

ON USING MONTE CARLO METHODS FOR SCHEDULING

Samarn Chantaravarapan
Ali Gunal

Production Modeling Corporation
Suite 1006 Parklane Towers West
3 Parklane Boulevard
Dearborn, MI 48126, U.S.A.

Edward J. Williams

2330 Engineering Complex
Industrial & Manufacturing Systems Engineering
University of Michigan – Dearborn
4901 Evergreen Road
Dearborn, MI 48128, U.S.A.

ABSTRACT

Monte Carlo techniques have long been used (since Buffon's experiment to approximate the value of π by tossing a needle onto striped paper) to analyze phenomena which, due to their complexity and/or stochasticity, are beyond the reach of closed-form equations. Basic examples of such studies are estimating the probability that military field communications will remain intact in the face of attack or the number of fish in an irregularly shaped lake. Likewise, scheduling is a necessity for the planning, control, and implementation of increasingly large projects in manufacturing, civil construction, military operations, and many other fields. We provide a framework for applying scheduling algorithms based on Monte Carlo simulation, to provide a scheduler, who inevitably confronts numerous uncertainties, an inexpensive and a highly customizable tool that can be utilized in a common spreadsheet environment.

1 INTRODUCTION

The supervisor of complex, multifaceted work must pay close attention to the scheduling and interrelationships (such as precedence and priority) among numerous tasks which may collaborate (successive tasks all directed toward an objective) or compete for priority and scarce resources (Lawrence and Zanakis 1984). Not only can effective scheduling yield cost savings and productivity increases, but also it can yield many other benefits (saving more lives in a hospital or reducing a university's need to expand facilities (Stevenson 2005). Numerous and relatively standardized approaches to scheduling are documented in sources such as (Pinedo 2002) and (Morton and Pentico 1993). Nevertheless, these methods typically have limited ability to adapt dynamically to changing conditions such as an unexpected raw material shortage, breakdown of a machine, sickness of a skilled worker, or rapid changes in priority driven by the marketplace. Since Monte Carlo simulation is intrinsically well equipped to support decision-making when confronting uncertainty (Aburdene 1988),

many industrial engineers and analysts have developed a variety of problem-specific approaches applying its power to dynamic scheduling within stochastic systems. For example, (Levchenkov and Gorobetz 2003) developed a method of scheduling "appointments to travel" within a conveyor system frequently receiving competing demands for service. (Dangelmaier, Franke, and Scheideler 2003) provide an algorithm which helps ostensibly competing logistics companies schedule resource usage to provide globally improved response to customers' collective demands. Likewise, (Mosca, Queirolo, and Tonelli 2002) have successfully attacked a job-sequencing problem within a semi-automated (hence relatively highly stochastic) production process using simulation. In this paper, we extend work of this type by providing a more generic and adaptable methodology and framework within which complex scheduling problems can be analyzed with the help of Monte Carlo methods. These methods, first introduced by von Neumann and Ulam, have historically been viewed with either undue optimism or undue pessimism; the pendulum of aggregate opinion is returning to moderate optimism. Justifications for this optimism are the steady increase in computer processor power and the increasingly dispersed awareness that understanding the overall structure of a particular problem is a prerequisite to attacking it effectively with Monte Carlo methods (Marchuk 1981).

2 BACKGROUND OF PROBLEM AND PREVIOUS METHODS OF ANALYSIS

We investigate a crew-scheduling problem at a plant manufacturing components for aircraft. At present, schedulers assign orders to workstations using a spreadsheet, usually requiring one to two days to complete a schedule. A scheduling template was designed in an EXCEL[®] spreadsheet. Initially, operators assign orders to complete at their due dates; then orders are rearranged manually in order to smooth the manpower requirements on a daily basis. There is a significant amount of heuristic knowledge of the floor operations, which schedulers utilize in building

schedules. The main design philosophy of our solution concept is then to automate the most tedious parts of drawing a schedule while still allowing the scheduler to apply his/her knowledge of how schedules should be constructed. Therefore, our goal is to help the scheduler reduce the scheduling time from three days to one.

We designed a solution that would automate the process of creating an initial schedule. We wanted to have an efficient yet simple algorithm to speed up the process of creating an initial schedule. The scheduler would then improve this schedule based on factors not easily accommodated in our algorithm.

We have explored several approaches, including mixed-integer programming and packaged scheduling software. Given time and budget considerations, we decided to build a simple tool integrated with the existing spreadsheet approach. Significant advantages were ease of use, little implementation effort, and low training requirements.

An approach based at least in part on random searches was considered a promising place to begin our investigations. Intuitive justification for this decision lies largely in the realization that many highly effective algorithms, such as genetic ones, admit randomness as an urgently needed escape hatch from local (but non-global) optima (Gen and Cheng 2000). At the core of the solution was an efficient search algorithm that strived to minimize an objective function of a normalized penalty function, allowing us to consider not only manning requirements but also inventory.

To measure the worker load, we use the variance of number of workers required daily. The scheduling horizon is eight weeks. The variance measures the deviation of number of workers daily from the average number of workers required over the eight-week period. A small variance implies that the number of workers required daily fluctuates little. Strictly minimizing this variance can cause some orders to be scheduled much earlier than they are needed. Therefore another criterion we added to the model is the average inventory holding days per order. We assume that the orders will be shipped at their due dates. Therefore, if orders are completed earlier, then an inventory holding cost is incurred. Therefore, to minimize the inventory cost, we try to schedule orders as close to their due dates as possible.

3 METHODOLOGY

3.1 Spreadsheet

We modified an existing EXCEL® spreadsheet to solve this problem. We develop the solution algorithm using a Visual Basic for Applications (VBA) routine. The spreadsheet interface is shown in Figure 1.

To make the algorithm accessible in a user-friendly manner, we developed built-in toolbars, as shown in Figure 2. The **Import Data** button imports order information from the client’s data file, which is in a comma-delimited

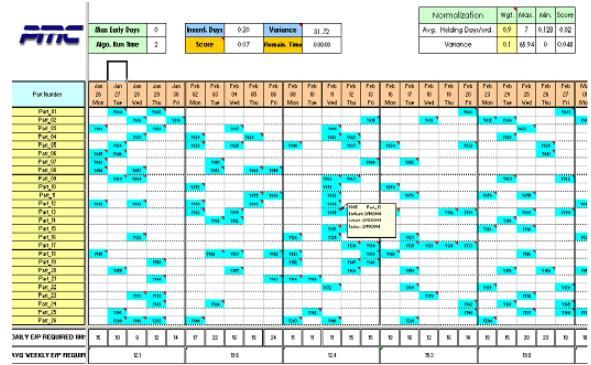


Figure 1: Spreadsheet Interface

(CSV) format. The **Revert** and **Save** buttons are useful when manual moves are performed. Operators may save the schedule from time to time during improvement. The Revert button recalls the last saved schedule into the spreadsheet. The **Clear** button is designed to erase all orders from the schedule. Three effective intelligent search algorithms under the **Optimization** button help operators in search of a better schedule. The details of these improvement-seeking algorithms will be discussed in the next section. Operators may specify the running time for these algorithms. The longer the running time is, the better the schedule tends to be.

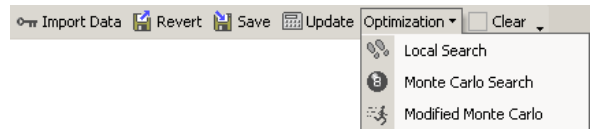


Figure 2: Built-In Toolbars

Furthermore, the spreadsheet gives users the ability to move or schedule orders manually. Operators can select an order and move to any position desired, and then press the **Update** button to update the schedule. This action causes a scan of the existing schedule and highlights any problematic assignments in various colors. An order with yellow shading indicates an early order, whereas an order with red shading implies that the order is assigned after its due date. An order with purple shading reminds operators that the order is assigned to the wrong part, as may happen when orders are manually assigned. The Comment feature included provides order information, such as part number, order number, earliest date and due date.

3.2 Search Algorithms

One of the goals of scheduling was to level the number of workers required daily and to minimize the average inventory holding days per order. There are several ways to deal with these multi-objective problems. One way is to optimize one objective function first, and then attempt to improve the second objective function while maintaining the

first objective function value. Alternatively, the second approach is to improve both objective functions simultaneously. However, as both objective functions have different units, we chose to use a normalization technique to combine them. This method provides flexibility to consider criteria other than those considered in this study.

To use this normalization technique, the maximum and minimum objective function values are necessary. In the case of the average inventory holding days per order, the minimum value can be obtained by scheduling all orders by their due dates. Similarly, assigning orders by their earliest dates provides the maximum value. In case of the variance value, the minimum value is set to 0, while the maximum value is obtained from the initial schedule.

The score (sc) of each objective function can be calculated using the following formula:

$$sc(i) = \frac{\text{current} - \text{min}}{\text{max} - \text{min}}.$$

Therefore, the modified objective function (net score) can be expressed as:

$$\sum_{i=1}^2 w(i).sc(i).$$

To obtain an initial schedule, orders are assigned by their due dates in the schedule. An order is assigned to its due date first. If the date is unavailable, then the order will be assigned to the date before the due date. It is noticeable that the due-date-first rule obtains a schedule with small average inventory holding days and highly unbalanced worker load. To improve the schedule, three effective search algorithms are described next.

The first algorithm, namely the Local Search, seeks the best possible date within its time range constraint. The algorithm starts by selecting an order randomly, and searching for the date with the best improvement, starting from the earliest date to its due date. If the schedule is thus improved, then the schedule is updated. Similar to the first algorithm, the second algorithm, the Monte Carlo Search, begins by randomly selecting an order. However, instead of trying every possible date, the Monte Carlo Search selects a date between its earliest date and its due date randomly and performs the interchange process. If improvement occurs, then the schedule is updated.

In the second algorithm, for each order, we assume that all feasible dates have equal probabilities of selection. However, one of our objective functions is to minimize the inventory holding days per order. Thus, it is reasonable to assign different probabilities to different feasible dates. In other words, a date close to an order's due date should be selected with higher probability than a date close to its earliest date. Using this approach constrains orders to be assigned as close to their due dates as possible. This modi-

fied algorithm is called the Modified Monte Carlo Search. For example, if the time range of an order is three days, the due date should have probability of $\frac{3}{6}$, while the earliest date should have the probability of $\frac{1}{6}$. If improvement materializes, then the schedule is updated.

4 A CASE STUDY

We performed a small experiment to investigate the algorithms' performance. We assigned equal weight (0.5) to both objective functions and ran each algorithm for 15 minutes and observed the results. Note that the sum of all weights must be equal to 1. Figure 3 shows that the Monte Carlo Search outperforms other algorithms, while the local search seems to have the worst performance of all. When we assigned higher weight to worker load variance (0.9), Figure 4 shows that the Monte Carlo Search still remains the best algorithm. It is observed that if the computer run time is our constraint, Modified Monte Carlo Search outperforms Local search. Otherwise, in a long run, the Local Search may obtain a better schedule than the Modified Monte Carlo Search does. Figure 5 demonstrates the results when assigning the weight of 0.9 to the average inventory holding days per order objective function. Since orders in the initial schedule, assigned by the due dates, already obtain a low value of the average inventory holding days, we see little improvement in Figure 5.

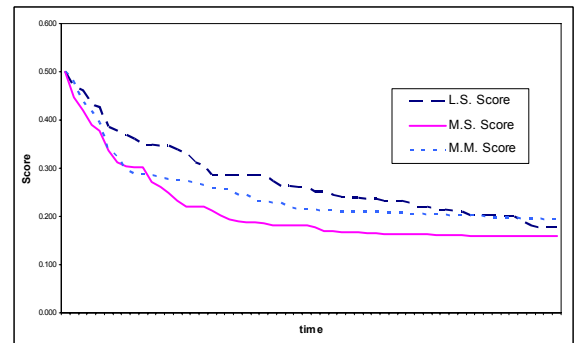


Figure 3: Weights (0.5, 0.5)

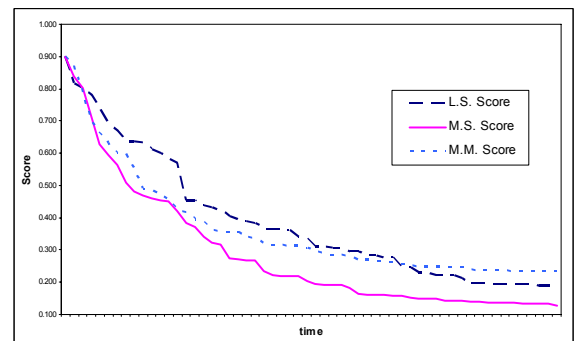


Figure 4: Weights (0.1, 0.9)

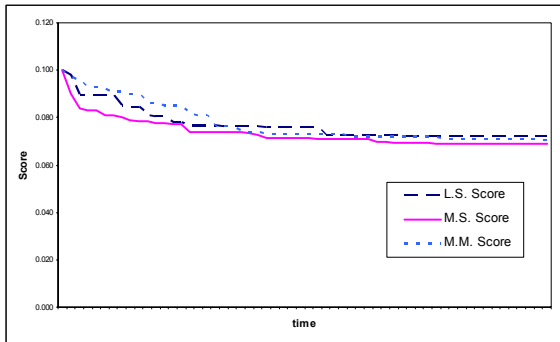


Figure 5: Weights (0.9, 0.1)

Furthermore, we tested the performance of the algorithms when early jobs are allowed. We set the maximum early days to 5. That is, an order may be assigned up to 5 days before its earliest date. It is observed that the performance of the Modified Monte Carlo Search improves noticeably. Furthermore, Figure 6 shows that if weights are assigned equally to both objective functions, the Modified Monte Carlo Search may outperform the other two algorithms. When the weights vary (Figures 7-8), the Modified Monte Carlo Search performs almost equivalently to the Monte Carlo Search, if not better. One reason for this phenomenon is the increase in the time-range constraint. If the time range is wide, the Modified Monte Carlo Search tends to be more effective in selecting a better date than Monte Carlo Search. If a longer running time is permitted, the Modified Monte Carlo Search may outperform the Monte Carlo Search. Furthermore, when the time range increases, the Local Search requires more time to search for the best date for each order.

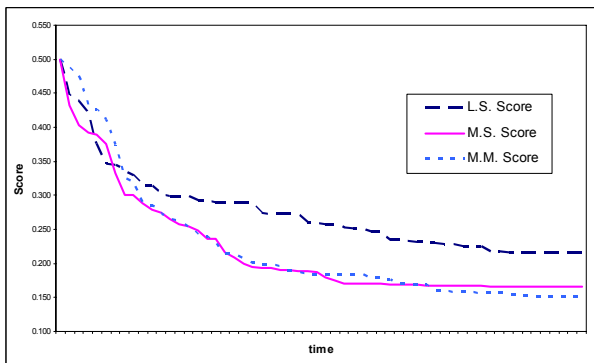


Figure 6: Weights (0.5, 0.5) with Max. Early days = 5

Although there is no guarantee that these algorithms can deliver optimal solutions, the results seem to be very acceptable, as the number of workers required daily shows no dramatic fluctuations. Furthermore, a better schedule can be obtained if longer running time is permitted. Nevertheless, we tested these algorithms with only one dataset. Additional tests are required to draw firmer conclusions concerning the performance of these algorithms.

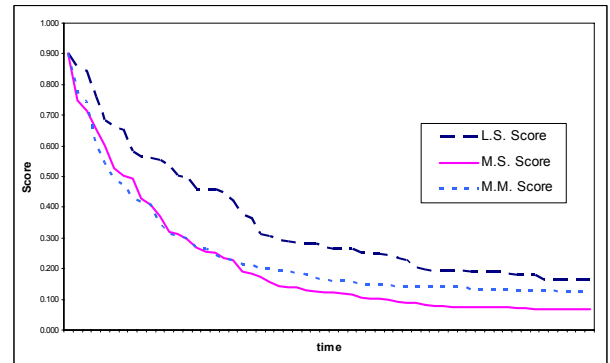


Figure 7: Weights (0.1, 0.9) with Max. Early days = 5

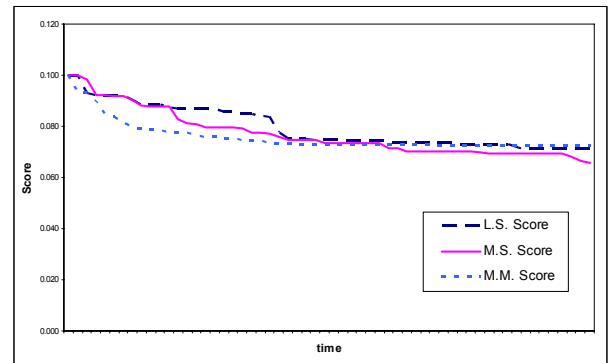


Figure 8: Weights (0.9, 0.1) with Max. Early days = 5

There is also room to improve the solution speed by choosing better initial schedules. Nevertheless, the purpose of this paper is not to provide the reader the best algorithm of all. Instead, we intend to give the reader an example of how to implement similar algorithms in similar problems.

5 CONCLUSIONS AND INDICATED FUTURE WORK

The goal of this paper is to describe a practical solution that is facilitated by the flexibility of Monte Carlo methods. Combined with a simple and user-friendly approach, the described solution cost-effectively satisfied a need for an improved scheduling tool. We show that an Excel[®] spreadsheet is an alternative powerful tool in solving such complex scheduling problems. As most users are already familiar with this spreadsheet tool, training becomes easier and requires less time to understand how to use the model. We find that the Monte Carlo method provides a convenient way to consider multiple objectives within a single search algorithm for similar size problems.

We show an example of using a spreadsheet in solving a real-life worker scheduling problem. We designed our spreadsheet similar to the existing spreadsheet with additional built-in toolbars. Users have the ability to assign the schedule manually and automatically. Furthermore, we developed several effective algorithms using Visual Basic

for Applications (VBA). These promising algorithms provide consistently good results, although not necessarily optimal, within an acceptable running time. Additionally, several academically effective search algorithms, such as genetic algorithm and simulated annealing, may be studied in search of a near-optimal schedule.

REFERENCES

- Aburdene, Maurice F. 1988. *Computer Simulation of Dynamic Systems*. Dubuque, Iowa: Wm. C. Brown Publishers.
- Dangelmaier, Wilhelm, Hubertus Franke, and Peter Scheideler. 2003. "A Parallel Procedure for a Distributed Logistics Scheduling." In *Proceedings of the 1st International Industrial Simulation Conference*, eds. Juan Carlos Guerri, Ana Pajares, and Carlos Palau, 233-237.
- Gen, Mitsuo, and Runwei Cheng. 2000. *Genetic Algorithms and Engineering Optimization*. New York, New York: John Wiley & Sons, Incorporated.
- Lawrence, Kenneth D., and Stelios H. Zanakis. 1984. *Production Planning and Scheduling: Mathematical Programming Applications*. Norcross, Georgia: Industrial Engineering and Management Press.
- Levchenkov, Anatoly, and Mikhail Gorobetz. 2003. "The Algorithm of Negotiation for Software Agents for the Open Conveyor Schedule in Logistics Tasks." In *Proceedings of the International Workshop on Harbour, Maritime and Multimodal Logistics Modelling and Simulation*, eds. Yuri Merkuryev, Agostino G. Bruzzone, Galina Merkuryeva, Leonid Novitsky, and Edward Williams, 142-148.
- Marchuk, G. I. 1981. *Methods of Numerical Mathematics*, 2nd edition. New York, New York: Springer-Verlag New York Incorporated.
- Morton, Thomas E., and David W. Pentico. 1993. *Heuristic Scheduling Systems with Applications to Production Systems and Project Management*. New York, New York: John Wiley & Sons, Incorporated.
- Mosca, Roberto, Filippo Queirolo, and Flavio Tonelli. 2002. "Job Sequencing Problem in a Semi-Automated Production Process." In *Proceedings of the 14th European Simulation Symposium*, eds. Alexander Verbraeck and Wilfried Krug, 343-347.
- Pinedo, M. 2002. *Scheduling – Theory, Algorithms, and Systems*, 2nd edition. Englewood Cliffs, New Jersey: Prentice-Hall Incorporated.
- Stevenson, William J. 2005. *Operations Management*, 8th edition. New York, New York: The McGraw-Hill Companies, Incorporated.
- in June 2003. He was graduated with a Ph.D. degree in Industrial Engineering from Texas Tech University, Texas, in 2002. He also holds bachelor's and master's degrees in industrial engineering from Chulalongkorn University, Thailand, in 1994, and University of Pittsburgh, Pennsylvania, in 1997, respectively. In 2003, as a visiting assistant professor, he taught Operations Management classes at University of Alabama in Huntsville, Alabama, for one semester before joining PMC. His interests include simulation, supply chain analysis, and complex scheduling problems. He is familiar with several related analytical software tools, such as SIMUL8, GPSS/H, Enterprise Dynamics, and ASPROVA. His email address is <schantar@pmcorp.com>.

AUTHOR BIOGRAPHIES

SAMARN CHANTARAVARAPAN joined Production Modeling Corporation (PMC) as an Applications Engineer

ALI GUNAL serves as the Director of Services at Production Modeling Corporation. He received his Ph. D. degree from Texas Tech University, Texas in Industrial Engineering in 1991. Ali has been involved in consulting engagements with companies like Sony, Samsung, Volkswagen of America, Nestle, Sara Lee, and Medtronic in areas of supply chain management and manufacturing improvement through the application of information technology. In addition, he has an extensive background in application of simulation and other analytical methods to manufacturing management problems. Prior to joining PMC, he worked at i2 Technologies as a senior consultant, and later served as an Operations Research Specialist for the State of Washington, where he developed a simulation system for modeling and analysis of civil law suit litigations. He is familiar with several simulation systems including AutoMod, Arena, Quest, Robcad, and Igrip. His email address is <agunal@pmcorp.com>.

EDWARD J. WILLIAMS holds bachelor's and master's degrees in mathematics (Michigan State University, 1967; University of Wisconsin, 1968). From 1969 to 1971, he did statistical programming and analysis of biomedical data at Walter Reed Army Hospital, Washington, D.C. He joined Ford Motor Company in 1972, where he worked until retirement in December 2001 as a computer software analyst supporting statistical and simulation software. After retirement from Ford, he joined Production Modeling Corporation, Dearborn, Michigan, as a senior simulation analyst. Also, since 1980, he has taught evening classes at the University of Michigan, including both undergraduate and graduate simulation classes using GPSS/H™, SLAM II™, SIMAN™, ProModel®, SIMUL8®, or Arena®. He is a member of the Institute of Industrial Engineers [IIE], the Society for Computer Simulation International [SCS], and the Michigan Simulation Users' Group [MSUG]. He serves on the editorial board of the *International Journal of Industrial Engineering – Applications and Practice*. During the last several years, he has given invited plenary addresses on simulation and statistics at conferences in Monterrey, México; İstanbul, Turkey; Genova, Italy; and Rīga, Latvia. He has just served as Pro-

gram Chair of the 2004 Summer Computer Simulation Conference, and is serving as Program Chair for the 2005 IIE Simulation Conference and the 2005 Summer Computer Simulation Conference. His e-mail address is: <williame@umdsun2.umd.umich.edu> and his university web-page is <<http://www.personal.umd.umich.edu/~williame>>.