
Development of an Interactive Simulation of Steel Cord Manufacturing for Industrial Engineering Education

Behlul Saka [1], Gurdal Ertek [2], Can Hulusi Turkseven [3]

[1] Department of Industrial Engineering
4207 Bell Engineering Center
University of Arkansas, Fayetteville, AR 72701, U.S.A.

[2] Sabanci University, Faculty of Engineering and Natural Sciences,
Orhanli, Tuzla, 34956, Istanbul, Turkey

[3] 17817 Coit Road, Apt 4307, Dallas, TX 75252, U.S.A.

Abstract

We developed an interactive simulation program to be used in industrial engineering education, based on an earlier simulation study of a steel cord manufacturing plant. In the class project, the students are asked to design strategies/algorithms for finding the optimal values of operational decision variables by using the program.

Submission areas: **Systems Modeling and Simulation, Manufacturing Systems Engineering, Production Planning and Control, Quality Control and Management**

Introduction

In this paper we present an interactive simulation software that can be used to introduce students to simulation and simulation optimization within the context of manufacturing. We extend an earlier study by Turkseven and Ertek (2003) that involves the design and development of a customized simulation program for modeling the bunching operations in a steel cord manufacturing plant for a particular product. They describe the development of a C++ program that evaluates the quality and productivity impacts of various operational policies, especially selection of spool sizes at each stage of the bunching operations. The tool that we developed takes their program as the simulation engine and allows users to interactively experiment with different scenarios through a GUI (Graphical User Interface). The tool has been used in an introductory industrial engineering course, within a project that required the students to come up with strategies/algorithms to find the optimal values of spool lengths.

Manufacturing Simulation and Optimization

Manufacturing systems have traditionally been the main focus of industrial engineers. The industrial engineer uses computer-aided mathematical methods including optimization, simulation, statistics, queueing theory, which are referred to as “Operations Research” techniques, to improve/optimize the performance of systems. Many successful applications of industrial engineering in manufacturing can be found in the literature, including those that model and improve large, complex systems.

Simulation can be defined as "the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system (Shannon, 1975)." Manufacturing simulation begins with a careful analysis of how the real-world system operates, mapping the processes involved. System properties such as the times required for various tasks are modeled using statistical techniques and characterized in the form of tables and functions, including statistical distributions. A computer model is built and validated to simulate the system being modeled. Verification of the computer model itself is also carried out. The results of simulation experiments, obtained through running the model, are statistically analyzed and alternative system configurations/operational rules are evaluated against the current system, with the goal of improvement. By its nature, simulation is a "descriptive" method, meaning that it does not search through possible solutions to the problem, but only evaluates the performance of the system -in terms of performance measures- for a given solution. Simulation optimization aims at transforming the simulation into a "prescriptive" method that searches for optimal solutions.

Simulation optimization is viewed by some researchers as the most significant new simulation technology in recent years (Law and Kelton, 2000, Section 12.6). It is "the process of finding the best input variable values from among all possibilities without explicitly evaluating each possiblity (Carson and Maria, 1997)." There are many techniques of simulation optimization, including gradient based search, stochastic optimization, response surface methodology, metaheuristics, statistical techniques and hybrid algorithms (Carson and Maria, 1997). Commercial simulation packages enable modelers to develop simulation models and also provide facilities to carry out simulation optimization (Fu, 2001). With due consideration to the importance of simulation optimization, we have elected to design an educational tool and the related project around the topic.

Related Studies

Simulation Optimization

Carson and Maria (1997) present an early review of applications of simulation optimization. Recent literature reports applications including process scheduling (Cave et al. 2002), plant loading (Thomas et al., 2002), and assembly line planning (Altiparmak et al., 2002).

Educational Tools

Engineering education literature contains several examples of simulation software used in the classroom, including software for teaching construction technology (Li and Love, 2002), process control (Young et al., 2001), and industrial processes (Kuriyen et al., 2001).

Examples of educational software related with optimization include the work of Kahn-Jetter and Sasser (1997), where spreadsheet-based machine design optimization is performed, and Khaisongkram and Banjerdponghchai (2003), where Matlab based GUIs are developed to teach linear controller design via convex optimization.

Simulation-based tools are also used for training purposes in industry: Pfeil et al. (2002) report development of a simulation-based productivity training course at Visteon's Sterling Plant. The main objectives of the course are to emphasize the importance of preventive maintenance and of continuously observing bottleneck operations. Bruzzone et al. Bruzzone et al (2002) present a system to train operators for real-time manufacturing control, based on a case study that involves a company that manufactures locomotive pipes and electrical pipes.

Possible Educational Approaches

A manufacturing simulation study is typically carried out in one of the following ways:

- Using spreadsheets such as MS Excel, StarOffice Calc
- Using commercial simulation software such as Automod (Banks, 2000), Arena (Kelton et al., 2003)

- Using simulation languages such as SIMAN (Pegden et al., 1991)
- Implementing the simulation in a general purpose programming language such as C++, Java

Teaching of manufacturing simulation and simulation optimization can be carried out in relation to each of the listed simulation modeling approaches. We present and discuss each of these alternatives below. Then we describe the implementation details and design considerations of our approach.

Spreadsheet Simulation

Spreadsheets are widely accepted and used in industry for engineering design and planning. Ingolfsson and Grossman (2002) and Hill (2002) report use of spreadsheets to teach manufacturing simulation in the classroom. Besides their popularity and acceptability, spreadsheets have the advantage of being easy to use. However, they are not without disadvantages. Seila (2002) lists limitations of spreadsheets: First of all, only simple data structures are available. Secondly, with spreadsheets complex algorithms are difficult to implement. Thirdly, spreadsheets are slower than other alternatives. Finally, data storage is limited and static. Typically simulation models created using spreadsheets are static, creating difficulties in implementing cases where data changes dynamically.

Commercial Simulation Software

The students may be asked to carry out simulation analysis of a model pre-built in a commercial simulation software. The advantage of this approach is that the students also become familiarized with the commerical software itself and are consequently equipped to solve challenging real world problems through the usage of that software.

The disadvantages of this approach are as follows:

- The commercial software might have to be purchased, typically at a significant cost.
- It could be time consuming for students to learn the software.
- The instructor/teaching assistant has to be experienced enough in using the software to provide support and guidance.

The students may also be asked to build a simulation optimization interface using a programming language or using an optimization engine that accompanies the simulation software (see Carson and Maria, 1997, Section 4, for a discussion of these software products). For example, Rogers (2002) reports experiences involving the use of the OptQuest optimization engine for Arena software.

Simulation Languages

The students may be asked to program a simulation optimization engine that interacts with a model written in a simulation language. Simulation languages have historically been used extensively, and their applications are still reported (Smith et al., 2002). However, the trend in the field of simulation is much more towards using simulation software which enables user-friendly modeling under a windows environment rather than using simulation languages.

In this approach, even though the students would learn a great deal about both programming and the simulation language, they would face the challenge of implementing the communication between the optimization and simulation engines, besides the challenges of programming. Meanwhile, the instructor or the teaching assistant would need to be proficient in the simulation language and its communication with an external optimization engine (written in a programming langauge), as would be the case in teaching with simulation software.

Coding Simulation-based Optimization Algorithms

The students may be provided with the piece of code that performs the simulation and evaluates the performance measures. Then the students could be asked to implement simulation-based optimization techniques (Gosavi, 2003) using a general purpose programming language.

The advantage of this approach is that the students would develop a strong understanding of the concepts and the implementation details of each of the optimization techniques. Since the development of the code would also require careful consideration of data structures, the students would also sharpen their computer skills in general and have a chance to practice their knowledge of Computer Programming and Data Structures. Computer skills are essential in Industrial Engineering as in other engineering disciplines.

This teaching alternative is by far the most time consuming one, and the assignment may eventually turn out to be a mission impossible for some industrial engineering students who do not possess strong computing skills.

Steel Cord Manufacturing

Steel cord is typically used as the main reinforcement material in manufacture of steel radial tires. It strengthens the tire to provide fuel savings, long mileage, safety and comfort. The manufacture of steel cord takes place through continuous processes where wire semi-products are stored on discrete inventory units, namely "spool"s (Figure 1).

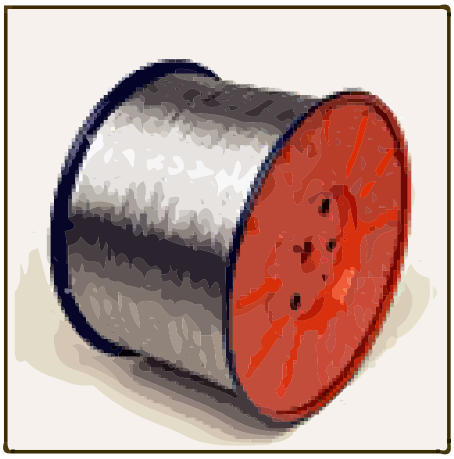

Figure 1. Spool on which Wire is Wound

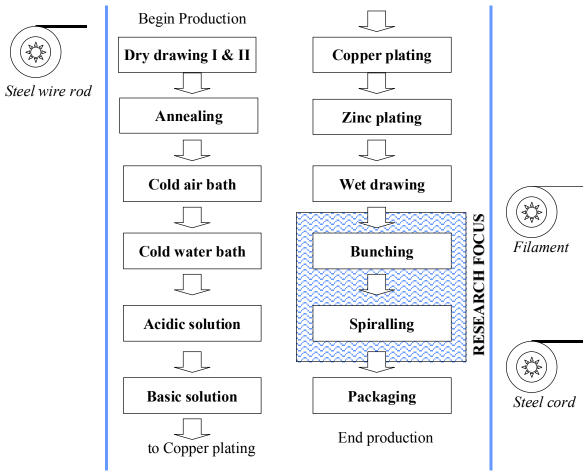

Figure 2. Production Processes in Steel Cord Manufacturing

In steel cord manufacturing incoming raw material, the "steel rod wire", is thinned into "filament"s which are used in successive bunching operations to construct the "steel cord" final products (Figure 2). Between every bunching operation, the intermediate wire products are wound onto spools of varying capacities (in the scale of thousands of meters).

Filaments coming out of wet-drawing are wound on spools and are referred to as “payoff”. Payoff becomes the raw material for bunching and spiralling operations. At each bunching operation, bunched wires enter as “core” to be bunched with a new layer of payoff (filaments) to form “take-up”. The “take-up” in turn becomes the “core” for the following bunching operation (Figure 3).

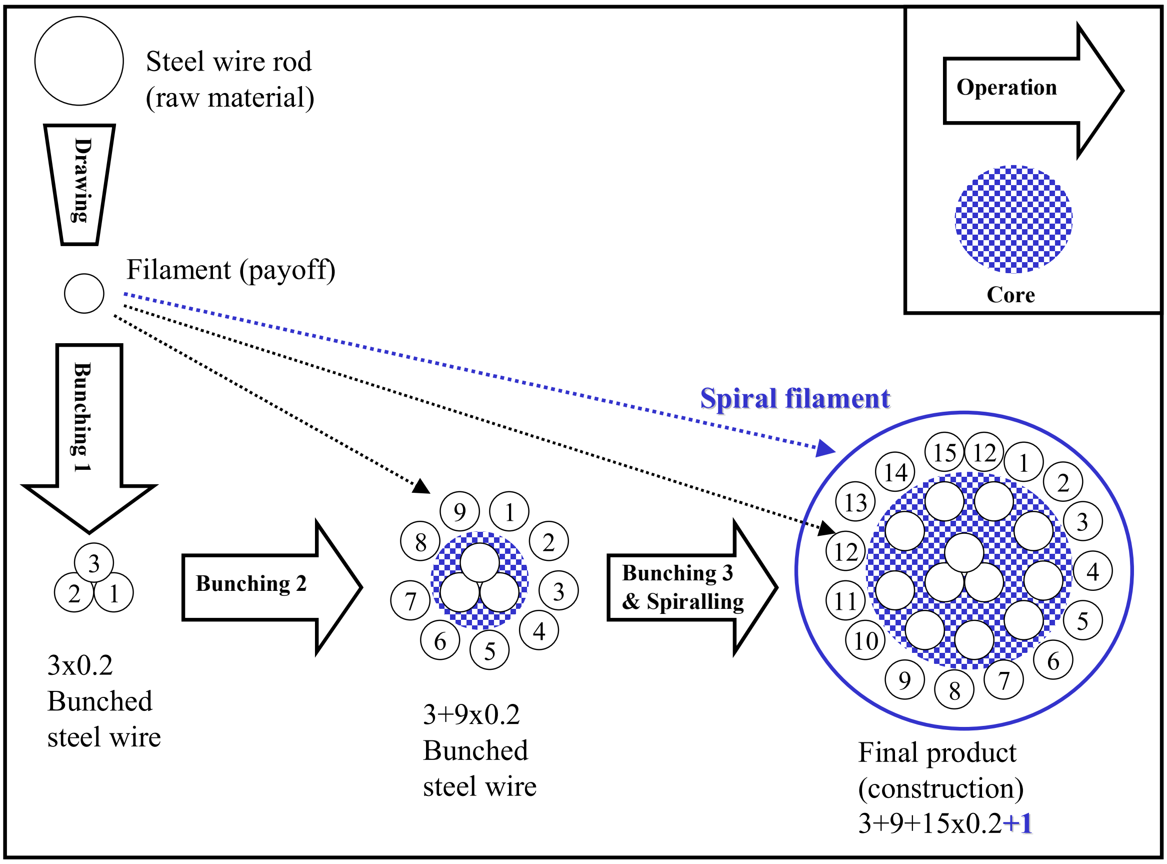

Figure 3. Production Processes in Steel Cord Manufacturing

The final steel cord product is obtained by spiralling a single filament after the final bunching operation, and is referred to as “construction”. Figure 3 illustrates the cross-section of the wire semi-products at various bunching stages in manufacturing of construction “3+9+15x0.2+1”. The naming convention for labeling constructions (and semi-product bunched wires) uses a “+” sign to denote each additonal bunching operation. The construction “3+9+15x0.2+1” is obtained by bunching 3 filaments of length 0.2mm in the first bunching operation, then 9 filaments, and then 15 filaments at 0.2mm, followed by a single spiralled filament.

As the spool of core and spools of payoff are used in a bunching operation, any of the spools may run out first. The time it takes for this run-out is a function of the spool lengths and production rates of the machines, besides other factors, some of which are discussed below. As run-out takes place, the bunching machine gradually slows down and finally stops. A setup is performed by a skilled operator to feed the next spool with the same kind of wire (core or payoff) into the machine. Payoff or core spool is tied at the wire location where the machine had stopped, and production in that machine restarts. Since the stopping takes place gradually, a certain amount of wire is typically wasted at every "change-over". This tying of changed spools results in a knot, which is an undesired situation. When the take-up spool (the spool on which the semi-product wire out of a bunching operation is wound) is completely full, a change of take-up is performed. Besides knots due to spool changes, "wire fractures", seemingly random breaks of the wire due to structural properties, may also result in a considerable number of additional knots. By tagging an information card on each spool the locations of knots can be recorded.

After the spiralling operation the steel cord is cut into specified lengths and wound onto final spools, which are eventually packaged for customers. Tire manufacturers prefer that the spools with the final cuts of steel cords contain no knots at all. Final spools that contain knots, namely "rejected spools", are classified as second quality and are sold at a very low price. Therefore, it is an important management objective to decrease the number of knots and the number of rejected spools.

The motivation of the simulation study has been to identify improved operating policies, specifically "optimal" spool lengths for each bunching operation, such that quality and productivity are improved. Both of these two performance measures can be improved if the number of rejected spools (spools containing knots) is reduced.

The optimal spool lengths are constrained to be within a certain percentage of the current spool lengths. This constraint is imposed by plant managers as a result of the strategy of making gradual changes over time, as opposed to rushing in radical changes in short time periods. One other reason for such constraints is the impact on other operational measures. For example, selecting the spool lengths that are too short would lead to prohibitively frequent payoff or core changes, and increased operator costs.

Some production issues are almost unique to this particular type of manufacturing: An example is that the locations of knots are reversed at every spool change. When a wound spool of length h with knot locations $(k_1, k_2, ..., k_n)$ is fed into the bunching operation, the unwinding results in knot locations $(h-k_n, ..., h-k_2, h-k_1)$. Turkseven and Ertek report programming the simulation with a general-purpose language due to this and other complexities that would be next to impossible to reflect using spreadsheets and would have to be custom-programmed if a simulation language or modeling software were used.

Literature on steel cord manufacturing is not extensive, since it is a very specialized type of production, and the systems required for steel cord manufacturing are produced and installed by only a handful of companies in the world. Almost all research on steel cord manufacturing is from materials science literature. We refer the readers interested in operational aspects to the following three studies: Thomas et al. (2002) report optimization of operations in a steel wire manufacturing company. Mercankaya (2003) develops an optimization-based decision support system for steel cord manufacturing. Turkseven and Ertek (2003) present the simulation program that our study is based upon.

The Tool Developed

Our educational tool allows the administrator (the professor or teaching assistant) to design a customized problem setting for the student through the stand-alone SteelCordSimulator_Administrator.exe program (Figure 4) by setting general parameters for the plant. The output of the SteelCordSimulator_Administrator.exe program is a binary "problem definition file" that contains plant parameters such as percentage of defects, number of simulation experiments to be carried out, simulation length and most importantly the permitted ranges for spool lengths. The problem definition file also includes the current spool lengths and the related simulation results. The students are provided with this problem definition file, which they load into the stand-alone SteelCordSimulator_Student.exe program (Figure 5). The students carry out experiments by typing in spool lengths, running the simulation by clicking the "Start Simulation" button and obtaining the performance measures (number of full spools, number of rejected spools, rejected wire in meters) for the specified set of spool lengths. The plant settings are visible to the student but are disabled, since they should only be set by the administrator. The students can scroll through the experiments that they perform through a user-friendly interface, save the current file into an "experimental results file", and export the experimental results to a text file for further analysis in a spreadsheet or a statistics software.

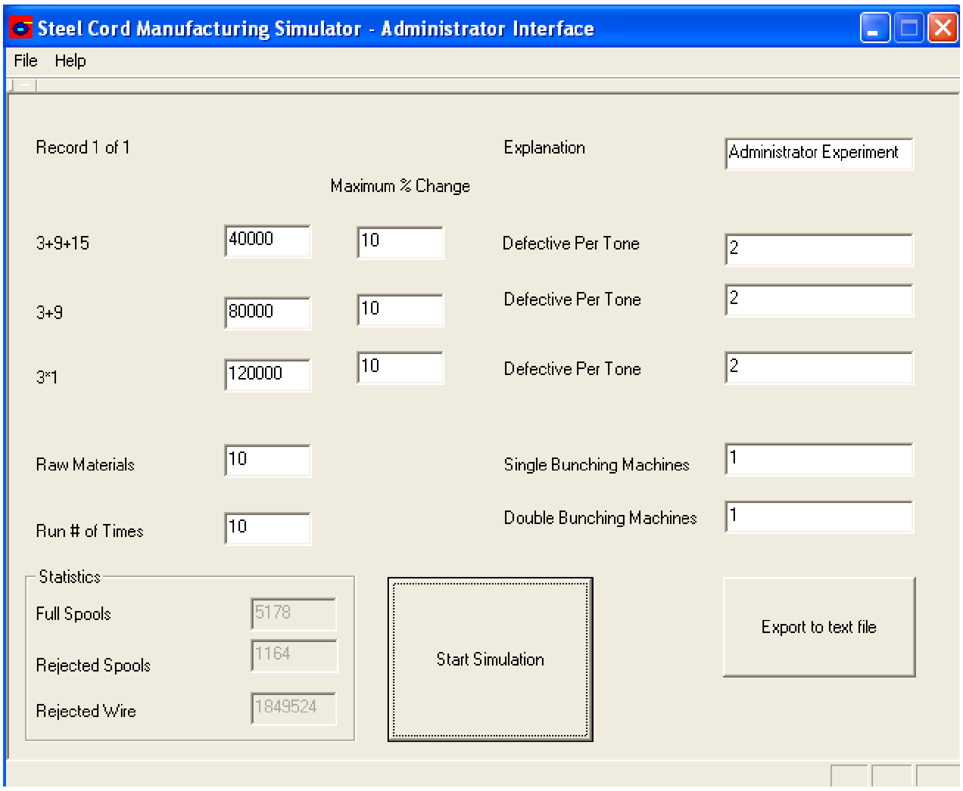

Figure 4. The Administrator's Interface

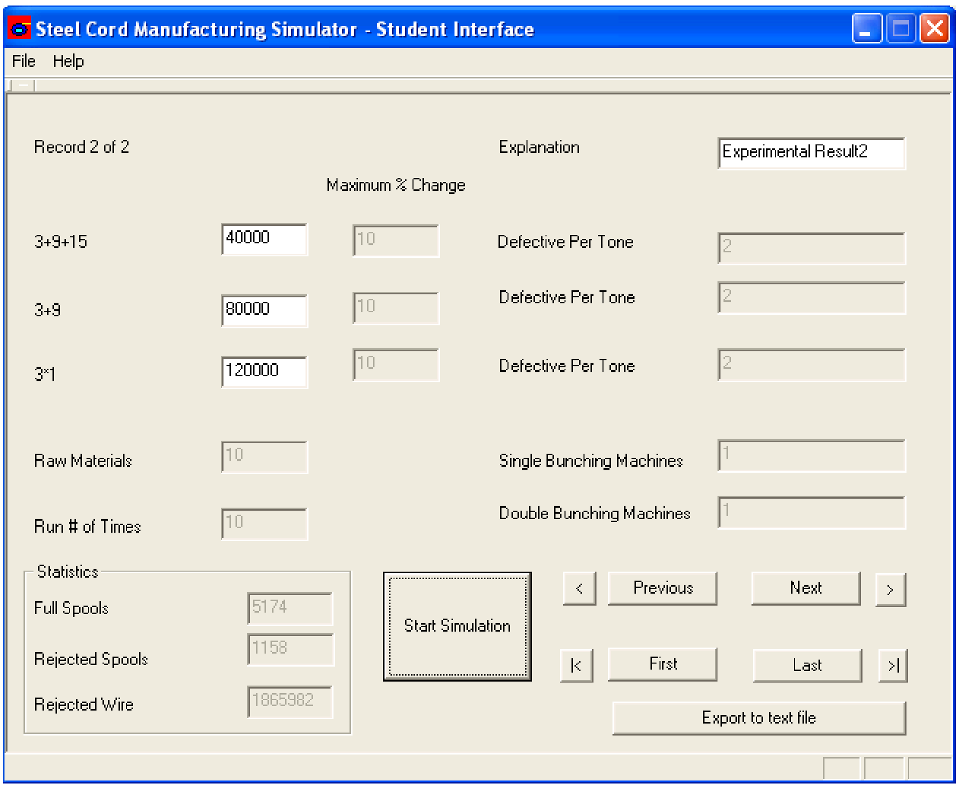

Figure 5. The Student's Interface

This design with seperate customized interfaces for the administrator and the student is similar to the design of Kuriyan et al. (2001), where the authors develop a web-based simulation of an industrial process with instructor and student interfaces.

The simulator is based upon the simulation code of Turkseven and Ertek (2003) written in C++, which evaluates the performance measures for a given selection of spool lengths under a particular plant setting. We built the GUI (Graphical User Interface) using MS Visual C++. Net platform under MS Windows Operating System. The installation program was created using Clickteam Install Creator software[1].

Classroom Experience

The earlier version of the developed tool was used in an introductory industrial engineering course, in a project that asked the students to come up with strategies/algorithms to find optimal spool lengths. Early on in the project, most of the students realized that there is a nonlinear relation between the simulation inputs (spool lengths) and outputs (performance measures). They recognized that carrying out more experiments increased the chances of finding better solutions.

The algorithms that the student groups suggested were simplistic. Many groups designed their experiments by selecting n equally-spaced values for each of the full spool lengths, thus carrying out n^3 experiments. This was not really a stimulating algorithm in terms of the goals of the project. The second approach that some groups followed was fixing one or two of the spool lengths and playing with the values of the other spool length(s). One creative group plotted a 3-D graph of the inputs, and let the size of the datapoint in the graph denote one of the performance measures (the number of full spools).

The projects were mainly evaluated on the basis of % increase obtained in the number of full spools compared to the current system, the sophistication involved in developing the algorithms, and clarity in writing the project reports. The % increase achieved in the number of full spools changed from 0.03% to 4.31%, with an average of 3.10%. A histogram of the % increases is given in Figure 6.

[1] http://www.clickteam.com

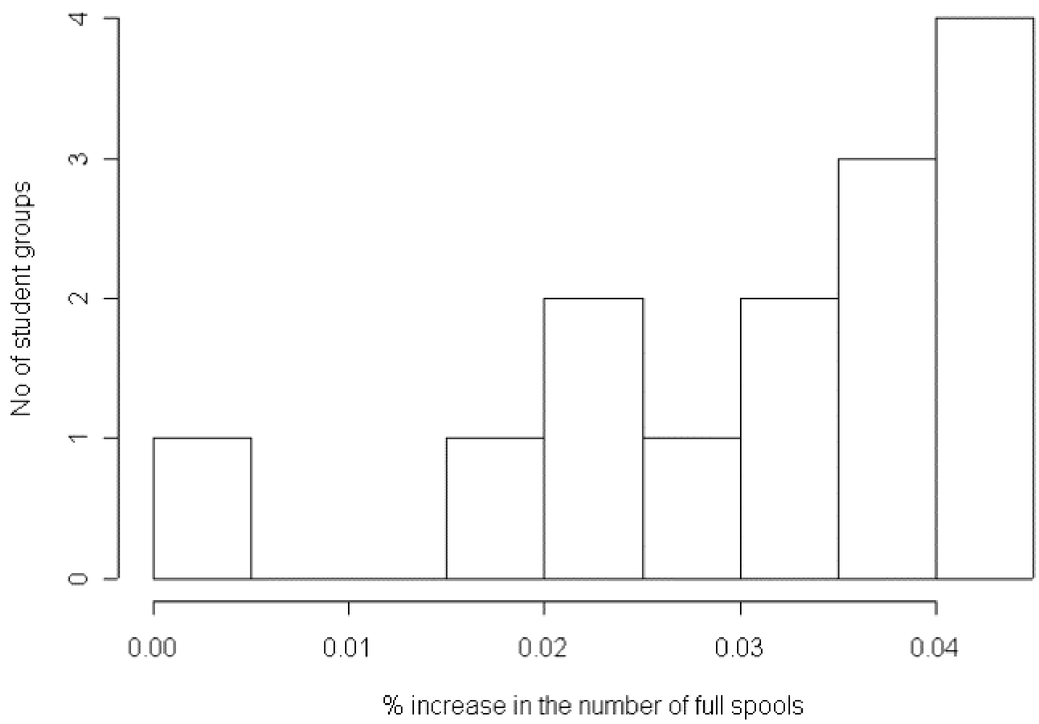

Figure 6. Histogram of % Increase Achieved by Student Groups
in the Number of Full Spools

In future applications, students could be encouraged to develop more sophisticated strategies/algorithms to solve the problem, present their algorithms in pseudo-code and effectively use charts to obtain deeper insights.

Conclusions

The educational tool that we present in this paper is based on a case study from industry, and introduces the students to an interesting real world problem in manufacturing. Real world problems typically require thousand even millions of decision variables. Yet, the selected problem involves only three decision variables, preventing the student from getting lost in the complexity of the problem. From the project description, the students learn the operational issues, objectives and constraints in the steel cord plant. They gain experience in developing algorithms as they try to optimize the spool lengths at different stages of the bunching operation. The tool is easy to install, learn and use, allowing students to carry out numerous experiments. We have implemented a special interface for the administrator (the instructor or teaching assistant), making it possible to set new parameters for the problem, allowing assignment of the project in subsequent years.

Acknowledgement

We thank Turgut Uzer for allowing us to share our experiences with academia. We thank Tonguc Unluyurt, Alpay Taralp and Gordon Dobie at Sabanci University for reading the draft of the paper and providing helpful comments and suggestions. In our study, we extensively used the online Proceedings of Winter Simulation Conference (WSC), with full papers available on the internet[2]. We thank INFORMS College on Simulation for making the proceedings available online.

References

Altiparmak, F., Dengiz, B. and Bulgak, A. A. (2002) Optimization of buffer sizes in assembly systems using intelligent techniques. *Proceedings of 2002 Winter Simulation Conference*, 1157-1162.

Banks, J. (2000) *Getting Started with Automod*. Autosimulations, Inc., Bountiful, Utah, available at http://www.autosim.com/academic/gswaproductpage.asp

Bruzzone, A. G., Giribone, P. and Revetria, R. (2002) Simulation as educational support for production and logistics in industrial engineering. *Proceedings of 2000 Winter Simulation Conference*, 1613-1618.

Carson, Y. and Maria, A. (1997) Simulation optimization: Methods and applications. *Proceedings of 1997 Winter Simulation Conference*, 118-126.

Cave, A., Nahavandi, S. and Kouzani, A. (2002) Simulation optimization for process scheduling through simulated annealing. *Proceedings of 2002 Winter Simulation Conference*, 1909-1913.

Fu, M. C. (2001) Simulation optimization. *Proceedings of 2001 Winter Simulation Conference*, 53-61.

Gosavi, A. (2003) *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Kluwer.

[2] http://www.informs-cs.org/wscpapers.html

Hill, R. R. (2002) Process simulation in Excel for a quantitative management course. *INFORMS Trans. on Educ.*, Vol. 2, No. 3.

Ingolfsson, A. and Grossman, T. A. (2002) Graphical spreadsheet simulation of queues. *INFORMS Trans. on Educ.*, Vol. 2, No. 2.

Kahn-Jetter, Z. L. and Sasser, P. A. (1997) Using spreadsheets for studying machine design problems involving simulation. *Comput. Appl. in Eng. Educ.*, Vol. 5, No. 3, 199-211.

Khaisongkram, W. and Banjerdponghchai, D. (2003) MATLAB based GUIs for linear controller design via comvex optimization. *Comput. Appl. in Eng. Educ.*, Vol. 11, No. 1, 13-24.

Kelton, W. D., Sadowski, R. P. and Sturrock, D. (2003) *Simulation with Arena*. 3rd Ed., McGraw-Hill.

Kuriyan, K., Muench, W. and Reklaitis, G. V. (2001) Air products hydrogen liquefaction project: building a web based simulation of an industrial process. *Comput. Appl. in Eng. Educ.*, Vol. 9, No. 3, 180-191.

Law, A. M. and Kelton, W. D. (2000) *Simulation Modeling and Analysis*, 3rd Ed. McGraw-Hill, New York.

Li, H. and Love, P. E. D. (2002) Use of visual simulation in construction technology education. *Comput. Appl. in Eng. Educ.*, Vol. 10, No. 3, 161-166.

Mercankaya, B. (2003) *Sales forecasting and production planning of BEKSA in the context of supply chain management*, Technical Report, Sabanci University, Istanbul, Turkey.

Patchong, A., Lemoine, T. and Kern, G. (2003) Improving car body production at PSA Peugeot Citroen. *Interfaces*, Vol. 33, No. 1, 36-49.

Pegden, C. D., Sadowski, R. P. and Shannon, R. E. (1991) *Introduction to Simulation Using Siman*, McGraw-Hill.

Pfeil, G., Holcomb, R., Muir, C. T. and Taj, S. (2000) Visteon's Sterling plant uses simulation-based decision support in training, operations, and planning. *Interfaces*, Vol. 30, No. 1, 115-133.

Rogers, P. (2002) Optimum-seeking simulation in the design and control of manufacturing systems: Experience with OptQuest for Arena. *Proceedings of 2002 Winter Simulation Conference*, 1142-1150.

Seila, A. F. (2002) Spreadsheet simulation. *Proceedings of 2002 Winter Simulation Conference*,17-22.

Shannon, R. E. (1975) *Systems Simulation – The Art and Science*, Prentice-Hall.

Smith, J. S., Li, Y. and Gjesvold, J. (2002) Simulation-based analysis of a complex printed circuit board testing process. *Proceedings of 2002 Winter Simulation Conference*, 993-998.

Thomas, J. , Todi, J. and Paranjpe, A. (2002) Optimization of operations in a steel wire manufacturing company. *Proceedings of 2002 Winter Simulation Conference*, 1151-1156.

Turkseven, C. H. and Ertek, G. (2003) Simulation modeling for quality and productivity in steel cord manufacturing. *Proceedings of 2003 Winter Simulation Conference*,

Young, B. R., Mahoney, D. P. and Svrcek, W. Y. (2001) Real-time computer simulation workshops for the process control education of undergraduate chemical engineers. *Comput. Appl. in Eng. Educ.*, Vol. 9, No. 1, 57-62.