Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

AN INTERACTIVE DECISION SUPPORT SYSTEM USING SIMULATION-BASED OPTIMIZATION AND DATA MINING

Ingemar Karlsson
Amos H.C. Ng
Anna Syberfeldt
Sunith Bandaru

School of Engineering Science
University of Skövde
P.O. Box 408, SE-541 28 Skövde, SWEDEN

ABSTRACT

This paper describes a decision support system (DSS) built on knowledge extraction using simulation-based optimization and data mining. The paper starts with a requirements analysis based on a survey conducted with a number of industrial companies about their practices of using simulations for decision support. Based upon the analysis, a new, interactive DSS that can fulfill the industrial requirements, is proposed. The design of the cloud-based system architecture of the DSS is then described. To show the functionality and potential of the proposed DSS, an application study has been performed for the optimal design of a hypothetical but realistic flexible production cell. How important knowledge with respect to different preferences of the decision maker can be generated as rules, using the new Flexible Pattern Mining algorithm provided in the DSS, will be revealed by the results of this application study.

1 INTRODUCTION

Manufacturing processes often involve several complex machines and procedures which together form a complex system. Such systems can be difficult to analyze and understand without any aid. Simulation has been used to improve and understand manufacturing processes in several different areas, for example: workforce planning, supply chain management and capacity planning (Jahangirian et al. 2010). Simulation studies can help a company to answer what-if questions, but in order to know which parameters are best suited for the manufacturing system, an optimization technique has to be applied. When combined with optimization technologies, simulation has a potential to generate data as a basis for decisions made by decision makers (Pehrsson, Ng, and Stockton 2013).

The combination of simulation and optimization is called simulation-based optimization where an optimization algorithm uses a simulation model to iteratively evaluate solutions to search for the best solution for the problem (April et al. 2003). Simulation-based optimization can be used in several types of applications, for example manufacturing (Amaran et al. 2014). This method can generate a lot of data, which together with other data the company has for their manufacturing process, contains a lot of information that can be used to improve the process. Because of the large amount of data and the complex nature of manufacturing systems, it is difficult to manually unravel the information. Through data mining, this can be done in an automatic manner to help the user in retrieving useful information. Data mining is the process of automatically extracting knowledge hidden in large data sets with the help of computers (Fayyad, Piatetsky-Shapiro, and Smyth 1996).

Data mining has been used to analyze several areas in manufacturing (Choudhary, Harding, and Tiwari 2008) and its addition to simulation-based optimization has proven to be powerful in generating valuable

978-1-4673-9743-8/15/$31.00 ©2015 IEEE

information that can be used to make decisions (Dudas et al. 2014). In order to make full use of the combination of simulation-based optimization and data mining they can be integrated into a DSS. The combination of the two methods can also be automated into a streamlined process so that manually transferring files between separate software packages can be avoided. Because of the potential of such a DSS, manufacturing companies show interest but their requirements on the DSS are not evident.

This paper investigates the requirements that manufacturing companies have in regard to a DSS that uses simulation-based optimization and data mining methods. It also introduces the implementation and evaluation of a novel DSS designed to help decision makers in making decisions confidently, based on the aforementioned requirements. The paper is structured as follows: Section 2 discusses the requirements of a DSS. Section 3 describes an architecture for the DSS. An application study is presented in Section 4 and the paper is concluded in Section 5.

2 REQUIREMENTS OF A DECISION SUPPORT SYSTEM

In this section a survey that was sent to several organizations to gain more knowledge about requirements for a DSS is described and analyzed. The section also discusses and summarizes the requirements.

2.1 Survey

To understand how simulation engineers use simulation and simulation-based optimization and how they elicit knowledge from the data generated by these methods, a survey was sent to a group of companies and organizations. The survey asks what general requirements there are for a DSS using simulation-based optimization and data mining. The selected companies and organizations are research partners of the University of Skövde and are known to be active users of simulation. They varied in size, from large manufacturing companies running worldwide businesses to small and medium enterprises. The survey was sent to 21 simulation engineers working in 8 different manufacturing companies and organizations that utilize simulation models in their work. Of the 21 who received the survey, 15 answered from 7 of the companies and organizations.

For the survey 19 questions were constructed, divided into two sections. The first section asks the respondents about their experiences and how they currently work with simulation and simulation-based optimization, as well as their experiences in eliciting knowledge from the data generated by simulation models. In the second section the respondents are asked about how they want to be able to work in the future, what features of a DSS they are missing and so on. The survey (in Swedish) is available on the web at http://research.i5r.se.

2.2 Requirement Analysis

The survey shows that the companies solve a multitude of problems using different simulation software. All the respondents answered that they use simulation for improving their processes, whereas the second most common answer was to use simulation results as decision basis for new investments. Other areas there the respondents use simulation models are production planning, personnel scheduling, supply chains, competence improvement and energy consumption. The majority of the respondents answered that they use simulation-based optimization together with their simulation models.

The respondents mention that experiment design is an important function in simulation-based optimization. With a proper experiment design, the simulation-based optimization can be initialized with promising solutions either by the user or through a design of experiments (DOE) method. Population-based metaheuristic optimization algorithms work better if they can start with a diversified set of initial solutions which leads to more effective optimizations (Talbi 2009). Online user interaction with simulation-based optimization where the user can guide the search process to interesting areas can also lead to faster convergence. The majority of the respondents answered that they are interested in being able to choose the degree to which they interact with the system.

Because the respondents want different degrees of interaction with the DSS, it has to provide a way for the user to create a workflow that tells the software how and when it should start the optimization and data mining processes. The system needs to have a graphical user interface (GUI) where the decision maker can create their workflow and to provide the settings that are needed for the components in the workflow.

Another important subject for the respondents is cooperation by working on the same data, like simulation models and optimization results. All the respondents claim to share data with their co-workers. Simulation results and simulation models are regularly shared. Those who use simulation-based optimization share optimization settings, results from the optimization and complete analysis based on output data. Documents are also shared among co-workers in the same project like requirement specifications, project specifications and project goals.

The survey also shows that the respondents use several methods for analyzing the optimization results to discover knowledge. The majority of the respondents use manual methods by studying the raw data together with the aid of data plots, like scatter and parallel coordinate plots (PCP), but they clearly indicate their desire for some automatic knowledge extraction functionality.

According to the answers to the survey the simulation models, the respondents work with contain information sensitive by the company. All respondents answered that their simulation models and data, always or sometimes, contain sensitive information. This is an important aspect about the system that it must be secure both from external and internal unauthorized access.

Most of the respondents run their simulations and optimizations on their own workstations. Because of their computationally intensive nature, these processes can make the process very slow. By using dedicated virtual computers, e.g. using cloud computing for these processes, the user can free resources for doing other things on the workstation. If the simulation evaluations are performed in parallel on several computers, the optimization process can run much faster. Utilizing several computers requires investments and a support organization and other resources, like space and electricity.

Based on the answers to the survey a list of requirements for a DSS is compiled. The requirements are summarized in Table 1 with a motivation as to why they are needed.

2.3 Comparison with Existing Decision Support Systems

To see how earlier work supports simulation-based optimization and data mining, a careful literature review of papers published in the last five years reveals four related studies describing similar solutions. Rossetti and Chen (2012) have developed an architecture that uses cloud computing techniques to run a supply chain network simulation. The authors mention that the simulations take a long time to run and by running them in parallel in a cloud environment, it is possible to get the result from the simulations faster when compared to running them on a single workstation. Melouk et al. (2013) present a decision support tool using simulation-based optimization to optimize a steel production process. By utilizing a simulation model and a standalone optimization package the authors show that by using simulation-based optimization they can reduce the cost of the steel production process they used in their case study. Another DSS based on a cloud computing environment is proposed by Taylor et al. (2014) who aim to put simulation software in the cloud. Through scalability, cloud computing offers the possibility of using the web-based version of the authors' simulation software by any users. Ng et al. (2011) also present an internet-based DSS where a simulation software can evaluate solutions both in a distributed and parallel manner. This increases the precision and speed of the DSS. In summary, none of the DSSs discussed fulfill all the requirements that have been previously identified. A comparison is summarized in Table 2.

Besides the DSSs presented in the literature, there are also generic software capable of running simulation-based optimization that can integrate with different simulation software. Examples of such software are modeFRONTIER® from Esteco and HEEDS® MDO developed by Red Cedar Technology. In these software the users can analyze the results from simulation-based optimization with the help of visualization. For an automated analysis the user has to export the data and use other software capable of such automated analysis.

Table 1: Summary of requirements for a new DSS.

Requirement	Motivation
Support for multiple simulation softwares	Problems may require different simulation software
Support for multiple optimization algorithms	There exists no optimization algorithm that is best for all types of problems
Experiment design	Initial solutions to optimization algorithm
	Ability to start from earlier optimizations
Automated decision support process	Easier use of the decision support system
	Less monitoring
Support for sharing data to promote cooperation between users	Faster and more efficient decisions
Accessible from company network with a client/server architecture	Multiple users can work on the same data
	Users can do other work on their workstations while doing optimizations
Secure storage and transfer of data	Simulation models and data derived from them can contain industrial secrets
Manual knowledge extraction methods	Simulation engineers can manually extract data
Automatic knowledge extraction methods	The simulation engineers wants to be supported in the process of extracting knowledge from the results of simulation-based optimization
	Can be used to automatically limit the optimization problems
Running simulation models in parallel	Faster and more efficient decisions because of faster optimizations
Ability to run in a cloud environment	More efficient use of resources
	No need for own hardware

Table 2: Supported requirements in earlier studies. ● = supported, – = not supported

Requirement	Rossetti (2012)	Melouk (2013)	Taylor (2014)	Ng (2011)
Support for multiple simulation software	–	–	●	●
Support for multiple optimization algorithms	–	–	–	●
Experiment design	–	–	–	–
Automated decision support process	–	–	–	–
Support for sharing data to promote cooperation between users	●	–	–	●
Accessible from company network with a client/server architecture	●	–	●	●
Secure storage and transfer of data	–	–	–	–
Manual knowledge extraction methods	●	●	●	●
Automatic knowledge extraction methods	–	–	–	–
Running simulation models in parallel	●	–	–	●
Ability to run in a cloud environment	●	–	●	–

3 DECISION SUPPORT SYSTEM

This section describes a system architecture for a DSS that is based on the requirements identified earlier in the paper.

3.1 System Architecture

A system architecture for a DSS using simulation-based optimization and data mining has been developed to fulfill the requirements. The major components of the DSS are illustrated in Figure 1. The DSS architecture is based on a client/server design. The server consists of a group of components that together supply the major computational functionality for the DSS. In the client application, the users can create and manage the workflows using the workflow designer graphical user interface (GUI) as shown in Figure 2.

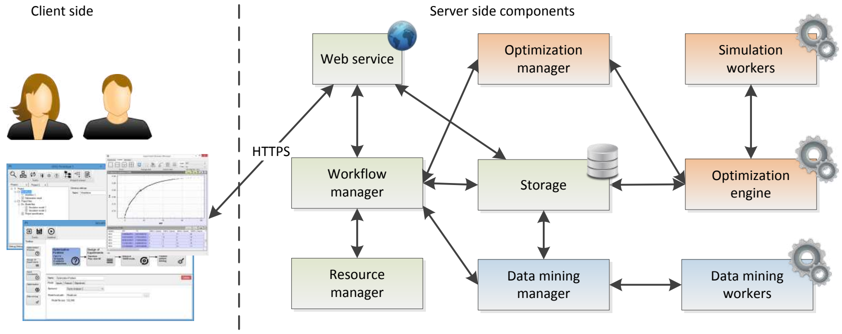
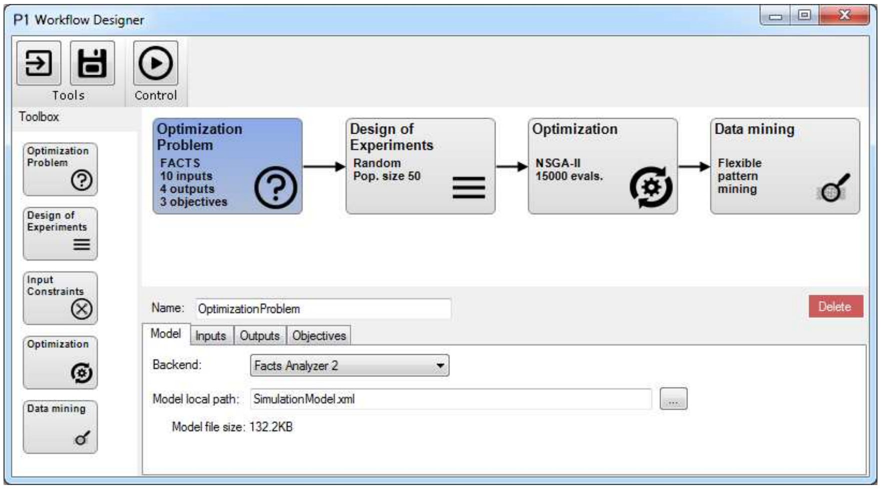


Figure 1: Architecture of the decision support platform.



Name: OptimizationProblem

Backend: Facts Analyzer 2

Model local path: SimulationModel.xml

Model file size: 132.2KB

Figure 2: Screenshot of a prototype workflow GUI.

The GUI lets the user use the processes and build the workflow to extract knowledge from a simulation model. A workflow is stored in a structure called project along with the results from the optimizations and data mining processes. In the project the users can also store files like simulation models, project plan and so on. The projects are managed in the GUI and can be shared between users depending on their permissions on the project. The optimization and data mining processes are run in the server.

The server components are divided into three groups: core, optimization and data mining components. The core components are: web service, workflow controller, resource manager and storage which constitute the core functionalities. The web service which acts as an interface for the client applications can be exposed to a company network or directly to the internet. This makes the DSS accessible from more than one computer and a user can sign in on any computer that has the client application installed. The resource manager component of the DSS keeps track of the resources needed by the optimization and data mining processes. In a local environment this component is not used, but when installed in a cloud environment the resource manager spawns virtual computers when needed to execute parallel and distributed simulation runs and data mining processes. The last one of the core components is the storage component that stores the data that the user uploads or creates with the DSS. This component consists of a SQL database and a file storage sub-component where the SQL database stores the project information, optimization problems, workflows, the optimization results and so on. The file storage stores the simulation models as well as other files that the user uploads. The files are accessible to all the users who have permissions to the files through the client application to support the requirements on collaborative project work.

The three optimization components provide the decision-support system with simulation-based optimization functionality. The optimization manager handles the optimization engines which are the second component in this group. The optimization uses the third component, simulation worker, to evaluate solutions using the simulation model. To support multiple simulation software, the simulation worker has an interface to the simulation software that can be replaced by the DSS in runtime as needed.

The data mining process is handled by the data mining manager and data mining worker. The data mining manager keeps track of the data mining workers and allocates work to the data mining worker that is free. The data mining worker has the data mining algorithms and downloads the data needed from the storage component.

3.2 Implementation of Requirements

The evaluation of simulation models is the core of the DSS and the support for integration between different types of simulation software and the simulation worker component, makes the system generic. To integrate a new simulation software the only component that needs development is the interface between the simulation worker and the simulation software. With support for different simulation software the user can optimize different types of problems. Since there is no optimization algorithm that performs well for all types of problems (Wolpert and Macready 1997), the DSS needs to support a set of optimization algorithms suiting different problems.

To automate the analysis, a workflow concept is introduced. A workflow is designed by the user and the workflow manager in the DSS uses the workflow to start and set up the simulation-based optimization and data mining process accordingly. This makes the analysis automatic. In case the user's interaction is needed, the user is notified in the client application that a running workflow needs supervision to be able to continue. An example of such interaction is the user-input to data mining algorithms.

Multiple data mining algorithms are supported by the DSS that has an interface in the data mining worker component which makes it possible to integrate several algorithms in the system. This gives the DSS the possibility to automatically generate knowledge obtained through data from the simulation-based optimization. Manual knowledge extraction methods are supplied to the user through the experiment analysis GUI in the client application.

During and after the analysis the users need to share data with their co-workers. This requires that other users can access the project from their workstations, based on the client-server design. By running the processes on a server, multiple users can access the core DSS components simultaneously as well as access other users' data. The protection of the data in the DSS is an important aspect identified by the requirements analysis. The data transferred to and from the client application use an encrypted version of the HyperText Transfer Protocol (HTTP). Through encrypted data transfer between the server and the client application, eavesdropping is made harder.

By running the simulation worker component in parallel on multiple servers, the DSS has the ability to evaluate more than one evaluation at a time which drastically reduces runtime. The DSS can run on ordinary servers or virtually in a cloud-computing environment. Running the DSS in a cloud environment has several advantages, most importantly, no physical servers are needed for the company and the company only needs to pay for the resources needed for the moment. To access the scalable and elastic properties of a cloud computing environment, the DSS is designed to contain the resource manager component. This component makes sure that the optimization and data mining processes have the required resources within the limits set by the company.

4 APPLICATION STUDY

An application study has been conducted to give an example of how the DSS can be used. This section describes the application study and how the DSS supports it.

4.1 Simulation Model and Optimization Problem

The application study is based on a machining cell model for studying the use of multi-objective optimization and data mining for production systems design. The model is an extended version of the one presented in Ng et al. (2011). Although the model is a hypothetical one, it is realistically designed based on our experiences in working with real-world complex models and decision making situations in industry. This model makes use of three so-called Selection objects in the simulation software FACTS Analyzer (or FACTS hereafter) (Ng et al. 2007) that can enable several design alternatives to co-exist in a model. By changing the values of the corresponding decision variables, the optimization module in FACTS can then switch the selection objects into various combinations. The analysis of such a model could then be made in an efficient way by letting the optimization algorithm to seek the optimal combination of design alternatives with their optimal settings together with the optimal values of other decision variables (e.g. buffer spaces), in a single optimization run.

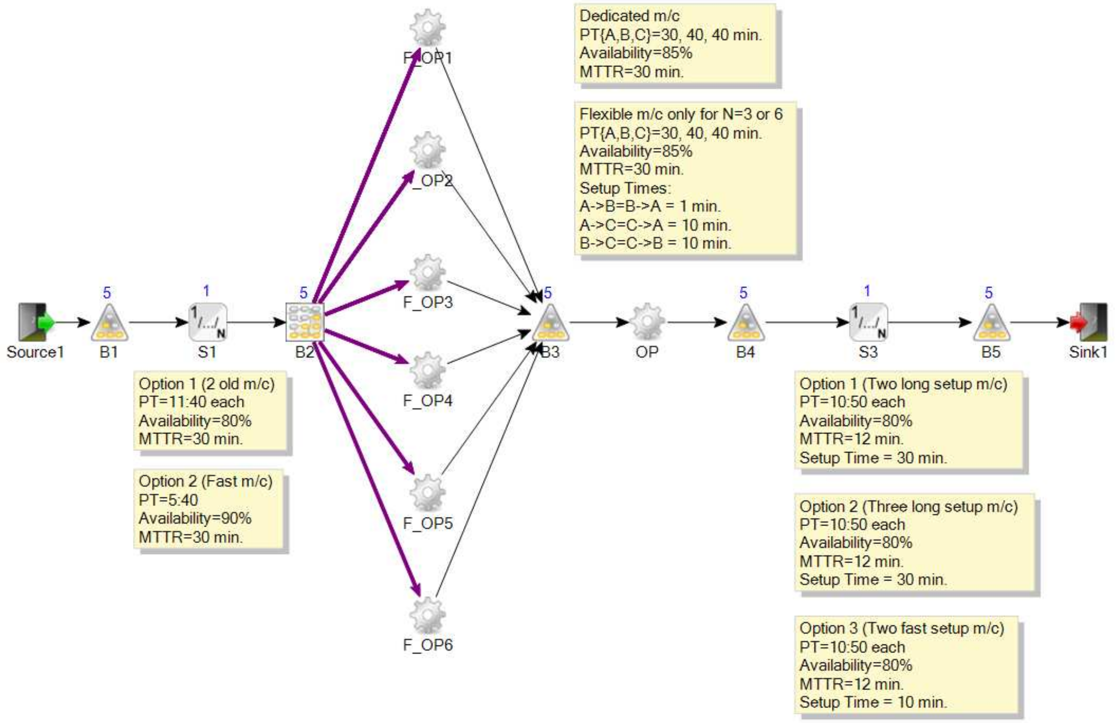


Figure 3: Simulation model depicting the made-up production system.

Table 3: Decision variables, their ranges and associated cost for calculating Payoff time.

Decision variable	Range	Investment cost ($)
B_i, $i = 1,...,5$	$\{1,2,3,\ldots,50\}$	1000 / buffer
S_1	$\{1,2\}$	500000 if $S_1 = 1$ (two old m/c), else 900000
N_2	$\{1,2,3,4,5,6\}$	Flexible m/c 500000 each; dedicated 250000
M_3	$\{1,2\}$	Flexible if $M_3 = 1$, else dedicated
M_6	$\{1,2\}$	Flexible if $M_6 = 1$, else dedicated
S_3	$\{1,2,3\}$	{100000, 150000, 120000}

Figure 3 illustrates how alternative designs can be easily built in FACTS with two selection objects (S1 and S3) and an additional flow selection controlled by two variables M_3 and M_6 (explained below). S_1 constitutes the choice of having one fast machine instead of two slow machines in parallel, which is a common decision-making problem faced in industry when considering the replacement of old, slow machines, with a new, fast machine. S_3 is a selection among three options: (1) two machines with longer setup times; (2) three machines with longer setup times; (3) two machines with shorter setup times. When compared to the model presented in Ng et al. (2011), the current model is more complex as there are two decisions that have to be made for the operations at the second stage, between Store2 and Buffer3, namely, (1) number of machines (N_2) and (2) if these machines are flexible machining centers that can process multiple part types or they are dedicated machines that are capable of processing only one specific part type. Since this model involves three product variants (i.e. A, B and C), to have their own dedicated machine means that this can only happen when the number of machines in question is either 3 or 6. In this case, M_3 and M_6 are the two variables that represent if dedicated (non-flexible) machines are used (in this case their value = 2) when $N_2 = 3$ and 6, respectively. In other words, flexible machines, which are double in investment costs than dedicated machines (see Table 3), will be set for all other combinations of N_2, M_3 and M_6. For example, a candidate solution being found in the optimization, with $N_2 = 6$ and $M_6 = 1$, represents an option of using 6 flexible machines. With the flexible machines, the three variants are allowed to be flexibly processed in any of the machines but with the cost in increasing setup times during variant switch, as well as higher investment cost. Therefore, apart from the decision on whether to replace the old machine in the first stage, the decision maker also needs to evaluate whether it is cost-effective to invest in flexible machines in the second stage.

Given that payoff time (year) can be calculated as total investment / profit per year, this decision-making problem can be easily formulated as a single objective optimization problem by minimizing the payoff. Assuming the profit by selling each product = $20 and there are 52 weeks' production per year. If P_w is the weekly production, then the payoff time is calculated as shown in (1).

$$Payoff = \frac{\text{Total investment cost}}{P_w \times 20 \times 52} \tag{1}$$

With an aim to demonstrate how multi-criteria decision support can be done efficiently using multi-objective optimization and data mining, the problem has been solved as a three-objective optimization problem: $f_1(\mathbf{x}) = \min(Payoff(\mathbf{x}))$, $f_2(\mathbf{x}) = \max(TH(\mathbf{x}))$ and $f_3(\mathbf{x}) = \min(WIP(\mathbf{x}))$, where $\mathbf{x} = [B_1,B_2,B_3,B_4,B_5,S_1,N_2,M_3,M_6,S_3]$ represents a solution vector in the decision space, $TH =$ system throughput and $WIP =$ Work-In-Process of the production line. It can be easily seen that while TH and WIP are simulation outputs evaluated through simulation runs, $Payoff$ is a variable by mixing other input and output variables. Table 3 gives the ranges of the decision variables and their associated costs to show how the total investment is calculated.

With the detailed data provided in Figure 3 and Table 3, other researchers can easily verify and compare with the optimization and data mining results presented in this paper. As will be shown in the next section, it can be seen that such multi-objective optimization study can provide the knowledge that Payoff year, Throughput and Work-In-Process are three conflicting objectives. The post-optimality analysis through data mining will then reveal properties of the efficient solutions that constitute the different preference regions indicated by the decision maker using the DSS.

Flexible Pattern Mining (FPM) was used as the data-mining algorithm in this study. FPM is an extension of the popular sequential pattern mining (SPM) algorithm (Agrawal and Srikant 1995). SPM was originally developed for extracting significant buying patterns (also called frequent item-sets) from market basket data (customer transactions). FPM extends the concept of market basket data to optimization data by considering each solution as a customer and each variable as a transaction containing individual items (variable values). While SPM looks for a sequence of items from any of the transactions, FPM enforces that these items (variable values) come from the same transaction columns (variables). Thus, significant patterns of discrete variable values can be found. FPM can also generate rules instead of patterns by transforming the optimization data into a truth matrix containing columns corresponding to logical operators $<$, $=$ and $>$ applied over all available discrete choices of all variables.

4.2 An Application Study of the DSS

A workflow has been created to run the optimization problem in the DSS. The workflow, which is shown in Figure 2, consists of four nodes, optimization problem, design of experiments (DOE), optimization and data mining. The first three nodes are fully automatic when they are configured in the client application but the chosen data mining process, FPM, requires a set of solutions to be selected to be able to extract knowledge.

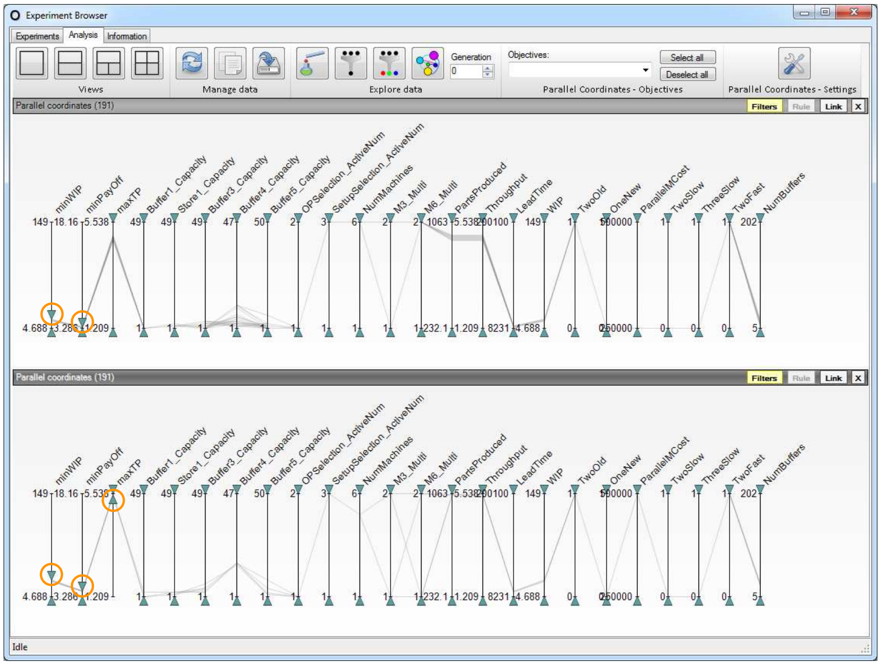
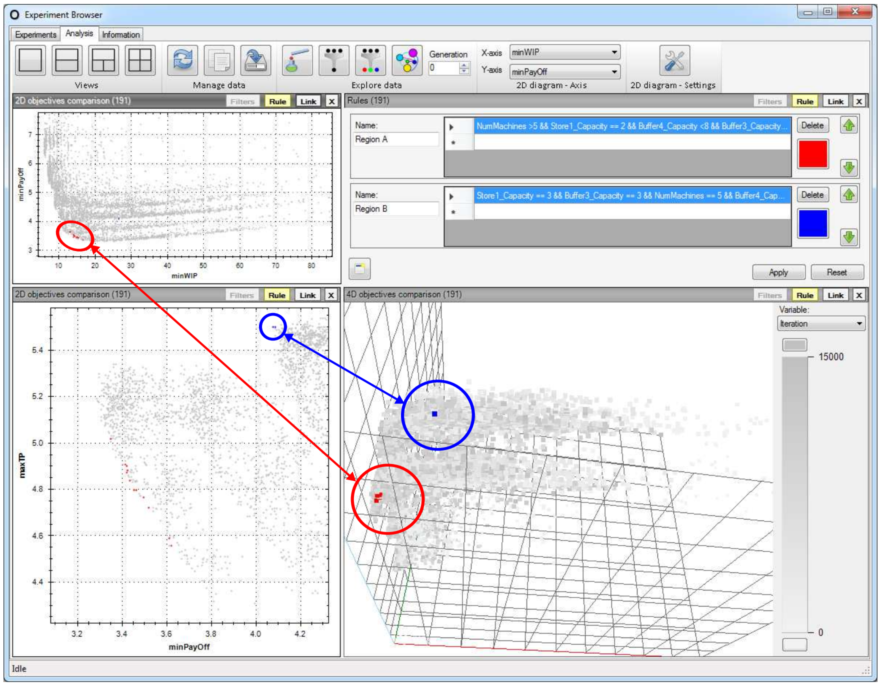


Figure 4: Selection of evaluations for data mining.

The interactive feature of the DSS can be illustrated through this example as the decision maker may choose between two preferences which from the results of the optimization have been shown to be impossible to satisfy at the same time: (A) lowest possible payoff period or (B) highest possible TH. From the data plots, e.g. the PCPs as shown in Figure 4, it is clear that the solutions which give the highest TH do not give the shortest $Payoff$. Here a trade-off situation has to be resolved by the decision maker, probably based on the forecasted production demand. In order to acquire deeper knowledge about what configurations lead to the two preferred regions of interest (i.e. A or B), a data mining algorithm is called



Experiment Browser

Generation 0

X-axis minWIP

Y-axis minPayOff

Rules (191)

Name: Region A

NumMachines >5 && Store1_Capacity == 2 && Buffer4_Capacity <8 && Buffer3_Capacity...

Name: Region B

Store1_Capacity == 3 && Buffer3_Capacity == 3 && NumMachines == 5 && Buffer4_Cap...

Variable: Iteration

Idle

Figure 5: Screenshot on client application experiment analysis GUI.

by identifying the preferable solutions. As shown in Figure 4, the DSS allows the user to do this by filtering away the non-preferred solutions by adjusting the limit bars in the PCPs. In case A (upper plot), only solutions with lowest $Payoff$ and WIP are selected while solutions with highest TH and lowest possible $Payoff$ and WIP are selected in case B (lower plot).

The shape of the Pareto-optimal front clearly shows that the three objectives are in conflict with each other. Particularly interesting is that the multi-objective optimization can locate a point in the objective space that shows the WIP value when the $Payoff$ is minimized (see the red circle in Figure 5).

Table 4: The rules extracted by using FPM in the two preferred regions.

Preferred region	Rule acquired by FPM
A: Lowest Payoff time	$S_1 = 1 \wedge S_3 = 3 \wedge N_2 = 6 \wedge M_6 = 2 \wedge B_2 = 2 \wedge B_3 < 3 \wedge B_4 < 8 \wedge B_5 < 3$
B: Highest TH with lowest possible Payoff and WIP	$S_1 = 1 \wedge S_3 = 3 \wedge N_2 = 5 \wedge B_1 = 1 \wedge B_2 = 3 \wedge B_3 = 3 \wedge B_4 < 16 \wedge B_5 > 5$

The data mining results reveal some very interesting and important knowledge in the form of rules generated by FPM based on the preferences of the decision maker as shown in Table 4. If this study were for a real-world decision-making scenario:

- Irrespective of whether lowest $Payoff$ or highest TH is the target, two parallel machines in stage 1, represented by $S_1 = 1$ and two fast setup machines in stage 3 ($S_3 = 3$) are preferred to any other combinations.
- Interestingly, 5 flexible machines ($N_2 = 5$) give higher TH than 6 dedicated machines (as indicated by $N_2 = 6 \wedge M_6 = 2$), but less economical (i.e. longer $Payoff$ period). On the other hand, 6 flexible machines do not give higher TH, only higher investment cost, so that no efficient solution possesses the attribute $N_2 = 6 \wedge M_6 = 1$.
- Apart from the difference between 6 dedicated machines and 5 flexible machines at processing stage 2, the difference between lowest $Payoff$ and highest TH lies on the required buffer spaces.

In this case, the rules generated by FPM have given the decision maker some specific details: it is B_4 and B_5 that matter and should be increased if higher TH is the preferred target; the size of B_1, B_2 and B_3 should be kept small in all cases to keep lower WIP and investment cost.

5 CONCLUSIONS

This paper describes how simulation-based optimization and data mining can be integrated in a DSS. To study how companies work with simulation and simulation-based optimization and how they extract knowledge using these methods, a survey was conducted at 8 companies and organizations. Their responses were used to identify requirements for a DSS that uses the aforementioned methods. A system architecture for such a DSS and how it fulfills the requirements is described.

To show how the proposed system architecture can help the user in the decision making, an application study was conducted which analyzed a simulation model depicting a hypothetical, but realistic production system. The flexible production cell consisted of machines and buffers, both in parallel and series which have alternative configurations with different costs. Simulation-based optimization was used to minimize the payoff time, minimize work in progress and maximize throughput for the production system. To extract knowledge from the data, a data mining algorithm, in this case FPM, has been applied on the data generated from the optimization which revealed valuable knowledge about the system.

To the authors' knowledge there is no DSS that currently fulfills all the requirements given by the industry. The proposed DSS enables the users to use simulation-based optimization and data mining in an efficient way without the need for any combination of tools that only supports one of the methods. Since all data is stored in a project structure and since the DSS is accessible through the company network, the users can share data with their co-workers.

The next step is to validate the system on a real simulation model and evaluate how the DSS is used by a simulation engineer in a manufacturing company. In a company environment the protection of data is vital and this also applies to the simulation models and the data related to them. When running the DSS in a cloud environment provided by a third party, the data is leaving the company and therefore security is a very important aspect. Before a DSS can be fully implemented, the security issues and their solutions should be investigated more thoroughly. The simulation-based optimization and data mining processes may generate a lot of data. Retrieving this data can take a considerable amount of time which is undesirable. How the data should be stored and transferred to the user in an efficient manner is also an area that has to be further investigated.

REFERENCES

Agrawal, R., and R. Srikant. 1995. "Mining Sequential Patterns". In *Proceedings of the Eleventh International Conference on Data Engineering, 1995*, 3–14.

Amaran, S., N. V. Sahinidis, B. Sharda, and S. J. Bury. 2014, November. "Simulation Optimization: A Review of Algorithms and Applications". *4OR* 12 (4): 301–333.

April, J., F. Glover, J. Kelly, and M. Laguna. 2003. "Practical Introduction to Simulation Optimization". In *Proceedings of the 2003 Winter Simulation Conference*, edited by S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, 71–78.

Choudhary, A. K., J. A. Harding, and M. K. Tiwari. 2008. "Data Mining in Manufacturing: A Review Based on the Kind of Knowledge". *Journal of Intelligent Manufacturing* 20 (5): 501–521.

Dudas, C., A. H. Ng, L. Pehrsson, and H. Boström. 2014. "Integration of Data Mining and Multi-Objective Optimisation for Decision Support in Production Systems Development". *International Journal of Computer Integrated Manufacturing* 27 (9): 824–839.

Fayyad, U., G. Piatetsky-Shapiro, and P. Smyth. 1996. "From Data Mining to Knowledge Discovery in Databases". *AI Magazine* 17 (3): 37.

Jahangirian, M., T. Eldabi, A. Naseer, L. K. Stergioulas, and T. Young. 2010. “Simulation in Manufacturing and Business: A Review”. *European Journal of Operational Research* 203 (1): 1–13.

Melouk, S. H., N. K. Freeman, D. Miller, and M. Dunning. 2013. “Simulation Optimization-based Decision Support Tool for Steel Manufacturing”. *International Journal of Production Economics* 141 (1): 269–276.

Ng, A., J. Bernedixen, M. Moris, and M. Jägstam. 2011. “Factory Flow Design and Analysis Using Internet-enabled Simulation-based Optimization and Automatic Model Generation”. In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 2176–2188.

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), 2007*, 879–885.

Pehrsson, L., A. H. C. Ng, and D. Stockton. 2013, December. “Industrial Cost Modelling and Multi-Objective Optimisation for Decision Support in Production Systems Development”. *Computers & Industrial Engineering* 66 (4): 1036–1048.

Rossetti, M. D., and Y. Chen. 2012. “A Cloud Computing Architecture for Supply Chain Network Simulation”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1 –12.

Talbi, E.-G. 2009. *Metaheuristics: From Design to Implementation*, Volume 74. John Wiley & Sons.

Taylor, S. J. E., T. Kiss, G. Terstyanszky, P. Kacsuk, and N. Fantini. 2014. “Cloud Computing for Simulation in Manufacturing and Engineering: Introducing the CloudSME Simulation Platform”. In *Proceedings of the 2014 Annual Simulation Symposium*, ANSS ’14, 12:1–12:8: Society for Computer Simulation International.

Wolpert, D., and W. Macready. 1997. “No Free Lunch Theorems for Optimization”. *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82.

AUTHOR BIOGRAPHIES

INGEMAR KARLSSON is a Ph.D. student at the University of Skövde. He has a B.Sc. in Computer Science and a M.Sc. in Automation Engineering from University of Skövde. His research interests include simulation-based optimization, data mining and cloud technologies. His email address is ingemar.karlsson@his.se.

AMOS H.C. NG is a Professor of Production and Automation Engineering at the University of Skövde, Sweden. He holds a Ph.D. degree in Computing Sciences and Engineering. His main research interest lies in applying multi-objective optimization for production systems design and analysis. His email address is amos.ng@his.se.

ANNA SYBERFELDT is a senior researcher at the University of Skövde, Sweden. She holds a Ph.D. in Computer Science from the De Montfort University, UK and a Master’s degree in Computer Science from the University of Skövde, Sweden. Her research interests include simulation-based optimization, artificial intelligence, metaheuristics, and advanced information technology with applications in logistics and manufacturing. Her email address is anna.syberfeldt@his.se.

SUNITH BANDARU is currently a postdoctoral researcher at University of Skövde, Sweden. He obtained his Ph.D. in the area of evolutionary optimization and machine learning from Indian Institute of Technology Kanpur, India. His research interests are knowledge discovery, data-mining and machine learning, multi-objective optimization, evolutionary algorithms, simulation-based optimization and metamodeling. His email address is sunith.bandaru@his.se.