produce more, consume less

Testing results: Machine Demo

Date: 24/09/2025

# TECHNICAL TESTING RESULTS

Durning the final testing stages of EvoAPS, it was important to validate the results using different data models that replicated the different scheduling environments typically seen.

These included:

- Single-operation process
- Finite resource calendar patterns
- Multiple-operation processes (Operation 10, 20, 30 etc.)
- Secondary constraints
- Sequence-dependent changeovers
- Internal and external material dependencies
- Operational dependencies through a presented BoM (Bill of Materials)
- Resources that could process multiple operations at the same time (such as cooking or curing ovens and heat machines)

During the testing, rather than test each element independently, models were constructed that could reflect combinations of these points, so that real examples could be tested.

The same data models were constructed with varying numbers of operations so that the speed of achieving a result could also be tested.

Different strategies were tested in all models to see how different strategies affected the results and could be used to create a schedule.

Model tests were compared against Siemens Opcenter APS to give a benchmark. This was partly so we were able to validate the results easily, but mainly because Opcenter APS has a long history of creating very good scheduling results, making it an ideal benchmarking tool. We also have over 30 years of implementation experience with Opcenter APS and are able to utilise advanced scheduling logic to build what is referred to as a 'trained model'. A 'trained model' is where an implementor configures Opcenter APS scheduling during an implementation to achieve a specific desired result. This could be built up using multiple scheduling rules, but the logic would always be heuristic-based, and so no other result would be considered.

Models were tested against both due date order scheduling and advanced scheduling logic.

# MODEL 1 – GENERAL MACHINE DEMO

Demo model used for showing potential customers examples of how scheduling issues can be overcome.

| Operation Count | Multiple Operation | Secondary Constraints | Sequence dependant changeovers | BoM dependencies | Material dependencies | Ovens |
|---|---|---|---|---|---|---|
| 1260 | Yes | No | No | Yes | No | No |

The model is designed to replicate scheduling at a fictitious machine building factory. Orders are represented by multiple operations, and have a BoM attached so that operations have dependencies on other production orders.

There are no other constraints.

# RESULTS COMMENTS

The improvements seen against the baseline result show a very big improvement in the end time of time, with a difference of 2 days 2 hours and 15 mins. But there were two more operations that were now late, but the overall number of later orders remained unchanged.

There was also a big improvement on the finishing time of the schedule against the trained model. This was slightly less (as we would expect) at 2 days and 1 hour. But the trained model has 29 less late operations and 6 less late orders.

This raises the question as to which is the better answer? Does the business prefer a schedule that gives better utilisation or one that is more customer service level driven, but at the sacrifice of machine utilisation?

The results show that due to the fact this is a larger dataset we can use the strategies to achieve results that maybe would not be considered normally. This would then present the business with a choice to make rather than having only a single choice in schedule to distribute.

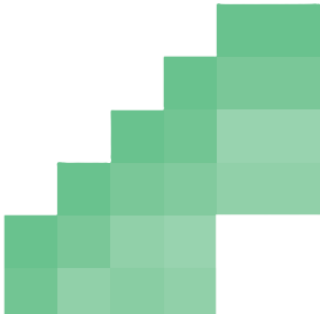
## RESULTS FROM BASE LINE TEST - SCHEDULE BY DUE DATE

| | Baseline test | EvoAPS | | |
|---|---|---|---|---|
| Schedule Creation method | Due Date | Demo Strategy | Difference | Improvement |
| Schedule Start | 10/01/2000 09:00 | 10/01/2000 09:00 | - | - |
| Schedule End | 24/02/2000 15:45 | 22/02/2000 13:30 | 2 days 2 h 15 mins | Yes |
| Make Span | 45 day 6 h 45 mins | 43 days 4 h 30 mins | 2 days 2 h 15 mins | Yes |
| Total Operations | 1260 | 1260 | - | - |
| Late Operations | 1050 | 1052 | +2 | No |
| Late Orders | 224 | 221 | -3 | Yes |
| Total Setup Time | 0 mins | 0 mins | 0 Mins | - |

## RESULTS FROM THE ADVANCE SCHEDULING LOGIC

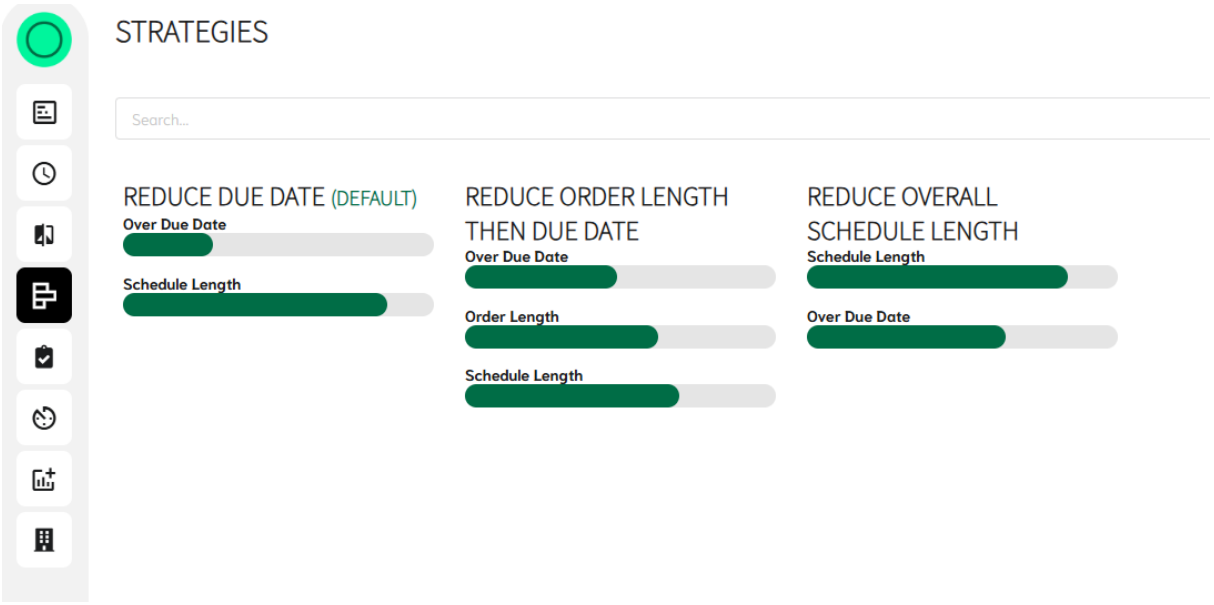
| | Opcenter APS | EvoAPS | | |
|---|---|---|---|---|
| Schedule Creation method | Advanced logic | Demo Strategy | Difference | Improvement |
| Schedule Start | 10/01/2000 09:00 | 10/01/2000 09:00 | - | - |
| Schedule End | 24/02/2000 14:30 | 22/02/2000 13:30 | 2 days 1 h | Yes |
| Make Span | 45 day 5 h 30 mins | 43 days 4 h 30 mins | 2 days 1 h | Yes |
| Total Operations | 1260 | 1260 | - | - |
| Late Operations | 1017 | 1052 | +29 | No |
| Late Orders | 215 | 221 | +6 | No |
| Total Setup Time | 0 mins | 0 mins | 0 mins | - |

As we can see improvements were made against both the baseline test and the advance scheduling logic.

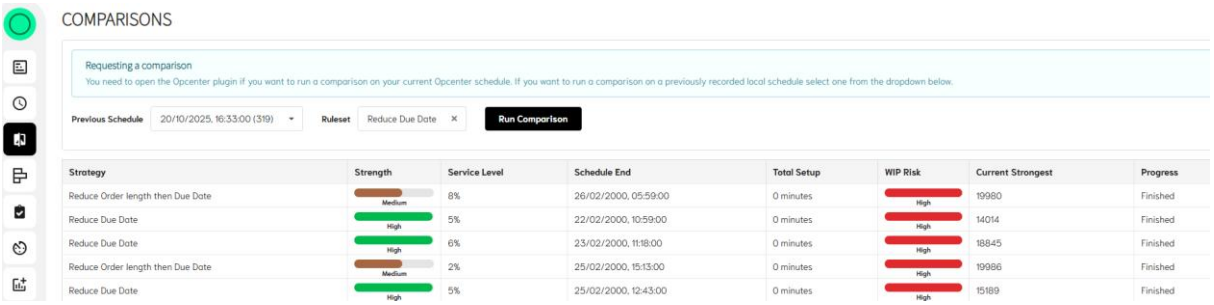


# TESTING PROCESS

The process of testing in EvoAPS is simply done by building a strategy and running a set of data against this. In this case a few different approaches were used.




Two strategies were created, Demo Strategy and Kudos Demo Starter. These would be selected, the evolutionary process run and the results returned. These results would be analysed, the strategy adjusted as required and the evolutionary process re-run. Multiple strategies can be run concurrently to keep testing time to a minimum.



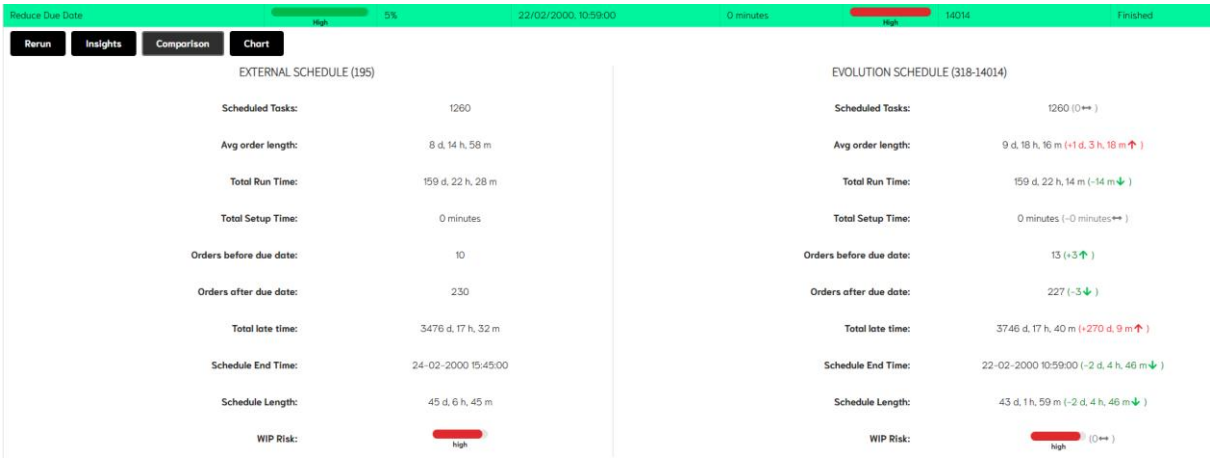

The 2 strategies above took on average around 13 minutes run. Each strategy ran for 100,000 generations creating an average of 7,500 different schedules per minute.

# VIEWING RESULTS IN EVOAPS

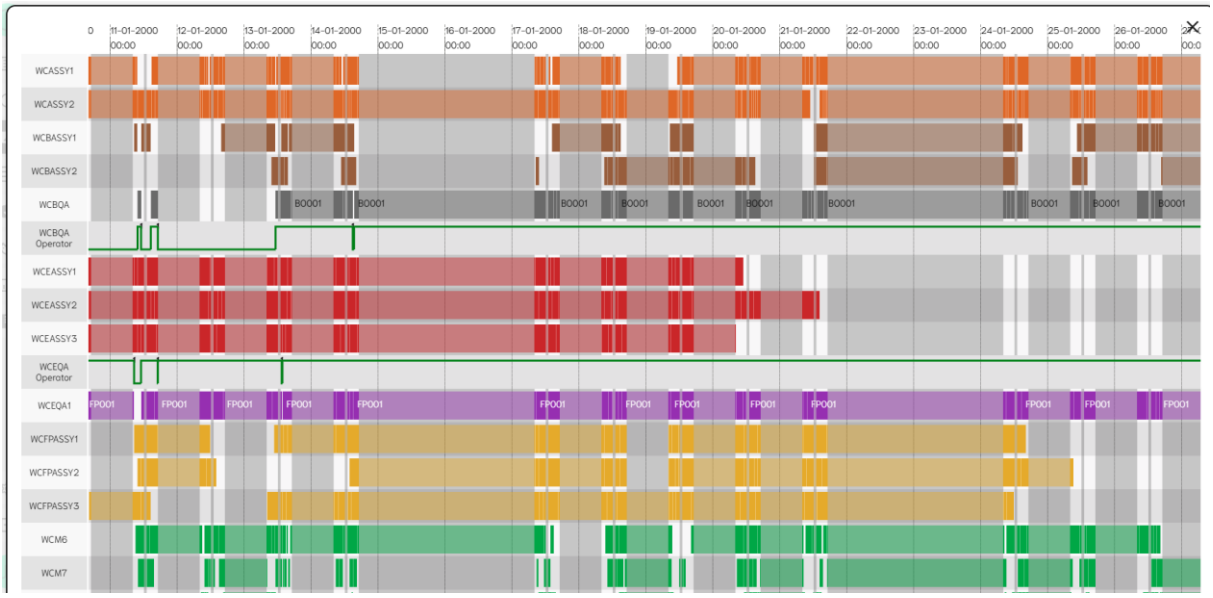
The differences between the two schedules can be viewed using the 'Comparison' feature.

The comparison shows a schedule that is currently in place in the system—displayed in the left-hand column (in this case, Opcenter APS)—alongside the results from EvoAPS in the column on the right.

The differences between the two schedules are shown in brackets, with green highlighting an improvement and red showing a negative change.



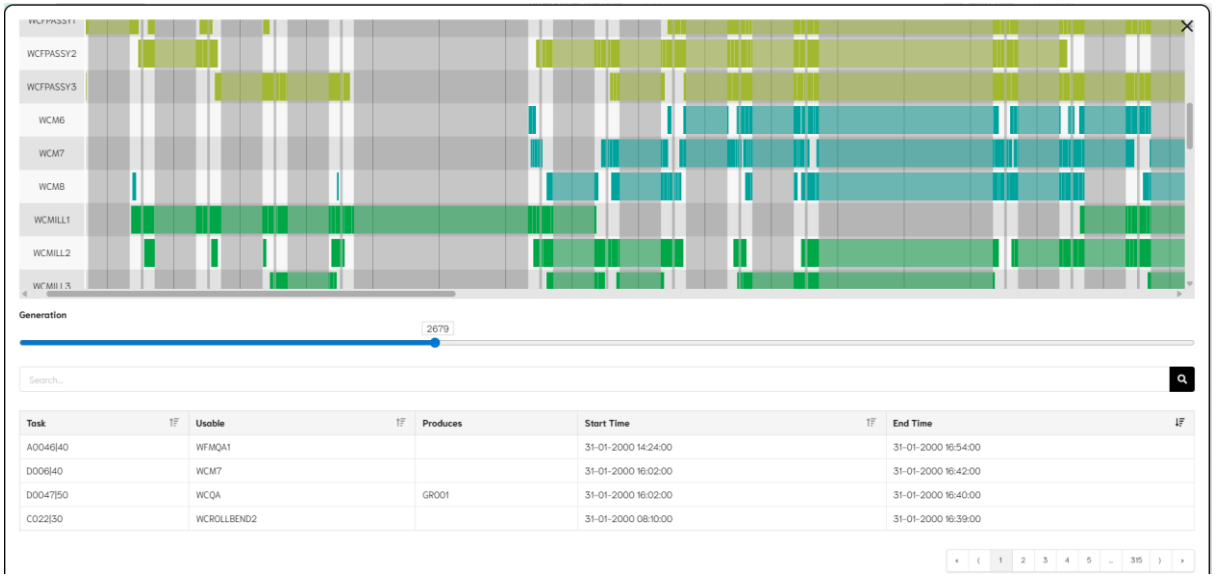

A simple 'read only' Gantt chart is also available for viewing the final schedule.



The generation history showing when and how improvements in the overall schedule are achieved, can be viewed in EvoAPS. This allows any improvements found by the evolutionary algorithm can be viewed and the strategies and their impact on results to be more easily understood.

As the number of completed schedules increases, and the improvements made, these can be seen visually on the Generation History graph below. The graph shows each element of the strategy represented along the generation timeline and indicates when a change to that element occurs. Combined, these elements make up the overall fitness result of the schedule.


More detail on each generation is available in both a Gantt chart and tabular format by selecting the ‘Show Generation History’ button. The user can scroll between the generations and see the chart and the table update in line with each change.



Are you ready to re-think scheduling?