Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

A WEB-BASED SIMULATION OPTIMIZATION SYSTEM FOR INDUSTRIAL SCHEDULING

Marcus Andersson
Henrik Grimm
Anna Persson
Amos Ng

Centre for Intelligent Automation
University of Skövde
PO Box 408, 541 28, SWEDEN

ABSTRACT

Many real-world production systems are complex in nature and it is a real challenge to find an efficient scheduling method that satisfies the production requirements as well as utilizes the resources efficiently. Tools like discrete event simulation (DES) are very useful for modeling these systems and can be used to test and compare different schedules before dispatching the best schedules to the targeted systems. DES alone, however, cannot be used to find the "optimal" schedule. Simulation-based optimization (SO) can be used to search for optimal schedules efficiently without too much user intervention. Observing that long computing time may prohibit the interest in using SO for industrial scheduling, various techniques to speed up the SO process have to be explored. This paper presents a case study that shows the use of a Web-based parallel and distributed SO platform to support the operations scheduling of a machining line in an automotive factory.

1 INTRODUCTION

In general, scheduling concerns "the allocation of resources over time to perform a collection of tasks" (Baker 1974). In practice, scheduling refers to "the determination of a set of orders, which will be processed by the resources during a short-term period (day, week, etc.)" (Kiran 1998). When all numeric quantities (processing times, due dates etc.) are known in advance, the scheduling problem can be classified as a deterministic scheduling problem. In contrast, numerical quantities are stochastic in a stochastic scheduling problem. A static problem is when jobs are assumed to be available at time 0, and a dynamic problem is when a sub-set of jobs has a non-zero release or ready time. According to Kiran (1998), scheduling problems can be defined into four different categories: static stochastic, static deterministic, dynamic deterministic or dynamic stochastic and can be addressed by three basic approaches:

- Mathematical optimization approaches
- Dispatching rules and simulation-based approaches
- Artificial Intelligence (AI) based approaches, such as meta-heuristic search methods.

Mathematical optimization approaches try to find the optimal schedule by analytical mathematics. There are different techniques that may be used depending on the problem to be solved. For example, linear programming can be used in many scheduling optimization problems, given that the objective function and the constraints can be defined as linear equations (Kiran 1998). A major drawback of the mathematical approaches is their limitation in representing real-world scheduling problems that are usually dynamic and stochastic in nature. In order to formulate a solvable mathematical model, unexpected events, such as machine break-down, rush orders and tools shortage that occur frequently cannot be taken into account. At the same time, the complex mathematical techniques involved induce low acceptance to actual industrial users when compared with, e.g. dispatching rules.

Basically, a dispatching rule is a rule of thumb that gives priority to a job among other jobs at a specific stage. It is the most common approach used to solve industrial scheduling problems because of its computationally efficiency and easy implementation. However, dispatching rule based approaches do not try finding optimal schedules, but rely on knowing that one, or a combination of scheduling rules would statistically perform better in producing satisfactory performance. In general industrial practice, dispatching rules are applied with no consideration to system stochastic behaviors. Nevertheless, steady-state performance with consideration of system dynamics and stochasticity of dispatching rules can be made by combining dispatching rules with simulation, which is commonly referred as simulation-based scheduling.

Simulation modeling has the capability to represent complex real world systems in detail. They are also very

1-4244-1306-0/07/$25.00 ©2007 IEEE

useful for communicating details, such as a scheduling situation, because of the visual aids that simulation brings. A simulation model built for scheduling is quite different when compared to an ordinary simulation model. Simulation models are generally used for the design and analysis of an existing or proposed system, while simulation-based scheduling is used for on-going operation and control of the system and the ultimate output is a detailed operation plan. Simulation-based scheduling approaches are derived from the group of dispatching rule based approaches. In a simulation-based approach several dispatching rules might be used at different stages in order to make a decision. A scheduling system could be used as a tactical tool to determine how jobs are to be planned for the next scheduling period (Koh, Souza and Ho 1996; Kiran 1998).

Simulation-based scheduling might, however, include much user intervention in order to manually test different schedules, and it is no guarantee that a good schedule is found. In order to automatically search for optimal solutions, a scheduling problem can be solved by using simulation-based optimization (SO) in which the simulation model is integrated with meta-heuristic search methods, such as genetic algorithms (GA). By integrating these two techniques the performance of different schedules in a complex production system can be accurately forecasted by the simulation model whilst the optimization makes it possible to find a “optimal” schedule, provided that such a schedule can be found in a limited computational time. With the observation that long computing time may prohibit the interest in using SO for decision-making support and/or operations scheduling, Ng et al. (2007) have developed a Web-based parallel and distributed simulation platform, which is called OPTIMISE (OPTIMization using Intelligent Simulation and Experimentation), to enable computationally expensive simulation to be run in parallel and distributed processors.

This paper presents a case study of the application of applying SO to the scheduling of a camshaft machining line in an automotive factory in Sweden, using the Web-based parallel and distributed computing platform OPTIMISE. The rest of the paper is organized as follows: Section 2 gives a brief overview on the background of the case study and its scheduling problem. Section 3 introduces the GA as the meta-heuristic method used in the SO process. Section 4 presents the Web-based computing platform OPTIMISE. The industrial scheduling application developed using the OPTIMISE framework is presented in Section 5. Section 6 summarizes the optimization results so far in this study. Conclusions and future work can be found in Section 7.

2 CASE STUDY AT VOLVO CARS

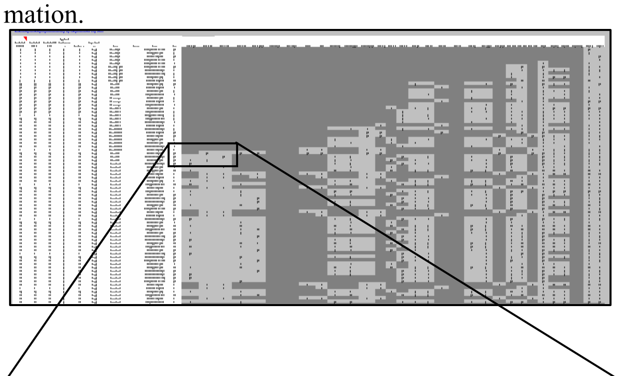
This section gives the background of the real-world scheduling problem and the simulation model developed to help users to evaluate different schedules through an input data interface.

2.1 Background

Volvo Cars Engine in Skövde, Sweden, is responsible for supplying engines for Volvo cars and some of Ford’s car models. The camshaft machining line needs to increase the overall throughput of the system, but the problem is that scheduling of the line is highly complex and is very difficult for a good schedule to be found by the current shop floor control system. The line consists of 14 different machine groups and each group has one to seven parallel machines; there are in total 34 machines in the whole line. However, unlike an ordinary flow shop with parallel machines, each machine has its own processing time, physical capability and limitations, as well as variability in terms of failure and setup. At the same time, there are many different types of products and each type of product has its own path through the line. The machining line is semi-automated with robots that feed machines inside cells, but the loading, unloading and the decision of when or where to process different types of parts is decided by operators at each work area. Currently, batches of different product variants are scheduled to start in a specific order, in order to keep the variants in the finished goods stock above a security level over time, but without much consideration on the sequence dependent setup times. The security levels are important because these make it possible for the line to feed camshafts downstream even if a major breakdown of a machine occurs. Even though different batches are prioritized in a specific order, the operators usually re-schedule them to minimize the number of setups. The consequence of these manual decisions in the machining line is that some machines might be optimized, but not the overall performance of the line, especially when unexpected events, like urgent orders, machine failures occur.

2.2 Input Data Interface

A simulation model was developed to represent the existing line in order to be able to answer production and scheduling related questions. The users of the simulation model are not expected to be simulation experts and therefore they will work from a custom made graphical user-interface (GUI). The GUI, built in Microsoft Excel, is important because it enables the users to flexibly test many different options in the settings and some attributes of the simulation model, without the need to open the simulation model. The GUI includes production input data such as machine cycle times, availability, setup settings, buffer capacity, etc. in order to be able to easily configure and run production related tests. It also includes different settings that are connected to the scheduling of the line such as start conditions and schedules for different parts of the line, see Figure 1. Details of the line, including carts, are also modeled. The cart is a resource that the camshaft are transported on between operations. It is possible to change the number of carts between operations in the GUI, without modifying the model. It is also possible to set which machines that should be available or not depending on current status of the line, define variant path through operation groups, planned stops on certain machines or the whole line due to maintenance, meetings, etc. When compared to classical scheduling methods, simulation-based scheduling can flexibly and dynamically take into account this information.



		OP Group 1			OP Group 2	OP Group 3
Batch	Size	OP 20/30-1:2	OP 20/30-3:4	OP 20/30-5:6	OP 35	OP 40
1	150		150		150	150
2	150			150		
3	150	50	100			
4	150	150				
5	50			50		
6	450	200	250			
7	500			500	500	500
8	600		300	300		600
9	400	200	200			

Figure 1: A screenshot of a schedule in GUI

The GUI allows and helps the user to manually fill in a schedule of the line and has a capability to control if the user has made a valid plan. In Figure 1, each row is one batch of a specific variant that is scheduled over several machines. The first rows represent Work-In-Process (WIP) in the line and these are only to be scheduled from its start position. The WIP at the end of the line allocates the first row followed by succeeding batches upstream in the line. The new batches, which are to be started, are allocated beneath the WIP in a prioritized order. After this step it is possible to run a macro to color each cell in the matrix in order to help the user to easily identify in which machines it is possible to schedule each batch (light-grey fields), which depends on a batch's start position, variant type and machine related settings. The user can fill in the matrix to make a schedule and run a macro to check whether it is valid. Another feature is that it is possible to re-prioritize batches at different parts of the line in order to run the production more efficiently. For example, it might be desirable to hold some batches at a specific stage in order to reduce the number of setups. It is possible to run the simulation model in deterministic or stochastic mode by the corresponding setting in the GUI.

3 THE OPTIMIZATION ALGORITHM

The optimization strategy implemented is based on a GA. GAs are population based search algorithms inspired by theories from natural evolution. The basic idea behind these algorithms is that a population of individuals represents possible solutions of a given problem. Through recombination of solutions, offspring are created, forming a new generation of the population. Some of the solutions are better suited for the problem and these are given more opportunities to reproduce and pass their desirable behavior to the next generation, similar to natural selection. New generations of the population are evolved until a sufficiently good solution is found.

GAs have been proven to be very flexible and reliable in searching for global solutions (Baesler and Sepúlveda 2001) and also capable of solving complex scheduling problems (Azzaro-Pantel 1998). Their characteristics making them suitable for solving multi-objective simulation-based problems (Eskandri, Rabelo and Mollaghasemi 2005) and they can easily be coupled with any discrete-event simulation models, in contrast with some other heuristic methods which are more suitable only to certain problems (Azzaro-Pantel 1998).

The rest of this section describes the GA implemented for solving the multi-objective optimization problem considered in this paper.

3.1 Representation of Solutions

The GA encodes possible solutions as genomes and each genome instance represents a single solution to the problem – in this case an operation schedule. In many applications, the efficiency of GAs is determined mainly on how the domain problem is encoded in the genome and the representation has therefore been considered carefully in this study.

The genome for this problem is designed to represent the schedule presented in Figure 1. The genome is implemented as a matrix in which each row corresponds to a specific batch that is scheduled over several operations. The internal order of the different variants is determined by a sequence number, which is also part of the genome. Table 1 below shows an example of a simplified genome with two batches scheduled on two operation groups. The task of the GA is to fill in the light-grey cells of the matrix while meanwhile considering all constraints. The cart size is a user-defined constant set to be 50.

Table 1: Example of genome.

		Operation group 1			Operation group n		
Batch	Sequence number	Op1	Op2	Op3	Op1	Op2	Op3
batch$A$	2	50	50			50	50
batch$B$	1		50	50	50	50	

3.2 Genetic Operators

A first population of 50 candidate solutions is generated satisfying all the constraints. It is possible to use an initial batch sequence order defined by the user and randomly generate the rest. The carts are randomly allocated for the allowed operations. A uniform crossover operator is used in which a new solution is created by taking each row of the matrix randomly from one of the two parents. The crossover operator only applies to operations and not sequence numbers; those are simply inherited from the parents. To maintain the genetic diversity from one generation to the next, some of the offspring solutions are mutated.

3.3 Fitness Function

A fitness function quantifies the quality of solutions by assigning a fitness value to each of them corresponding to their performance. An essential objective for the camshaft machining line is keeping the security levels of the variants in the finished goods stock. The production planners and other experts of the system would rather not like to go under the security levels, but minor shortage could be allowed if a schedule can increase overall system throughput at the same time. A large shortage is not desired because of the risk it conveys. Not only is the performance mean important because a high variation could cause major losses; hence, a robust schedule is requested. Based on quadratic loss function (Sanchez 2000), the expected loss of shortage (S) in Equation (1), is a good approximation in the representation of this objective.

$$S(loss) = w\left(\sigma^2 + \left(Y(x) - \tau\right)^2\right) \tag{1}$$

where, $Y(x)$ is the performance mean and σ is the standard deviation for variable settings x, τ is the target value, and w is the objective weight. The quadratic loss function, penalizes small values of deviation to target little, but penalizes large ones much. In this case a one-sided quadratic loss function is used, because a loss only occurs if a shortage of the security levels takes place. Some of the product variants are more important than others when considering preventing a shortage of the security level. Therefore, an objective weight is used to prioritize the product variants to each other. The performance mean is measured in a percentage scale, where zero means that the security level were kept, 100% means that the quantity of this product variant in the finished goods stock was empty, and more than 100% means that there was a deficit of this variant in the finished goods stock at some time. The 18 different product variants have their own weight according to its relative importance to the overall objective function. As long as the security levels of the product variants have no or a small shortage, the most important objective for the camshaft machining line is to increase the overall through-put of the system. Not only is the performance mean important because a high variation could cause major losses. Therefore the objective function for throughput (T), Equation (2), was created according to the values of production planners and other experts of the system. The objective function was formulated into a minimization objective and can be written as:

$$T = w\left(k\sigma + \left(\tau - Y\left(x\right)\right)\right) \tag{2}$$

where, $Y\left(x\right)$ is the performance mean and σ is the standard deviation of variable settings x, τ is the target value, k is a constant, and w is the objective weight. The target value is set to theoretically max throughput (τ = 225) in order not to exceed it. The function has a constant, k = 0.5, because the performance mean is more important than performance variation according to the experts of the system, but still robustness is essential.

The fitness value (F) assigned in this case is based on a combination of throughput and shortage and is calculated using the Equation:

$$F = w_t\left(0.5\sigma_t + \left(225 - Y(x)_t\right)\right) + \sum_{s=1}^{18} w_s\left(\sigma_s^2 + Y(x)_s^2\right) \tag{3}$$

where, w_t is the objective weight of throughput, σ_t is the standard deviation of throughput, $Y(x)_t$ is the performance mean of throughput, w_s is a variant's objective weight of shortage, σ_s is a variant's standard deviation of shortage, and $Y(x)_s$ is a variant's performance mean of shortage.

4 OPTIMISE: A WEB-BASED SO ENVIRONMENT

OPTIMISE is a Web-based parallel and distributed computing platform that supports multiple users to run experiments and optimizations with different simulation systems. The platform is designed to be multi-tier client/server based in which all complex components, including various meta-heuristic search algorithms, neural network based meta-models, deterministic/stochastic simulation systems and the corresponding database management system are integrated in a parallel and distributed platform and made available for general users to easily access, anytime, anywhere, through Web Services technology (Ng et al. 2007). As shown in Figure 2, at the heart of the OPTIMISE architecture is a number of optimization engines, surrounded by a set of OPTIMISE Server Components which spread across three tiers: (1) Web Server; (2) Optimization and (3) Simulation subsystem. In a SO application supported by the OPTIMISE framework, the optimization engine (OptEngine) in the optimization tier is the most important component because they provide the core functionality for major algorithmic processing and act as the hubs for coordinating other functions, including data logging and metamodeling. Server components can be accessed by client applications through consuming the OPTIMISE Web services, hosted by the Web server as shown in Figure 2. The Web server listens to the XML requests and acts accordingly.

The Optimization Manager (OptManager) is Windows service that listens to the request from the Web Server to launch different OptEngines according to the preferences and parameters specified by the users through client applications. In the OPTIMISE server, simulation components are decoupled from the core server components in order to support parallel simulation. Different simulation systems can be connected to the SimManager homogenously via SimAgents. When a SimAgent is launched it will be registered at the SimManager as an available resource. The current dispatching strategy is that an idle SimAgent will pick up a job that is pending in the message queue of the SimManager and launch the targeted simulation system.

In principle, any applications that consume the Web services provided by OPTIMISE can be called an OPTIMISE client application. In order to supply the required data needed for running SO for the industrial scheduling problems, the GUI mentioned in Section 2.2 was extended to connect to the OPTIMISE Web services for launching SO for the camshaft machining line. On the other hand, there are some generic applications developed for the monitoring/control of the OPTIMISE Server Components and manage optimization project data. By generic, it means that they are not specific developed for a particular application. One of such an application that can be very useful for a wide range of SO applications is OPTIMISE Browser, with which it is possible to browse new and historical optimization data from OptDB, using advanced data plotting and analysis. How the OPTIMISE framework and client applications are used for production planners to do weekly production scheduling in the industrial case study is the topic of the next section.

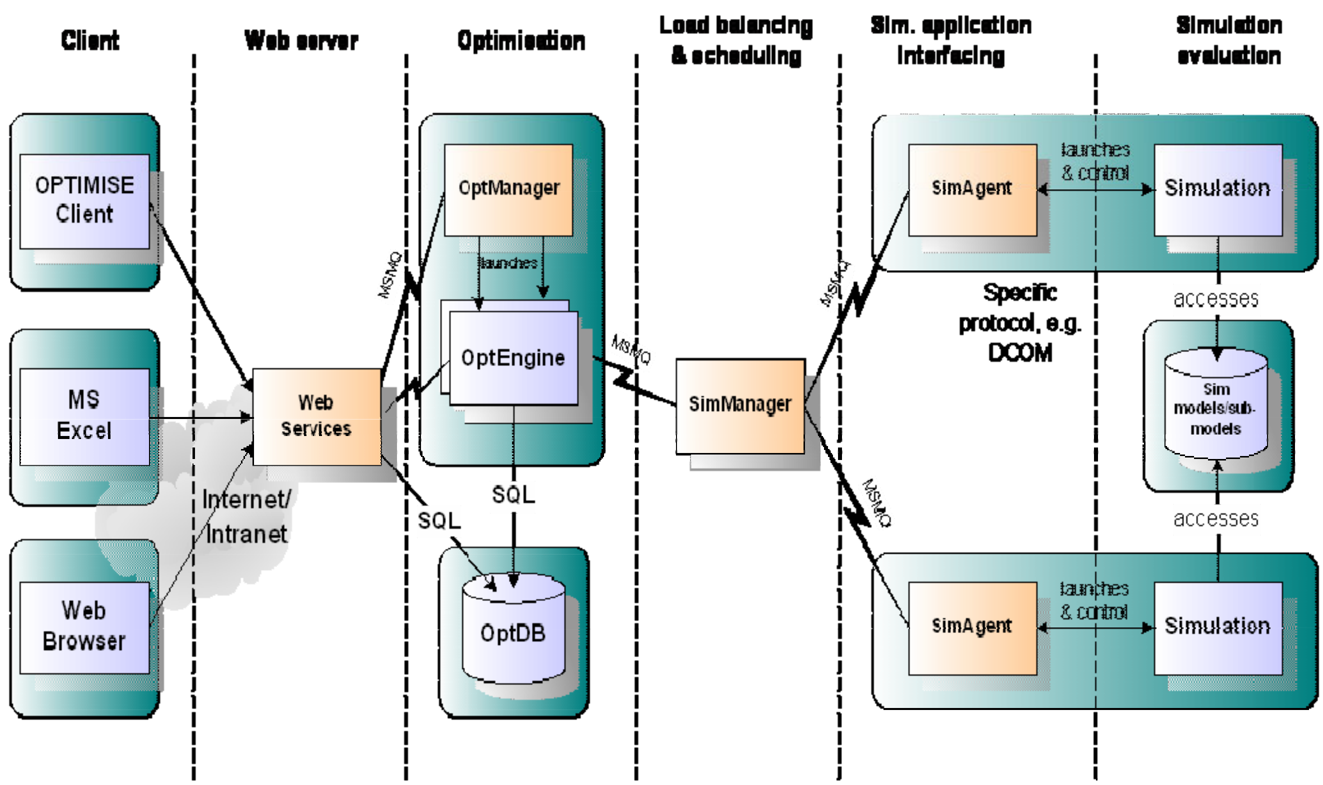


Figure 2: The system architecture of OPTIMISE

5 OPTIMISE FOR INDUSTRIAL SCHEDULING

5.1 Using OPTIMISE Browser

With OPTIMISE Browser, it is possible to see different data plots, studying the optimization progress and generate Gantt charts for each individual evaluations. Figure 3 provides the screen shots of the information that can be viewed using OPTIMISE Browser. When the SO is finished the user can use the best operation schedule for the real production line. Not only are the output data related to the fitness function important for the industrial users to view, but also other production data are very useful for their daily work in, e.g. decision making. Large amounts of data is stored in the OptDB and can be viewed from the OPTIMISE browser. Production personnel might for example use forecasted inactive periods for planned maintenance using the Gantt chart. To view utilization of machines and stock levels over time makes it easier for the production planner to see which machines that are possible bottlenecks for the period and which variants that has a critical stock level and need to be followed up during the day.

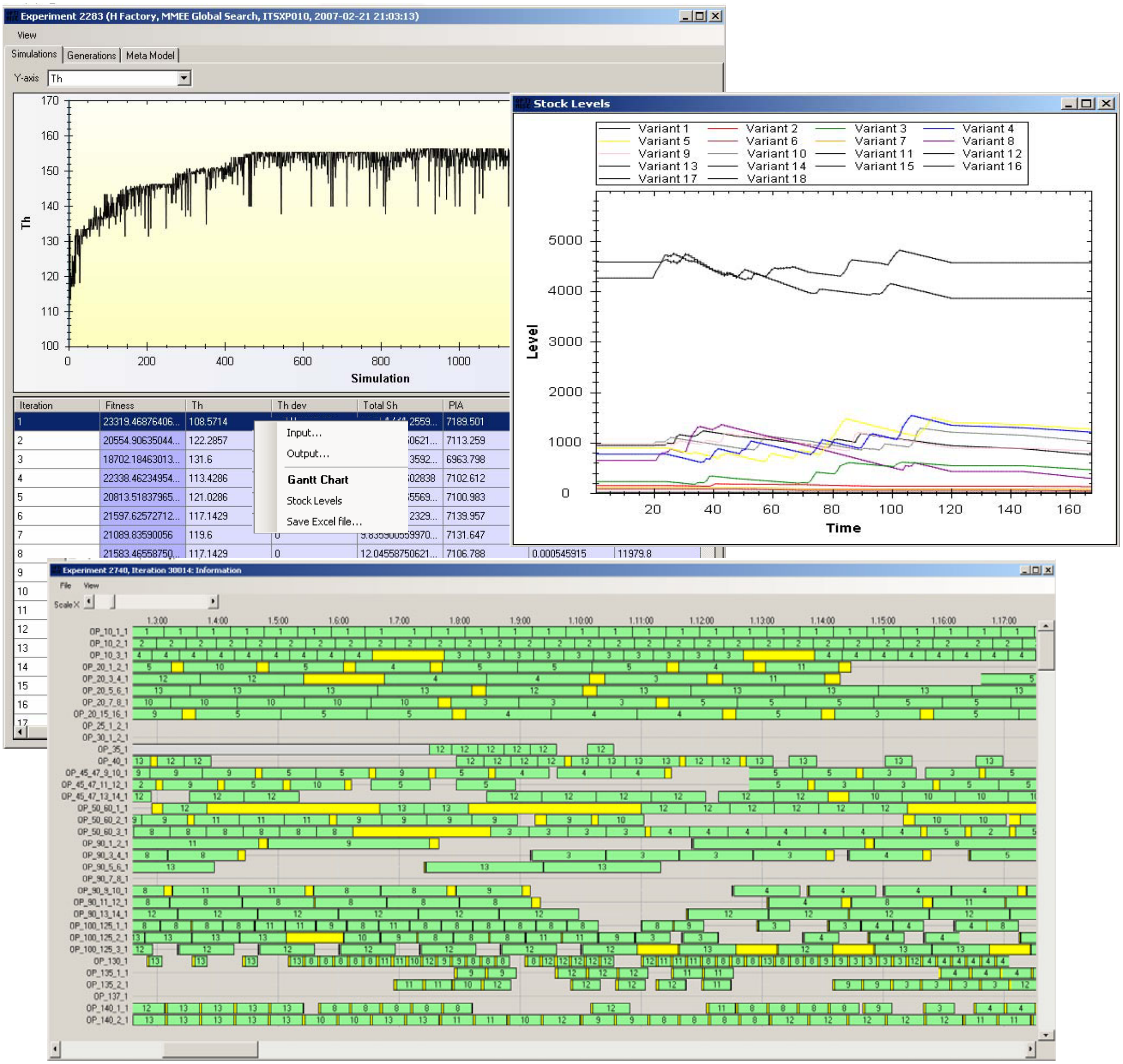


Figure 3: Screenshot of output data in OPTIMISE Browser

5.1.1 Optimization Progress

Some of the output data, stored in the database, are more important in a scheduling point of view for the experts of the camshaft machining line, and are therefore presented in graphs. It enables the users to be able to observe the progress of the SO process. Mean values and standard deviation are presented in graphs of fitness, throughput, shortage, lead time, and WIP, but if necessary it is possible to present more output data in graphs, as the data is stored in the database. The graph shown at right upper corner in Figure 3 is a screenshot from the OPTIMISE browser displaying the SO process of the throughput value of a scheduling scenario. By comparing the graph of the fitness value with the graphs of throughput and shortage it is possible to see how the optimization is carried out. At first the optimization has focus on producing the right variants at the right time by decreasing the shortage, because a large shortage penalizes the fitness function much, and then the focus is on increasing the throughput.

5.1.2 Gantt Charts

For each evaluation, a Gantt chart can be generated in order to study the status of machines (processing, setup, failure, shift out, planned stop, and nothing to produce) and jobs allocated to them in the scheduling period. Here the user can see how a schedule affects each machine over time. This will bring additional information to the operators at the machining line that they did not have before. The production planner can see how a schedule affects the line, the shift leaders and production engineers can plan the number of operators needed, plan for extra work such as testing of new variants, plan maintenance of machines, and plan for other events such as meetings.

The Gantt chart at the bottom in Figure 3 shows the schedule from one evaluation in the simulation model. In order to make it useful for experts of different domains it is possible to configure how the Gantt chart has to be displayed. The zoom function allows the user to watch a period of interest. It is possible to show or hide status bars to make certain activities clearer. The user may also select which machines that should be displayed or not, in order to customize the Gantt chart for individual needs. Information is attached to activities showing which variant number and batch number an activity has. Different batches in the Gantt chart and variant number information listed below make chart clearer to see. The Gantt chart bars can be colored according to user specifications, where in default the colors are based on activity types. For example, the user can specify the color according to batch number in order to be able to easily follow a batch throughout its operation steps. To make it even clearer it is possible to choose to show or hide each batch in the Gantt chart.

5.1.3 Stock Levels

Due to the importance of handling demand fluctuations and unexpected events, such as machine break downs and product defect in quality, it is important to keep track of the security stock levels of different product variants over time. Therefore, the FGI levels are logged over time during each evaluation, see Figure 3. This enables the production personnel to see which variants that has a critical stock level and need to be followed up during the day.

5.1.4 Export Input and Output Data

If necessary, it is easy to export input and output data from the OptDB through OPTIMISE browser to excel, which enables the user to analyze and plot customized graphs in excel. If the user wants to visualize a schedule other than just using the Gantt chart it is easy to export the schedule to the input data GUI and launch a simulation run manually. Through the animation capability of the simulation software, the user can check the dynamic product flow (e.g. blocking, buffer levels) by running the simulation model with the predetermined schedule.

5.2 User Scenarios

To run an optimization, the user first defines the scheduling scenario, starts the optimization and then receives the results through OPTIMISE browser. To define the scheduling scenario certain information is needed. At each new scheduling process the user sets the simulation-based scheduling period to one or two weeks, in order to have overlapping periods (see Figure 4) due to changes to the status of the line and demand changes.

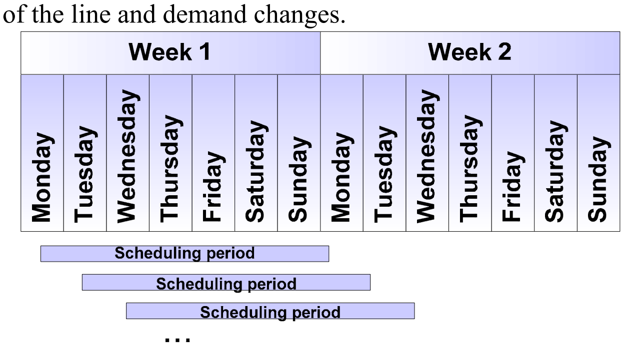


Figure 4: Overlapping scheduling periods

The overlapping periods is purposed to avoid the risk of generating sub-optimal schedule just for the first day/days of the scheduling period. At each scheduling process the user defines the scheduling scenario by collecting information from different systems (see Figure 5). The production planner needs to collect the information listed below:

- Scheduling scenario information: the user sets the scheduling period length, calendar information such as shifts and planned maintenance, simulation replications, and fitness function used into the input data GUI.
- Production line status: the user checks the status of machines, counts WIP of different variants and put this data into the input data GUI.
- Information system: the user checks the current number of variants in FGI, and collect demands and put this information into the input data GUI.

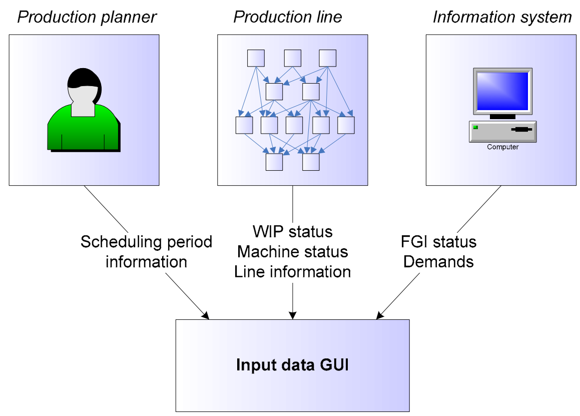


Figure 5: Input data collection for a scheduling period

Depending on the status of the line, the user may either gather WIP information automatically from the simulation model or collect the information directly from the shop floor control system, if possible. The information gathered for the scheduling scenario is used for creating a new schedule of which the optimization is based on. It is possible to do this manually or automatically by using a built-in macro in the input data GUI. The macro allocates WIP and the start of new batches according to demand over time.

When the user starts the optimization the input data file is sent to the optimization engine at the server site via File Transfer Protocol (FTP). Microsoft Message Queuing (MSMQ) is used for the communications among various OPTIMSE Server Components. At the client site the user can watch the progress of the optimization using OPTIMISE Browser. When the SO is finished the user can use the best operation schedule for the real production line.

6 RESULTS

Several SO scenarios have been carried out and so far the results look very promising. Evaluation of results shows that the application is successful in finding good solutions according to the fitness function. When compared to a manual approach, once the optimization process has started there is no user intervention needed. Furthermore, it enables many different users to easily access the results as these will affect different parts of the line differently. The SO in this case study does not guarantee that the optimal schedule is found, but on the other hand this is usually not needed in industrial practice in which it is more important in finding a good and reliable schedule. Engineers and other experts of the system see great potential in the application as it not only gives them a good schedule, but also give them complementary information required to run their production system more effectively.

A SO was carried out for a scenario of five days of production representing a difficult scheduling problem that could be encountered by the shop floor. Figure 6 shows the moving average of the fitness value and throughput per hour with a window size of 50 for 2000 evaluations. In this scenario the fitness value is rapidly improved in the first 1000 evaluations and after approximately 2000 evaluations the fitness value has been improved by more than 80% compared to the first population. In the beginning of the optimization the most important objective is to decrease shortage of some variants due to their tight deadlines, but after a while throughput gains more and more importance. The throughput is increased by approximately 18 % after 2000 evaluations compared to the first population. Total computing time for the SO process was significantly reduced by running simulations in parallel utilizing the Web-based parallel and distributed computing platform OPTIMISE.

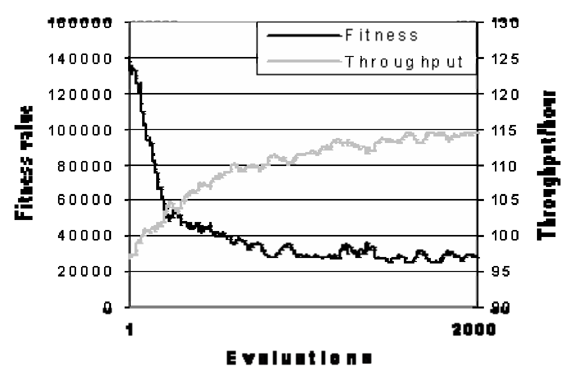


Figure 6: Moving average of fitness value and throughput

7 CONCLUSIONS AND FUTURE WORK

This paper presents a successful implementation of a Web-based simulation optimization system for industrial scheduling of a machining line in an automotive factory in Sweden. A DES model was built in order to represent the complexity of the existing line and to provide the user with valuable information and to be able to answer scheduling related questions. The model was incorporated into an SO application framework, which makes use of parallel and distributed simulation, through a Web services interface, to speed up the optimization process. The optimization algorithm developed is based on a Genetic Algorithm (GA). Initial results have shown that the application is successful in finding good solutions according to the fitness function, within a limited computation time. Future work is on efficiency enhancement, to further develop the GA implementation, investigate the possibility of a surrogate model, etc. Another important task to be investigated is robustness concerning both the fitness function and the GA implementation.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Knowledge Foundation (KK Stiftelsen), Sweden, for the provision of research funding and Volvo Cars Engine for their collaborative inputs to this case study.

REFERENCES

Azzaro-Pantel, C., L. Bernal-Haro, P. Baudet, S. Domenech, and L. Pibouleau. 1998. A two-stage methodology for short-term batch plant scheduling: discrete-event simulation and genetic algorithm. *Journal of Computers and Chemical Engineering* 22(10):1461-1481.

Baesler, F. F., and J. A. Sepúlveda. 2001. Multi-Objective Simulation Optimization for a Cancer Treatment Center. In *Proceedings of the 2001 Winter Simulation Conference,* ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 1405-1411. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Baker, K. R. 1974. *Introduction to Sequencing and Scheduling*. New York: John Wiley and Sons, Inc.

Eskandari, H., L. Rabelo, and M. Mollaghasemi. 2005. Multiobjective Simulation Optimization Using an Enhanced Genetic Algorithm. In *Proceedings of the 2005 Winter Simulation Conference*, 833-841. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kiran, A. S. 1998. Simulation and scheduling. In *Handbook of Simulation*, ed. J. Banks, 677-717. New York: John Wiley and Sons, Inc.

Koh, K-H., R. Souza and N-C. Ho 1996. Database driven simulation/simulation-based scheduling of a job-shop. *Simulation Practice and Theory* 4:31-45. Elsevier.

Ng, A., H. Grimm, T. Lezama, A. Persson, M. Andersson, and M. Jägstam. 2007. Web Services for Metamodel-Assisted Parallel Simulation Optimization. In *Proceedings of The IAENG International Conference on Internet Computing and Web Services (ICICWS'07)*, 879-885. Hong Kong.

Sanchez, S. M. 2000. Robust design: Seeking the best of all possible worlds. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 69–76. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

MARCUS ANDERSSON is a Ph.D. candidate at the University of Skövde, Sweden and De Montfort University, U.K. He received his BSc and MSc degrees in Simulation Engineering and Manufacturing Management from the University of Skövde and Loughborough University, UK, respectively. His research is within simulation-based optimization, simulation-based scheduling and manufacturing system analysis using discrete event simulation. His e-mail address is <marcus.andersson@his.se>.

HENRIK GRIMM is a Systems Developer at the University of Skövde, Sweden. He received his BSc and MSc degrees in Computer Science from University of Skövde. His research interests include computer simulation, artificial intelligence, and distributed systems. His e-mail address is <henrik.grimm@his.se>.

ANNA PERSSON is a Ph.D. candidate at University of Skövde, Sweden and De Montfort University, U.K. She holds a Master's degree in Computer Science from University of Skövde. Her research interests include artificial intelligence, simulation-based optimization, and efficiency enhancement techniques for simulation-based optimization. Her e-mail address is <anna.persson@his.se>.

AMOS H. C. NG is a Senior Lecturer at the University of Skövde, Sweden. He holds a B.Eng. degree and a M.Phil. degree, both in Manufacturing Engineering from the City University of Hong Kong and a Ph.D. degree in Computing Sciences and Engineering from De Montfort University, Leicester, U.K. He is a member of the IEE and a Chartered Engineer in the U.K. His research interests include agent-based machine control systems and simulation-based optimization. His e-mail address is <amos.ng@his.se>.