Virtual Commissioning for an Overhead Hoist Transporter   
in a Semiconductor FAB

**Sang H. An, Sang C. Park**

Dept of IE, Ajou University, San 5,

Woncheon-dong, Yeongtong-gu, Suwon, Korea

**Abstract**: Presented in the paper.

**Key words:** HILS, OHT, Virtual commissioning, Control software verification

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*Correspondence: Sang C. Park ([scpark@ajou.ac.kr](mailto:scpark@ajou.ac.kr))

Department of Industrial Engineering

Ajou University

San 5, Woncheon-dong, Yeongtong-gu

Suwon 443-749, Korea

**1. INTRODUCTION**

Increasing global competition and a shift towards customer-centric markets in most industries poses challenges for manufacturing enterprises. Their production and logistic processes must have a high level of logistic quality, expressed by KPIs such as average throughput time or mean tardiness, to satisfy customers’ demands while incurring low logistic costs to ensure competitive-ness. This requires the careful design of operational planning and control processes. Positioning an enterprise and choosing or developing the most suitable planning and control methods is a difficult and time-consuming process, currently often performed manually in a trial-and-error fashion.

Manufacturing environments are often too complex to consider all important attributes, because the requisite information comes from multiple sources and sensors, and also because much of the underlying logics of the operation might be implicit and challenging to capture intuitively. Thus, extraction of implicit knowledge from given schedules enables smart factories to automate the development of dispatching rules while reflecting various attributes.

✁ Figure 2. Commissioning configurations ✁

Automated Guided Vehicles (AGVs), a kind of mobile robot for material handling, have been regarded as one of the most promising technologies and applied to various shop floors and warehouse logistics for material supply operations owing to high degree of autonomy and flexibility (Vis, 2006; Michalos, Kousi, Makris, & Chryssolouris, 2016; Kousi et al., 2016; Demesure et al., 2018; Wang, Zhang, & Zhong, 2020; Kousi, Koukas, Michalos, & Makris, 2019).

Dispatching rules classified a vehicle-initiated task and a workcenter-initiated task assignment (Tanchoco 1984). Depending on the situation, traditional dispatching rules are used to dispatch AGVs using only single method or multi mixed methods. These rules are First Come First Served (FCFS), Shortest Travel Distance (STD), Earliest Due Date first (EDD), Longest Waiting Time (LWT), Nearest Vehicle First (NVF), Maximum Queue Size (MQS), etc. To solve dispatching problem, some cases adopted reinforcement learning. **Reinforcement learning is a machine learning method that can constantly adjust agent’s behavior through trial and error** (Kaelbling, Littman & Moore, 1996). **[Scheduling problem]** An reinforcement learning based approach for a multiple-load carrier scheduling problem (Chen, Xia, et al. 2015), and they proposed Q() model improve throughput and reduce travel cost. Vehicle-initiated task assignment approach production scheduling problem using Q-learning algorithm (Wang and Usher, 2005). In dynamic job shop scheduling problem approach using reinforcement learning. **The other effective cases**, Dispatching solution based on Q-learning, wherein the functions are approximated with a neural network (NN), can be used to reduce the complexity inherent to centralized learning. Because a Q-learning-based solution is easy to control for a discrete time, coordination with discrete event simulation (DES) has great value (Park, Huh et al. 2019; Gabel and Riedmiller 2008; Gosavi 2009). Wang et al. (2015) proposed dueling networks with two separate estimators for the state value function and state dependent action advantage function. The dueling networks separately learn V(s), which is determined only by the state, and the advantage A(s, a), which is determined according to actions, to derive Q(s, a). This approach has the advantage of being able to divide the information of the Q-function into the portion determined only by the state and that determined according to actions. In contrast to a deep Q-network (DQN), it learns the combined weights that lead to V(s) at every step regardless of the action. Moreover, it requires fewer episodes than a DQN to complete learning, resulting in better performance as the number of action types increases (Wang et al. 2015; van Hasselt, Guez, and Silver 2016; Nair et al. 2015; Gosavi 2009).

**Multi-agent Reinforcement Learning.** Multi-agent reinforcement learning has been applied in domains like collaborative decision support systems. Different from the single agent reinforcement learning (RL), multi-agent RL needs the agents to learn to cooperate with others. It is generally impossible to know other policies since the learning process of all agents is simultaneous. Thus for each agent, the environment is non-stationary [3]. It is problematical that directly apply the independent reinforcement learning methods into the multi-agent environment. There are several approaches proposed to relieve or address this problem, including sharing the policy parameters [8], training the Q-function with other agents’ policy parameters [31], centralized training [17] and opponent modeling [2, 26]. Besides, there are also some methods which use explicit communication to offer a relatively stationary environment for peer agents [7, 9, 30]. In the large-scale multi-agent systems, the nonstationary problem will be amplified. To address this problem, Yang et al. [38] proposed a novel method which converts multi-agent learning into a two-player stochastic game [28] by applying mean field theory in multi-agent reinforcement learning to make it possible in large-scale scenarios. Since the mean-field MARL method only takes a mean field on states/actions input into consideration, it ignores the agent interactions. Our proposed method provides another way to enable large-scale multi-agent learning and retain the interactions between agents, which makes agents receive global feedback from the next moments and adjust their strategies in time. Furthermore, our proposed method provides a backward stationary learning method and has a rapid reaction to the feedback from the environment.

**2. PROBLEM FORMULATION**

State, s is the geo-coordinates of the driver and time-of-day (in seconds) when the driver is dispatched for a trip order, i.e. s := (l, t), where l is the GPS coordinates pair (latitude, longitude) and t is time. Note that it could be different from the actual origin of the trip where the passenger stands at. Moreover, s may contain additional contextual features at (l, t), such as statistics of demand, supply, and order fulfillment within the vicinity of (l, t), denoted as f. In this case, s can be extended from (l, t) to (l, t, f). We also differentiate the time for weekday and weekend. For the rest of the paper, we denote the l and t components of a state s by sl and st respectively.

Action, a is the assignment of a particular trip to the driver, which is simply defined by the trip destination and drop-off time. Let the current state := (, ) be the driver’s location, time and the context when the trip is assigned, and the next-state is the drop-off location, time and context. Then, the action is . The space of all eligible actions is denoted by A.

Reward, r is the total fee collected for the trip and is a function of s and a.

An episode is one complete day, from 0:00am to 23:59pm. Hence, a terminal state is a state with t component corresponding to 23:59pm. We set s1 in all those transitions where the trip crosses midnight to be terminal state.

State-action value function, Q(s, a) is expected cumulative reward that the driver will gain till the end of an episode if he/she starts at state s and takes an action a. Mathematically, , where S, A, and R are stochastic variable version of s, a, and r respectively; T is the number of transition steps till the terminal state, and is the discount factor for the future rewards. We discretize the time space into steps of 10 minutes and is multiple powers of the time steps that an order stride across.

Policy, is a function that maps a state s to a distribution over the action space (stochastic policy) or a particular action (deterministic policy). The greedy policy with respect to a learned is given by .

State value function, : expected cumulative reward that the driver will gain till the end of an episode if someone starts at state s and follows a policy . Assuming that a greedy policy w.r.t. the Q function is used, the state value.

2.1. State representation

**State raw observation full state**

State at t-step is separable term represented multi-matrix form meaning 3-channel image (127 by 127). Environment clip image is program screen of human level. And Feature Image show AGVs, Current Job, and Simulation attributes as color image. conclude Red channel image is represented job information. Green Channel image is represented each link’s driving constraints such as direction, velocity, and rotation. Blue channel image is represented sequence information about time-horizon AGV routing left.

Finally, mask make block unnecessary region. It’s predefined static matrix.

2.2. Reward representation

Many researcher are trying well-made reward function, several good reward function design cases, ‘Deep Mimic’ and ‘GAIL(**dd**)’ are used exponential form into kinematic models. If demonstrate agent get valuable experience by non-linear functions. Also, we refer multi attributes rule, selected throughput, waiting time, and mileages. We make non-negative reward function.

Reward in t-step.representation operating level , selected job’s waiting time and selected AGV’s mileage . Reward function imitate our objective function merged maximize production, minimum waiting time and line balancing.

Operation level matched maximize production can get highly score when t-step production achieve 80% level of target production . Selected job’s waiting time matched minimum waiting time should keep over 0.2 score. Selected AGV’s mileage matched line balancing means a standard deviation about all AGVs mileages. When selected AGV’s mileage is 1.7 (about 95%) over, is converged to 0.

2.3. Action representation

Action represents the dispatching rule of the AGV’s system and is defined by a encoding value Dispatching rules in action are First Come First Served (FCFS), Shortest Travel Distance (STD), Earliest Due Date first (EDD), Longest Waiting Time (LWT), Nearest Vehicle First (NVF), and Maximum Queue Size (MQS).

✁ Figure 3. OHT design & production procedures ✁

**3. EXPERIMENTS**

As sh

**4. DISCUSSION AND CONCLUSIONS**

In a large FAB,

**ACKNOWLEDGEMENT**

**This work was supported by the technology innovation program (20002772) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).**

**REFERENCE**

Anglani, A., Grieco, A., Pacella, M., Tolio, M. (2002). Object-oriented modeling and simulation of flexible manufacturing system: a rule-based procedure, Simulation Modeling Practice and Theory, 10, 209-234.

Chang, DS., Park, SC. (2018). Configuration space-based discrete event system specification formalism for a smart factory with real-time flexibility, Concurrent Engineering: Research and Applications, 26(3): 265-275

Drath, R., Weber, P. and Mauser, N. (2008). An evolutionary approach for the industrial introduction of virtual commissioning, *IEEE International Conference on Emerging Technologies and Factory Automation*: 5-8.

Gans, N. R., Dixon, W. E., Lind, R., Kurdila, A. (2009). A hardware in the loop simulation platform for vision-based control unmanned air vehicles, Mechatronics, 19, 1043-1056.

Ham, WK., Oh, JW., Cho, KH., Park, K., Park, SC. (2019). New modeling formalism for the energy simulation of conveyor systems, Computers & Industrial Engineering, 128, 180-191.

Hibnio, H. Inukai, T. and Fukuda, Y. (2006). Efficient manufacturing system implementation based on combination between real and virtual factory. *International Journal of Production Research*, **44** (18): 3897–3915

Hoffman, P., Maksoud, T. M. A., Schuman, R. and Premier, G.C. (2010). Virtual Commissioning of Manufacturing Systems a review and new approaches for simplification*, Proceedings 24th European Conference on Modeling and Simulation*.

Hsieh, CH., Cho, C., Yang, T., Chang, TJ. (2012). Simulation study for a proposed segmented automated material handling system design for 300-mm semiconductor fabs, Simulation Modeling Practice and Theory, 29, 18-31.

Huang, H., Yeh, C. (1999) Development of a virtual factory emulator based on three-tire architecture, *IEEE Int. Conf. On Robotics & Automation*, Detroit Michigan, 2434-2439.

Kim, J., Yu, G., Jang YJ. (2016). Semiconductor FAB layout design analysis with 300-mm FAB data: “Is minimum distance-based layout design best for semiconductor FAB design?”, Computers & Industrial Engineering, 99, 330-346.

Kim, T. G. (1994). *DEVSIM++ User’s Manual*, Department of Electrical Engineering, KAIST, Korea.

Klingstam, P., Gullander, P. (1999). Overview of simulation tools for computer-aided production engineering, Computers in Industry, 38, 173-186.

Kong, SH. (2007). Two-step simulation method for automatic material handling system of semiconductor fab, Robotics and Computer-Integrated Manufacturing, 23(4), 409-420.

Lee, CG., Park, SC. (2014). Survey on the virtual commissioning of manufacturing systems, Journal of Computational Design and Engineering, 1(3), 213-222

Park, SC. (2005). A methodology for creating a virtual FMS model, Computers in industry, 56(7), 734-746

Park, SC., Chang M. (2012). Hardware-in-the-loop simulation for a production system, International Journal of Production Research, 50(8), 2321-2330

Park, SC., Park, CM., Wang, G. (2008). A PLC programming environment based on a virtual plant, International Journal of Advanced Manufacturing Technology, 39, 1262-1270

Seo, JC., Chung, YH., Kim, BH., Park, SC. (2016). Backward capacity-filtering for electronic Fabs, Production Planning & Control, 27(11), 925-933.

Zeigler, B. P. (1984). *Multifacetted modeling and discrete event simulation*, Academic Press, Orland.