Manufacturing Letters 30 (2021) 1–4

Contents lists available at ScienceDirect

Manufacturing Letters

journal homepage: www.elsevier.com/locate/mfglet

An implementation of a reinforcement learning based algorithm for factory layout planning

Matthias Klar *, Moritz Glatt, Jan C. Aurich

Institute for Manufacturing Technology and Production Systems, TU Kaiserslautern, Germany

ARTICLE INFO

Article history:
Received 31 May 2021
Received in revised form 28 July 2021
Accepted 4 August 2021
Available online 25 August 2021

Keywords:
Reinforcement learning
Machine learning
Layout planning
Factory planning
Optimization

ABSTRACT

Factory layout planning is a recurring and time consuming process since multiple often conflicting planning objectives have to be considered simultaneously. Inadequately planned layouts however can significantly impede the operation of a factory. Recent studies from other disciplines have shown the potential of reinforcement learning to solve complex allocation problems. Consequently, this paper presents a reinforcement learning based approach for automated layout planning. In particular, a first implementation of the algorithm using Double Deep Q Learning is presented and used to solve an allocation problem with four functional units and the transportation time as an optimization criterion. The algorithm generated an optimized layout within 8,000 episodes of training and showed promising potential for more comprehensive applications in the future.

1. Introduction

Factory planning is a systematic, objective-oriented process, structured into a sequence of phases, extending from the setting of objectives to the start of production [1]. A central element within factory planning is the generation of a layout that includes a coarse spatial representation of functional units (e.g. machines) and consequently predefines multiple characteristics of the future plant, like the throughput time, utilization, and area demand [2]. Since efficient layout planning can reduce the material handling costs in manufacturing by 10 to 30 percent, an optimized layout planning process is of high importance [3]. However, the generation process of a factory layout is highly complex due to a variety of partially conflicting planning objectives that need to be considered [4].

In particular, manual planning approaches are often used in practice. Due to the resulting effort, often only a few layout variants are generated and evaluated against each other. Besides, manual planning is characterized by the often subjective decisions of the planner, hence the best possible factory layout is not always found [5]. Consequently, approaches that automate this process bear potential to support planners by generating better layouts in a shorter time. By using exact optimization methods (e.g. Branch-and-Bound), the optimal layout can be found. However, these approaches require exponential rising computational capacities with rising problem sizes and therefore show limited practical applicability. As a recent review shows, most approaches apply approximation methods like *meta*-heuristics (e.g. genetic algorithms or simulated annealing) [6]. These approaches provide valuable solutions with good suitability. However, they mostly rely on boundary conditions and objective functions that need an analytical formulation. This usually requires assumptions and simplifications, which is challenging since it requires a high expert knowledge [7]. Optimizing a layout with regards to efficient material handling is mostly performed based on the distance and handling intensity between workstations, which is only one example for commonly used simplifications and assumptions [8]. Besides, only a few approaches utilize artificial neural networks (ANNs). One approach considers 5 machines. However, the possible positions of the machines are predefined [9]. Consequently, this approach does not provide the required flexibility to adequately generate a factory layout.

These challenges can potentially be addressed by Reinforcement learning (RL), a class within machine learning that aims at learning a policy that enables the selection of a reward maximizing, situation depending action [10]. RL offers potentials, where the solution space of a problem is too large to be exhaustively described or the problem is difficult to be modeled analytically and where the results of decisions are observable. Using RL, digital models can be trained to master complex games such as Othello, while outperforming human players and existing approximative

Abbreviations: ANN, artificial neural networks; RL, reinforcement learning.

* Corresponding author at: P.O. Box 3049, 67653 Kaiserslautern, Germany.
E-mail address: matthias.klar@mv.uni-kl.de (M. Klar).

https://doi.org/10.1016/j.mfglet.2021.08.003

methods, without analytical modeling but instead by exploratory observation of actions (move) and resulting consequences, displayed by the reward (e.g. win or loss) [11]. Furthermore, compared with supervised and unsupervised learning approaches, RL does not require extensive training data in advance, which is hard to provide in the case of layout planning. Since the layout planning problem has strong similarities with the present fields of RL applications it is necessary to transfer and use the described potential to solve the layout planning problem.

2. Reinforcement learning algorithm

To address this research gap, a RL framework is developed to learn the placement of functional units in a given layout to optimize the planning objective. The framework of the algorithm is shown in Fig. 1 and consists of two central elements: the environment and the agent. The framework is presented in the following Subsections 2.1 and 2.2 followed by its application in a first Use Case with reduced complexity in Subsection 2.3.

2.1. Agent

The algorithm is trained to select the best possible action based on the current situation. In this case, the problem is an allocation problem in which all functional units are to be placed, one after another. Therefore, the layout is represented as a grid with n positions. The algorithm selects at which position the bottom left corner of the functional unit is placed and if it is allocated at a 0° or 90° angle. This leads to the following action space:

$$A_t = \begin{pmatrix} Position\ 1;\ Rotation\ 0^\circ \\ Position\ 1;\ Rotation\ 90^\circ \\ Position\ 2;\ Rotation\ 0^\circ \\ \vdots \\ Position\ n;\ Rotation\ 0^\circ \\ Position\ n;\ Rotation\ 90^\circ \end{pmatrix}$$

To obtain a stable and converging learning behavior [12], the agent is built as a Double Deep Q Learning (DDQL) Agent using experience replay and target networks [13]. The epsilon value, which defines the ratio between exploration and exploitation, is 0.4 in the first training episode and linearly decreases to a value of 0.1 within 8,000 episodes of training.

2.2. Environment

The environment provides the necessary information to the agent and consists of two parts, the state representation, and the reward function.

The state representation served as an input for the ANNs. The input can be divided into the representation of the layout and the functional unit that needs to be placed next. The layout is represented by a grid with n positions. Each position of the grid can have one of the following states: the position is free, occupied by a wall, occupied by one of the m functional units, or reserved as a lane. The functional units are represented by their shape, denoted by x and y dimensions. Additionally, information about which of the m functional units will be placed next is given.

Second, the agent receives a reward depending on which action is selected in which situation (state). The calculation of the value of each planning objective can be done analytically and by simulation. The computed value of each planning objective is normalized and weighted to its prioritization to compute the reward the agent receives.

2.3. Use Case

The algorithm is implemented for a relatively simple Use Case to demonstrate its ability to optimize layouts. The Use Case is shown in Fig. 2. Four functional units need to be placed next to the lane where the material is transported by AGVs. The layout is optimized regarding a single planning objective, the transportation time. The transportation time is calculated based on the distance, which is displayed by the number of quadrants between the

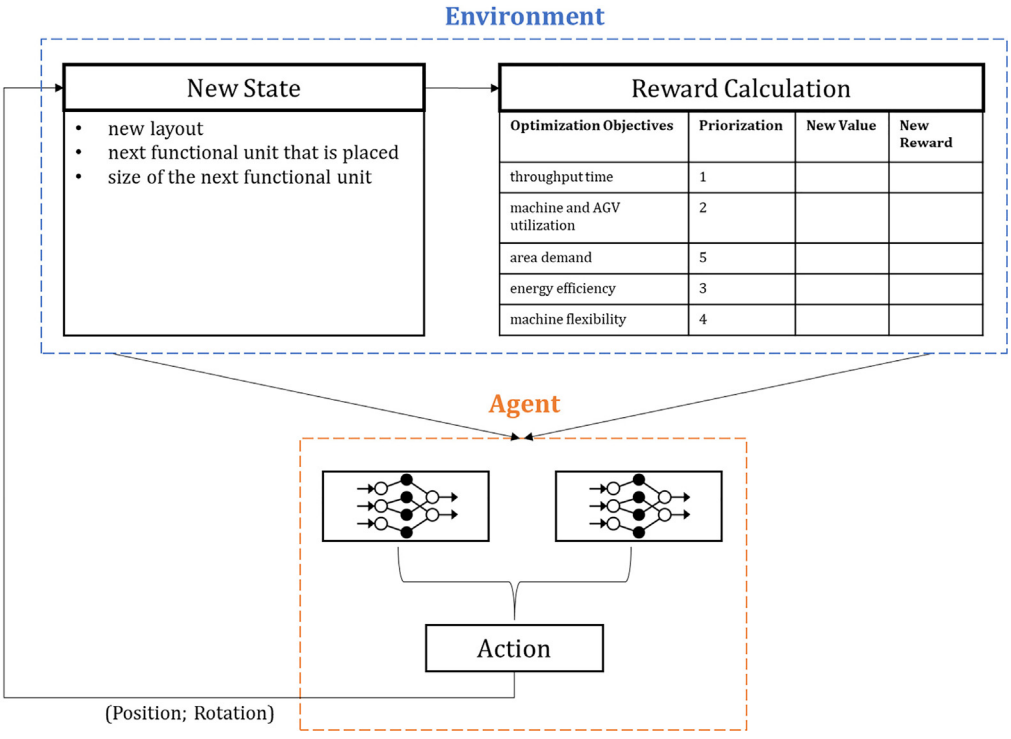

Fig. 1. Framework of the reinforcement learning algorithm.

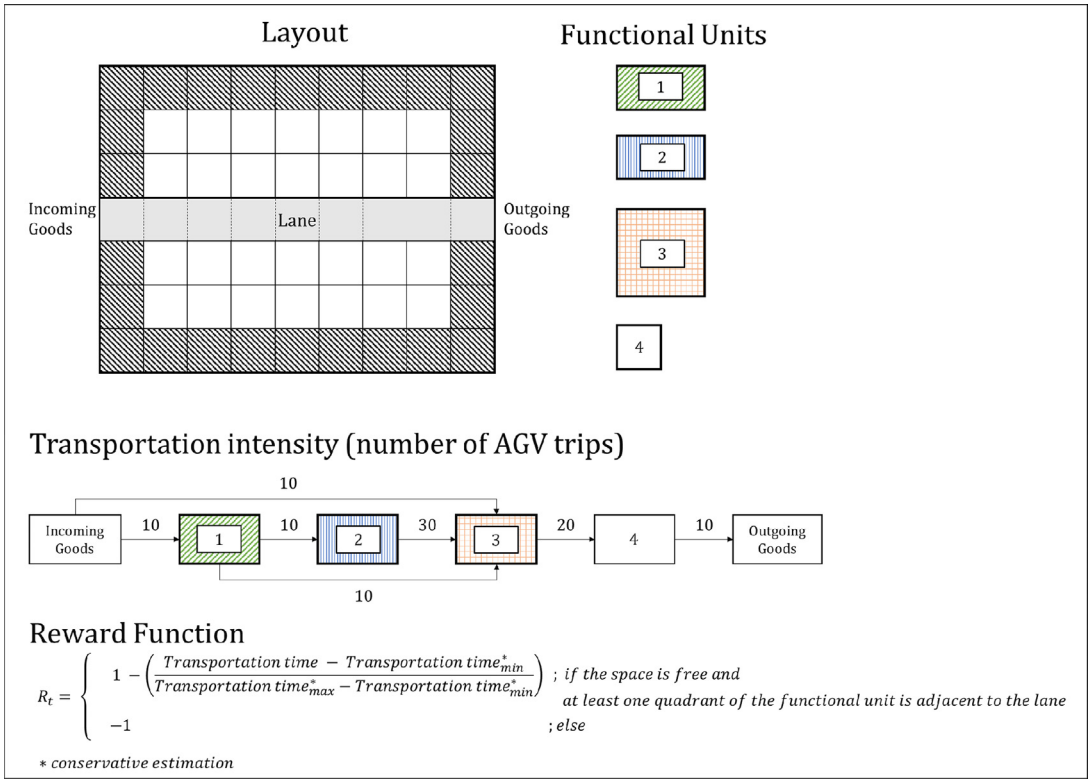

Fig. 2. Use Case.

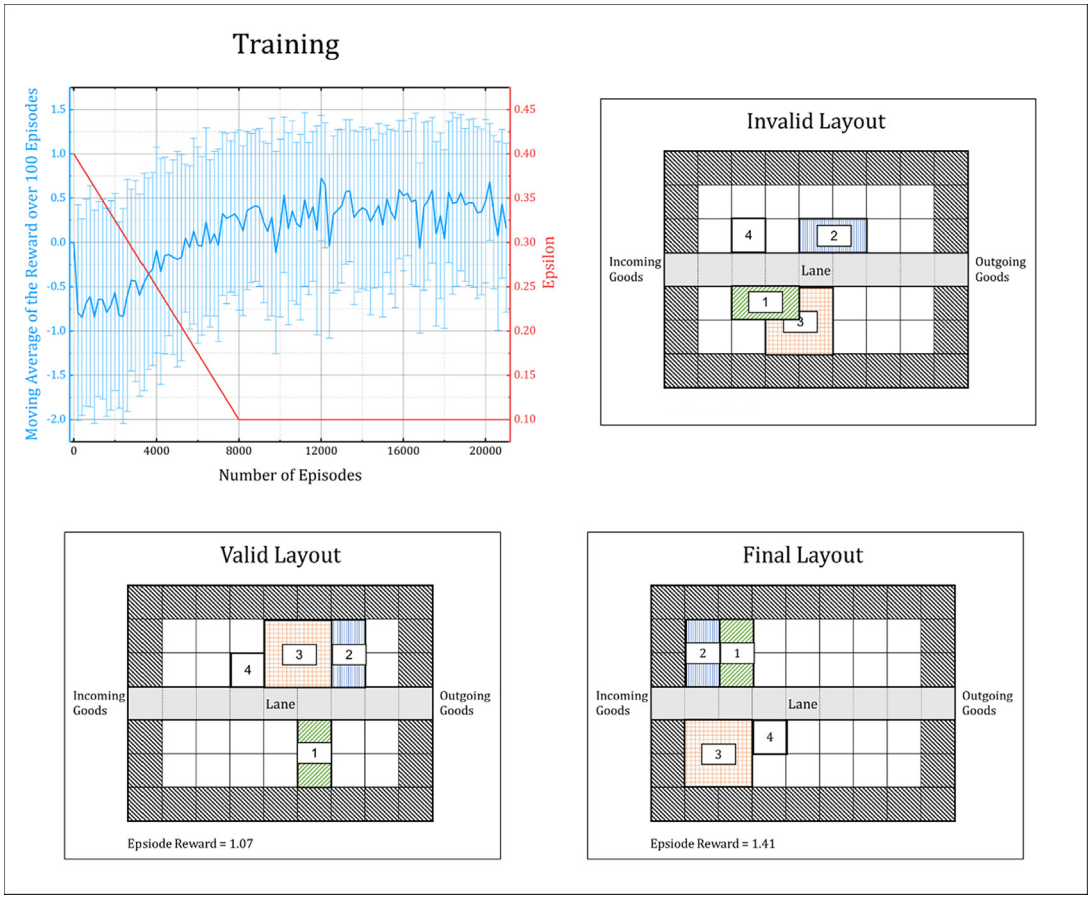

Fig. 3. Training and Final Layout.

functional units, and the transportation intensity given by the number of AGV trips. According to the reward function, a lower transportation time will result in a higher reward. The training process of the algorithm and the final result is visualized in Fig. 3. The high exploration rate (epsilon = 0.4) at the start of the training leads in combination with the untrained algorithm to a negative moving average reward since more invalid than valid actions are selected. The decreasing exploration rate and the ongoing training process result in an improved selection process and consequently better layouts. After 8000 episodes of training, the average reward alternates between 0 and 0.7. The training is completed since no significant increases in the average reward function can be observed. During training, the currently best solution is stored. The final layout with a reward of 1.41 and selected layouts within the training process are shown in Fig. 3. The quality of the solution increases over time since functional unit 3, which has the highest incoming transportation intensity, is allocated as a central element of the material flow in the final layout. Furthermore, the functional units are allocated near the entrance since the transportation intensity of incoming goods is higher than the transportation intensity of outgoing goods. Consequently, RL show suitability to generate a factory layout without the need for extensive analytic modeling of objective functions and boundary conditions. Nevertheless, the scalability needs to be further investigated, to address problems with a bigger size and match the performance of existing approximative methods.

3. Conclusion

This paper presents an RL-based approach for factory layout planning that can be used in the early stages of layout planning. After presenting the general framework of the algorithm, it is trained based on a Use Case, in which four functional units are placed in a given grid. The planning objective is the minimization of the transportation time. After the training, the algorithm can place the functional units in an optimized way by positioning the functional unit with the highest transportation intensity as a central element of the layout.

In future research, the complexity of the allocation problem will be increased, by integrating currently missing restrictions like requirements regarding media supply and disposal as part of the environment. In addition, the algorithm will be extended by a multiobjective optimization approach. Furthermore, it is important to incorporate dynamic effects that result from the variability, interconnectedness, and complexity of a material handling system, which largely defines its efficiency. In the regarded context, discrete-event simulations will be integrated into an RL environment to directly determine the quality of a layout and to be the foundation of an RL-based approach for factory layout planning. Furthermore, the number of functional units will be increased. Consequently, the reward function, as well as the selected RL method, needs to be further investigated since DDQL suffers from time-consuming training especially when facing large action spaces [14].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 252408385 – IRTG 2057.

References

[1] VDI. Factory Planning: Planning Processes (5200); 2011.
[2] Grundig C-G. Fabrikplanung: Planungssystematik - Methoden - Anwendungen. 6th ed. München: Carl Hanser Verlag; 2018.
[3] Francis RL, McGinnis LF, White JA. Facility layout and location: An analytical approach. 2nd ed. Englewood Cliffs, N.J.: Prentice-Hall; 1992.
[4] Azevedo MM, Crispim JA, Pinho de Sousa J. A dynamic multi-objective approach for the reconfigurable multi-facility layout problem. J Manuf Syst 2017;42:140–52. https://doi.org/10.1016/j.jmsy.2016.12.008.
[5] Bénabès J, Bennis F, Poirson E, Ravaut Y. Interactive optimization strategies for layout problems. Int J Interact Des Manuf 2010;4(3):181–90. https://doi.org/10.1007/s12008-010-0100-x.
[6] Hosseini-Nasab H, Fereidouni S, Fatemi Ghomi SMT, Fakhrzad MB. Classification of facility layout problems: a review study. Int J Adv Manuf Technol 2018;94(1–4):957–77. https://doi.org/10.1007/s00170-017-0895-8.
[7] Robinson S. Simulation: The practice of model development and use. 2nd ed. London: Palgrave Macmillan; 2014.
[8] Hu B, Yang B. A particle swarm optimization algorithm for multi-row facility layout problem in semiconductor fabrication. J Ambient Intell Human Comput 2019;10(8):3201–10. https://doi.org/10.1007/s12652-018-1037-3.
[9] Tsuchiya K, Bharitkar S, Takefuji Y. A neural network approach to facility layout problems. Eur J Oper Res 1996;89(3):556–63. https://doi.org/10.1016/0377-2217(95)00051-8.
[10] Sutton RS, Barto A. Reinforcement learning: An introduction. Cambridge, MA, London: The MIT Press; 2018.
[11] van Eck NJ, van Wezel M. Application of reinforcement learning to the game of Othello. Comput Oper Res 2008;35(6):1999–2017. https://doi.org/10.1016/j.cor.2006.10.004.
[12] Bui Y-H, Hussain A, Kim H-M. Double deep Q -learning-based distributed operation of battery energy storage system considering uncertainties. IEEE Trans. Smart Grid 2020;11(1):457–69. https://doi.org/10.1109/TSG.516541110.1109/TSG.2019.2924025.
[13] van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. AAAI 2016;30(1).
[14] Sewak M. Deep Reinforcement Learning. Singapore: Springer Singapore; 2019.