IE3100R Systems Design Project

Development of an Optimization Model to determine the Most Efficient Strategy for Autonomous Mobile Robot Delivery

NUS Supervisor: A/Prof. Ng Kien Ming   Industry Supervisor(s): Yokeshbabu Vijayaragunathan, Joanne Chan, Liu Zhepei

Group Members: Ling Cheng Kiat | Sherwin Tieon Wei Jian | Toh Wei Lun | Yang Xuan Lin Eliza

micron NUS National University of Singapore

Problem Description

In the Micron warehouse, Autonomous Mobile Robots (AMRs) are deployed to fulfill daily material requests originating from various workstations within the semiconductor fabrication plant. Currently, there is an **absence of intelligent batching or optimization strategies in their current dispatching logic**. Orders are fulfilled on a first-come, first-served basis, with priority levels ranging from 1 (highest) to 3 (lowest) and AMRs are dispatched individually for each order.
These limitations give rise to several key inefficiencies:

1. **Excessive movement** and **increased energy consumption** as each AMR would need to travel for every single order
2. **Inefficient utilization of available capacity** and an **increase in the total number of trips** required to meet daily demand
3. **Higher order fulfilment lead times** as orders are completed sequentially rather than simultaneously
4. **Failure to meet time constraints** as orders with tight deadlines may not be prioritized efficiently
5. **Path congestion** and **traffic delays** due to high frequency of robot dispatches

Objectives

This project aims to address the identified inefficiencies by **minimizing the total time required for Autonomous Mobile Robots (AMRs) to pick and deliver items** within a future warehouse layout. To achieve this, the project implements algorithmic solutions that optimize order batching based on two key objectives:

1. **Minimizing the total distance travelled** by AMRs during the picking process, thereby reducing energy consumption and operational time.
2. **Minimizing delivery tardiness** by ensuring that orders are fulfilled as close as possible to their required due dates to improve service levels and responsiveness.

Key Skills

Front End: Python Programming, Dashboarding

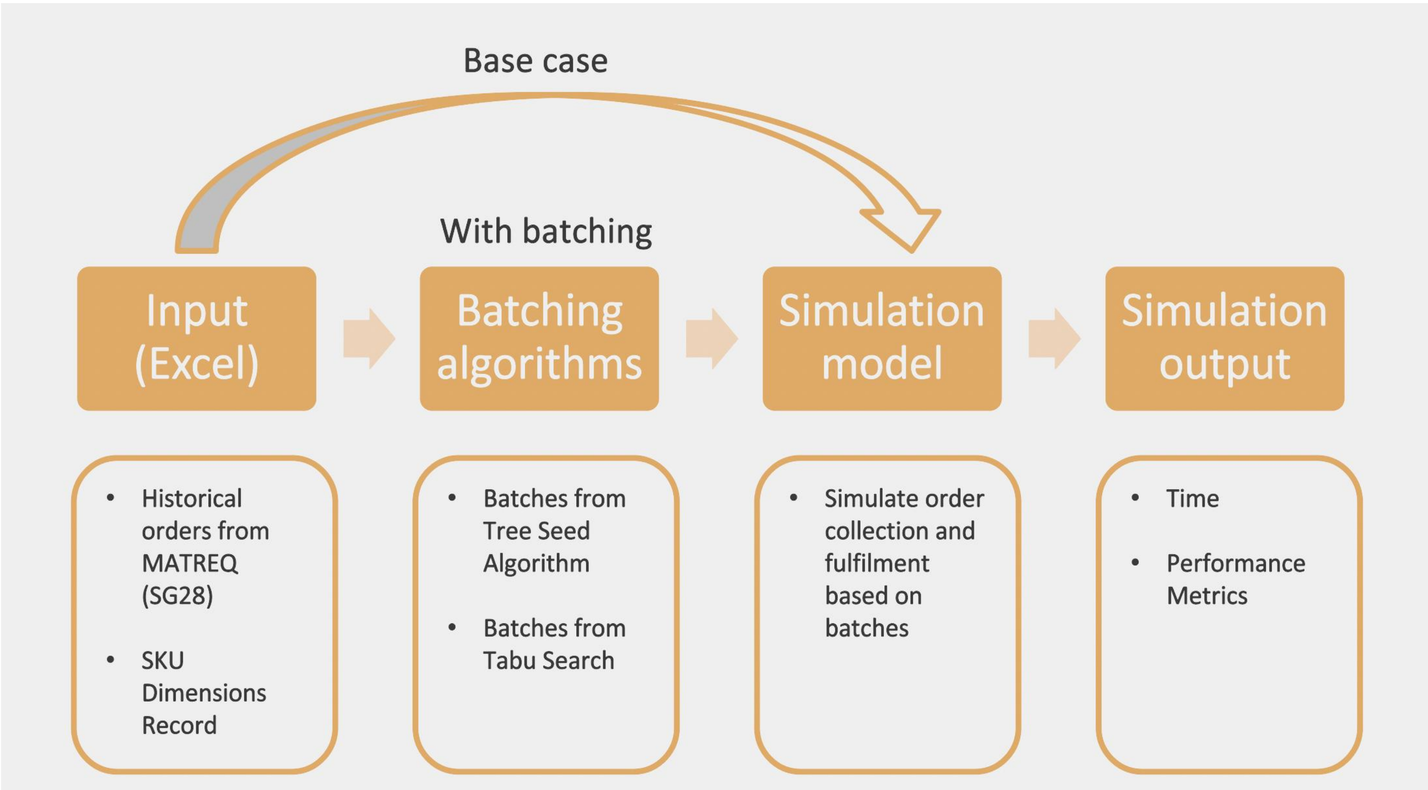
Streamlit

Back End: Python Programming, Discrete Event Simulation, Excel, Data Structures

Project Management

AGILE Methodology
Scrum Framework

Approach



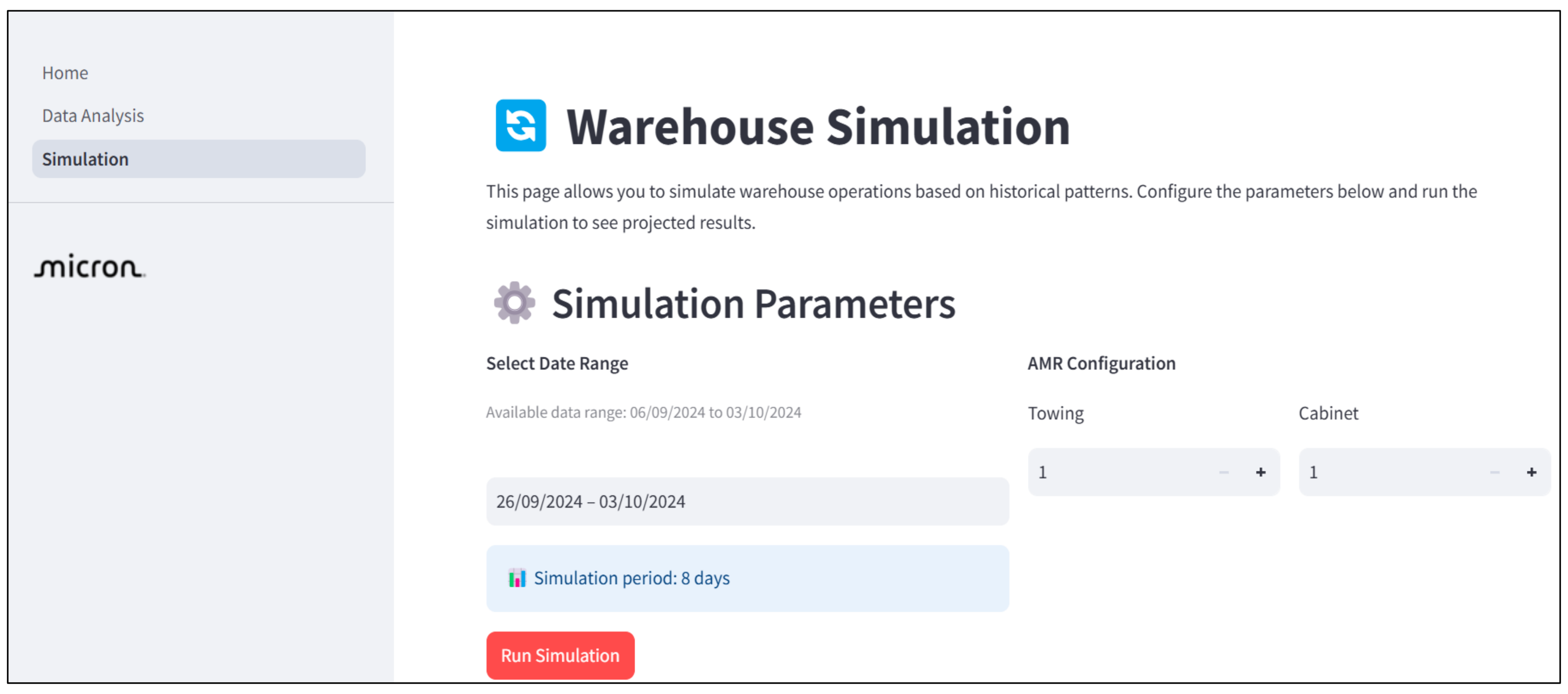
Input (Excel): The input stage of our solution involves the extraction and preprocessing of relevant warehouse operational data, which serves as the foundation for both batching optimization and simulation modelling. 4 key datasets are used in this phase — Material-VLM (Vertical Lift Module) Mapping, Historical Orders from MATREQ, SKU Dimensions Record and Warehouse Distance Matrices.

Batching Algorithms: We propose a hybrid optimization approach, combining the speed of simple heuristics with the solution quality of metaheuristics. The process begins with the application of a greedy heuristic, which constructs an initial feasible batching solution by grouping orders based on either spatial proximity or earliest due dates. Followed by two metaheuristic algorithms. Greedy heuristic provides a fast, feasible initial solution, reducing overall search time while metaheuristic increases the likelihood of identifying high-quality solutions by avoiding local optima.

Simulation Model: Our simulation model replicates Micron's warehouse operations under different AMR configuration scenarios. It translates our batching algorithms into measurable outcomes by simulating AMR movements, order fulfilment processes, and resource utilization patterns. We use the model to simulate three cases — no batching, batching with Tabu Search and batching with Tree Seed Algorithm.

Simulation Output: From our simulation model, we obtain detailed time-based outputs, allowing us to compute key performance metrics — order lead time, order fulfilment rate, AMR utilization rate, and idle time. These metrics serve as the basis for evaluating and comparing the effectiveness of the batching algorithms, helping us determine which method yields better operational efficiency.

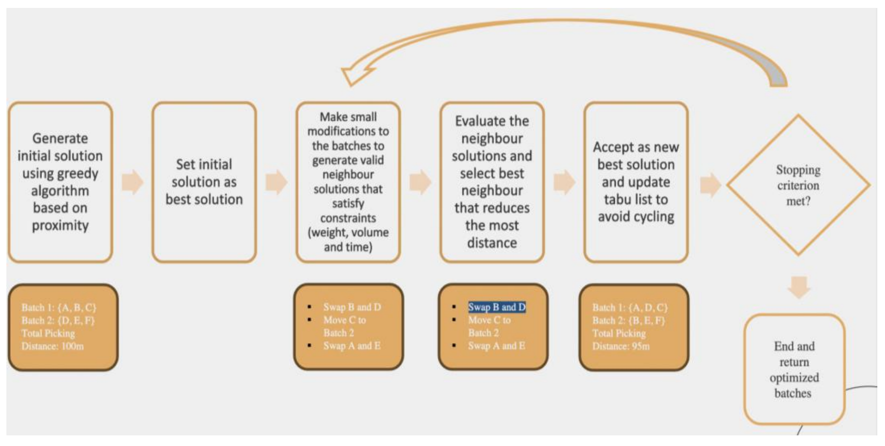
Front End



A web-based interface that transforms complex warehouse optimization into an accessible decision-support tool for operations managers. The front-end features a responsive dashboard that allows users to:

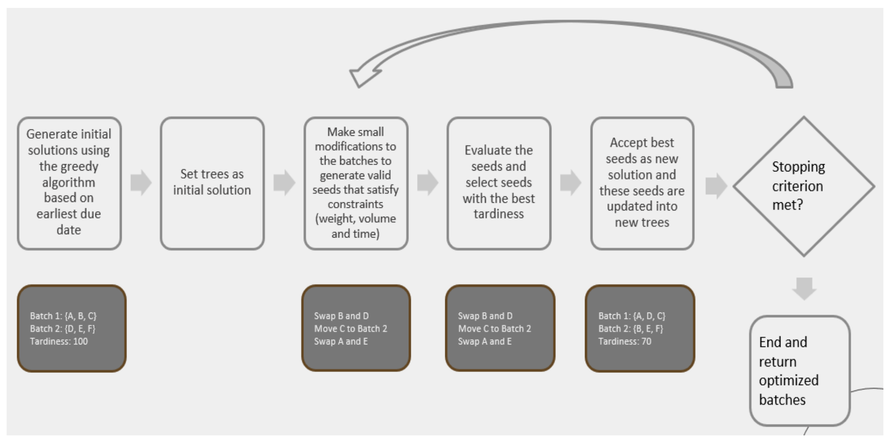
- Configure simulation parameters including date ranges based on available data in the MYSQL database and AMR specifications
- Monitor key performance metrics such as order fulfillment rates and AMR utilization
- Compare optimization algorithm performance (Tabu Search vs. Tree Seed Algorithm)
- Generate actionable insights through visualization

Back End



Tabu Search (TS)

TS is designed to iteratively explore the neighborhood of a current solution in search of improved outcomes. TS escapes local optima using a Tabu List, a short-term memory structure that records recent moves and temporarily prohibits the algorithm from reverting them. By doing this, it encourages the exploration of new regions within the solution space, thereby enhancing the chances of identifying high-quality solutions.



Tree Seed Algorithm (TSA)

TSA is inspired by the natural process of seed dispersal in trees. Each tree in the algorithm represents a potential solution, while its seeds represent variations of that solution. TSA operates by generating new candidate solutions (seeds) from each existing solution (tree), evaluating their quality based on the objective function, and selecting the best-performing candidates for the next generation. This iterative process enables TSA to explore efficiently a diverse range of configurations and avoid premature convergence.

Constraints

- Cabinet AMR weight & volume threshold
- Each order can only be assigned to 1 run, and each batch can only be assigned to 1 AMR
- Each towing AMR can only handle one order at a time

Rolling Horizon

To handle the dynamic nature of order arrivals in the warehouse, we also adopted a Rolling Horizon approach in which orders are batched at regular intervals, rather than waiting for all orders in a day. At each interval, orders will be grouped using our batching algorithms, and batches will be updated continuously as new orders arrive. This enables dynamic, real-time batching as it closely reflects how operations run in real-life, making it more practical and realistic for implementation.

Simulation Model

Our discrete-event simulation mimics Micron's warehouse operations using SimPy, replicating two distinct AMR types with different capabilities: Towing AMRs that transport single keal cases and Cabinet AMRs that handle multiple materials to different locations. The model represents the warehouse as a directed graph where vertices function as operational locations and edges encode navigable paths.

Key Results

Batching Algorithm Comparison

	Batching Algorithm	AMR utilisation rate (%)	PR1 Smart WH Order Fulfilment Lead Time (min)	PR2 Smart WH Order Fulfilment Lead Time (min)	PR3 Smart WH Order Fulfilment Lead Time (min)	Order Fulfilment according to delivery priority (%)
0	No Batching	28.99	32.59	56.57	59.74	93.86
1	Tabu Batch	17.47	21.07	42.02	42.74	95.71
2	Tree Seed Batch	21.43	20.40	51.62	65.51	97.44

Recommendation

From 29/09/2024 to 30/09/2024 - 2 days, with 3 towing AMRs and 4 cabinet AMRs, it is recommended to have **Tree Seed Batch**.

Batching Algorithm Comparison

	Batching Algorithm	AMR utilisation rate (%)	PR1 Smart WH Order Fulfilment Lead Time (min)	PR2 Smart WH Order Fulfilment Lead Time (min)	PR3 Smart WH Order Fulfilment Lead Time (min)	Order Fulfilment according to delivery priority (%)
0	No Batching	27.02	37.12	48.75	43.18	94.29
1	Tabu Batch	16.85	25.12	40.29	36.27	96.12
2	Tree Seed Batch	20.22	29.29	48.47	53.78	95.06

Recommendation

From 27/09/2024 to 29/09/2024 - 3 days, with 3 towing AMRs and 4 cabinet AMRs, it is recommended to have **Tabu Batch**.

In general, batching improves performance compared to base cases (no batching). Based on our results, in some situations, the Tabu Search Algorithm is the best at optimising performance, while in other situations, the Tree Seed Algorithm performs better. For each case, we determine the best performance in order of fulfilment rate, lead time, and AMR utilisation rate. The rationale is that orders are users' needs, so we should fulfill these demands. Hence, we can achieve main project objective of minimising total time needed for AMRs to pick the items in the future warehouse layout.

Future Work

1. **Multi-Objective Optimization**
   - While this project focused primarily on minimizing travel distance and order tardiness, real-world warehouse operations often involve trade-offs among multiple competing objectives. Future iterations of this project could incorporate multi-objective optimization techniques to optimize for these diverse KPIs simultaneously. This would enable more holistic decision-making and provide operators with a set of optimal solutions tailored to specific operational priorities.
2. **Multi-Agent Path Planning**
   - The current fixed route scheduling approach, while computationally efficient, lacks the dynamic adaptability essential for real-world warehouse environments. We propose implementing a hybrid path planning framework that combines the optimality of Conflict-Based Search (CBS) with the efficiency of prioritized planning. This would involve developing a hierarchical planning architecture where high-priority AMRs receive route optimization precedence