

From Production to Dispatch



Using Simulation to Design Resilient Packaging Operations

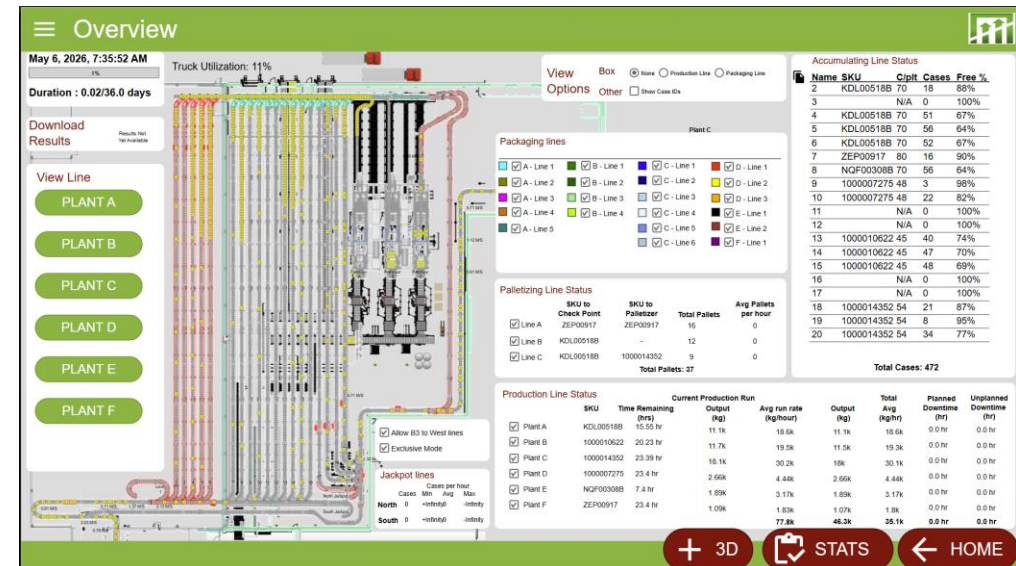
JBV Consulting

Jaco-Ben Vosloo & Devon Cowling



Presentation contents

1. Company and project background
2. Process flow overview
3. The need for simulation
4. Benefits of using AnyLogic
5. Demonstrations
 - I. Single run
 - II. Using AI for model development
 - III. Sensitivity analysis
 - IV. Automated unit testing
6. Verification and validation
7. Results
8. Q&A



1) Company and project background

Golden Veg: The fictitious frozen vegetable company

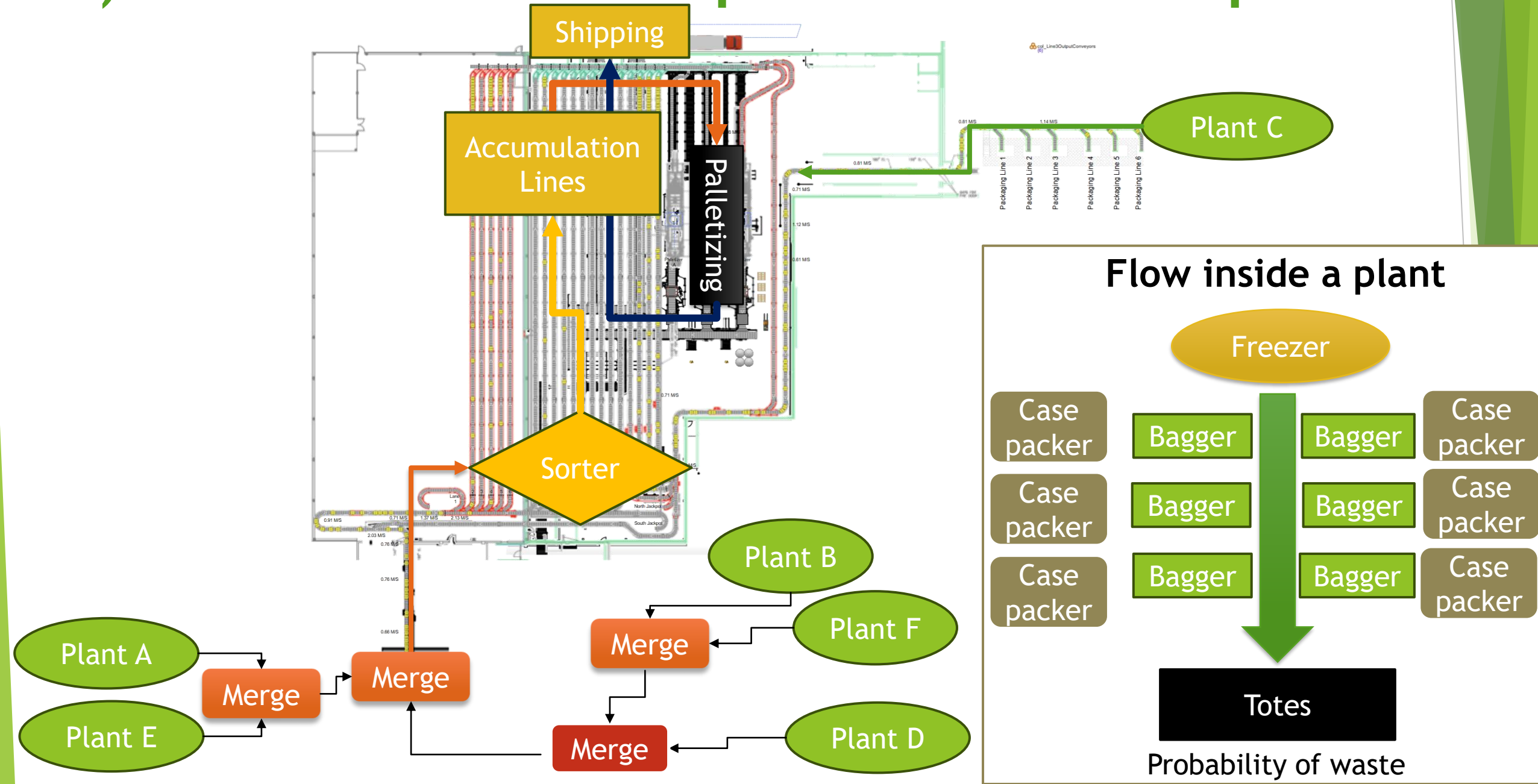
- Specializes in processing premium mixed vegetables for national distribution.
- Operations encompass end-to-end production: from washing, sorting, mixing and flash-freezing to packaging, palletization and cold storage.

Project background:

- **Background:**
 - The company is planning significant production increases to one of the plants and have various ideas and initiatives to improve packaging and palletizing performance
- **The key questions:**
 - Where are the true system constraints?
 - Which improvements deliver meaningful system-wide benefits rather than local optimization?
 - Can the plant handle future production line upgrades?



2) Material flow from production to dispatch



3) The need for simulation

The need	Why?	Benefit of simulation
Determine achievable throughput	Individual capacities ignore shared constraints	Quantifies true overall capacity, identifies constraints
Capture system interactions	Interconnected lines cannot be assessed independently	Simulates merging flows, queues and downstream capacity constraints
Assess downtime impact	Equipment stoppages disrupt downstream flow	Models realistic congestion and buffering due to unavailability
Visually demonstrate impact	Summary statistics hide physical bottlenecks	Makes knock-on effects visible through queues and buffer usage
Test improvements before investing	Clarifies if new investment will provide a return on investment	Validates changes without disrupting production, or incurring costs

4) Benefits of using AnyLogic

Standard flow objects: Material Handling Library

- Utilizes standard flowchart objects for fast development and reduced time on verification.

2D & 3D Animation

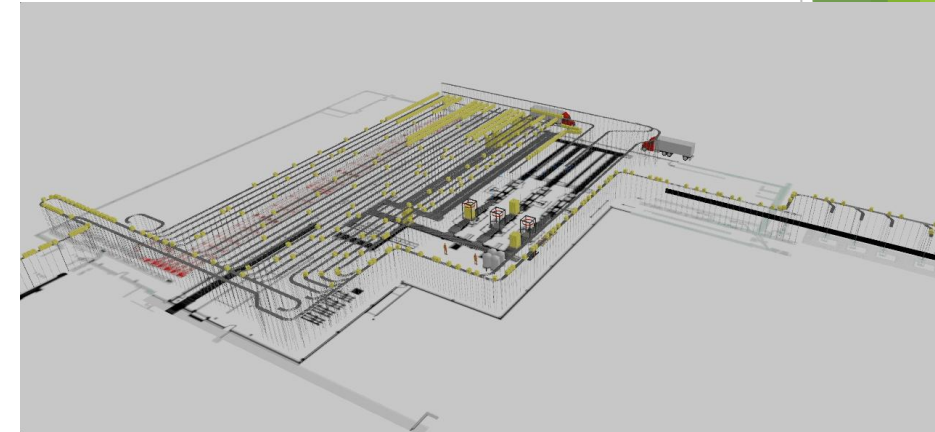
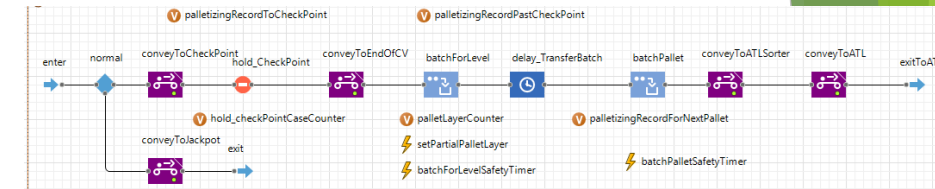
- Detailed animations and charts assists with validation, enabling stakeholders to verify results

Java Code Integration

- Seamlessly integrates with standard Java libraries, used for detailed logic, unit testing, and logging outputs

Standalone application behaviour

- Export model as stand-alone application for testing and analysis by clients and use for future analysis
- Download of results: standard web interactions



```
import org.junit.jupiter.api.extension.ExtendWith;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.MethodSource;
import org.junit.jupiter.api.Assertions;
import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.parallel.Execution;
import org.junit.jupiter.api.parallel.ExecutionMode;
import java.util.stream.Stream;

10 //
11 // Unit tests that load scenarios directly from Excel files instead of relying on
12 // hard-coded fixtures. The Excel data is loaded into the database in a
13 // thread-safe way so multiple tests can run concurrently without corrupting the
14 // shared database instance. Thread safety is centralized in
15 // (Link ExcelScenarioTestSupport) so any other Excel-driven test class can
16 // reuse the same lock/root agent instead of duplicating logic.
17 //
18 @Execution(ExecutionMode.CONCURRENT)
19 @ExtendWith(UT_Palletizing_PartialPalletTest.class)
20 public class UT_Palletizing_PartialPalletTest {
21
22     //
23     // Specify as many Excel files as you like by adding new (Link ExcelScenarioTestSupport.ExcelScenarioCase)
24     // entries to (Link ExcelScenarioCases()). Each case defines its own assertion
25     // and so expectations stay isolated per file.
26     //
27     // -> If you want a separate test class for different Excel suites, you can
28     // instantiate the test class and still call
29     // (Link ExcelScenarioTestSupport.runScenarioFromExcel(String)) to share the
30     // same lock/root agent for safe database access.
31     //
32     private static Stream<ExcelScenarioTestSupport.ExcelScenarioCase> excelScenarioCases() {
33         return Stream.of(
34             new ExcelScenarioTestSupport.ExcelScenarioCase(
35                 "Scenarios/Partial Pallets/Partial Pallet Line 1 - Issues.xlsx",
36                 () -> Assertions.assertEquals(1899, 0, result.getValue(OutputType.TOTAL_POUNDS_PRODUCED), 2, "Pounds produced"),
37                 () -> Assertions.assertEquals(16379, 0, result.getValue(OutputType.TOTAL_BAGS_PRODUCED), 2, "Bags produced"),
38                 () -> Assertions.assertEquals(1727, result.getValue(OutputType.TOTAL_CASES_PRODUCED), 2, "Cases produced"),
39                 () -> Assertions.assertEquals(19, result.getValue(OutputType.TOTAL_PALLETS_PRODUCED), 2, "Pallets produced"),
40                 () -> Assertions.assertEquals(119, result.getValue(OutputType.TOTAL_JACKPOT_CASES), 2, "Jackpot cases")
41             )
42         );
43     }
44 }
```

Download
Results

Summary

Detail

5.1) Model demo - single run

From Production to Dispatch

Single Run

Demonstration Video



Jaco-Ben Vosloo & Devon Cowling



5.2) Using AI for model development

From Production to Dispatch

Using Artificial Intelligence

Demonstration Video



Jaco-Ben Vosloo & Devon Cowling



5.3) Model demo - sensitivity analysis

From Production to Dispatch

Sensitivity Analysis

Demonstration Video



Jaco-Ben Vosloo & Devon Cowling



5.4) Automated unit testing

From Production to Dispatch

Automated Unit Testing

Demonstration Video



Jaco-Ben Vosloo & Devon Cowling



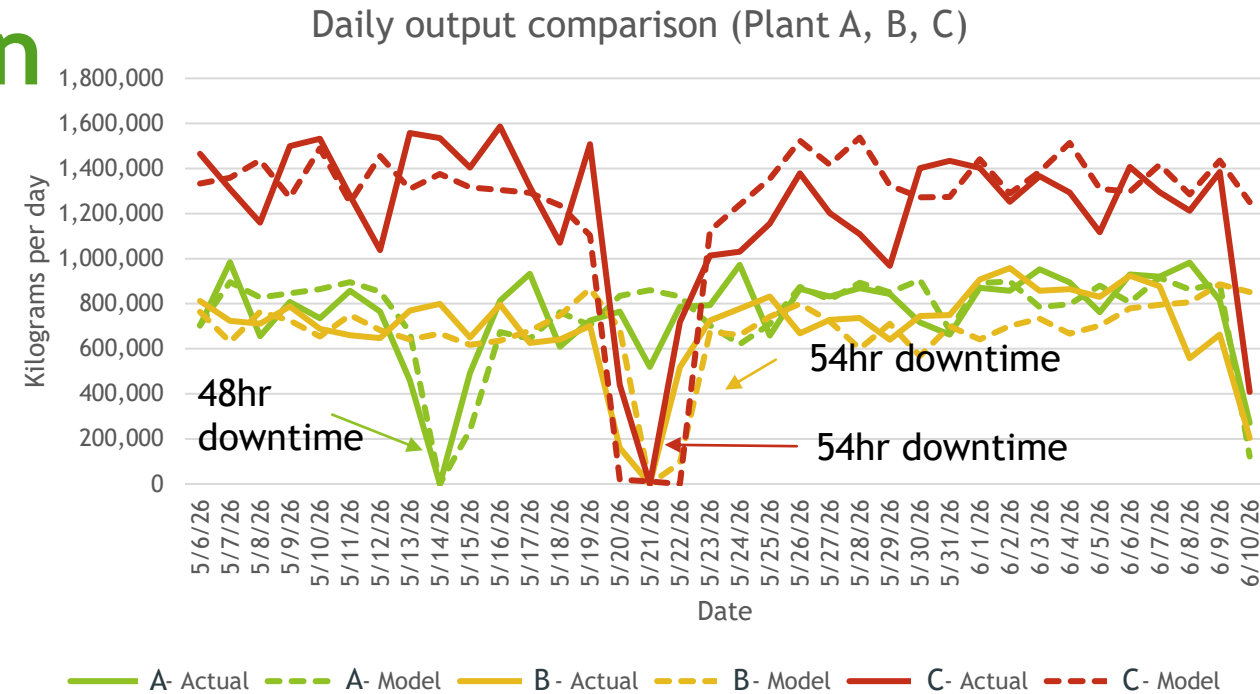
Verification and validation

6. Verification & validation

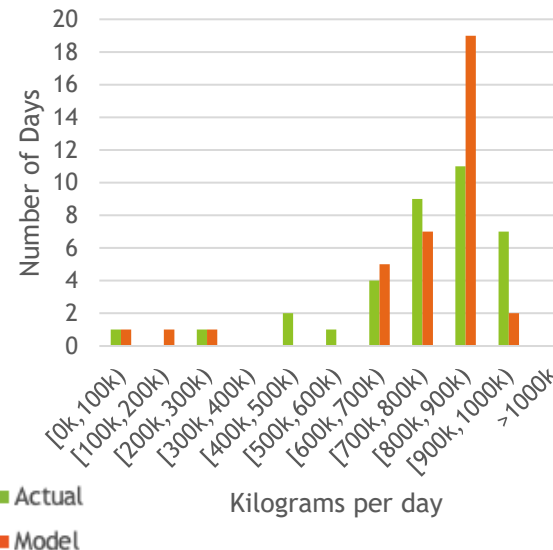
Verified and validated using several complementary methods:

- Daily output compared with historical operating data per day for each area.
- Output distributions compared to confirm observed range and frequency of actual results
- Efficiency and production ratio metrics used multiple runs to confirm statistically stable and valid results.
- Model logic, operating assumptions and process behavior reviewed with operational personnel and subject-matter experts.

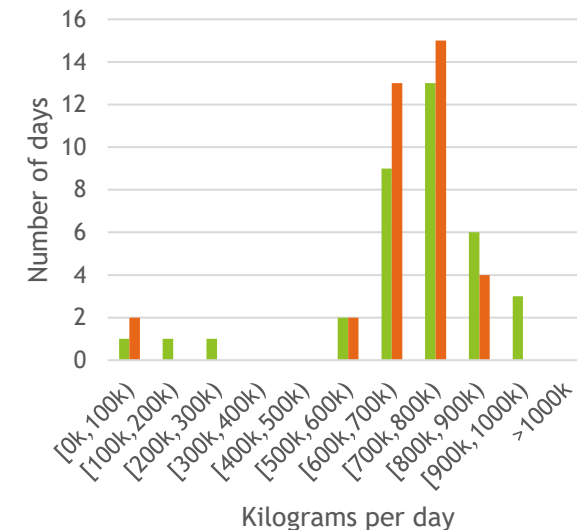
Conclusion: The model is sufficiently accurate for comparing scenarios and identifying capacity constraints.



Plant A output distribution analysis



Plant B output distribution analysis



7. Overall results

1. Distribute bagging load

- Material flow distributed evenly across all packaging lines
- No benefit - minor increase in availability but at the expense of some more toting

2. Upgrade Plant BFD merge

- Upgrade BFD merge
- Improves Plant B output, reduces Plant A slightly due to more more blockages at Main merge caused by increased flow

3. Recirculation

- Add recirculation line for each production line
- Improves Plant A and Plant B output. Significantly reduces toting

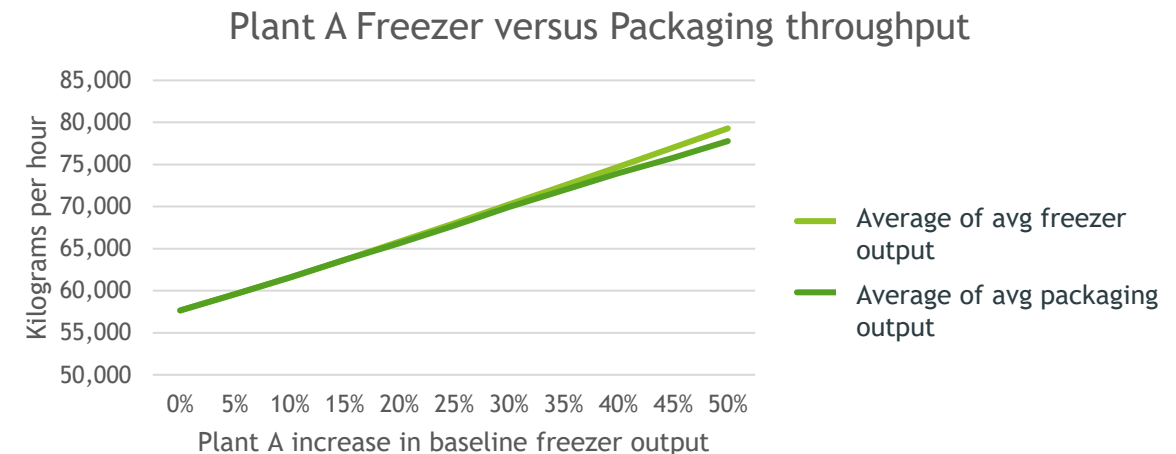
4. Add pack line at Plant B

- Add an additional packaging line at Plant B - assign all rework
- Significant reduction in toting, minor increase in throughput due to waste reduction

5. Upgrade plant A

- Increase production line throughput by up to 50%.
- First bottleneck appears at Plant A packaging - not palletizing

Parameter	Base Case (6 month)
KG: Output	109.1 M
Plant A	27.0 M
Plant B	24.5 M
Plant C	44.5 M
Pack Line Availability	
Plant A	86%
Plant B	83%
Plant C	94%
KG: Toted	5.5 M
Plant A	1.6 M
Plant B	2.6 M
Plant C	0.16 M



Questions



Jaco-Ben Vosloo

jacoben@theanylogicmodeler.com

[LinkedIn](#)



Devon Cowling

devon@theanylogicmodeler.com

[LinkedIn](#)