Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

A COMPARISON OF THREE OPTIMIZATION METHODS FOR SCHEDULING MAINTENANCE OF HIGH COST, LONG-LIVED CAPITAL ASSETS

Terry M. Helm
Steve W. Painter
W. Robert Oakes

Los Alamos National Laboratory
Los Alamos, NM 87545, U.S.A.

ABSTRACT

A range of minimization methods exist enabling planners to tackle tough scheduling problems. We compare three scheduling techniques representative of "old" or standard technologies, evolving technologies, and advanced technologies. The problem we address includes the complications of scheduling long-term upgrades and refurbishments essential to maintaining expensive capital assets. We concentrate on the costs of being able to do maintenance work. Using a standard technology as the baseline technique, Constraint Programming (CP) produces a 50-yr maintenance approach that is 31% less costly. Genetic Programming produces an approach that is 60% less costly.

1 INTRODUCTION

Simulation "is an established method" for assessing performance capabilities and operational domains for a wide range of industries (Splanemann 2001). Production and production-related operations planners often turn to simulation technologies to assist operators in dealing with complex production requirements particularly in highly constrained environments (Wang and Handschin 1999). Simulation technologies permit the planner/engineer to quickly assess numerous operational alternatives representing useful solution domains that otherwise cannot be "seen" (Contaxis *et al* 2000; Bretthauer *et al* 1998).

One of the most critical production-related decisions facing industrial planners is when to do what. It is well recognized, for example, that production effectiveness is symbiotically dependent on timely maintenance as well as on-time product delivery (Kelleher 1997). The ubiquitous enterprise schedule of events is an important and essential part of industrial activities because they tend to "drive" production systems. Consequently, the usefulness of production-related simulations is inseparably dependent on robust, meaningful production implementation schedules.

Many "optimizing" algorithms have been developed for scheduling essentially all production-related activities including maintenance (Rong-Ceng and Shih 2000; Filipic and Zupanic 1999). There have been little, if any, comparative assessments of the effectiveness of the different classes of algorithms in terms of solution robustness. This paper compares the robustness (as measured by hours) of three simulation optimization "engines" commonly used to assess complex scheduling options applicable to maintenance scheduling.

2 PROBLEM SCOPE AND BACKGROUND

The problem examined in this paper is representative of the scheduling challenges associated with very expensive capital assets that must be periodically maintained over long design lives (*e.g.*, 50 years). One measure of the business robustness of the maintenance schedule is the maintenance costs required to meet all maintenance objectives. Although the total maintenance costs expended over the lifetime of the capital assets is often used as a measure of maintenance robustness, the problem investigated in this paper focuses on minimizing the cost of the **capacity** to perform maintenance work.

The extraordinarily long lifetimes of many expensive capital assets require that maintenance activities be effectively managed over a relatively long time. System maintenance requirements that oscillate widely from time-to-time are not as desirable as those that remain reasonably consistent. This is true because the capacity required to meet peak maintenance equipments can be an underused, costly burden during non-peak periods.

The peak or maximum maintenance requirements is a direct measure of the maintenance capacity that must be in place to perform the required work. Consequently, the best and most efficient maintenance schedule over a many year period is one that minimizes the "peaks" in maintenance capacity requirements over the capital assets lifetimes.

Maintenance capacity costs are directly related to the maximum number of maintenance hours expended in any given time period (annually in this case) assuming that maintenance hours properly reflect facility, material and labor requirements. Hours were chosen as the common analysis variable for ease of comparison between the algorithms of interest.

We analyze three representative classes of algorithmic methods. Specifically, the methods classes are (1) quasi-manual using standard spreadsheet technology, (2) a modern Constraint Programming-based software package, and, (3) a genetic program written specifically for this application. The first method represents a rules-of-thumb approach. Method two represents a recent application of a standard mathematical technique. Method three represents an advanced capability entering the workplace. We assess how well each approach accomplishes the problem goals, i.e., minimizing the peak hours required for upgrades/refurbishment in any year of the 50 year lifetimes of the capital assets. The measure of "goodness" is determined by comparing the peak annual hours calculated by each of the three approaches to meet time-dependent production goals. The best maintenance schedule is one that produces the lowest peak maintenance hours in any year and, thus, has the lowest installed capacity costs.

3 BRIEF PROBLEM SUMMARY

We assign representative characteristics to the maintenance scheduling problem to symbolize actual requirements for large capital assets investments. Table 1 summarizes the capital assets maintainable component characteristics. The first column in Table 1 is the component identification. The second column defines the number of component units by type and the initial fabrication date of the components (shown in parentheses.) The third column lists the estimated range of design lifetimes in integer years. The fourth column shows the number of maintenance work hours required for each individual component.

Table 1: Component/Unit Maintenance Characteristics

Component ID	Number of Units (fab date)	Design Life	Per Unit Work Hours	Packageable ?
P.1	6 (1/75)	20-25	1200	no
P.2	12 (4/75)	26-30	800	yes
P.3	5 (1/74)	35-37	2000	yes
P.4	20 (6/76)	25-28	1080	yes
P.4a	6 (9/74)	21-25	600	yes
P.5	45 (11/74)	17-20	500	yes
P.6	16 (12/75)	45-55	550	no
P.7	2 (1/79)	31-39	1550	yes
P.8	10 (3/75)	15-20	975	yes

The components shown in Table 1 are unique and separable from one another for maintenance purposes. However, component **units** are indistinguishable from one another, have the same fabrication date, and must be maintained at the same maintenance intervals. The number of units by component is specified as part of the problem definition to show that the component workload is functionally dependent on the number of units defining each component. The **total** component work hours for any maintenance schedule is simply the product of the number of units and the per unit work hours.

The final column of Table 1 states the "packageability" of each component. Packageability means that at the time a component is selected for maintenance operations (including all of its units), it may be more efficient to combine it with other component maintenance activities. Efficiency is reflected in reduced total work hours for the "packaged" components/units. These maintenance efficiencies are available to the planner. However, other maintenance constraints such as responsibly managing component/unit lifetimes may preclude packaging opportunities. The acceptable component packaging options are shown in Table 2. For example, the units of component P.4a may be combined with the units of components P.2 and P.4 for maintenance purposes provided all other conditions are met. The fact that the number of component units for P.4a (6 units) differs from P.2 (12 units) and P.4 (20 units) permits scheduling schemes that take advantage of differing unit work hours since work may be performed on the units of the "package" in any order.

If the maintenance planner decides to jointly maintain two components (including all component units) during the same maintenance period, then the number of work hours (the sum of the total work hours for both components) required to perform the "packaged" maintenance operations is reduced by 33%. If three components are maintained at the same time, the sum of the component work hours is reduced by 60%.

Table 2: Permitted Component Packaging

Component	Packagable ?	May be Grouped with
P.1	no	
P.2	yes	(P.3 and P.7) OR (P.4 and P.4a)
P.3	yes	(P.2 and P.7) OR (P.8)
P.4	yes	(P.2 and P.4a) OR (P.8)
P.4a	yes	(P.2 and P.4)
P.5	no	
P.6	no	
P.7	yes	(P.2 and P.3)
P.8	yes	(P.4) OR (P.3)

The maintenance assessment period is 50 years beginning in the year 1988.

4 CONSTRAINTS

The constraints are summarized in Table 3. It is clear from Table 3 that acceptable solutions must assure that (1) component design lifetimes are never exceeded and that (2) maintenance operations before design life is reached are unacceptable. Since the goal is to minimize maintenance costs, it is advantageous to combine component maintenance activities whenever possible as outlined in Table 2.

Table 3: Constraints

• No part may exceed its maximum lifetime before being maintained
• A component may NOT be replaced before its minimum lifetime is reached
• "New" lifetimes are the same as "old" lifetimes
• All components are equally important, *i.e.*, maintenance priority is the same for all components.
• There are 8 work hours per day, 5 work days per week, and 50 work weeks per year.

5 OPTIMIZATION APPROACHES

5.1 Spreadsheet

The spreadsheet approach uses the capabilities of modern spreadsheet technologies to capture and display the rules-of-thumb approach. Asset components are scheduled row by row. Columns are used to represent each year of interest. Columns contain either a "1" or a "0" where

Column N = 1, maintenance scheduled,
Column N = 0, no maintenance.

The spreadsheet compares the proposed maintenance schedule against the constraints (see Table 3) to identify constraint violations and to highlight possible work hour enhancements available through component concurrent maintenance packaging (see Table 2.) A summary table calculates the total number of work hours and the maximum work hours in any year.

5.2 Genetic Programming

Both Genetic Algorithms (Goldberg 1989) and Genetic Programming (Koza 1999) fall under the broad category of evolutionary computation (Heitkotter and Beasley 2002). Genetic Algorithms (GA) and Genetic Programming (GP) are based on the ideas of natural selection and survival of the fittest.

The genetic algorithm (GA) approach is quite simple as follows:

1. Randomly generate an initial solution population
2. Evaluate these solutions for fitness
3. If time or iteration constraints not yet satisfied, then
4. Select parents (best solutions so far)
5. Recombine parents using portions of original solutions
6. Add possible random solution "mutations"
7. Evaluate new solutions for fitness
8. Return to Step 3.

When using Genetic Programming for assessing manufacturing schedules, we are faced with three challenges. They are (1) solution representation, (2) the proper fitness function, and (3) calculation run times.

1. *Solution Representation:* A simple example is f (x) = x^2. Since the goal is to maximize f(x), a solution could be a bit string such as 00011001. This string represents a number with the right most bits being the most significant. As the genetic algorithm runs, solutions with the more 1's in the right most positions yield a great value for f(x) and are selected as parents for the next generation. Representing a manufacturing schedule is much more complex than a simple bit string as represented a significant challenge.
2. *Fitness Function:* The fitness function for the manufacturing analysis is a "good" schedule. Good means cheaper (less work intensive) and one that doesn't violate any of the constraints.
3. *Calculation Run Times:* In some settings, the run times of an evolutionary algorithm can take extraordinarily long times to generate potential solutions. At least for the results reported in this paper, we did not experience run time limitations.

The goal is to generate a maintenance schedule that minimized the required annual work hours. Particular emphasis is placed on minimizing the overall capacity required to refurbish the capital assets. Each potential solution yields a schedule requiring a capacity. The rules, or fitness functions, used are as follows:

> A sum-of-squares algorithm is used to evaluate each potential schedule. Those solutions resulting in higher annual peaks are weighted as "worse" solutions. Relatively better solutions are chosen to become the parents of the next generation of solutions. A combination of checks is used to penalize the breaking of constraints and data structures are used to enforce constraints.

5.3 Constraint Programming (CP) Approach

We use the ILOG Solver (Lustig and Puget 2001) to construct and optimize a model using the component maintenance constraints of Table 1. Expressing the component maintenance constraints as Solver constraints is straightforward--virtually a direct match--and the simulation allows us a primitive understanding of this approach. However, we do not understand how to express the permitted component packaging efficiencies of Table 2 using the subject approach so that remains a challenge.

6 RESULTS

Figure 1 shows the average age of the capital assets by year beginning in year 1988. The average age is the weighted sum of the ages of all the asset's components. Since a number of the components are fabricated before 1988, the age of the assets is non-zero at the beginning of the problem.

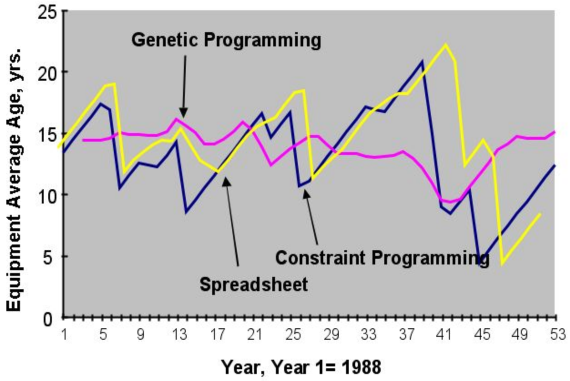

Figure 1: Comparison of Three Approaches

It is striking that the shape of the age-curve for the solution identified by the genetic program (see Figure 1) is significantly different from the age-curves produced by both the spreadsheet and the Constraint Programming (CP) algorithm methods. One would expect the component age profile to reflect the built-in strategy used to manage the maintenance requirements. If this is so, it follows that genetic programs may offer the more flexible option to optimizing constrained scheduling problems.

Figure 2 shows the total number of maintenance hours (over the problem time frame) for each of the approaches evaluated in this paper. We differentiate between total work hours and maximum annual work hours to reflect the difference between the cost of work and the cost of the capacity to do work. Annual work hours are a measure of required maintenance capacity/costs, i.e., typically component/units capacity per year.

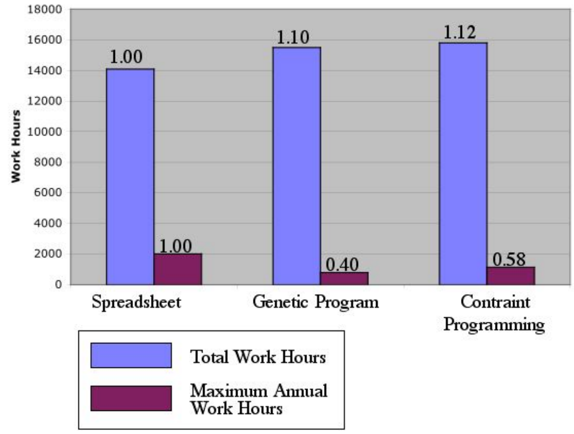

Figure 2: Work Hour Comparison

Our analysis reveals a potential tradeoff between work and work capacity in that minimizing installed capacity (including facilities, equipment and laborers) may result in more total work over a relative long period of time. The problem did not include controls or constraints on total work hours over the 50 years of interest.

The total number of hours expended for maintenance over the time period of interest (50 years) is lowest for the spreadsheet approach (see Figure 2.) It appears that the advanced techniques achieve lower installed capacity numbers, but result in more overall work during the same time period. We posit that the GP approach is unable to take full advantage of the "packaging" opportunities as a result of striving for minimal annual workload levels. The CP approach does not acknowledge component packaging efficiencies as noted in section 5.3 above.

Figure 2 also shows the maximum annual work hours (analogous to installed capacity) for each of the three methods analyzed. The figure shows that the genetic program found a solution producing maintenance workloads that were 60% less than those determined by the spreadsheet approach and 31% less than the CP-based approach. The very significant difference between the genetic program results and those produced by the spreadsheet approach may be attributable to the genetic program's built-in philosophical approach. However, more study is needed to determine the sensitivity of the results, if any, to the philosophical approaches used by three different methods of interest.

REFERENCES

Bretthauer, G., T. Gamaleja, E. Handschin, U. Neumann, and W. Koffman. 1998. Integrated Maintenance Scheduling System for electrical Energy Systems. IEEE Transactions on Power Delivery 13 (2): 655-660.

Contaxis, G.C., C. Delkis, D. Papachirstou, A. Adrachtas. 2000. Fuel Limitations and Environmental Constraints in Probabilistic Production Simulation Models. In *Proceedings of the 10th Mediterranean Electrotech-*

nical Conference (Lemesos, Cyprus, May 29-30). IEEE, Piscataway, NJ, 1039-1042.

Filipic, B. and D. Zupanic. 1999. Near-optimal Scheduling of Line Production With an Evolutionary Algorithm. In Evolutionary Computation Proceedings (Washington, D.C., July 6). IEEE, Piscataway, NJ, 1129.

Goldberg, D. 1989. *Genetic Algorithms*. Reading, MA: Addison-Wesley.

Heitkotter, J. and D. Beasley. 2002. The Hitch-Hiker's Guide to Evolutionary Computation. Available on line via <http://surf.de.uu.net/encore/www> [accessed January 22, 2002].

Kelleher, G. 1997. Scheduling and Rescheduling. In *IEEE Colloquium on Artificial Intelligence in Manufacturing* (London, UK, February 11). IEEE, Piscataway, NJ, 3/1-3/4.

Koza, J. 1999. *Genetic Programming*. Boston, MA: MIT Press.

Lustig, I.J. and J.F. Puget. 2001. Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming. *Interfaces* 31 (6): 29-53.

Rong-Ceng, Leou and Yih Shih-An. 2000. A Flexible Unit Maintenance Scheduling Using Fuzzy 0-1 Integer Programming. In *Power Engineering Society Summer Meeting* (Seattle, WA, July 16). IEEE, Piscataway, NJ, 2551-2555.

Splanemann, R. 2001. Production Simulation: A Strategic Tool to Enable Efficient Production Processes. *Chemical Engineering & Technology* 24 (6): 571-573.

Wang, Y. and E. Handschin. 1999. Unit Maintenance Scheduling in Open Systems Using Genetic Algorithm. In *Transmission and Distribution Conference Proceedings* (New Orleans, LA, April 11). IEEE, Piscataway, NJ, 334-339.

AUTHOR BIOGRAPHIES

TERRY HELM was been a group leader at Los Alamos for 16 years. He earned his B.S. in physics at Brigham Young University and M.S. from the Ohio State University in Nuclear Engineering and has worked in the nuclear industry for over 30 years. At Los Alamos, Terry's group has focused on supporting simulation-based modeling of the Laboratory's nuclear operations. Recently, he has become interested in analysis of decision processes as applied to system accidents and developing aids for better decision-making processes.

STEVEN PAINTER holds a B.S. from Brigham Young University in Design Engineering and M.S. from Rensselaer Polytechnic Institute in Computer Science. He joined Los Alamos National Laboratory in 1980 and has worked for Digital Asset Corporation and The Center for Adaptive Systems Applications. He has been involved in a wide variety of software projects including computer integrated manufacturing, discrete-event simulations, adaptive modeling and systems, and software process improvement.

WILLIAM R. (ROB) OAKES joined Los Alamos National Laboratory in 1977. He holds a B.S. and a M.A. in mathematics from the University of New Mexico. Rob's background includes software development, scientific data acquisition, structural analysis, and complex applied physics analysis. He is interested in manufacturing systems with the goal of waste minimization and disposal optimization technologies.