Factory optimization using deep reinforcement learning AI

Shivam Duhan  
*Purdue University*, sduhan@purdue.edu

Chengming Zhang  
*Purdue University*, cmz@purdue.edu

Wenyu Jing  
*Purdue University*, jing7@purdue.edu

Mingqi Li  
*Purdue University*, li2154@purdue.edu

Follow this and additional works at: [https://docs.lib.purdue.edu/purc](https://docs.lib.purdue.edu/purc)

**Recommended Citation**  
Duhan, Shivam; Zhang, Chengming; Jing, Wenyu; and Li, Mingqi, "Factory optimization using deep reinforcement learning AI" (2019). *Purdue Undergraduate Research Conference*. 57.  
[https://docs.lib.purdue.edu/purc/2019/Posters/57](https://docs.lib.purdue.edu/purc/2019/Posters/57)

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
Factorio is the perfect environment for deep reinforcement learning as it supports extensive modification using an in-game debugging mode which allows our agent to interface with the game effortlessly. The reinforcement learning agent implements a policy of actions based on the reward function, iteratively optimizing towards goals specified by the user. A model-free Deep Q-network agent is used in conjunction with experience replay to learn the best policy (mapping from environment state to action) by sampling past experiences. The environment state is obtained by displaying important variables in a defined location and using OCR to read the values. The DQN agent then uses the best current policy to generate a set of actions, evaluate them using the reward function, and execute the best action.

The Deep Reinforcement Learning Artificial Intelligence Agent reads environment state by using OCR to decode important in-game variables being updated in real time. It then uses an MCTS-based neural network and a hand-crafted reward function to decide the best action and control the on-screen to execute that action. The best policy is decided and continuously updated by a Deep Q-Learning Agent. Various Deep Learning algorithms and agent designs are explored and compared - DQN, Double DQN, MCTS, combine value+policy network, varying degrees of exploration, etc. The Deep Reinforcement Learning Agent is expected to outperform humans in factory design, optimization, and resource management.