PURDUE UNIVERSITY | The Data Mine

VALBRUNA GROUP — SYNERGIES FOCUSED ON THE FUTURE

Dynamic Production Scheduling : Optimizing Steel Flow

Maria Afif, Ben Fager, Isaac Lee, Cai Osemwota, Matthew Murphy, Luke Wells, Jason Xie

INTRODUCTION

Valbruna is a leading producer of stainless steel and nickel alloys since 1925. Producing 300,000+ tons per year for industries like infrastructure, engineering, military, etc.

Problem → Steel orders compete for machines, which causes delays and slows production

Purpose → Build a scheduling tool that prioritizes jobs based on deadlines, demands, and material availability

Impact → Improve production flow, reduce delays, facilitate optimal decision-making

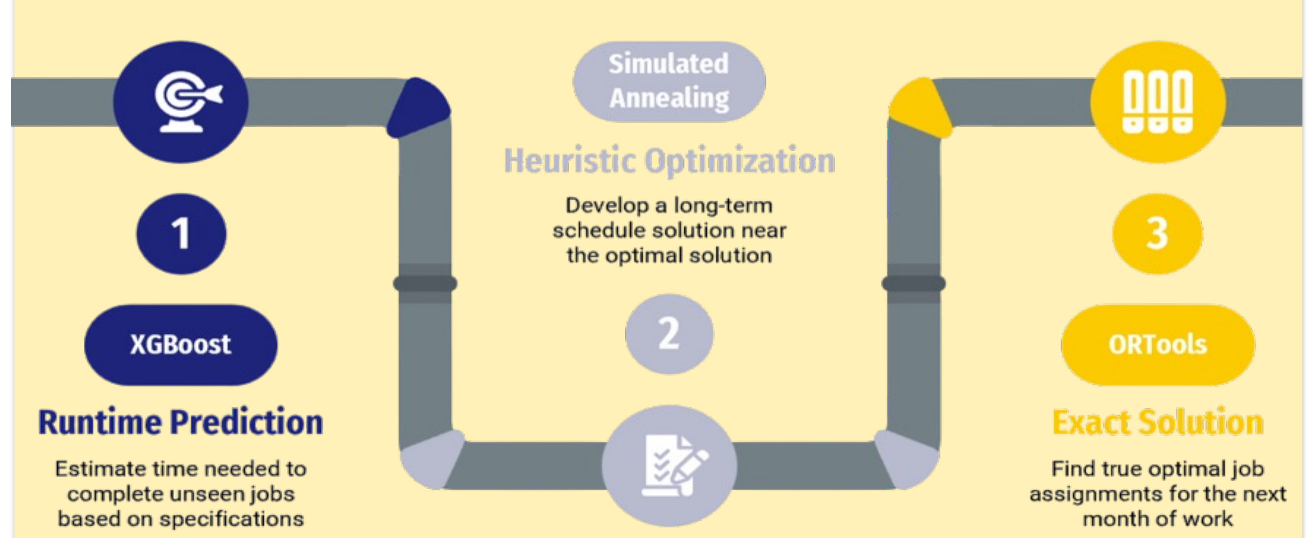
METHODOLOGY

❖ **Problem Modeling**
- Steel production → Job shop scheduling problem with flexible machines that run concurrently
- Job = sequence of operations with constraints resulting in a finished product
- Included "phantom" machines for operations that require time but no machine (transfers, cooling, testing, etc.)

▪ **Method**
- Machine Learning predicts runtimes
- Combine two scheduling methods:
  - Simulated Annealing / Heuristics → fast schedule
  - OR-Tools → optimized schedule

❖ **Tools**
- Python language
- Constraint programming: OR-Tools
- Runtime Prediction: XGBoost, Random Forest
- Data Processing: pandas, NumPy, JSON
- User Interface: Dash, Plotly
- Visualization: matplotlib, seaborn

The Scheduling Pipeline



1. Runtime Prediction

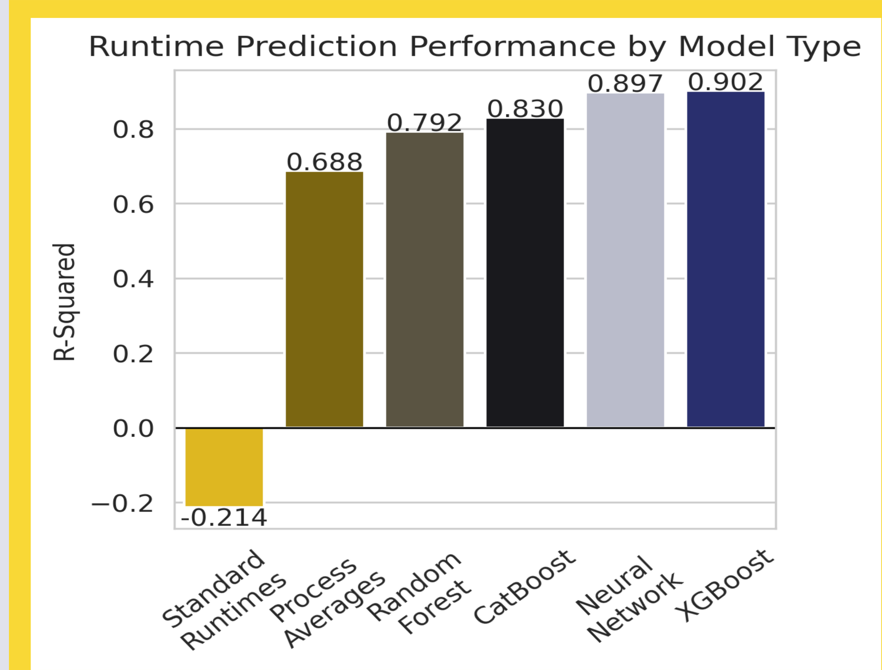


Figure 1. Runtime Prediction Performance

Figure 1:

❖ **Runtime prediction performance:**
- Tested 5 models
- Compared using R^2 score (proportion of explained variance)
- XGBoost was best ($R^2 = 0.902$)

Figure 2:

❖ **Model Prediction Accuracy**
- Compares predicted vs actual run times
- Run time = time for a machine to complete an operation

❖ **Result:**
- Over 94% of predictions are within ±2 hours of actual time → very good

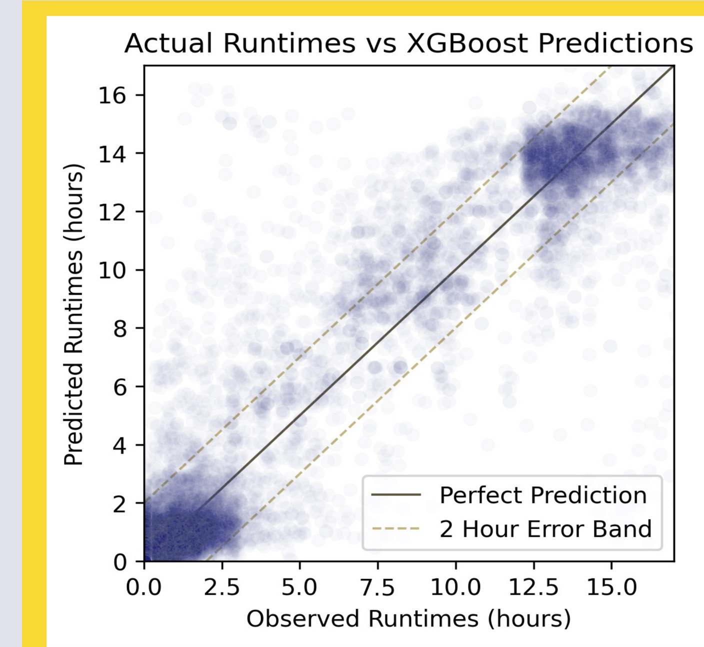


Figure 2. Observed Runtimes vs XGBoost Predictions

2. Scheduling

❖ **Initial Problem:**

Finding the perfect schedule for all the jobs:
a. Would take too long to produce
b. Wouldn't be practical for common sudden changes like machine breakdowns

❖ **Our Solution:**

Use a hybrid approach combining two methods:

1. Heuristics (SA)
❖ Builds a good quick overall schedule
❖ Works well with large numbers of jobs
❖ Uses little computer power and memory

2. OR-Tools (Constraint Programming)
❖ Refines short term schedule, more efficient outcome
❖ Makes sure all real-world constraints are satisfied
❖ Uses a lot more resources

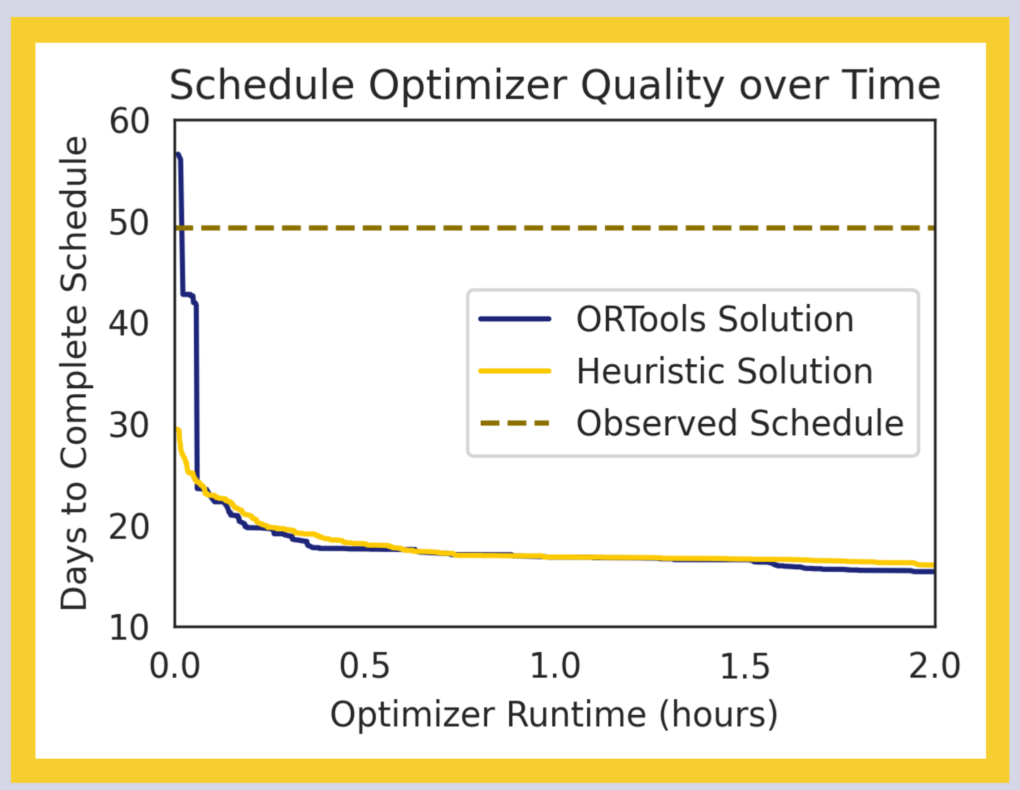


Figure 3. Quality of Heuristic vs OR-Tools

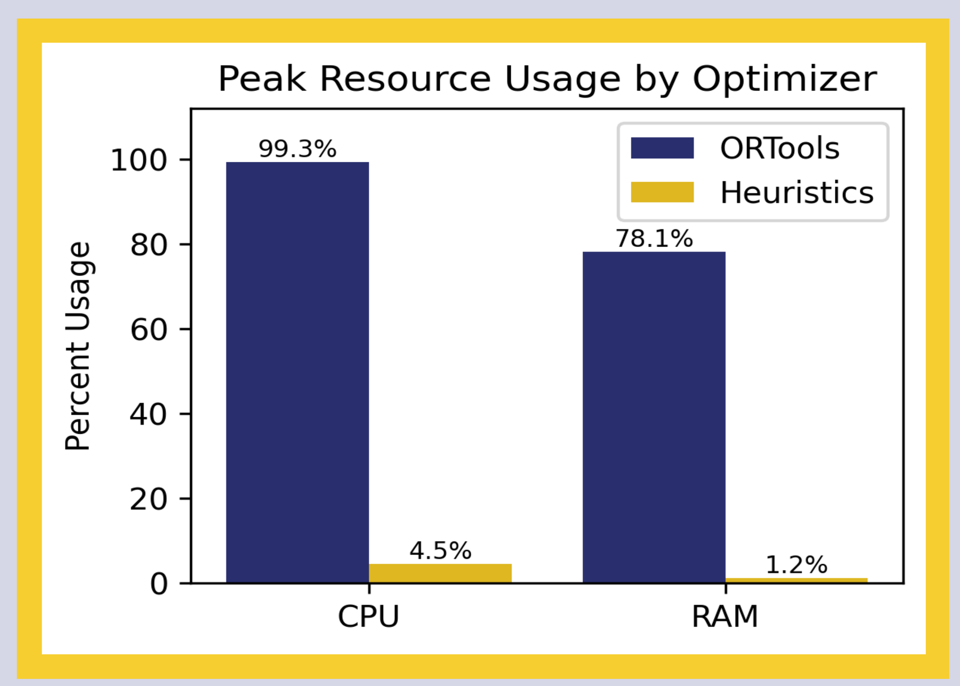


Figure 4. Resource Usage by Optimizer

Benchmark performance:
24-core CPU with 32GB RAM

3. Results

Figure 5. Final Daily Schedule

Comes with interactive features (Figure 5)
- Hovering over job shows details (task ID, start time, duration, deadline)
- Helps monitor progress and spot delays

❖ **Job Prioritization**
- Using a weighted lateness metric
- Accounts for:
  - ✓ Deadlines
  - ✓ Customer importance
  - ✓ Production efficiency

❖ **Realistic Model**
- Uses predicted runtimes as input instead of fixed times
- Satisfies real conditions
- Handles multiple jobs while respecting machine constraints

CONCLUSION

Problem: Inefficient scheduling = delays, wasted potential and $

Solution: Built a scheduling system using ML (XGBoost) + optimization (Heuristics + OR-Tools)

Result: Accurate runtime predictions and better weekly schedules

Impact: Less delays and late fees, better machine use, faster and smarter decisions, handles large amount of jobs

To sum up:
❖ Heuristics = fast + flexible
- big picture

❖ OR-Tools = precise + accurate
- details

Together:
- ✓ Schedules quickly produced
- ✓ High performance
- ✓ Balance for speed and precision
- ✓ Realistic constraints

FUTURE GOALS

Full deployment:
❖ Rewrite an interactive scheduling tool using React.js for better scalability
❖ Update model:
- Add updated real-time data
- Explore more models to reduce prediction error (e.g., reinforcement learning)
- Include machine downtime, maintenance and worker availability

❖ New features:
- Evaluating tool that automatically regenerates schedule when problem occurs
- Predict and incorporate setup times

Going Beyond:
❖ Drag and drop tool to change job assignments
❖ Live schedule updates with smart alerts (e.g., late warnings, change suggestions
❖ Incorporate optimization for energy usage and cost (not just time)

REFERENCES AND ACKNOWLEDGEMENTS

❖ Product Owner: Valbruna Stainless (Fort Wayne)
- Special thanks to representatives Joseph Hawkins, Jerrid Bilderback, Josh Patton, Mason Spenn

❖ Faculty & Project Manager: Dr. Yihao Deng, Collin Garmon
❖ Nate Sodolski, Team Member Fall 2025
❖ Facilitating Organization: Purdue University's Data Mine

The Data Mine Corporate Partners Symposium 2026