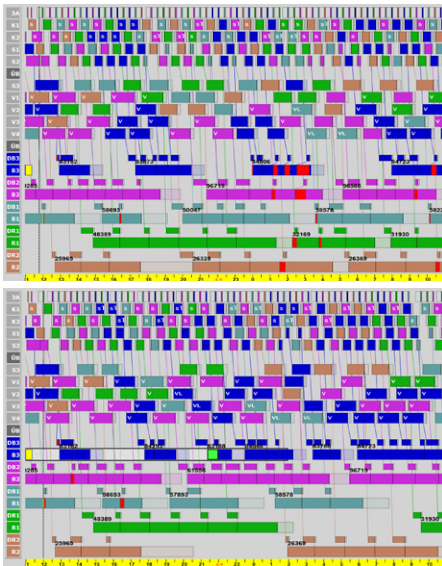


HKM Optimizes Just-in-Time Steel Manufacturing Schedule



Manually reviewed plant schedule (top) and plant schedule automatically optimized with MATLAB genetic algorithms (bottom). The optimized schedule minimizes schedule conflicts (in red), meets delivery dates, and achieves the target utilization rate.

The Challenge

Optimize a steel production process to enable consistent, just-in-time delivery

The Solution

Use MATLAB and Global Optimization Toolbox to maximize throughput of more than 5 million tonnes of steel annually

The Results

- Algorithm development accelerated by a factor of 10
- Optimization time cut from 1 hour to 5 minutes
- Customer satisfaction increased

To manufacture more than 5 million tonnes of steel annually and deliver each order just in time, the Hüttenwerke Krupp Mannesmann (HKM) plant must operate on a tightly controlled schedule. Each phase of the process is carefully timed and orchestrated. Tons of coke, pig iron, scrap, and other raw materials must arrive at the appropriate machinery exactly when needed, blast furnaces must be stoked to temperatures of at least 1450 degrees Celsius, and the steel must be cast before it has time to cool. In the making of pig iron, the stages of the process can be handled through linear production, but in the steel plant, logistics and customer requirements add complexity.

To maximize throughput while meeting these operating constraints, HKM engineers developed an automated scheduling system in MATLAB®. “MATLAB enabled us to rapidly develop a system for global optimization of our steel process, deploy it as a Java component in our production environment, and run it as a computing cluster,” says Alexey Nagaytsev, project manager at HKM. “With MATLAB we can easily make changes to incorporate new constraints and scale the system to meet new demands.”

The Challenge

Steel production is a complex process, with more than 150 discrete steps and numerous constraints imposed by the capabilities of the equipment, the availability of raw materials, environmental regulations, and the demands of multiple customers. Customers typically require delivery of their orders in precisely scheduled stages. In HKM’s

previous process, dispatchers managed scheduling for the entire plant manually. While manual operation does a good job of maximizing throughput, it cannot fully account for the delivery schedule imposed by customers.

The role of dispatcher required such extensive knowledge of the process that only a few employees were qualified to fill it. If either dispatcher were absent, the business incurred significant risk.

To mitigate this risk, HKM needed an automated system capable of maximizing plant production and meeting the delivery schedule. “We wanted the system to provide objective results for a complex optimization problem in minutes,” says Nagaytsev. “The system had to be flexible enough to accommodate changing constraints. It also had to integrate with our existing Java software and be available 24/7. Lastly, we wanted to deploy the system quickly while keeping development costs low.”

The Solution

HKM used MATLAB and related optimization, parallel computing, and application deployment products to develop a complete plant scheduling system that supports efficient, just-in-time delivery of its steel products.

Using MATLAB and Global Optimization Toolbox, Nagaytsev quickly prototyped the core optimization module. The module uses a genetic algorithm to search for global solutions that will maximize plant efficiency based on customer requirements as well

“C++, Java, or third-party optimization solutions would have required us to spend significantly more time in development or to simplify our constraints. Only MATLAB provided the flexibility, scalability, development speed, and level of optimization that we required.” — ALEXEY NAGAYTSEV, HKM

as on time, temperature, and equipment usage restrictions.

To accelerate the optimization process, Nagaytsev used Parallel Computing Toolbox™ and MATLAB Distributed Computing Server™ to execute the optimization algorithm on a 16-worker Linux® computing cluster.

Nagaytsev generated plots in MATLAB to visualize intermediate results from the algorithm, which he used to create a custom genetic algorithm variant.

He used MATLAB Builder™ JA and MATLAB Compiler™ to create a Java™ class from the MATLAB code, which was then deployed as a Java component in HKM's existing scheduling application.

The MATLAB based Java component retrieves up-to-date constraint and status information from the main system, computes an optimized schedule, and then returns the results to the main system, where they are accessed by the dispatchers.

HKM dispatchers are currently using MATLAB based optimizations to plan schedules for the next 24 to 48 hours, as well as longer-term schedules that span up to 4 weeks.

The Results

Algorithm development accelerated by a factor of 10. “With MATLAB we completed a working version of the optimization algorithm in just 20 days,” says Nagaytsev. “Implementing a comparable algorithm in Java would have taken about 10 times longer and cost 10 times as much.”

Optimization time cut from 1 hour to 5 minutes. “Our process is so complex that initially it took about an hour to complete a full optimization,” notes Nagaytsev. “With minimal code changes we set up the application on our cluster using Parallel Computing Toolbox and MATLAB Distributed Computing Server. Now we have results in about 5 minutes.”

Customer satisfaction increased. “In some cases the automated system has produced a solution that our dispatchers did not consider,” says Nagaytsev. “They use the objective results produced by MATLAB and Global Optimization Toolbox to continuously improve the process and reduce risk. Our customers are more satisfied because they can get their material exactly at the time they requested it.”

Industry

- Industrial automation and machinery

Application Areas

- Data analysis
- Algorithm development
- Mathematical modeling
- Parallel computing
- Desktop and Web deployment

Products Used

- MATLAB®
- Global Optimization Toolbox
- MATLAB Builder™ JA
- MATLAB Compiler™
- MATLAB Distributed Computing Server™
- Parallel Computing Toolbox™

Learn More About HKM

www.hkm.de/english