance of the coordinates in comparison

Two approaches were applied to determine the significance of the OE coordinates: summation and averaging of the importance of all coordinate features. In the summation approach (shown in blue), the importance of all coordinate features is added together, revealing the greatest impact of the spatial arrangement on prediction accuracy. In contrast, the averaging approach (shown in red) calculates the mean importance of each individual coordinate feature. This analysis indicated that, on average, individual coordinate features have less influence on prediction accuracy compared to other features, such as the number of OEs. Furthermore, it suggests that while the overall spatial arrangement of the OEs is highly significant for the model's predictions, the specific positions of individual elements carry less weight. The model seems particularly sensitive to changes in the collective layout rather than to variations in the location of single OEs.

5.2. Validation

The network can only be considered validated with respect to the hypothesis if, beyond demonstrating prediction accuracy, it also shows a performance advantage. This involves comparing the time it takes to simulate a layout with the time required for the ANN to estimate throughput. In the experiments conducted, a linear and proportional relationship between simulation duration and simulated time was observed, allowing for extrapolation and thus making longer simulations unnecessary. For comparison with the ANN, this finding implies that the simulation duration must serve as a benchmark, where a similarly linear increase in throughput begins to emerge in relation to simulated time. This point represents the minimum required simulation duration. To determine this threshold, an experiment was conducted showing the throughput over the simulation time. The results of this experiment are shown in Figure 11 (Eschemann et al., 2024).

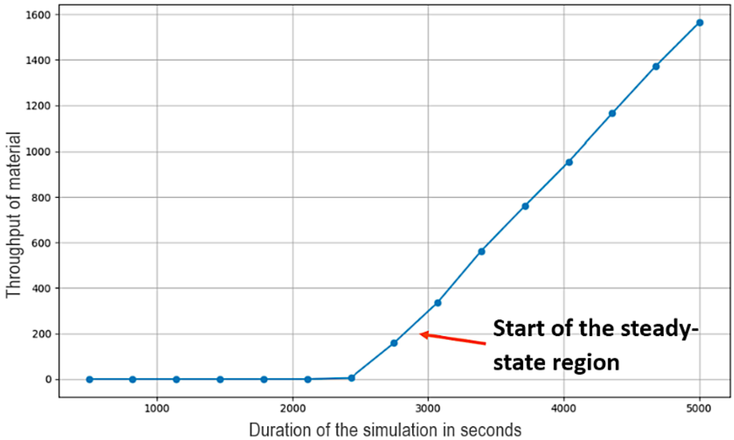

Figure 11. Development of the throughput over the simulation duration (Eschemann et al., 2024)

The evaluation reveals that throughput at the output warehouse is first measured after ~2500 seconds, with linearity emerging between 2500 and 3500 seconds. To account for fluctuations, a simulation duration of around 3500 seconds is set as the minimum necessary time for accurate results. For the performance comparison, the factory configurations with the lowest and highest complexity are considered. On the same hardware, the trained ANN achieves an average prediction speed of 0.026 milliseconds per layout. Under the most favourable conditions, and based on prior investigations, the factory simulation requires ~30 milliseconds per layout at the minimum possible simulation duration. This makes the ANN ~1,154 times faster than an event-driven simulation.

While it is theoretically possible to optimize and accelerate the simulation for parallel processing and more efficient hardware utilization, such optimizations would come with significant costs, development efforts, and uncertain outcomes. It is unlikely that these optimizations would fully bridge the substantial performance gap. Additionally, the ANN results were obtained from a dataset featuring five different factory configurations, whereas the simulation result reflects the most favourable value from a single configuration.

Therefore, this evaluation demonstrates that the AI-based approach has a systematic advantage over conventional event-driven simulations in terms of speed and applicability across multiple factory configurations. To further assess the model's generalization capabilities, two experiments were conducted. In the first experiment, the ANN was tasked with predicting the simulated throughput for a factory configuration with eight OEs. The dataset was divided into five batches, with each batch simulated for a different duration, ranging from 5,000 to 15,000 simulated seconds. This analysis revealed that the model performs better for layouts with longer simulation times, likely because these allow for a more extended period in the steady-state phase. In contrast, shorter simulations, especially for more complex configurations, tend to display higher variability in the initial stages, leading to less accurate predictions.

In the second experiment the ANN was trained on a factory configuration containing 20 OE, see Figure 1. A dataset of 100,000 layouts was created and used for training following the same procedure as before. In the first step, each OE was assigned a unique colour to help the ANN distinguish its role in the production flow. After generating the layouts, the time required for the factory configuration to reach a steady-state throughput was examined, showing that the more complex layout configuration requires on average 9,000 seconds until the first products reach the output warehouse. Based on this analysis, the simulation is reset after 10,000 seconds to capture throughput, with the total necessary simulation time for a factory configuration with 20 OEs determined to be 12,000 seconds. Using this benchmark, throughput simulations were carried out for all 100,000 layouts. The distribution of the model's predictions compared to the actual test data in the setup for 20 OE appeared similar to that shown in Figure 6. On the training data, the ANN achieved an R^2-score of ~0.97, with a validation score of ~0.95. The result on the test dataset was ~0.92, demonstrating the model's ability to predict simulated throughput for a complex factory configuration. The denormalized throughput values to be predicted ranged from a global minimum of 1 to a global maximum of 11,454. The confidence band indicates homoscedasticity, showing that the model maintains consistent performance across the entire range of throughput values.

6. Conclusions and final remarks

This study explored the application of AI to the FLP, an area that has seen extensive research since the 1960s but has only recently begun to incorporate AI. Historically, AI approaches in this field have been considered less advantageous, which is why there has been limited exploration. Three research questions were formulated and addressed to utilize AI for finding optimized factory layouts. Literature reviews revealed that while many AI-based techniques offer different methods for layout optimization, no comprehensive or directly applicable system currently exists. The main challenge lies in the lack of data and models that are suitable for optimizing layouts. The conclusion highlighted those integrative approaches, combining metaheuristic techniques with AI, tend to lead to improved layouts.

In this study, an event-driven simulation was integrated with a concatenated ANN architecture, consisting of both MLP and CNN models, allowing a supervised learning approach. The simulation was employed to generate labeled training data. The AI model processed visual layout representations and tabular data to predict the simulated throughput.

The evaluation demonstrated that the ANN was accurate in predicting simulated throughput and ~1,154 times faster than simulation. While the simulation time increased linearly with more complex layouts, the ANN maintained its speed advantage. A feature analysis revealed that the ANN benefited primarily from the additional tabular data, which contributed to approximately 60% of its accuracy. Among these data, the model heavily relied on coordinate information. One limitation of this approach is its dependence on an initial dataset generated via simulation.

In summary, the study showed that an ANN can be optimized with self-generated data to solve specific problems. This suggests that, in theory, an AI platform with programming capabilities could train itself on automatically generated data to specialize in specific problem-solving tasks. Future developments are expected to focus more on the integration of large language models (LLMs) and generative adversarial networks (GANs). These approaches could, unlike the current method, generate layout options that meet predefined throughput goals, potentially eliminating the need for a metaheuristic algorithm. This would represent a fully AI-based solution to the FLP.

References

Azimi, P., & Soofi, P. (2017). An ANN-based optimization model for facility layout problem using simulation technique. *Scientia Iranica*, *24*(1), 364–377. https://doi.org/10.24200/sci.2017.4040

Ball, R., & Tunger, D. (2005). *Bibliometrische Analysen-Daten*, *Fakten und Methoden-Grundwissen Bibliometrie für Wissenschaftler*, *Wissenschaftsmanager*, *Forschungseinrichtungen und Hochschulen* (Bd. 12). Forschungszentrum Jülich.

Bochmann, L. S. (2018). *Entwicklung und Bewertung eines flexiblen und dezentral gesteuerten Fertigungssystems für variantenreiche Produkte* [PhD Thesis]. ETH Zurich.

Brocke, J. vom, Simons, A., Niehaves, B., Niehaves, B., Reimer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process. In *ECIS 2009 Proceedings*. 161. https://aisel.aisnet.org/ecis2009/161

Burggraef, P., Wagner, J., & Heinbach, B. (2021). Bibliometric study on the use of machine learning as resolution technique for facility layout problems. *IEEE Access*, *9*, 22569–22586. https://doi.org/10.1109/ACCESS.2021.3054563

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, *31*(2), 255–267. https://doi.org/10.1016/j.arcontrol.2007.04.001

Eschemann, P., Krauskopf, J. E., Sauer, J., & Zernickel, J. S. (2021). Optimizing factory layouts with supervised genetic algorithm. In *Modelling and Simulation 2021 – The European Simulation and Modelling Conference*, *ESM 2021* (pp. 49–56).

Eschemann, P., Nieße, A., & Sauer, J. (2024). Determining throughput of factory layouts with neural networks. In *Industrial Simulation Conference*, *ISC 2024*.

García-Hernández, L., Pérez-Ortiz, M., Arauzo-Azofra, A., Salas-Morera, L., & Hervás-Martínez, C. (2014). An evolutionary neural system for incorporating expert knowledge into the UA-FLP. *Neurocomputing*, *135*, 69–78. https://doi.org/10.1016/j.neucom.2013.01.068

Garcia-Hernandez, L., Salas-Morera, L., Pierreval, H., & Arauzo-Azofra, A. (2018). A novel hybrid multi-criteria decision-making model to solve UA-FLP. In *International Symposium on Distributed Computing and Artificial Intelligence* (pp. 292–299). https://doi.org/10.1007/978-3-319-94649-8_35

Hosseini-Nasab, H., Fereidouni, S., Ghomi, S. M. T. F., & Fakhrzad, M. B. (2018). Classification of facility layout problems: A review study. *The International Journal of Advanced Manufacturing Technology*, *94*(1–4), 957–977. https://doi.org/10.1007/s00170-017-0895-8

Hou, Y., Yano, K., Suga, N., Webber, J., Nii, E., Higashimori, T., Denno, S., & Suzuki, Y. (2022). A study of throughput prediction using convolutional neural network over factory environment. In *2022 24th International Conference on Advanced Communication Technology* (*ICACT*) (pp. 429–434). https://doi.org/10.23919/ICACT53585.2022.9728893

Ikeda, H., Nakagawa, H., & Tsuchiya, T. (2022). Towards automatic facility layout design using reinforcement learning. *Annals of Computer Science and Information Systems*, *32*, 11–20. https://doi.org/10.15439/2022F25

Jaber, A. Q., Hidehiko, Y., & Ramli, R. (2007). Machine learning in production systems design using genetic algorithms. *International Journal of Computational Intelligence*, *4*(1).

Klar, M., Ruediger, P., Schuermann, M., Gören, G. T., Glatt, M., Ravani, B., & Aurich, J. C. (2024). Explainable generative design in manufacturing for reinforcement learning based factory layout planning. *Journal of Manufacturing Systems*, *72*, 74–92. https://doi.org/10.1016/j.jmsy.2023.11.012

Rummukainen, H., Nurminen, J. K., Syrjänen, T., & Numminen, J.-P. (2018). *Machine learning from prior designs for facility layout optimization*.

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 843–852). https://doi.org/10.1109/ICCV.2017.97

Tam, C., & Tong, T. K. (2003). GA-ANN model for optimizing the locations of tower crane and supply points for high-rise public housing construction. *Construction Management and Economics*, *21*(3), 257–266. https://doi.org/10.1080/0144619032000049665

Tsuchiya, K., Bharitkar, S., & Takefuji, Y. (1996). A neural network approach to facility layout problems. *European Journal of Operational Research*, *89*(3), 556–563. https://doi.org/10.1016/0377-2217(95)00051-8