Augmenting High-Fidelity Simulation with Python to Enable Warehouse Optimization

BUSINESS PROBLEM

Target's supply chain infrastructure is ever changing as the company works to reduce costs, improve service, and expand operations. Simulation is central to this effort - it lets the team analyze equipment and processes virtually before changes are made, guiding design decisions and de-risking capital investments. Target relies exclusively on Emulate3D, a commercial platform that produces detailed physics-based models but suffers from long run times, slow GUI-driven model building, and a steep learning curve. This thesis tests whether augmenting Emulate3D with Python simulation can unlock analyses the team cannot practically run today.

DATA SOURCES

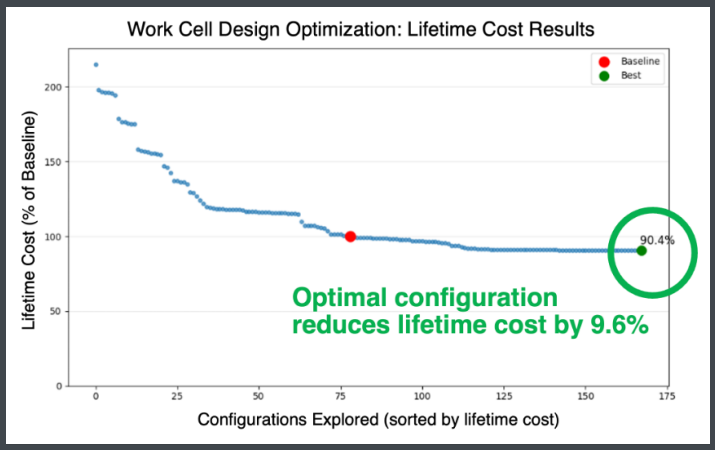
The primary data sets were CSV files of carton-level records from two real inbound receiving processes at Target: a 132-trailer Modified Manual dataset used for validation, and ART line operational data used for optimization. Cartons were captured by scan tunnels on the conveyor belts feeding each work cell, providing carton dimensions, weights, and trailer associations.

Data Types and Format

Categorical (nominal), Numeric, Structured

APPROACH

The thesis takes a case-based approach, building Python models of two inbound carton receiving processes at Target: Modified Manual and ART (Automated Receiving Technology). The work proceeds in three stages. First, it establishes Python's equivalency to Emulate3D, using an existing Emulate3D model of Modified Manual as the benchmark. Second, it uses both models to demonstrate the capabilities faster simulation unlocks - sensitivity analysis, NPV-based work cell optimization, and stochastic modeling. Third, it presents a framework for when Target should use Python, Emulate3D, or both.



Gideon Feifke
MIT Leaders for Global Operations, Class of 2026

Leaders for Global Operations

IMPACT

The thesis established that Python-based simulation is a viable partner to Emulate3D for Target, and produced an implementation framework that lets the team capture that value on future projects.

The case studies revealed that Python ran up to 327x faster than Emulate3D while producing equivalent results, turning multi-day workloads into multi-hour ones. That speed made design optimization tractable: paired with a net present value lifetime cost methodology, Python identified configurations that reduce total cost by 5.3% for the Modified Manual process and 9.6% for the ART line - representing millions in savings over the life of the equipment. Sensitivity analysis at scale and stochastic modeling also became routine, mapping how throughput responds to design parameters and exposing effects like throughput degradation under task time variability.

The framework that emerged is not a recommendation to replace Emulate3D, but a guide to melding code-based and commercial simulation into a coherent workflow - using each tool where it fits and using them together where neither alone suffices. Until recently, this hybrid approach was difficult to staff, but generative AI has lowered the barrier to code-based simulation enough that an engineer with basic Python knowledge can contribute in weeks. For any organization that has built its simulation practice around a single commercial tool, this work offers a roadmap to expand its analytical capabilities by leveraging code-based simulation with Python.

DRIVERS	Target's supply chain is in constant flux - new facilities, retrofits, and process redesigns all rely on simulation to de-risk capital investment. Expanding the simulation team's analytical capabilities directly improves the rigor of these decisions, helping the team identify better-performing, lower-cost designs before capital is committed.
BARRIERS	This work was novel within Target - all infrastructure had to be built and validated from scratch, with no internal precedent for code-based simulation. Designing an implementation framework that leveraged the strengths of both Python and Emulate3D, rather than positioning them as competitors, was an additional challenge.
ENABLERS	Design team leadership was highly supportive and open to exploring new approaches. Generative AI was a substantial enabler - it accelerated code development across the project and made it practical to explore a wide breadth of analytical capabilities in a short timeframe, despite the author having no prior simulation experience.
ACTIONS	A Python model of the Modified Manual process was built to demonstrate equivalency with Emulate3D, surfacing likely flaws in the existing Emulate3D model in the process. A second model was built for ART; together they enabled sensitivity analysis, NPV-based optimization that identified cost-reducing configurations, and stochastic modeling. An implementation framework was developed to guide tool selection on future projects.
INNOVATION	Rather than replacing a commercial simulation tool, this work shows how Python-based simulation can complement it - unlocking optimization, stochastic modeling, and sensitivity analysis that were previously impractical. The implementation framework is replicable and tool-agnostic, and is now staffable for any team because generative AI has reduced the historical software-engineering barrier to code-based simulation.
IMPROVEMENT	Python ran up to 327x faster than Emulate3D, turning multi-day experimentation workloads into multi-hour ones. NPV-based optimization identified configurations that reduce lifetime cost by 9.6% for the ART line and 5.3% for Modified Manual - millions in savings over the equipment life. An implementation framework was delivered to guide when Target should use Python, Emulate3D, or both.
BEST PRACTICES	Generative AI was a powerful accelerant for exploring an unfamiliar tool and methodology - without it, building this Python pipeline from scratch would not have been feasible in the timeframe. Equally important, technical findings require an accompanying implementation framework to drive real organizational impact.
OTHER APPLICATIONS	Simulation is widely used across retail and logistics to support warehouse and facility design, and many teams face the same tension between commercial tools and code-based alternatives that motivated this work. For any of them looking to expand their analytical capabilities, the framework developed here offers a replicable starting point.