DEC 12 2000

SANDIA REPORT

SAND2000-2990
Unlimited Release
Printed December 2000

Advanced Production Planning Models

Production Planning for a Project Job Shop, with Application to Disassembly, Evaluation and Maintenance of Nuclear Weapons

Dean A. Jones, Craig R. Lawton, Edwin A. Kjeldgaard, Stephen T. Wright, Mark A. Turnquist, Linda K. Nozick, and George F. List

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

RECEIVED
DEC 22 2000
OSTI

SAND2000-2990
Unlimited Release
Printed December 2000

Advanced Production Planning Models

Production Planning for a Project Job Shop, with Application to Disassembly, Evaluation and Maintenance of Nuclear Weapons

Dean A. Jones, Craig R. Lawton, and Edwin A. Kjeldgaard
Stephen T. Wright
Critical Infrastructure Surety Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0451

Mark A. Turnquist and Linda K. Nozick
School of Civil and Environmental Engineering
Cornell University
Ithaca, NY 14853

George F. List
Department of Civil Engineering
Rensselaer Polytechnic Institute
Troy, NY 12181

Abstract

This report describes the innovative modeling approach developed as a result of a 3-year Laboratory Directed Research and Development project. The overall goal of this project was to provide an effective suite of solvers for advanced production planning at facilities in the nuclear weapons complex (NWC). We focused our development activities on problems related to operations at the DOE's Pantex Plant. These types of scheduling problems appear in many contexts other than Pantex — both within the NWC (e.g., Neutron Generators) and in other commercial manufacturing settings. We successfully developed an innovative and effective solution strategy for these types of problems. We have tested this approach on actual data from Pantex, and from Org. 14000 (Neutron Generator production). This report focuses on the mathematical representation of the modeling approach and presents three representative studies using Pantex data. Results associated with the Neutron Generator facility will be published in a subsequent SAND report. The approach to task-based scheduling described here represents a

significant addition to the literature for large-scale, realistic scheduling problems in a variety of production settings.

Contents

Figures

Tables

Executive Summary

This report describes the innovative modeling approach developed as a result of a 3-year Laboratory Directed Research and Development project. The overall goal of this project was to provide an effective suite of solvers for advanced production planning at facilities in the nuclear weapons complex (NWC). We focused our development activities on problems related to operations at the DOE's Pantex Plant. These types of scheduling problems appear in many contexts other than Pantex – both within the NWC (e.g., Neutron Generators) and in other commercial manufacturing settings. We successfully developed an innovative and effective solution strategy for these types of problems. We have tested this approach on actual data from Pantex, and from Org. 14000 (Neutron Generator production). This report focuses on the mathematical representation of the modeling approach and presents three representative studies using Pantex data. Results associated with the Neutron Generator facility will be published in a subsequent SAND report. The approach to task-based scheduling described here represents a significant addition to the literature for large-scale, realistic scheduling problems in a variety of production settings.

The essence of the resource-constrained, multi-project planning/scheduling problem is to determine when tasks should be scheduled during a given analysis period. The resulting solution must generate two types of outputs, the most important of which is the task schedule. A schedule of resource assignments must also be produced resulting in output on how a set of resources is to be used in a given time period.

One of the most innovative aspects of our approach is to represent the problem using a set of continuous variables, rather than integer variables. In previous solution approaches, integer variables are used to decide whether or not a task begins in a given time period. This results in a mixed-integer programming problem. Our approach formulates the problem differently using actual start times (as continuous variables) to represent task start times.

With this new formulation, we have created a unique way of interpreting and formulating math programs for a class of resource-constrained scheduling problems, resulting in distinct advantages over existing techniques. First, its time periods can have variable lengths. This means that calendar boundaries can be matched precisely, resolving a significant implementation barrier. Second, the principal choice variables are the starting times for each task, rather than using a period index for a start time, as is done in integer programming approaches; we have shown that this allows more efficient solution strategies. It appears that this approach to task-based production scheduling will be a major breakthrough for large-scale, realistic scheduling problems in a variety of production settings resulting in a new generation of advanced production planning models.

Due to the inherent difficulty of this problem, exact procedures are ineffective for typical problems encountered in practice. Therefore, we have focused on heuristic methods to produce approximately optimal solutions. Our solution procedure is based on Generalized Bender's Decomposition (GBD), which decomposes the problem into a master and a sub-problem. In this decomposition scheme, the sub-problem is a linear program that optimizes the assignment of resources to tasks, given start times, while the master problem optimizes over the start times.

This solution procedure is a heuristic because we do not solve the master problem exactly. Testing has shown that this procedure is capable of identifying near optimal solutions quickly.

Introduction

This report describes the innovative modeling approach developed as a result of a 3-year Laboratory Directed Research and Development project. The overall goal of this project was to provide an effective suite of solvers for advanced production planning at facilities in the nuclear weapons complex (NWC). We focused our development activities on problems related to operations at the Department Of Energy's (DOE) Pantex Plant. These types of scheduling problems appear in many contexts other than Pantex – both within the NWC (e.g., Neutron Generators) and in other commercial manufacturing settings.

Production planning in make-to-order operations frequently must focus on the problems of assigning resources for small production lots (often of size one), each of which has specific requirements for a sequence of processing steps on different machines, or by people from different crafts, etc. In these environments, the production planning problem bears considerable resemblance to resource-constrained project scheduling, such as is practiced in construction operations or other task-oriented situations. In fact Morton and Pentico (1993) refer to this type of operation as a "project job shop," to emphasize the connections between the traditional views of job shop scheduling and project scheduling.

As this research evolved, several formulations were developed and tested. Of these, two formulations held the most promise – the ν-variable and the s-variable. The ν-variable formulation is documented in SAND99-2095, "The Pantex Process Model: Formulations of the Evaluation Planning Module." This document is dedicated to the description of the s-variable formulation and the results of testing this formulation on Pantex-related data.

The DOE's Pantex Plant in Amarillo, Texas, where nuclear weapons are evaluated, maintained, and/or dismantled requires a production planning model that must handle a few hundred tasks with no less than 30 types of facilities and approximately one hundred multi-certified (90-100 possible certifications) technicians. We have developed a heuristic method, based on Generalized Benders' Decomposition designed to plan production activities in a "project job shop" environment. The new formulation is a mathematical programming model with a linear objective function and both linear and complementarity constraints. The complementarity constraints make the feasible region non-convex. In the decomposition scheme, the sub-problem is a linear program that optimizes the assignment of resources to tasks, given the start times, while the master problem optimizes over the start times. This method provides approximate solutions to the problem because we use a heuristic procedure to solve the master problem. Ideas developed in this model are transferable to other production planning and project scheduling situations; we are continuing the exploration of these applications.
The Pantex facility is where dismantlement operations, weapon evaluation, and maintenance activities for U.S. nuclear stockpile occur. The evaluation/maintenance activities are the application focus of the methods described in this paper, but it is important to emphasize that both types of activities occur in the same facilities at Pantex. To simplify terminology, we will refer to both evaluation and maintenance activities under the single term, "evaluations."

An evaluation *job* involves partial disassembly of a weapon, one or more tests on components, and in most cases re-assembly and replacement in the stockpile. Each job consists of a set of tasks, many of which have precedence constraints. The tasks vary widely in duration, and may have both earliest feasible start times and latest allowable finish times. Each task must be performed in a facility that meets specific requirements, and requires a crew (usually two people) that is certified to perform that task. In some cases, *parent jobs* can spawn *daughter jobs*, as shown in Figure 1. The daughter job begins only after its enabling task is completed in the parent job.

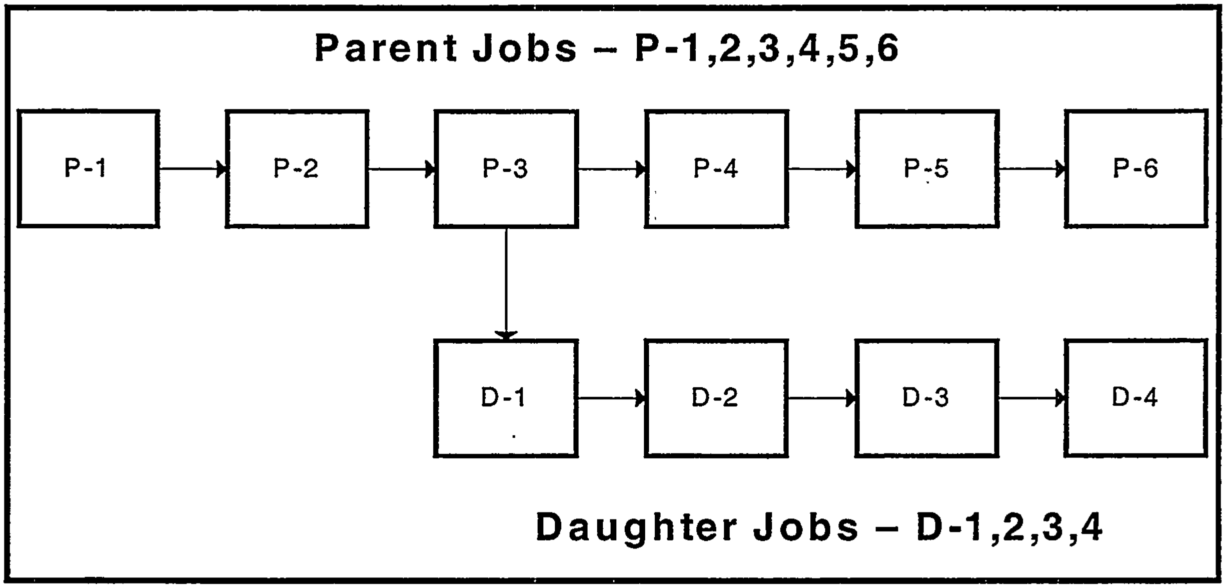

Figure 1. Parent and daughter jobs.

Facilities at Pantex vary in capabilities, and a higher-level facility (i.e., one with greater capabilities) can be substituted for a lower-level facility to perform a specific task. Higher-level facilities are generally more rare than lower-level ones, so such substitution has implicit costs, but it is an available option.

Technicians normally have several different active certifications, so they are partially substitutable. Each task must be assigned to the required number of technicians, all of whom have the required certification, but any given technician is usually able to perform more than one type of task.

Much of the workload is known in advance. That is, there is a set of annual evaluation requirements that specifies that certain jobs must be done during the year; that is, certain numbers of specific weapon types must undergo particular tests, etc. Some of these jobs have very wide time windows, while others are much more restrictive.

The production planning problem is to determine when specific jobs should be scheduled so as to meet all evaluation requirements, be within the given time window constraints, and not over-schedule available resources (facilities and technicians). All else being equal, we would prefer to leave available resources later in the year to account for contingencies and unexpected workload, but the minimization of makespan (in the sense often used in production scheduling) is not necessarily the primary measure of effectiveness. In actual practice at Pantex, the initial specification of the problem is often infeasible (i.e., there is no feasible way to meet the specified deadlines with the available resources). The primary focus of the production planning exercise is determining which deadlines must be negotiated with external customers (e.g., the Department of Defense) in order to reach feasibility.

While some of the details of this application are very specific to the unique nature of operations at Pantex, the general problem of scheduling tasks subject to externally prescribed deadlines, precedence requirements and resource constraints, with varying degrees of substitutability among different resources, is a problem with much broader applicability. Our approach has been to view the "Pantex problem" in a relatively general way, so that the solution procedure developed can be applied in other contexts.

Background

The traditional formulation of the resource-constrained project-scheduling problem is shown in (1)-(7), and is denoted as problem (*P*). This integer programming formulation was originally proposed by Pritsker, et al. (1969), and captures in a compact way the essence of a problem that has been studied by many subsequent authors.

This formulation includes the following decision variables:

C_j	=	completion period for task j ($j = 1, \ldots, J$)
y_{jt}	=	1 if task j completes in time period t ($t = 1, \ldots, T$); 0 otherwise;
x_{jt}	=	1 if task j is active in period t; 0 otherwise

and the following input parameters:

$w_j(Cj)$	=	cost or weight associated with completing task j at time C_j
d_j	=	duration (number of periods) for task j
$P(j)$	=	set of predecessor tasks for task j
r_{jk}	=	amount of resource k consumed by task j in any period in which it is active
M_{kt}	=	amount of resource k available in period t.

$$\min \sum_{j=1}^{J} w_j(C_j) \tag{1}$$

subject to:

$$C_j = \sum_{t=1}^{T} t y_{jt} \quad \forall j \tag{2}$$

$$C_j \geq C_i + d_j \quad \forall i \in P(j), \ \forall j \tag{3}$$

$$x_{jt} = \sum_{u=t}^{t+d_j-1} y_{ju} \quad \forall j, t \tag{4}$$

$$\sum r_{jk} x_{jt} \leq M_{kt} \quad \forall k, t \tag{5}$$

$$\sum_{t=1}^{T} y_{jt} = 1 \quad \forall j \tag{6}$$

$$x_{jt}, y_{jt} \in \{0,1\} \quad \forall j, t \tag{7}$$

It is important to note that the planning horizon is divided into discrete time periods and the key decision variables indicate the time period in which each task is completed (an equivalent formulation can be written using variables that specify the period in which each task *begins*). For consistency of representation, it is necessary that the time periods be of equal length across the planning horizon, and the chosen length of the period determines the level of accuracy in the representation of resource consumption and availability, as well as the duration of tasks. If the problem contains tasks of widely varying duration, it may be necessary to use a large number of very short time periods, because each task must have a duration of at least one period and last an integer number of time periods. This creates computational difficulties, because the number of variables and constraints in the problem is directly related to the number of time periods.

Many authors have developed exact solution procedures for this problem formulation (or minor variations on it), but most of those authors have also pointed out that it is impractical to solve this integer-programming problem exactly for even moderately sized instances. For problems of the size generally experienced in practice, it is necessary to resort to heuristics.

General reviews of efforts to address problem (*P*) can be found in Morton and Pentico (1993), Özdamar and Ulusoy (1995) and Herroelen, *et al.* (1998). In this paper, we will not review all of the voluminous previous work on resource-constrained project scheduling, but we do want to focus attention on two particularly useful ideas from previous work that we have built on in our approach. First, previous authors seeking to develop exact solutions to problem (*P*) have used the idea of decomposing the problem and developing an iterative solution strategy. We also use a decomposition-based strategy, but with a different form of decomposition than has been used previously. Second, in the literature on heuristic solutions for scheduling there is considerable evidence to indicate that stochastic implementations of priority rules are more effective then their deterministic counterparts. We also use this idea as part of our solution procedure.

Weiss (1988) and Deckro, *et al.* (1991) develop decomposition-based solution procedures. Weiss (1988) uses a Danzig-Wolfe decomposition to solve the LP-relaxed version of a resource constrained multi-project scheduling problem. The master problem receives precedence-feasible schedules from the sub-problems and transmits resource prices (by period) to the sub-problems (which are the individual projects, or jobs, in the terminology employed at Pantex). Deckro, *et al.* (1991) developed a Lagrangian relaxation solution procedure. The key idea is that by relaxing the resource constraints, which are the only coupling constraints, the problem can be decomposed into a master problem and a sequence of sub-problems (one sub-problem for each project, or job). The variables in the master problem represent all the precedence feasible schedules for each of the projects. Solving the master problem directly is impractical; so the sub-problems are used to generate candidate feasible solutions (in essentially what is a column generation scheme). The Lagrangian variables are used in the sub-problems to encourage the generation of resource feasible schedules.

Some of t he most effective solution procedures for large problems in the literature are heuristics based on priority rules. Priority-based heuristics develop a schedule by adding activities one at a time to the schedule. A priority rule specifies (for a set of activities that are eligible to be scheduled at a particular point in the algorithm) the one to be placed on the schedule next. The priority values for each task can be based on a number of factors, including task duration, the difference between early and late start times, the number of successor tasks, etc. In multi-pass stochastic priority-based heuristics the priority rules are used to calculate selection probabilities. This means that the order in which tasks are added to the schedule is uncertain. For further discussion of multi-pass stochastic priority rules, see Cooper (1976), Drexl (1991), Drexl and Greenwald (1993), and Kolisch (1995, 1996), among others.

Model Formulation

In order to be useful in practice at Pantex, a production planning model must be able to handle a few hundred tasks (at least), with about 30 different types of facility resources, and 80-90 technicians who hold different combinations of 90-100 different *certifications* (qualifications to perform specific tasks). This problem is far beyond the scope of exact solution algorithms, and the substitution possibilities among facilities and technicians imply that enhancements to the basic formulation of problem (*P*) are necessary.

We have developed an approach that views the problem in a fundamentally different way, and this has opened new opportunities for a more effective solution. One of the principal features to distinguish this new formulation is that it has *continuous* variables s_j which designate the start time for each task j. This means that its time periods can have variable lengths, and that arbitrary time boundaries (daily, weekly, monthly, etc.) within which resource availability is measured can be matched precisely. To do this we use a set of parameters, h_t, that indicate when time period t ($t = 1, 2, \ldots, T$) begins (e.g., the number of working hours, days, etc., since the beginning of the planning horizon, measured as a real, not necessarily integer, value).

To make the connection between the h_t's and s_j's, consider a variable a_{jt}^{+} that represents the time from when task j starts until h_t, if task j starts <u>before</u> h_t, and is zero if task j starts <u>after</u> h_t, as shown in Figure 2. That is,

$$a_{jt}^{+} \quad = \quad \begin{cases} h_t - s_j & \text{if } s_j \le h_t \\ 0 & \text{if } s_j > h_t \end{cases} \qquad (8)$$

Similarly, we will define a_{jt}^{-} to be the time from h_t until task j starts, if task j starts after h_t, and zero if task j starts before h_t:

$$a_{jt}^{-} \quad = \quad \begin{cases} 0 & \text{if } s_j \le h_t \\ s_j - h_t & \text{if } s_j > h_t \end{cases} \qquad (9)$$

Then we can write a pair of equations that describe the relationship between s_j and h_t as follows:

$$s_j = h_t - a_{jt}^+ + a_{jt}^- \qquad \forall \quad t, j \tag{10}$$

$$a_{jt}^+ a_{jt}^- \;=\; 0 \qquad \forall \quad t, j \tag{11}$$

In a completely parallel way, we can define:

b_{jt}^+	=	time from when task j <u>ends</u> until h_t (if $s_j + d_j < h_t$)
b_{jt}^-	=	time from h_t until task j <u>ends</u> (if $h_t < s_j + d_j$)

and create the relationships:

$$s_j + d_j = h_t - b_{jt}^+ + b_{jt}^- \qquad \forall \quad t, j \tag{12}$$

$$b_{jt}^+ b_{jt}^- \;=\; 0 \qquad \forall \quad t, j \tag{13}$$

Figure 2 illustrates the definitions of a_{jt}^+, a_{jt}^-, b_{jt}^+ and b_{jt}^-, and shows three combinations of non-zero values. In the first case, task j starts and ends before h_t so a_{jt}^+ and b_{jt}^+ are both non-zero. In the second, task j starts and ends after h_t so a_{jt}^- and b_{jt}^- are both non-zero. In the third case, the task starts before h_t and ends after h_t, so a_{jt}^+ and b_{jt}^- are non-zero.

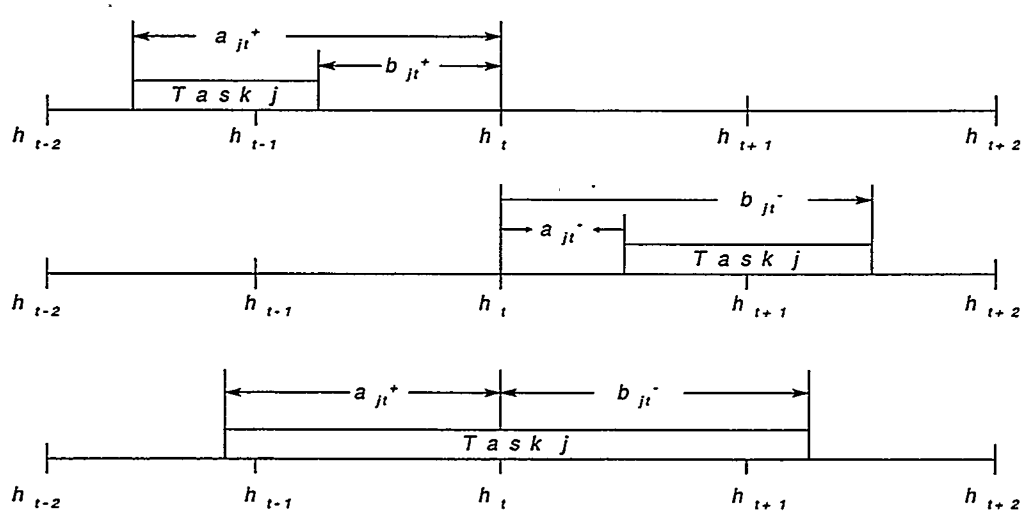

Figure 2. Task timings and time period boundaries.

We also define:

g_{jt}	=	length of activity for task j in time period t
d_j	=	duration of task j.

Note that the definition of d_j as the duration of task j is measured in continuous time, not as a number of periods, so task durations can be represented exactly without having a large number of discrete periods. Similarly, the activity levels for tasks within periods (g_{jt}) are continuous variables, so it is not necessary to assume that if a task is active in a period, it consumes resources for the entire period. The resource consumption can be matched to the portion of the period over which the task is active.

The task start times, s_j, are linked to the task activities by time period, g_{jt}, through the following constraint:

$$g_{jt} = \min(d_j, a^+_{j,t+1} - a^+_{j,t}, b^-_{jt} - b^-_{j,t+1}) \tag{14}$$

If task j begins and ends in time period t then d_j defines g_{jt}. Figure 3 helps illustrate the other two terms in (14).

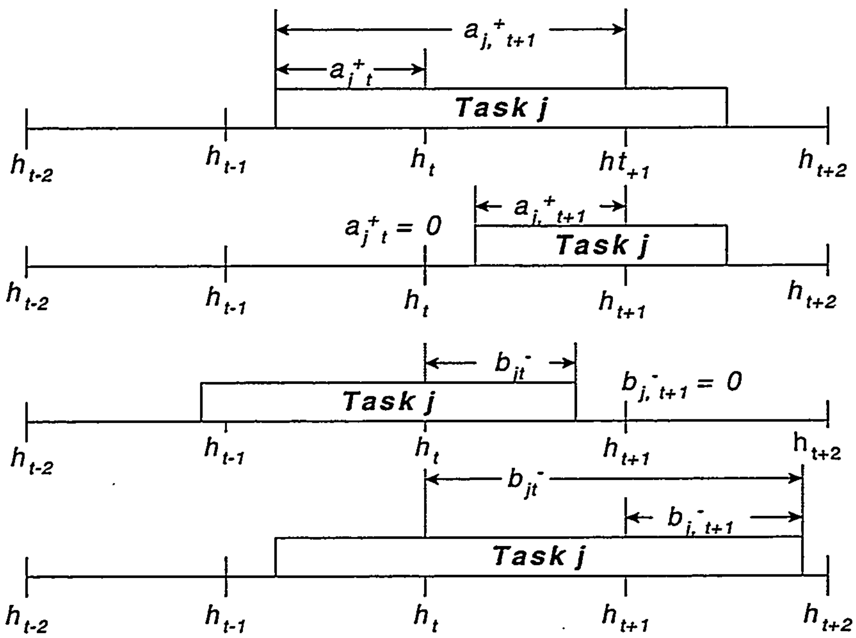

Figure 3. Determining g_{jt}.

In the first case, task j begins before time period t and ends after time period t, so a^+_{jt} and $a^+_{j,t+1}$ will both be positive and the value of $a^+_{j,t+1}$ - a^+_{jt} will be the length of time period t (i.e., h_{t+1} - h_t). In the second case, task j starts during time period t but continues after the end of period t. Now a^+_{jt} is zero, but $a^+_{j,t+1}$ is non-zero. Thus $a^+_{j,t+1}$- a^+_{jt} will equal $a^+_{j,t+1}$, which is the length of time that task j is active in time period t. In the third case, task j starts before time period t and ends during period t. In this instance, b^-_{jt} is non-zero, but $b^-_{j,t+1}$ is zero. Thus, b^-_{jt} -$b^-_{j,t+1}$ will equal b^-_{jt}, the length of time the task is active during time period t. In the last case, task j is active throughout time period t, so b^-_{jt} and $b^-_{j,t+1}$ are both non- zero and the difference, b^-_{jt} -$b^-_{j,t+1}$ is the duration of period t.

Variable substitutions can be used to simplify the formulation. We can solve for a_{jt}^{+} and b_{jt}^{+} in equations (10) and (12), and then substitute into (11) and (13) to produce the following:

$$a_{jt}^{-}(h_t - s_j + a_{jt}^{-}) = 0 \qquad \forall\ j,t \tag{15}$$

$$b_{jt}^{-}(h_t - s_j - d_j + b_{jt}^{-}) = 0 \qquad \forall\ j,t \tag{16}$$

We also know that:

$$s_j - a_{jt}^{-} \le h_t \qquad \forall \quad j,t \tag{17}$$

$$s_j - b_{jt}^{-} \le h_t - d_j \qquad \forall \quad j,t \tag{18}$$

and if we substitute into (14), we can write inequalities related to g_{jt}:

$$g_{jt} + a_{jt}^{-} - a_{j,t+1}^{-} \le h_{t+1} - h_t \qquad \forall \quad j,t \tag{19}$$

$$g_{jt} - b_{jt}^{-} + b_{j,t+1}^{-} \le 0 \qquad \forall \quad j,t \tag{20}$$

The need for variables a_{jt}^{+} and b_{jt}^{+} is thus eliminated. In what follows, we will simplify the notation and use a_{jt} and b_{jt} instead of a_{jt}^{-} and b_{jt}^{-} as the remaining variables in the model. In addition to (19) and (20), we know that the g_{jt}'s sum to the duration of the task:

$$\sum_t g_{jt} = d_j \qquad \forall \quad j \tag{21}$$

Expressions (15)-(21) define the relationships that must exist between s_j, a_{jt}, b_{jt} and g_{jt}.

We must also constrain each s_j to be between the earliest possible start time for task j, e_j and the latest possible start time f_j - d_j (i.e., the latest possible finish time, f_j, minus the task duration):

$$e_j \le s_j \le f_j - d_j \qquad \forall \quad j \tag{22}$$

We also ensure that the precedence relationships among the tasks are met:

$$s_k + d_k \le s_j \qquad \forall \quad k \in P(j), \quad \forall\ j \tag{23}$$

where $P(j)$ is the set of all predecessor tasks to task j.

The task activity variables, g_{jt}, determine the demand for resources. Generically, we will assume that tasks require resources of two types: facilities and people. In the application to evaluation planning at Pantex, this categorization of resources is very clear, and we have used the specific term "technician" to represent the personnel resources. In other applications, the facilities may include equipment or tools as well as physical space. In what follows, we will continue to use the terms facilities and technicians, consistent with the Pantex application, but recognizing that in other applications, different specific terms may be more appropriate.

Resource use is tied to task activity through constraints for facility and technician utilization. In the case of the facilities, we have:

$$\sum_{j \in J_k} g_{jt} = \sum_{i \in F_k} x_{ikt} + R_{kt} \quad \forall \quad k,t \tag{24}$$

$$\sum_{k} x_{ikt} \leq M_{it} \quad \forall \quad i,t \tag{25}$$

where:

k	=	facility configuration index
i	=	facility type index
J_k	=	set of tasks that require facility configuration k
F_k	=	set of facility types that can be configured as type k
x_{ikt}	=	time of facility type i used in configuration k in time period t
R_{kt}	=	demand shortage for facility configuration k in time period t
M_{it}	=	hours of facility type i available in time period t

For the technicians and certifications, we have a similar pair of constraints:

$$\sum_{j \in \Gamma_c} n_j g_{jt} = \sum_{e \in E_c} y_{ect} + Q_{ct} \quad \forall \quad c,t \tag{26}$$

$$\sum_{c} y_{ect} \leq H_{et} \quad \forall \quad e,t \tag{27}$$

where:

c	=	certification index
e	=	technician index
Γ_c	=	set of tasks that require certification c
E_c	=	set of technicians that hold certification c
n_j	=	number of technicians required for task j
y_{ect}	=	time that technician e uses certification c in time period t
Q_{ct}	=	demand shortage for certification c in time period t
H_{et}	=	availability of technician e in time period t

The objective is to minimize:

$$z = \sum_{i,k,t} \phi_{ik} x_{ikt} + \theta \sum_{k,t} R_{kt} + \Omega \sum_{c,t} Q_{ct} \tag{28}$$

The three terms impose penalties for:

- substituting facilities in subordinate configurations (using a facility of type i in configuration k has an associated penalty of ϕ_{ik}, which may be either zero or positive),
- using "pseudo-facilities" (reflecting facility shortages), and
- using "pseudo-technicians" (reflecting certification shortages).

This objective reflects the primary focus on finding a feasible way to meet externally imposed deadlines and workload using available resources. The weighting parameters (θ and Ω) on the "pseudo-resource" variables in the objective function are intended to be large, relative to ϕ_{ik}. In a solution to the problem, non-zero values of R_{kt} and/or Q_{ct} indicate shortages (infeasibilities) that must be resolved by Pantex management. If desired, a term can be added to the objective function to reflect makespan in completing all the jobs.

The overall problem formulation (which we will denote *P1*) is then: *minimize* (28) subject to (15)-(27) and non-negativity conditions on all variables. Except for (15) and (16), this is a linear program (LP). However, the complementarity conditions (15) and (16) make the feasible region non-convex, and complicate the process of finding a solution. Fukushima, *et al.* (1998) studied solution procedures for LP's with complementarity restrictions and developed a theoretical approach for solution, but their approach is computationally prohibitive for large problem instances, and we have therefore developed a different strategy.

Solution Procedure

Although problem (*P1*) has a non-convex feasible region, we can take advantage of the fact that most of the constraints are linear to our advantage in constructing a solution strategy. Our solution approach is based on Generalized Benders' Decomposition (GBD) as discussed in Geoffrion (1972).

The key idea of GBD is that some problems have a structure that allows identification of a set of complicating variables. If these complicating variables are temporarily fixed, the remaining problem is relatively easy to solve. We can then decompose the problem into one part where we search for a solution in the complicating variables (the master problem), and another part where we optimize over the non-complicating variables (the sub-problem), given values for the complicating variables. In each iteration, the algorithm produces an upper bound (from the sub-problem) and a lower bound (from the master problem) to the solution. The algorithm terminates when there is no reduction in the upper bound after a given number of iterations or when the upper and lower bounds are sufficiently close.

We use the task starting times, s_j, as the complicating variables. Implicitly, the a_{jt}, b_{jt} and g_{jt} variables are also part of the complicating set, but they are linked to the s_j variables, so that once the task start times are fixed, the a_{jt}, b_{jt} and g_{jt} values can be computed directly.

The sub-problem (for fixed values of the s_j's) is then as given in equations (29)-(34). It is simply an LP that assigns resources (facilities and technicians) to meet demands implied by the fixed task schedule. We have added the third term to the objective given in equation (29) to prevent the model from over-assigning resources to tasks. The values of g_{jt} (the activity levels for tasks, by time period) are fixed by having fixed the task start times, so the right-hand-side values in constraints (30) and (32) are known. The superscript l is used to signify the specific values of s_j and g_{jt} for the l^{th} iteration of the solution process. The variables μ_{kt} and κ_{ct} signify the dual variables associated with constraints (30) and (32), respectively.

$$\min z = \sum_{i,k,t} \phi_{kt} x_{ikt} + \theta \sum_{k,t} R_{kt} + \gamma \sum_{e,c,t} y_{ect} + \Omega \sum_{c,t} Q_{ct} \qquad (29)$$

s.t.

$$\sum_{i\in F_k} x_{ikt} + R_{kt} = \sum_{j\in J_k} g^l_{jt} \qquad \forall \quad k,t \qquad [\mu_{kt}] \tag{30}$$

$$\sum_{k} x_{ikt} \le M_{it} \qquad \forall \quad i,t \tag{31}$$

$$\sum_{i\in F_k} y_{ect} + Q_{ct} = \sum_{j\in \Gamma_c} n_j g^l_{jt} \qquad \forall \quad c,t \qquad [\kappa_{ct}] \tag{32}$$

$$\sum_{c} y_{ect} \le H_{et} \qquad \forall \quad e,t \tag{33}$$

$$x_{ikt}, y_{ect}, R_{kt}, Q_{ct} \ge 0 \qquad \forall\ j,t \tag{34}$$

The master problem is as given in equations (35)-(46).

$$\min \quad \nu_B \tag{35}$$

s.t.

$$\nu_B \ \ge \ \sum_{i,k,t} \phi_{ik} x^l_{ikt} + \theta \sum_{k,t} R^l_{kt} + \gamma \sum_{e,c,t} y^l_{ect} + \Omega \sum_{c,t} Q^l_{ct} + \sum_{kt} \mu^l_{kt} \left(\sum_{j\in J_k} g_{jt} - \sum_{i\in F_k} x^l_{ikt} - R^l_{kt} \right) + \sum_{ct} \kappa^l_{ct} \left(\sum_{j\in\Gamma_c} n_j g_{jt} - \sum_{e\in E_c} y^l_{ect} - Q^l_{ct} \right) \qquad \forall\, l = 1,2..,L \tag{36}$$

$$e_j \le s_j \le f_j - d_j \qquad \forall \quad j \tag{37}$$

$$\mathrm{s_k + d_k \le s_j \qquad \forall \quad j,k \ where\, k \in P_j} \tag{38}$$

$$\mathrm{s_j - a_{jt} \le h_t \qquad \forall \quad j,t} \tag{39}$$

$$\mathrm{s_j - b_{jt} \le h_t - d_j \qquad \forall \quad j,t} \tag{40}$$

$$\mathrm{g_{jt} + a_{jt} - a_{j,t+1} \le h_{t+1} - h_t \qquad \forall \quad j,t} \tag{41}$$

$$\mathrm{g_{jt} - b_{jt} + b_{j,t+1} \le 0 \qquad \forall \quad j,t} \tag{42}$$

$$\sum_t g_{jt} = d_j \qquad \forall \quad j \tag{43}$$

$$a_{jt}(h_t - s_j + a_{jt}) = 0 \qquad \forall\ j,t \tag{44}$$

$$b_{jt}(h_t - s_j - d_j + b_{jt}) = 0 \qquad \forall\ j,t \tag{45}$$

$$s_j,\ g_{jt}, a_{jt}, b_{jt} \ge 0 \quad \forall\ j,t \tag{46}$$

The master problem has no resource constraints, but it does have "prices" on resources (the κ and μ values) from the cuts produced by solving the sub-problem at each iteration.

The master problem still has the nonlinear constraints (44) and (45) that made the original problem difficult, but we can construct an approximate solution to the master problem with a relatively straightforward algorithm. The essential idea behind the solution procedure to the master problem is that at any iteration we have a series of constraints that are of the general form given in (47):

$$\nu_B \geq K^l + \sum_{kt} \mu_{kt}^l \left(\sum_{j \in J_k} g_{jt} \right) + \sum_{ct} \kappa_{ct}^l \left(\sum_{j \in \Gamma_c} n_j g_{jt} \right) \qquad \forall\, l = 1,2..,L \tag{47}$$

where K^l is a constant defined by equation (48). From (47) it can be observed that for a particular task we can calculate the incremental impact on ν_B of each feasible start time.

$$K^l = \sum_{i,k,t} \phi_{ik} x_{ikt}^l + \theta \sum_{k,t} R_{kt}^l + \gamma \sum_{e,c,t} y_{ect}^l + \Omega \sum_{c,t} Q_{ct}^l - \sum_{kt} \mu_{kt}^l \left(\sum_{i \in F_k} x_{ikt}^l + R_{kt}^l \right) - \sum_{ct} \kappa_{ct}^l \left(\sum_{e \in E_c} y_{ect}^l + Q_{ct}^l \right) \tag{48}$$

The heuristic for the master problem must produce start times for which ν_B is small. This can be done by ordering the tasks according to some metric that indicates the sensitivity of the solution to the placement of the task in the schedule. Tasks can then be placed in the schedule one at a time so that each incremental impact on ν_B is as small as possible. As each task is scheduled, the earliest possible start times and latest allowable finish times for the remaining tasks must be updated. In actuality, it is very difficult to construct a single metric to order the tasks.

Therefore, the approach taken is to define a robust metric and use that to generate multiple sets of task orderings. Each list will be similar in that the probability that a task appears either near the top or the bottom is a function of the value of the metric for that task, but variations in the exact ordering of the tasks will occur. For each list of orderings the master problem scheduling heuristic is run. The solution to the master at the end of the iteration is that set of start times for which the lowest value ν_B results.

Figure 4 illustrates the solution procedure for the master problem. Suppose there are J tasks. Further, suppose that we will generate N^* schedules to estimate the solution to the master problem in each iteration. The first step in generating a schedule, N, is to calculate for each task, j, the importance measure, I_j, which equals

$$I_j = (\beta_k + \xi_c n_j) d_j \tag{49}$$

where $$\beta_k = \sum_t \mu_{kt}^l \tag{50}$$

and $$\xi_c = \sum_t \kappa_{ct}^l . \tag{51}$$

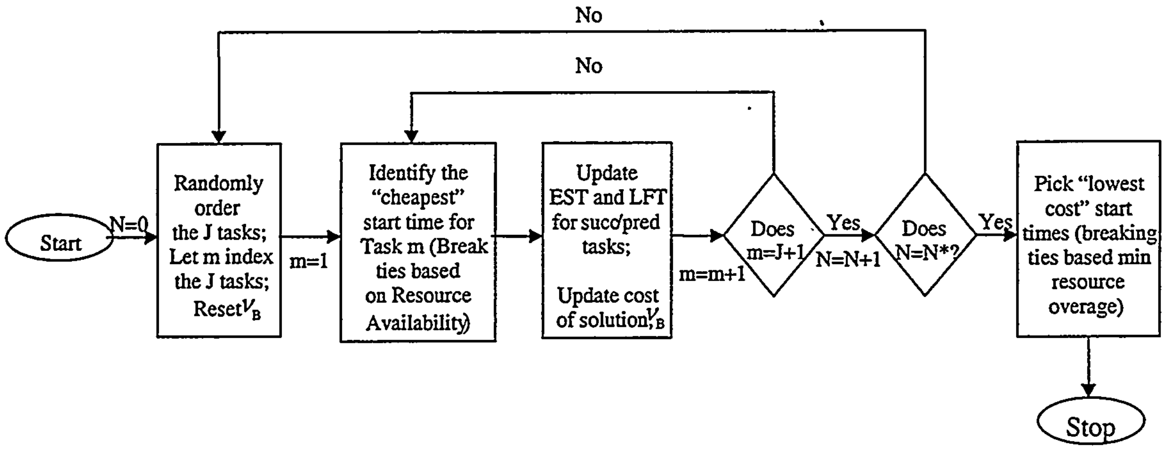

Figure 4. Solution procedure for the master problem.

The importance measure is an estimate of the cost of the resources required by the task. Based on the importance measure for each task, we then calculate the probability of selection for each task (to be placed on the ordered list) from equation (52).

$$p_j = \frac{\log(I_j + offset)}{\sum_i \log(I_i + offset)} \tag{52}$$

The *offset* is a user-defined parameter. If the offset is large relative to the values of I_j, the effect of the importance factor is reduced, and the ordering of the tasks becomes more random. If the *offset* is small, the task ordering is less random, and is more dictated by the I_j values.
Once a task has been selected and placed on the ordered list, its probability of selection p_j and its importance factor I_j are set to zero, and the selection probabilities for the remaining tasks are recalculated based on equation (52).

Next, the tasks on the ordered list are placed in the schedule in the order in which they appear in the list. When a task is to be placed in the schedule there may be multiple starting times for which the incremental effect on V_B is the smallest. In order to break ties, the time which is least constrained in terms of the needed resources is selected.

Once a task has been placed on the schedule, the earliest start and latest finish times for predecessor and successor tasks are updated. Once all the tasks have been scheduled, we have a value for V_B and an estimate of the resource overages for that random ordering of tasks. Once this process has been done N* times we have N* choices for solutions to the master, each of which has a value for V_B and an estimate of the resource overages associated with that schedule. The chosen approximate solution to the master is then the one with the lowest value of V_B . If there are multiple solutions with the same value for V_B the one with the lowest estimated amount of resource overages is selected.

The overall solution procedure alternates between solving a standard LP (the sub-problem) and executing the algorithm for the master problem. Each LP solution generates a new constraint in the master problem. In an exact implementation of GBD the algorithm terminates when the upper and lower bounds are deemed sufficiently close. In practice there is often a secondary stopping criteria, which is to terminate when a given number of iterations have elapsed with no reduction in the upper bound. Since our solution procedure to the master problem yields an approximate solution only, we use the secondary criterion (no improvement in a given number of iterations) as a stopping rule.

Because we are solving the master problem using an approximate algorithm, we may not obtain an exact solution, and hence may not have a true lower bound on the original problem. Thus, we will not be able to guarantee an optimal solution to the original problem, and we must view this procedure as a heuristic.

However, it is important to note that the approximate solution to the master problem is generated basically through a sorting process, so that heuristic runs very fast. The sub-problem LP is relatively easy to solve, and from iteration to iteration the only change is in the right-hand-side, so a "warm start" from the previous iteration solution is possible, making this a very fast process also. Thus, on the whole, the procedure offers a very rapid method of finding an approximately optimal solution to problem *P1* – rapid enough to offer the possibility of solving very large practical problems.

Studies

Throughout the course of this research, numerous studies have been performed to determine the effectiveness of the solution procedures. We present three such studies in this document:

1. A small-scale example designed to highlight the nature of the production planning context at Pantex and show how the model creates an effective scheduling solution.
2. A larger-scale example that illustrates how the model can be used effectively to identify the source of an infeasibility in the production requirements.
3. An example where we have extended the formulation and Bender's Decomposition solution procedure for *s*-variable to include makespan.

Study 1

The first study is a small-scale example consisting of six jobs that include 14 tasks, as shown in Figure 5. Although these particular jobs are hypothetical, this mixture of single-task jobs and multi-task sequences is typical of activities at Pantex.

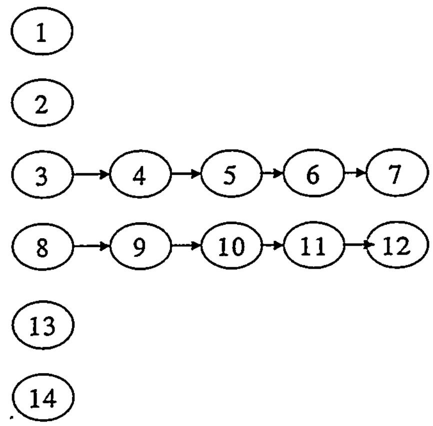

Figure 5. Study 1 Jobs and Task Precedence.

The 14 tasks require three different facility types and five different certifications. Table 1 gives the facility, crew size and certifications needed, in addition to the duration, earliest possible start time (EST) and latest allowable finish time (LFT) for each task, measured in hours. The tasks vary widely in duration, from a minimum of 3 hours to a maximum of 560 hours, typical of tasks in the Pantex plant. The job that includes tasks 3-7 has an earliest start time of $t = 220$ hours and a required completion time of $t = 600$ hours. These two limits plus the precedence relationships among the tasks dictate the available time windows for each of the five tasks. A similar structure, with a different required finish time, applies to the job that includes tasks 8-12. We assume for this example that there are three facilities available, one of each type, and that no facility substitution is possible. We also assume that there are five technicians, with varying certifications. Table 2 lists the certifications for each technician. Notice that, with the exception of certification 5, there are two technicians with each certification.

Table 1. Study 1 Task data

Task	Duration (hrs)	Facility Type	Cert.	# People	EST (hrs)	LFT (hrs)
1	560	1	1	2	0	1160
2	440	1	2	2	0	1160
3	14.4	2	3	2	220	534.4
4	3	3	5	1	234.4	537.4
5	4	3	5	1	237.4	541.4
6	55.6	2	3	2	241.4	597
7	3	3	5	1	297	600
8	14.4	2	3	2	220	734.4
9	3	3	5	1	234.4	737.4
10	4	3	5	1	237.4	741.4
11	55.6	2	3	2	241.4	797
12	3	3	5	1	297	800
13	40	1	4	2	100	520
14	40	1	4	2	0	1200

Table 2 – Study 1 Technician Data

Technician	Certification
1	1,4
2	1,2
3	2,3
4	3,4
5	5

Two features of this problem make it challenging. First, tasks 1 and 2 are very long tasks, each requiring facility 1. Tasks 13 and 14 are shorter tasks that also require facility 1. The solution must sequence these four tasks in a way that meets the feasible time window constraints, especially for task 13. Second, the technician resources are quite constrained. For instance, task 2, which requires two technicians with certification 2, cannot overlap with any task that requires certification 3, because technician 3 would be required for both of those tasks. Similarly, technician 4 is a common resource required by tasks that use certifications 3 or 4.

It is possible to schedule these tasks so that there are no shortages of technicians or facilities. The algorithm was initialized with the start time for each task equal to its EST. This solution generates facility and technician shortages of 585 and 724 hours, respectively. Much of this is due to the initial overlap of tasks 1, 2, 13 and 14.

Figure 6 illustrates the solution produced on the 11^{th} iteration. This solution is resource feasible. It is possible, with the resources specified, to implement this solution with no shortages. Figure 7 illustrates the facility and technician shortages by iteration. The resource shortages, reflecting the values of the objective function, do not decline monotonically as the algorithm proceeds. This is true of GBD in general, and the behavior exhibits itself in this example.

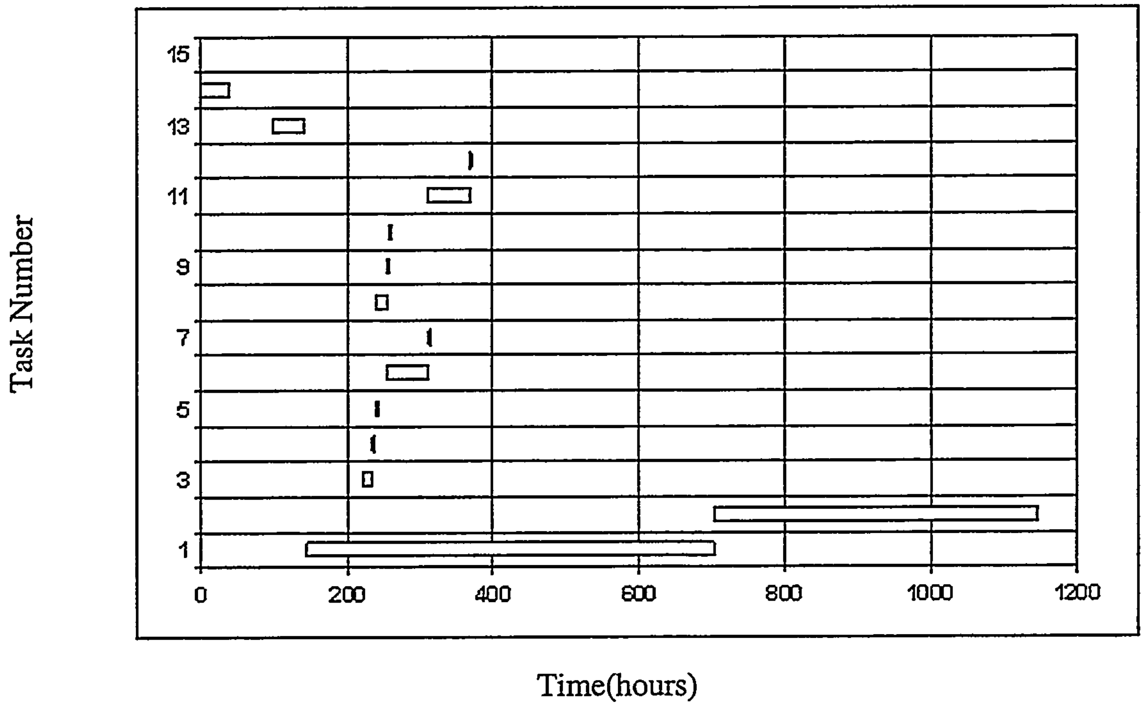

Figure 6. Study 1 Solution Gantt chart.

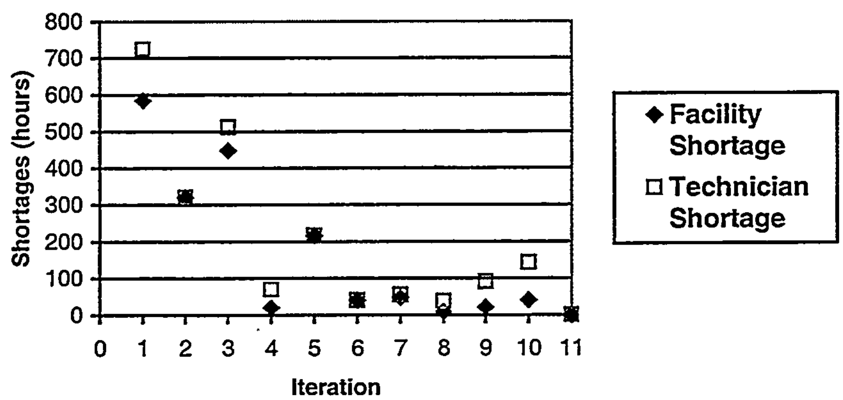

Figure 7. Study 1 Resource shortages by iteration.

Study 2

The second study is an example utilizing a larger dataset drawn from actual operations at Pantex. This example has 130 tasks to be scheduled over a planning horizon of one year. The tasks each require one of eight facility types and technicians who have one of 11 certifications. The resources available include 25 different individual facilities (across the eight types) and 47 technicians. There are 77 active technician-certification combinations, so on average, each technician has about 1.6 available certifications.

We have performed three experiments with this dataset. The first experiment was to identify a resource feasible schedule. As will be discussed, this is not possible because there is simply not enough time available in facilities of type two. In fact a simple inspection of the input data is sufficient to prove this. In the remaining experiments we relaxed the deadlines and push off some of the tasks that require this facility type. Through a simple inspection of the input data it is no longer obvious that there is not a feasible solution.

Figure 8 shows the facility and technician shortages for 30 iterations of the algorithm. The initial condition specified is all tasks starting at their EST values. In the first few iterations, the algorithm resolves the technician shortages successfully, and the facility shortage value falls to a little under 1000 hours, but progress then slows considerably. The best solution achieved is at iteration 25, with no technician shortage and 720 hours of facility shortage.

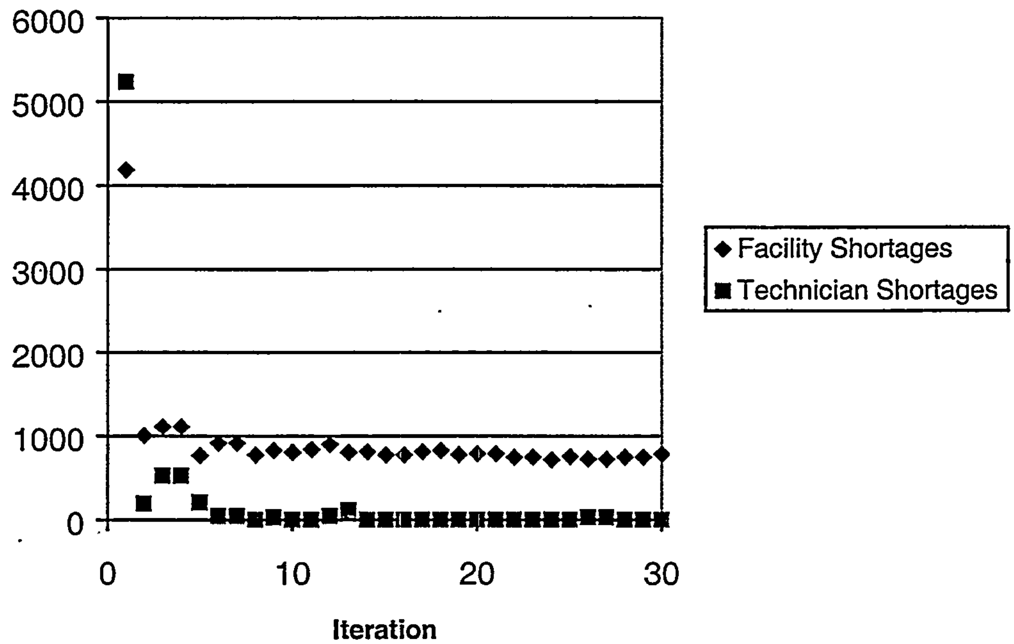

Figure 8. Study 2 Resource shortages by iteration.

Examination of the variables in this solution points to facility type 2 as the resource constraint that cannot be resolved. There are two individual facilities of type 2, and no substitution of other facility types is allowed for facility type 2. Thus, in the first u hours after the beginning of the analysis period, the maximum availability of type 2 facilities is $2u$ hours.

Of the 130 tasks, 43 of the 130 tasks require a type 2 facility, and many of them have latest allowable finish times (LFT values) that are early in the year. In fact, further analysis of these 43 tasks shows that there can be no feasible solution to the scheduling problem. By sorting these tasks in ascending order according to LFT, we can compute the total amount of work that must be accomplished in facility type 2 before time u ($0 \leq u \leq 2000$ hours). We can then compare this value with the total hours available ($2u$) up to time u. Figure 9 summarizes the results of this analysis. The key point to note from Figure 9 is that at $u = 784$ hours, there is a deficit of 712 hours of facility 2 time. Thus, without even considering other constraints on task scheduling, we can see that the best solution obtainable will have at least 712 hours of facility shortage. The solution obtained by the algorithm (720 hours of shortage) is thus likely to be very nearly the best solution available.

This points to a need for management to address the basic shortage of type 2 facilities, either by providing more facility availability (e.g., by planned overtime operation or an extra shift), or by negotiating a more relaxed set of required completion times for some tasks. In this case, Figure 9 also indicates that facility type 2 is likely to present difficulties even if some LFT values are relaxed, because it has such high utilization over the entire year. Over the year ($u = 2000$ hours), facility type 2 has only about 300 hours more available than are demanded (approximately 94% utilization). Thus, the analysis indicates the importance of finding a way to increase the amount of facility type 2 time available, or to decrease the overall required workload.

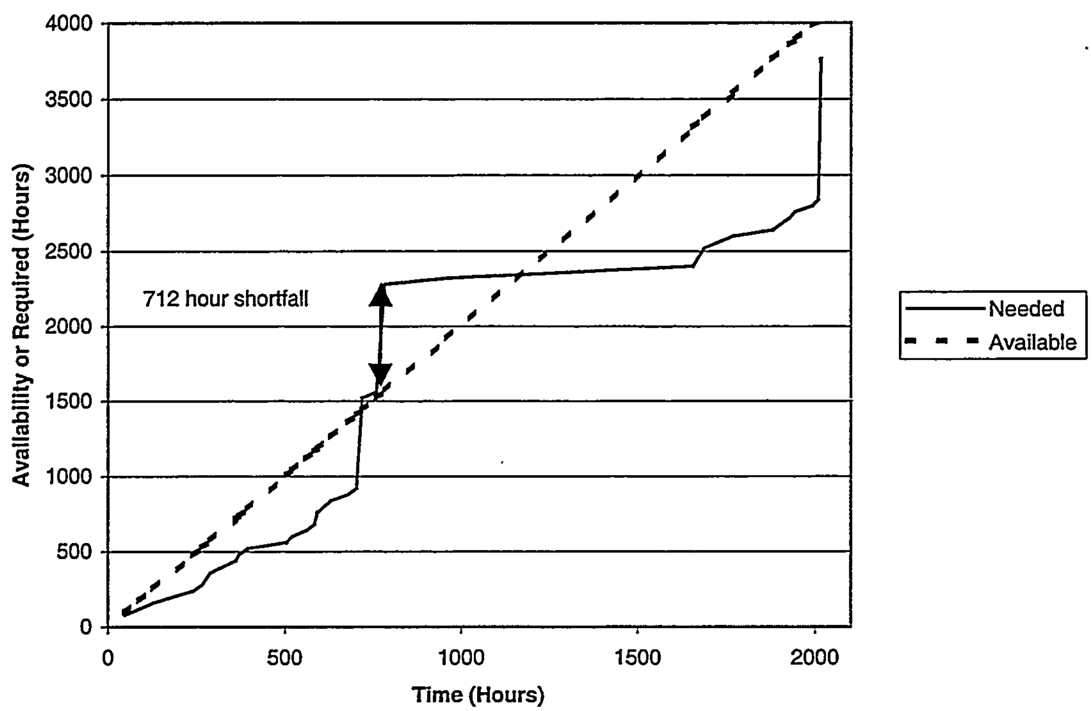

Figure 9. Study 2 analysis of aggregate demand and supply for facility type 2

Suppose we loosen up the deadlines on the work that requires facilities of type 2 and remove some of the workload rather then simply removing 720 hours of work that requires facilities of type 2 early in the year. In this experiment, the utilization of facility type 2 over the planning horizon has been reduced to about 86% (from approximately 94%). Two large tasks and a few smaller tasks have been pushed off beyond the planning horizon and some of the work which was originally required to be completed early in the year is now due later in the year. The reduction in utilization from 94% to 86% corresponds to the delay of about 320 hours of workload that requires facilities of type 2 which has been pushed off beyond the planning horizon. A simple inspection of the due dates for the workload, which requires facilities of type 2, no longer yields a clear understanding that the workload can not be accomplished with the existing resources. It might be possible to accomplish the workload but, as a result of more complicated interactions between the needs of the different tasks, there still may not be sufficient resources available to accomplish the workload.

Figure 10 shows the facility and technician shortages for 45 iterations of the algorithm. Again, the initial condition specified is all tasks starting at their EST values. Notice that in the first few iterations, the algorithm again resolves the technician shortages successfully, and the facility shortages value falls to about 200. The solution produced at iteration 9 is the best solution produced with 72 hours of facility overage. Over the next 36 iterations 5 more solutions are produced with facility overages of less then 100 hours. Based on an inspection of the data, the lower bound on the solution is still that there is a resource feasible solution so the algorithm has produced a solution, which is 72 hours above that lower bound.

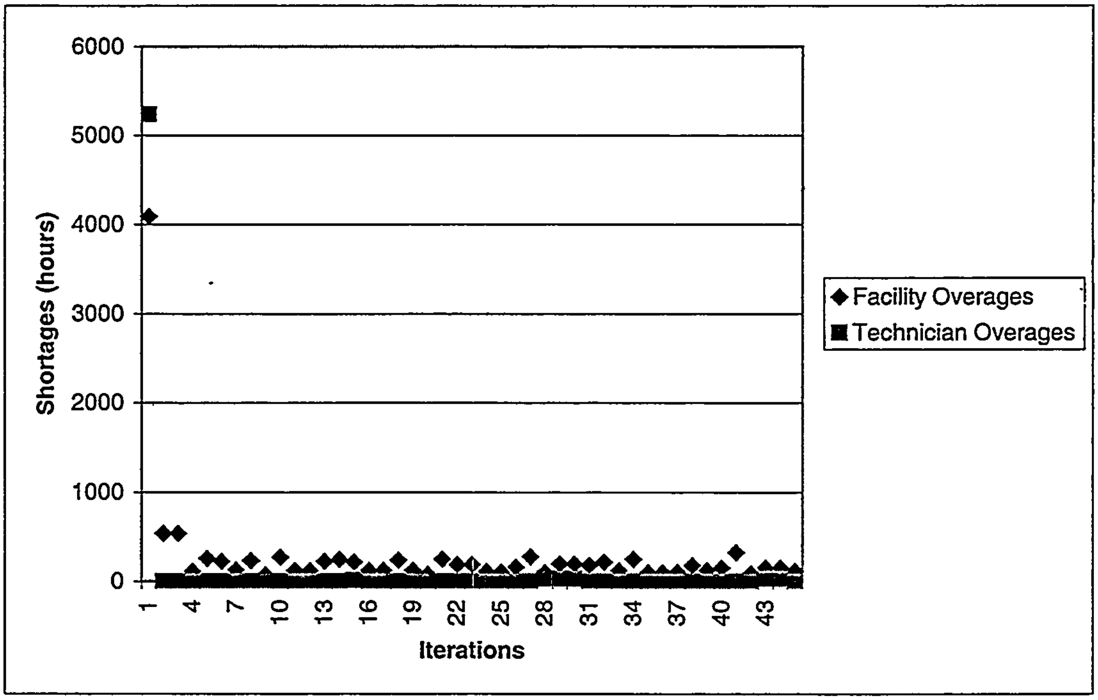

Figure 10. Study 2 Resource overages by iteration.

In an attempt to identify a resource feasible dataset the utilization on facilities of type 2 over the entire planning horizon has been reduced to about 80%. Additional reductions in the workload for the year that require facilities of type 2 have been made. The reduction in utilization from 94% to 80% corresponds to the removal of about 550 hours of workload that required type 2 facilities.

Figure 11 shows the facility and technician shortages for 27 iterations of the algorithm. Again, in the first few iterations, the algorithm again resolves the technician shortages successfully, and the facility shortages value falls to a little under 200 hours quickly. At iteration 27 a resource feasible solution is identified.

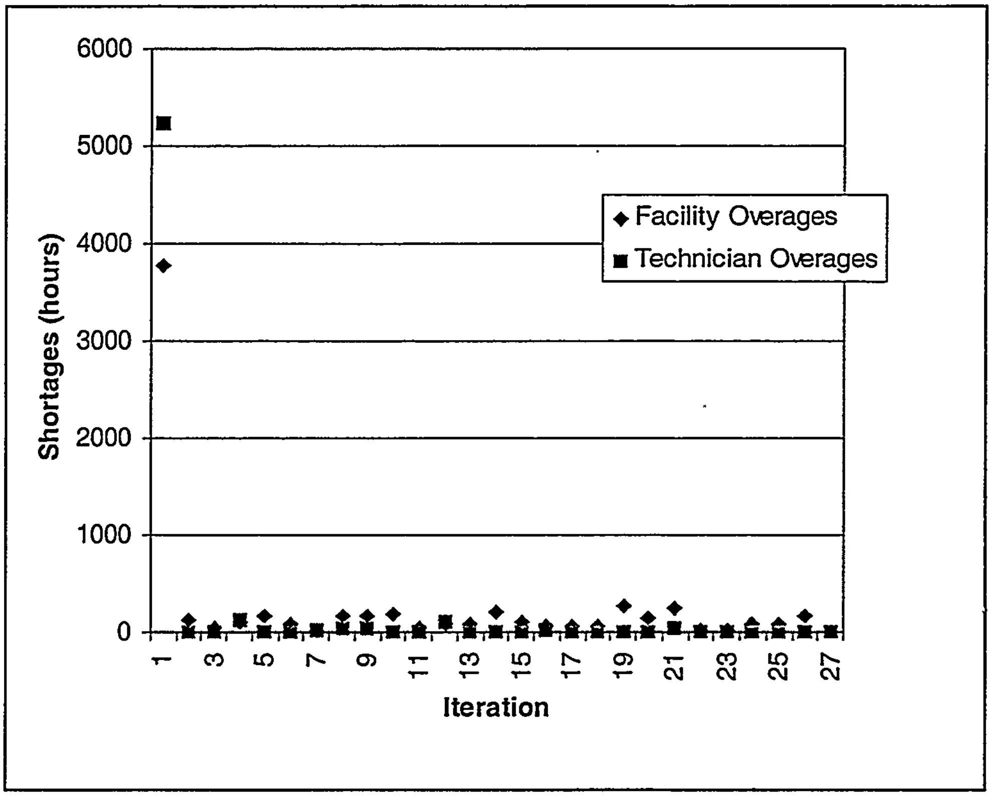

Figure 11. Study 2 Resource shortages by iteration.

Taken together, these two examples illustrate the value of the model in both finding good production plans when there are feasible solutions, and in identifying resource shortages that cannot be resolved by scheduling alone. The last example has illustrated how the model can be used to identify a useful trade-off between workload reductions and deadline extensions. These shortages must be addressed by management attention to reducing the overall workload, modifying the externally imposed scheduling constraints, or increasing the resource availability.

Study 3

There are many applications of the resource constrained project scheduling problem for which the key objective is to minimize the makespan of the collection of jobs. Therefore, we have extended the formulation and Bender's Decomposition solution procedure for *s*-variable to include makespan. The key change to the solution procedure is that when we are solving the master problem, if a potential start time for a task will cause the earliest start time for the terminal task of a job to slip, that start time is penalized by a user defined constant (α) multiplied by the length of the resulting slip. Notice that this will not cause any penalty to be added for start times that do not force the task onto the critical path of a job.

We have tested the algorithm on a small test problem consisting of 18 tasks across 6 jobs shown in Figure 12. Table 3 gives the task duration, facility and technician requirements, ESTs and LFTs. Table 4 gives the technician certifications. In this example there are 3 facilities of type 1 and 2 of type 2.

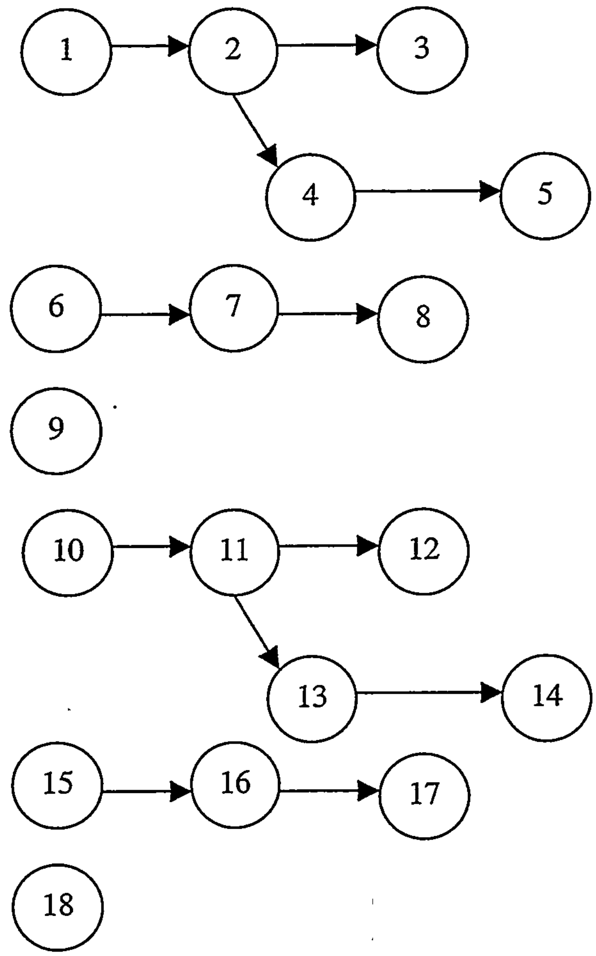

Figure 12. Study 3 Project Network

Table 3 – Study 3 Task Attributes

Task	Duration (hrs)	Facility	Cert	# People	EST	LFT
1	10	1	1	2	0	111
2	20	2	2	2	10	131
3	50	1	3	2	30	231
4	20	1	1	2	30	151
5	80	2	2	2	50	231
6	40	2	2	2	0	183
7	8	1	3	2	40	191
8	40	2	1	2	48	231
9	100	1	2	2	0	231
10	10	1	1	2	0	161
11	20	2	2	2	10	181
12	50	1	3	2	30	231
13	40	1	1	2	30	211
14	20	2	2	2	70	231
15	40	2	2	2	40	203
16	8	1	3	2	80	211
17	20	2	1	2	88	231
18	80	1	2	2	0	231

Table 4 – Study 3 Technician Attributes

Technician	Certifications
1	1,2
2	1,3
3	2,3
4	2
5	1,2
6	1
7	2,3
8	2,3
9	1,2
10	3

It is useful to notice that this problem, with the LFTs in Table 3, is not a tightly-constrained problem. The average utilization across all facilities of type 1 and of type 2 is about 50% as is the utilization across all the technicians. Of course this calculation does not include the impacts of precedence relationships.

We have run the algorithm 17 times, seven times with an α of 500 and ten times with an α of 0. By using an alpha of 0 we are effectively ignoring makespan in the solution. By using an α of

500 we are allowing tradeoffs to be made in the solution to the master between warnings of resource infeasibilities and task start times. We have not attempted to optimize the selection of α.

The results of the experiments are as given in the Table 5. Notice that when $\alpha = 500$, the makespan of the solutions are significantly shorter, but more iterations are needed to find a resource feasible solution. The solution for the experiments with an α equal to 500 is taken to be the first resource feasible solution produced. It is useful to notice that in aggregate terms, when the makespan is 183 hours the utilization on facilities of type 1 rises to about 69% and the utilization of facilities of type 2 rises to about 77%. Finally, the utilization across all technicians is about 72%.

Table 5 – Study 3 Experiments

Experiment	$\alpha = 500$		$\alpha = 0$	
	Makespan	# Iterations	Makespan	# Iterations
1	168	47	226	5
2	186	7	226	5
3	184	35	231	2
4	184	30	231	3
5	184	34	226	5
6	184	34	226	5
7	192	45	231	2
8			231	2
9			226	5
10			231	2
Average	183.1	33.1	228.5	3.6

Conclusions

We have developed a method to plan production activities in a "project job shop" production environment. The formulation has continuous variables that represent start times for the individual tasks, rather than using discrete variables representing whether or not a task begins or terminates in a given time-period. The continuous-variable formulation allows for time-periods of variable lengths and for natural mapping of the model's time-periods to specific applications' time-periods. The formulation was motivated by a need to schedule the disassembly, evaluation and maintenance activities of nuclear weapons, but the general formulation and solution procedure can apply more broadly to resource-constrained project-scheduling problems.

The formulation is a mathematical programming model with a linear objective function and linear constraints, but also includes complementarity constraints, which make the feasible region non-convex. A heuristic method based on Generalized Benders' Decomposition has been developed to provide solutions to this optimization problem. In the decomposition scheme, the sub-problem is a linear program that optimizes the assignment of resources to tasks, given the start times, while the master problem optimizes over start times. The method is a heuristic because the master problem is not solved exactly.

Resource-constrained project-scheduling problems have many variations. Our work focuses on situations in which there is a "best" facility for a given task but others can sometimes be substituted, albeit at a penalty. Tasks require technicians with specific certifications; however, technicians can have more than one certification. In the specific application at Pantex, one of the valuable uses of the model is determining when the set of externally imposed requirements is infeasible, so that management can focus on resolving issues related to those requirements. One of the examples in the previous section has illustrated that aspect of the application.

The ideas developed in this work are transferable to other situations in production planning and project scheduling. We are exploring these other applications. For example, in situations where resource substitutions are excluded, the sub-problem formulation changes, but it is still an LP. The details of the solution algorithm may vary, but the general procedure is the same.

An additional interesting extension is to include other measures of performance in the objective function. These additional measures might include makespan or terms that reflect "risk" in the schedule. In many instances of the resource-constrained project scheduling problem, the task durations are uncertain and the notion of identifying a "low-risk" schedule is important. Given the structure of the solution procedure, it is likely that a measure of schedule risk can be incorporated, and this is under investigation currently.

References

D.A. Jones, C.R. Lawton, M.A. Turnquist, G.F. List, 1999 *Formulations of the Evaluation Planning Module*, SAND99-2095. Sandia National Laboratories, Albuquerque, NM.

E.A. Kjeldgaard, D.A. Jones, G.F. List, M.A. Turnquist, J.W. Angelo, R.D. Hopson, J. Hudson, T. Holeman, 1999 *Swords into Plowshares: Nuclear Weapon Dismantlement, Evaluations, and Maintenance at Pantex.* INTERFACES.

J.F. Benders, 1962, Partitioning Procedures for Solving Mixed-Variables Programming Problems, vol. IV of Numerische Mathematik.

D. Cooper, 1976, Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation, Management Science, No.22, pp.1186-1194.

R. Deckro, E.P. Winkofsky, J. Hebert, and R. Gagnon, 1991, A Decomposition Approach to Multi-Project Scheduling, European Journal of Operational Research, No.51, pp.110-118.

A. Drexl, 1991, Scheduling of Project Networks by Job Assignment, Management Science, No. 37, pp. 1590-1602.

A. Drexl, and J. Grunewald, 1993, Nonpre-emptive Multi-Mode Resource Constrained Project Scheduling, IIE Transactions, No. 25, pp. 74-81.

M. Fukushima, Z. - Q. Luo, and J.-S. Pang, 1998, A Globally Convergent Sequential Quadratic Programming Algorithm for Mathematical Programs with Linear Comlementarity Constraints, Computational Optimization and Applications, No. 10, pp. 5-34.

A.M.Geoffrion, 1972, Generalized Benders Decomposition, Journal of Optimization Theory and Applications, Vol. 10, pp. 237-260.

W. Herroelen, B. De Reyck, and E. Demeulemeester, 1998, Resource-Constrained Project Scheduling: A Survey of Recent Developments, Computers in Operations Research, No.25, pp. 279-302.

R. Kolisch, 1996, Efficient Priority Rules for the Resource-Constrained Project Scheduling Problem, Journal of Operations Management, No.14, pp. 179-192.

R. Kolisch, 1995, Project Scheduling Under Resource Constraints: Efficient Heuristics for Several Problem Classes. Physica-Verlag, Germany.

T.E. Morton and D.W. Pentico, Heuristic Scheduling Systems. John Wiley & Sons, New York, 1993.

L.Özdamar, and G. Ulusoy, 1995, A Survey on the Resource-Constrained Project Scheduling Problem, IIE Transactions, No. 27, pp. 574-586.
A.A.B. Pritsker, L.J. Watters, and P.M. Wolfe, 1969, "Multi-Project Scheduling with Limited Resources: A Zero-One Programming Approach, Management Science, No.16, pp. 93-108.

E. Weiss, 1988, An Optimization Based Heuristic for Scheduling Parallel Project Networks with Constrained Renewable Resources, IIE Transactions, No. 20, pp. 137-143.

Distribution

John Hudson (1)
Pantex Plant
P.O. Box 30020
Amarillo, TX 79177

Professor George List (5)
Department of Civil and Environmental Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Professor Linda Nozick (5)
School of Civil & Environmental Engineering
Cornell University
Hollister Hall
Ithaca, NY 14853

MS 0451	J. Nelson, 6515 (1)
MS 0451	E. Kjeldgaard, 6515 (5)
MS 0451	D. Jones, 6515 (5)
MS 0451	C. Lawton, 6515 (1)
MS 0451	S. Wright, 6515 (1)
MS 0451	J. Bechdel, 6515 (1)
MS 0451	M. Turnquist, 6515 (1)
MS 0451	S. Wagner, 6515 (1)
MS 1110	C. Phillips, 9211 (2)
MS 1110	W. Hart, 9211 (2)
MS 9018	Central Technical Files, 8945-1
MS 0899	Technical Library, 9616
MS 0612	Review and Approval Desk, 9612 For DOE/OSTI