# PySCFabSimUI: Graphical User Interface for a Semiconductor Fab Simulator

**Benjamin Kovács, Pierre Tassel, Martin Gebser**

Univesity of Klagenfurt, Klagenfurt am Wörthersee, Austria
Benjamin.Kovacs@aau.at, Pierre.Tassel@aau.at, Martin.Gebser@aau.at

## Abstract

Novel artificial intelligence-based algorithms show superior performance on large-scale scheduling problems compared to traditional handcrafted methods. However, most methods tend to be data-hungry during the development, testing, and deployment phases. Simulations play particular importance in the prototyping process of machine learning-based methods, especially in the training and verification of the algorithms. Our vision is to develop a universal open-source simulator software to assist the development process of novel scheduling techniques from the first small-scale prototypes to the training and verification on real-world scale instances. The goal of the developed toolbox is to accelerate the research of industry-scale dynamic scheduling problems. We extend our open-source simulator with a visual interface. The interface helps in the development and analysis of novel algorithms and to enable their industrial applications. The extension visualizes the most important properties of the simulator and provides insights into the behavior of black-box dispatching strategies either on the global- or machine level in both normal or emergency operating scenarios.

## Introduction

Artificial intelligence (AI) based techniques are becoming increasing popular for solving large-scale scheduling problems. While integrated development environments (IDEs) accelerate the coding and deployment process for general-purpose software, integrated domain-specific toolboxes for scheduling algorithms are not available.

Simulations proved to be a handful option for training, testing, and validating dispatching agents, even on complex large-scale instances where the application of exact optimization techniques run into scalability issues. In our work, we provide an interactive simulator for the semiconductor industry. The goal is to extend an open-source simulator with visualizations of the factory's and the decision making agents' important features. We aim to develop a general tool that is applicable from small-scale toy-size prototyping problems to industry-size applications. We make use of a semiconductor production process model, one of the most complex practical constrained planning problems.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

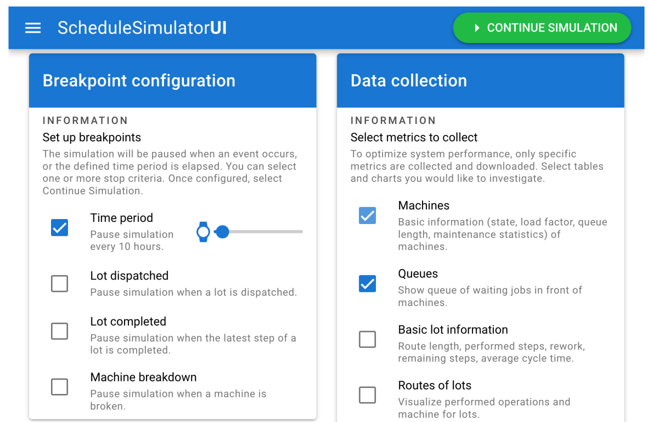


Figure 1: Configuration interface to set up breakpoints and collectable information.

Thus, problems such as the (flexible) job-shop can be considered as its simplified subproblem, making our tool applicable in diverse scenarios.

This demo presents a visualizer extension for an open-source simulator, that simulates a real-world scale semiconductor fabrication plant, modeled as a flexible job-shop problem with many advanced, industry-specific constraints. A video presentation[1] of this demo is publicly available.

## Our Tool

We use the *SMT2020* (Kopp et al. 2020), a large scale dataset with a realistic model involving a diverse selection of machine types (single, batching, cascading), the constraints related to machine dedications and timing, as well as the realistic setup, maintenance, and breakdown models. The large quantity of orders also makes the problems challenging. The re-entrant flow in the production routes offers opportunities for large performance gains with smart planning methods and dispatching strategies with a global view, compared to naive greedy algorithms that focus on the local optimums.

The dataset is simulated by our tool *PySCFabSim* (Kovács et al. 2022), a semiconductor fab simulator developed to conduct reproductible research on large-scale factory

[1]https://prosysscience.github.io/ICAPS_demo.mp4

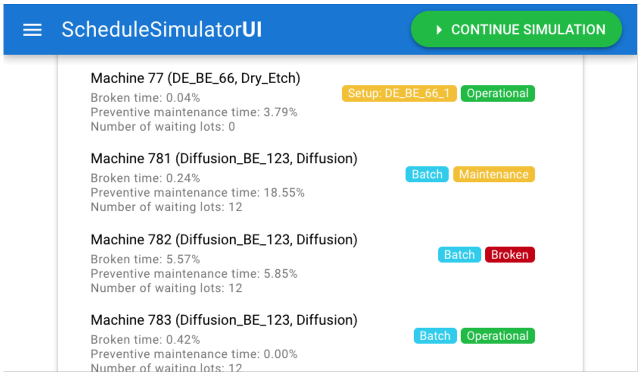


Figure 2: Machine details with type, state, setup, and operational metrics.

scheduling problem instances. The tool can be augmented with plugins, providing convenient way to extend it.

Our visual interface is available in the latest release bundle of *PySCFabSim*[2] (available July 2023), and can be installed as a plugin for the simulator.

**Configuration.** Once the plugin is installed, on the launch of the simulation, a web-based configuration interface opens up (Figure 1). The user can select time intervals (slider) or events for pausing the simulation – for example a when a lot is dispatched on a given machine or a breakdown occurs. To minimize interruptions, it is possible to assign conditions to stopping events based on values of variables, such as the name of the machine, machine group, product, or operation. The user can also pre-define the data to collect and charts to investigate, to optimize the performance of the data extraction algorithm. The settings can be saved in configuration files and loaded for further runs, to avoid frequent manual reconfigurations.

Once the configuration is complete, the user starts the simulation. The plugin installs event handlers for the simulator and starts collecting the selected metrics. The simulator then begins the execution of the virtual manufacturing process. When a breakpoint is reached, the obtained data is aggregated and the monitoring interface is launched. The plugin provides a data access and an intervention API for the UI, to monitor and alter the virtual factory.

**Factory state.** Considering the scale and complexity of the simulated models, only a part of the factory can be displayed on a screen (Figure 2).

Users can select *machines*, *workstations* (groups of same machines) or *machine families* (groups of similar machines) to visualize their features either separately or aggregated. For each machine, the queue length (optionally by lot type) and the active setup, the operations performed with latest setup, and statistics about the frequently performed operations are displayed. The simulator also collects information on longer horizon such as the utilization, breakdown-, preventive maintenance-, and setup times of machines. The metrics can be aggregated for machine families and groups, to obtain higher level insights of the process, for example to analyse the actual bottlenecks in the process. Once critical machines or machine groups are identified, the developer can adjust the decision making strategy for the given machine group. Finally, special characteristics of machines like parallelization (batching) or the current setup are also visible for the users.

To obtain detailed insights in the dispatching process, it is possible to inspect the queue of the machine after each dispatching decision was performed. The simulator computes important metrics used by dispatching strategies such as waiting time, urgency, and lateness of the lot. Based on these computed metrics, one can evaluate whether a custom dispatching or scheduling agent takes reasonable decisions for a given machine or family. For small scale problems, the user can take the manual control of some workstation, and select the dispatchable lot from the machine's queue.

For a *lot* the production routes can displayed, visualizing the re-entrant flow to verify constraints like machine dedications and time-based step couplings are satisfied. The routes for multiple lots can be displayed on a single chart, helping in the analysis of the black-box agents' machine allocation strategy. Basic lot metrics include priority, the number of processed steps with the waiting time between operations, and the number of remaining steps. Lot-related metrics such as the average cycle time, or the average number of operations can be aggregated for a product.

**Dispatching strategies.** A significant barrier in the real-world adaption of black-box decision making strategies is the difficulty in understanding the algorithms' behavior. For example, it is essential to analyze how these algorithms handle challenging corner cases, such as large-scale breakdowns or an overload of the factory with excess amount of orders.

Inspecting queues of waiting lots after the dispatching decisions, and the routes of re-entrant lots are already an important step towards more transparency about the decision making strategy of machine learning-based algorithms.

Our tool provides insights by automatically comparing the controller algorithm's decision making strategy to selected reference heuristics like first-in-first-out, critical ratio, or earliest due date. Comparison is possible either globally (for all machines) or for a selected machine or machine family, in the case when separate dispatching algorithms are learned for the equipments.

## Ongoing Applications

Our research group conducts experiments with evolution strategies, genetic programming and reinforcement learning using the simulator and the plugin introduced in this demo. The simulator's website[2] is continuously updated with ongoing work and publications.


## Acknowledgments

This work was partially funded by KWF project 28472, cms electronics GmbH, FunderMax GmbH, Hirsch Armbänder GmbH, incubed IT GmbH, Infineon Technologies Austria AG, Isovolta AG, Kostwein Holding GmbH, and Privatstiftung Kärntner Sparkasse.


[2]https://prosysscience.github.io/PySCFabSim-release/

## References

Kopp, D.; Hassoun, M.; Kalir, A.; and Mönch, L. 2020. SMT2020—A Semiconductor Manufacturing Testbed. *IEEE Transactions on Semiconductor Manufacturing*, 33(4): 522–531.

Kovács, B.; Tassel, P.; Ali, R.; El-Kholany, M.; Gebser, M.; and Seidel, G. 2022. A Customizable Simulator for Artificial Intelligence Research to Schedule Semiconductor Fabs. In *2022 33rd Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 1–6.