

Scheduling Workforce and Workflow in a High Volume Factory

Oded Berman • Richard C. Larson • Edieal Pinker

University of Toronto, Toronto, Ontario, Canada

*Center for Advanced Educational Services, Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139*

Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

We define a high volume factory to be a connected network of workstations, at which assigned workers process work-in-progress that flows at high rates through the workstations. A high rate usually implies that each worker processes many pieces per hour, enough so that work can be described as a deterministic hourly flow rate rather than, say, a stochastic number of discrete entities. Examples include mail processing and sorting, check processing, telephoned order processing, and inspecting and packaging of certain foods. Exogenous work may enter the factory at any workstation according to any time-of-day profile. Work-in-progress flows through the factory in discrete time according to Markovian routings. Workers, who in general are cross-trained, may work part time or full time shifts, may start work only at designated shift starting times, and may change job assignments at mid shift. In order to smooth the flow of work-in-progress through the service factory, work-in-progress may be temporarily inventoried (in buffers) at work stations. The objective is to schedule the workers (and correspondingly, the workflow) in a manner that minimizes labor costs subject to a variety of service-level, contractual and physical constraints. Motivated in part by analysis techniques of discrete time linear time-invariant (LTI) systems, an object-oriented linear programming (OOLP) model is developed. Using exogenous input work profiles typical of large U.S. mail processing facilities, illustrative computational results are included.

(Linear Programming; Scheduling; Manufacturing; Object-oriented; Queueing; Queueing Networks; Markov Chains)

Our focus is development of a model to schedule both workers and their work through a complex high volume factory. Our aim is to build an initial first bridge between simple one station queueing models that have been used to schedule bank tellers (and similar single node servers) and complex system models that take into account the complexity of an entire manufacturing floor. While we believe that a useful framing of a general model is created, we do not assert that the operational details and idiosyncrasies of any particular factory are represented in our model. To bring the work to that level of operational implementation may require a

large number of additional side constraints and (perhaps) additional types of "network nodes" within the model.

Our motivation for this research stems from a professional assignment that the second author had with the U.S. Postal Service (USPS) in 1991. That assignment dealt with scheduling workers within large Mail Processing Centers (MPCs), which are examples of high volume factories. The inputs to an MPC are both out-bound and inbound mail that require sorting. The MPC provides a sequence of required sorts to each piece of mail, delivering the sorted mail to loading docks for

transportation to either long haul transportation facilities (e.g., an airport), for outbound mail; or to local post offices, for inbound mail. The arrival pattern of mail over the course of a day is highly predictable and time-of-day dependent; the fully sorted mail must be on the loading docks prior to prespecified "dispatch deadlines." This fully sorted mail, appropriately bundled, represents the "manufactured product" of an MPC.

Other examples of high volume factories can be found in the "back rooms" of banks (e.g., in the processing of checks), in insurance companies (e.g., in the processing of claims), in order processing rooms associated with "800" numbers, in many different governmental offices dealing with the processing of various types of applications, and in food inspection and processing plants.

For simplicity in this paper we shall use the nomenclature of the MPC to motivate, illustrate and crystallize the concepts involved. An earlier version of the model, using data from three USPS MPC's, was instrumental in assisting the second author's expert testimony in 1991 before a five member labor/management arbitration panel in Washington, D.C. At that time USPS management could designate only 10% of its workforce in large facilities as part time and/or flexible. At least 90% of the workers had to be full time regularly scheduled 40-hour-a-week employees, working 8-hour shifts on scheduled work days. Based on the modeling work, arbitrators concluded that management of the USPS should contractually be allowed twice as much flexibility (i.e., up to 20%) in scheduling their personnel in MPC's (Cahn, Larson, and Berman 1992). That decision is currently saving the USPS hundreds of millions of dollars per year. This paper represents a generalization of that work. Yet, the improved model should be considered at this time for strategic planning only, not tactical day-to-day planning. An appropriate tactical model would have to consider myriad additional details including probabilistic items such as worker absenteeism.

1. Perspective on the Problem: Beyond the "Teller Paradigm"

Workforce scheduling has had a rich history in the literature of services industries, such as nursing (Warner

1972), bank tellers (Mabert 1977), telephone operators (Henderson 1977), and fast food servers (Glover 1986).

In the operations research literature the labor scheduling problem has been guided by what can be called the "bank teller paradigm," i.e., the scheduling of bank tellers to shifts during a work day that exhibits predictable varying demand. The demand or workload is assumed to have been determined empirically by methods such as that in Edie (1954). The objective is to minimize the amount of labor used while providing enough service to satisfy demand. Papers in the literature have run the gamut from the scheduling of days off in a cyclical weekly schedule, to overlapping shift scheduling during a day, taking into account meal and rest breaks. These problems are all formulated as either linear or integer programming models, some driven by the outputs of queueing models. Many of the early results obtained from these types of models are surveyed in Baker (1976). In all cases homogeneous and unlimited labor pools are assumed. In the simpler cases, efficient solution algorithms could be derived (Baker 1973); in the more complicated formulations, one of the major difficulties has been the integer nature of the problem. Various heuristics (Henderson 1976) have been suggested and attempts made to reduce the problem to a network flow problem (Bartholdi 1980). Bechtold and Jacobs (1990) present a new modeling approach for flexible break assignments which reduces the number of decision variables in the problem.

There have also been formulations that add significant realism to the problem. For example Warner (1972) and Emmons and Burns (1991) address a non-homogeneous labor pool with substitution of a limited number of employees of differing qualifications to cover shortages. We refer to this aspect as "job switching." Though job switching is introduced in a treatment of the nurse staffing problem, in Warner (1972) it too can be categorized by the bank teller paradigm. If one thinks of a nurse qualified for a particular ward of a hospital as a bank teller qualified for a particular financial transaction, the similarity of the situations becomes clear. Glover (1986) creates one of the most realistic or "general" formulations of the employee scheduling problem by considering a case with a nonhomogeneous labor pool of limited size with great flexibility in assignment of days off,

employee activity preferences, breaks, part time work, etc. Although complex, this formulation too is part of the bank teller paradigm, since it assumes that work performed by one employee has no influence on the rate at which work arrives to another employee who performs a different task.

The high volume factory can be viewed as a *network* of "bank tellers." "Customers" (i.e., units of work-in-progress) proceed through the various work stations of the factory, with different customers perhaps taking different routes. Any customer may enter the factory exogenously at any station and exit the factory at any station. Customer arrival profiles can follow any prescribed time-of-day pattern. Paths through the network are governed by a Markov chain, allowing for feedback (i.e., cycling) due perhaps to defective processing or work categorization. In many ways the model parallels that of a Jackson queueing network (Jackson 1957), but without stochasticity and steady state. In addition, invoking the "high volume" assumption, we assume that the various customer flows are large enough so that they may be accurately approximated as continuous ("relaxed") variables, not integers. By allowing the queue of customers, i.e., work-in-progress, at a workstation to grow in a buffer, one has the flexibility to smooth the workflow over the course of a day. Work does not arrive "downstream" from any particular workstation until, of course, the work is processed at that station; inventorying work at a workstation will delay flow of work to stations downstream and thereby delay demand for workers at downstream workstations. Workers, who in general are cross-trained, may work part time or full time shifts, may start work only at designated shift starting times, and may change job assignments at mid shift. Our task is to schedule "servers" (i.e., qualified workers) at the respective stations, and also to schedule the time progress of "customers" through the network so as to minimize labor costs while satisfying constraints such as required time windows for customers to emerge "fully serviced" from the network. In an actual factory, the "customer" at a workstation may be a subassembly or a piece of partially sorted mail or a food product being inspected or processed prior to packaging. A "customer" who emerges from the factory is a "fully manufactured item," which may mean a fully assembled product, a bundled and

fully sorted package of mail, or a packaged food product ready for shipping.

The modeling approach uses ideas of linear time invariant (LTI) systems analysis, due in part to the fact that the time progression of a Markov chain is governed by a set of linear difference equations (Sittler 1956, Howard 1971). When realism can be enhanced, we include extensive detail in the input data set, recognizing that data set size does not necessarily imply model intractability or conceptual complexity. Finally, the linear program derived from the modeling analysis could be thought of as an *object oriented LP (OOLP)*, due to the building block nature of the model; the resulting model can be thought of a set of interrelated LTI blocks (i.e., workstations), connected together by a network whose dynamics are governed in discrete time by LTI analysis. A user can construct and operate the LP in object language without ever having to see the detailed objective function and constraint equations imposed by the model. The ideas are compatible with many of the suggestions of Geoffrion toward building a "language for structured modeling" (Geoffrion 1987, 1989, 1992a, 1992b).

For reader convenience, a glossary of modeling terms is given in Exhibit 1.

2. Formulation of the Problem

2.1. Overview

The high volume factory contains n workstations (or stations), where at each workstation workers will be assigned to process work-in-progress (hereafter simply referred to as "work") that flows through the workstation. The workday is divided up into T equal length periods (e.g., 24 one-hour periods).

Work arrives exogenously at the factory during each time period and is presented to the appropriate work station(s) at the *beginning* of the next time period. We denote by b_{jt} the number of units of work (sometimes called "jobs") that arrive exogenously to station j and are presented there at the beginning of time period t , $t = 1, 2, \dots, T$. For instance, if $b_{47} = 66$, then 66 units of work arrive exogenously to station 4 during period 6 and "become ready" for processing by station 4 at the beginning of period 7. We assume a cyclic clock, so that b_{j1} is the exogenous work that arrives at workstation j

Exhibit 1 Glossary

- n = total number of workstations in the factory
 T = total number of equal length time periods during a working day
 b_{jt} = the number of units of work that arrive exogenously to station j and is presented there at the beginning of time period t , $t = 1, 2, \dots, T$
 B = total daily exogenous work input, or $B \equiv \sum_{j=1}^n \sum_{t=1}^T b_{jt}$
 ρ_{ij} = fraction of jobs processed at station i that are routed next to station j , $i, j = 1, \dots, n$
 \mathbf{H} = the set of all allowed shift lengths
 \mathbf{ST} = the set of all allowed starting times for shifts
 $M_{ki} = 1$ (0) if worker type k can (cannot) perform work at station i
 β_{ki} = units of work that worker type k can process per time period at station i
 (j_1, j_2) = a pair of stations j_1 and j_2 , representing a worker's workstation assignments for the first and second half of her shift, respectively
 A_k = the set of all (j_1, j_2) that are feasible for worker type k , i.e., $M_{kj_1} = M_{kj_2} = 1$
 $X_{k,(j_1,j_2),h,\tau}$ = number of workers of type k working the first half of the shift at station j_1 , the second half of their shift at station j_2 , for a shift of length h that starts at time period τ ; $k = 1, \dots, K$; $(j_1, j_2) \in A_k$; $h \in \mathbf{H}$; $\tau \in \mathbf{ST}$
 $C_{k,(j_1,j_2),h,\tau}$ = cost of a worker of type k working the first half of the shift at station j_1 , the second half of their shift at station j_2 , for a shift of length h that starts at time period τ ; $k = 1, \dots, K$; $(j_1, j_2) \in A_k$; $h \in \mathbf{H}$; $\tau \in \mathbf{ST}$
 $\beta_{k,(j_1,j_2),h,\tau,t}$ = number of units of work that a type k worker can process during period t , assuming the worker works the first half of the shift at station j_1 , the second half of their shift at station j_2 , for a shift of length h that starts at time period τ ; $k = 1, \dots, K$; $(j_1, j_2) \in A_k$; $h \in \mathbf{H}$; $\tau \in \mathbf{ST}$; $t = \tau, \tau + 1, \dots, \tau + h - 1$
 I_{jt} = total quantity of new work presented to station j at the start of period t
 $R_{k,t-1}$ = total work remaining at station j from period $t - 1$
 Y_{jt} = units of work in the buffer at station j at the start of period t
 W_{jt} = maximum number of jobs that can be processed by personnel assigned to station j during period t
 O_{jt} = "output" of station j during period t
 λ = the maximum allowed percentage of daily exogenous work that can be left in the system at the end of the day for processing "tomorrow"
 w_{jt} = maximum number of workers who are permitted to work at station j during period t
 γ_{jt} = capacity of buffer j during period t , measured in units of work
-

during period T and is presented to station j at the beginning of period 1.

Work can move from one station to another only at the *end* of each respective time period, being transported virtually instantaneously to another station, ready for processing at the *beginning* of the next time period.

2.2. Markovian Work Routing

The process by which work proceeds from station to station can be described by a network in which each node represents a station and each (directed) link represents a one-step path of work flow from one station to another. It is convenient to add one additional (dummy station) node $n + 1$ which "collects" all the final output from the stations.

We denote the fraction of work output that flows from station i to station j by p_{ij} , $i, j = 1, 2, \dots, n + 1$, where $\sum_{j=1}^{n+1} p_{ij} = 1$, $i = 1, 2, \dots, n$ and $p_{n+1,n+1} \equiv 1$. We allow $p_{ii} > 0$, reflecting a self loop at station i which in practice usually depicts processed work at station i that, due to some type of defect, must be reworked before it can be forwarded to station $j \neq i$. We assume that any such rework on defects must occur in a time period subsequent to the time period in which the work is initially processed at node i . That is, the same job cannot be both worked and reworked during the same time period.

While the flow of individual jobs is sufficiently large that we can treat all flows as deterministic quantities, the route of any individual job is probabilistic. This is due to the Markovian nature of the network of stations,

in which each station can be viewed as a state in a discrete state discrete transition Markov chain. Feasibility of a solution requires that the artificial state $n + 1$ is accessible from any starting state i (corresponding to the station of exogenous entry into the system), and since state $n + 1$ is a trapping state (i.e., $p_{n+1,n+1} = 1$), the Markov chain is ergodic with state $n + 1$ the only recurrent state; all other states are transient. This implies, with probability one, each piece of work will eventually reach state $n + 1$, i.e. leave the factory. The *path* taken by any individual job is independent of the values of the decision variables, assuming a feasible solution. However, the *time* that a job is resident in the system is highly dependent on the values of the decision variables; in general, larger in-residence times correspond to larger inventories in buffers.

The deterministic flows used in the optimization are in fact the expected values of integer-valued random variables corresponding to the merged operation of many identical Markov chains, each governed by the transition probabilities p_{ij} . An analysis of the validity of the deterministic assumption used in the optimization model would focus on the values of the coefficients of variation of these random variables. In large MPCs, for instance, these are typically less than 0.05, implying that the deterministic assumption is a reasonable one.

2.3. The Decision Variables

The objective is to assign workers by time and task to meet system constraints at minimum cost. Thus, the numbers of workers assigned, by category, represent the decision variables.

Each worker's time schedule consists of assignment to a shift of a given integer length and starting at a given integer time, where the respective integers correspond to numbers of time periods of the shift and the time period during which the shift is started, respectively. Let \mathbf{H} be the set of all allowed shift lengths (e.g., $\mathbf{H} = \{4, 6, 8, 12\}$) and \mathbf{ST} be the set of all allowed starting times for shifts (e.g., $\mathbf{ST} = \{4, 8, 12, 16, 20, 24\}$).

In a move away from the simple teller's paradigm, workers may be *cross-trained*. That is, some or all workers are able to do the work associated with two or more workstations. This flexibility is exploited at the mid shift break point, usually just after the meal break, at which point the worker may be switched from the station

worked before the break to another for which she is trained. A worker *type* is characterized by the set of stations for which she is trained to work *and* by the productivity levels that she can attain at each respective station. For instance, worker type 1 may be trained to work stations 1, 5, 6, and 10 (with given productivity levels), whereas worker types 2 and 3 may be trained only to work stations 1 and 7, with type 2 being "more productive" than type 3. A worker is said to be "trained on station j " if she has positive (i.e., nonzero) productivity at station j , at least during certain hours on certain shifts. Let M be a matrix with rows corresponding to worker types and columns corresponding to stations, where

$$M_{ki} = \begin{cases} 1 & \text{if worker type } k \text{ is trained on station } i, \\ 0 & \text{otherwise.} \end{cases}$$

Let (j_1, j_2) be a pair of stations where j_1 represents the station a worker is assigned to during the first part of her shift and j_2 is the station the worker is assigned to during the second part of her shift. Let A_k be the set of all (j_1, j_2) that are feasible for worker type k , i.e., $M_{kj_1} = M_{kj_2} = 1$.

Each worker type's hourly productivity depends on shift characteristics and is allowed to vary over the course of a shift; this flexibility allows one to adjust for partial or full time off during a period for meal break, travel between stations, set-up and set-down requirements during the starting and ending periods of shifts, etc. The productivity coefficient is

$$\beta_{k,(j_1,j_2),h,\tau,t} = \text{number of units of work that a type } k \text{ worker can process during period } t, \text{ assuming the worker works the first half of the shift at station } j_1, \text{ the second half of his shift at station } j_2, \text{ for a shift of length } h \text{ that starts at time period } \tau; k = 1, \dots, K; (j_1, j_2) \in A_k; h \in \mathbf{H}; \tau \in \mathbf{ST}, t = \tau, \tau + 1, \dots, \tau + h - 1.$$

This coefficient is equal to zero if M_{ki} is zero; i.e., if the worker is not qualified to work at station i , and is usually positive otherwise. For any particular period t the coefficient may be zero or a small number, reflecting a meal break, time in transit, etc. Otherwise, we assume that a qualified worker at station j can process many pieces of work per time period. For instance, a worker

assigned to an automated letter sorting machine in an MPC can process more than 3,000 letters per hour.

There is one set of explicit decision variables in the model, namely the numbers of workers of various types to schedule for tours and assign to workstations. We define these decision variables and their associated costs as follows:

$X_{k,(j_1,j_2),h,\tau}$ { $C_{k,(j_1,j_2),h,\tau}$ } = number of workers {cost of each worker} of type k working the first half of the shift at station j_1 , the second half of their shift at station j_2 , for a shift of length h that starts at time period τ ; $k = 1, \dots, K$; $(j_1, j_2) \in A_k$; $h \in \mathbf{H}$; $\tau \in \mathbf{ST}$.

The objective is to minimize the total cost of the system,

$$\text{Min} \sum_{k=1}^K \sum_{(j_1,j_2) \in A_k} \sum_{h \in \mathbf{H}} \sum_{\tau \in \mathbf{ST}} C_{k,(j_1,j_2),h,\tau} X_{k,(j_1,j_2),h,\tau}$$

where the sum is seen to be over all possible combinations of worker types, feasible pairs of stations for which workers are qualified to work, shift lengths and starting times.

A second set of decision variables, which are implicit, deals with workflow. There are (finite capacity) buffers at each workstation that can be used to inventory work so that not all work that arrives at a workstation at the start of time period t needs to be processed during period t . Some work can be left over at the end of period t , to be processed during period $t + 1$ or even later. In that sense, the optimal solution to the problem determines personnel scheduling with job assignments and workflow progression through the factory.

2.4. Model of a Workstation

The generic workstation is the building block of the model, the key "icon" in the object-oriented depiction of the high volume factory. We develop the model for the workstation in this subsection.

First we deal with work flowing into the workstation. The total quantity of work (jobs) at station j at the start of period t is the sum of the *new work* that is presented to station j and the *residual work* that has remained at station j from period $t - 1$.

We define I_{jt} , the *new work* at station j at the start of period t , as the sum of the exogenous work presented

to station j and the sum of all the work delivered from other work stations, excluding station j :

$$I_{jt} = \begin{cases} b_{jt} + \sum_{i \neq j} p_{ij} O_{i(t-1)}, & t = 2, 3, \dots, T, \\ b_{j1} + \sum_{i \neq j} p_{ij} O_{iT}, & t = 1, \text{ where} \end{cases} \quad (1)$$

$O_{i(t-1)}$ = the "output" at station i at the end

of period $t - 1$

= the number of units of work produced

at station i during period $t - 1$.

We define the *residual work* remaining at station j from period $t - 1$, $R_{j,t-1}$, as the sum of (i) the number of jobs processed during period $t - 1$ at station i that must be reworked at station j due to defects and (ii) the difference (if positive) between the quantity in the buffer at the beginning of period $t - 1$ and the amount processed during period $t - 1$:

$$R_{j,t-1} = p_{jj} O_{j(t-1)} + [Y_{j(t-1)} - O_{j(t-1)}], \text{ where:} \quad (2)$$

Y_{jt} = number of units of work in the buffer at station j at the start of period t .

Y_{jt} is the maximum potential quantity of work that can be processed at station j during period t . The actual quantity of work depends on the number and types of workers assigned to station j during period t .

Recognizing that the buffer content is the sum of the new work and residual work, we can write

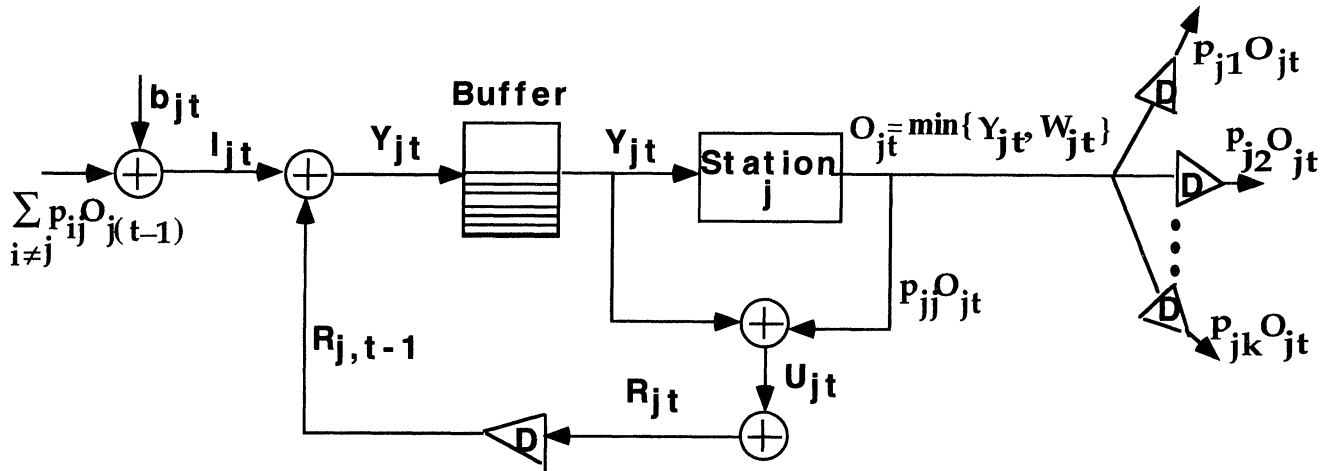
$$Y_{jt} = I_{jt} + R_{j,t-1}. \quad (3)$$

All of these relationships are shown schematically in Exhibit 2, in which a unit of delay is depicted by a triangular unit delay operator.

We now develop expressions reflecting the fact that the amount of output from a workstation j during a time period t , O_{jt} , will be either the maximum amount that the currently deployed workers can produce or the amount in the buffer (Y_{jt}), whichever is smaller. If we define

W_{jt} = maximum number of jobs that can be processed by personnel assigned to station j during period t ,

Exhibit 2 Schematic Diagram of Generic Workstation



then we can write

$$O_{jt} \leq Y_{jt}, \quad (4a)$$

$$O_{jt} \leq W_{jt}. \quad (4b)$$

The maximum number of jobs W_{jt} depends on the number of each type of worker that is assigned to station j during period t , and can be expressed as

$$W_{jt} = \sum_{k \in M_j} \sum_{h \in H} \left\{ \sum_{\tau \in F_{th}} \beta_{k,(jj),h,\tau} X_{k,(jj),h,\tau} + \sum_{n_2 \in W_k} \sum_{\tau \in G_{th}} \beta_{k,(jn_2),h,\tau} X_{k,(jn_2),h,\tau} + \sum_{n_1 \in W_k} \sum_{\tau \in Q_{th}} \beta_{k,(n_1j),h,\tau} X_{k,(n_1j),h,\tau} \right\} \text{ where } (5)$$

M_j is the set of qualified worker types for station j ,
 W_k is the set of stations for which worker type k is qualified,

F_{th} is the set of starting times such that a shift of length h includes period t , i.e., $F_{th} = \{\tau: t + 1 - h \leq \tau \leq t\}$,

G_{th} is the set of starting times such that the first half of shift h includes period t , i.e., $G_{th} = \{\tau: t + 1 - h/2 \leq \tau \leq t\}$, and

Q_{th} is the set of starting times such that the second half of the shift h includes period t , i.e., $Q_{th} = \{\tau: t + 1 - h \leq \tau \leq t - h/2\}$.

The entire workstation described in this section can be considered as one object or icon in an interactive

computer implementation of this modeling system. In computer parlance, the workstation object can be thought of as a large "macro."

2.5. Completed Work

All work that is completed during the day is artificially collected and stored at node $n + 1$. Consider a particular day: work is physically completed at the end of each of the periods $1, 2, \dots, T$, and is forwarded to node $n + 1$ at the start of periods $2, 3, \dots, T, 1$, respectively. Work completed at the end of period T (i.e., at the end of the day), is counted as the first completed work for the next day. In terms of defined variables, we note that

$$Y_{n+1,t} = \text{total number of units of work at node } n + 1 \text{ at the start of period } t \\ = \text{total work completed up to and including the start of period } t.$$

The total intraday work completed up to and including the start of period t is given by the usual recursion,

$$Y_{n+1,t} = \sum_{i=1}^n p_{i,n+1} O_{i(t-1)} + Y_{n+1,(t-1)}, \quad t = 2, \dots, T, \quad (6)$$

where we define the start-of-day boundary condition, which counts work processed during "yesterday's" last period as "today's" first period completed work,

$$Y_{n+1,1} = \sum_{i=1}^n p_{i,n+1} O_{iT}.$$

In LTI terminology node $n + 1$ is simply a summation operator that is reset at the start of each day; it too is an object that can be thought of as an icon or a macro.

2.6. Capacity Constraints

Most applications of a model of a high volume factory must include constraints on capacities of certain physical facilities to accommodate workers and/or work.

With regard to workers, in most factories there is a capacity limit for the number of workers who can simultaneously work at a workstation. For instance, for certain semi-automated mail sorting machines in U.S. postal facilities, there is a finite number of positions (each consisting of a chair, keyboard and mail input device) for workers. Once these positions are all filled, no additional workers can be accommodated at that station. We generalize this by defining

w_{jt} = maximum number of workers permitted at station j during period t .

Then we must have:

$$\sum_{h \in H} \left\{ \sum_{\tau \in F_{th}} X_{k,(jj),h,\tau} + \sum_{n_2 \in W_k} \sum_{\tau \in G_{th}} X_{k,(jn_2),h,\tau} + \sum_{n_1 \in W_k} \sum_{\tau \in Q_{th}} X_{k,(n_1j),h,\tau} \right\} \leq w_{jt},$$

$$j = 1, \dots, n; t = 1, \dots, T \quad (7)$$

With regard to work, we assume that each buffer j is capacity constrained, with perhaps a different capacity by time of day. Defining

γ_{jt} = capacity of buffer j during period t , measured in units of work,

we can write,

$$Y_{jt} \leq \gamma_{jt}, \quad j = 1, \dots, n; t = 1, \dots, T. \quad (8)$$

2.7. Time Window Constraints for Work Completion

We now consider time constraints for completing the work. For any feasible solution, the model behaves as a deterministic time-cyclic system, in which each "day of work" is exactly like every other day. Thus, if a solution is feasible, every day the amount of work *produced* by the factory is equal to the amount of work *presented* ex-

ogenously to the factory. Noting that the total daily exogenous work input to the system is $B \equiv \sum_{j=1}^n \sum_{t=1}^T b_{jt}$, for solution feasibility we must have $Y_{n+1,T} = B$.

But management is usually interested in the *speed* with which work proceeds through the factory. For instance, management might specify that 95% of the work brought to the factory each day must exit the factory by the end of the day. By "work brought to the factory," we mean the sum of exogenous work and work left over from the previous day. For the stated condition to be fulfilled, the system (excluding state $n + 1$) at the end of the day cannot contain as work-in-progress inventory more than 5% of the sum of exogenous inputs and work left over from the previous day. At each station j , the work left over from the previous day in period 1 is $Y_{j1} - b_{j1}$. Thus the total work in the system remaining from one day to the next is $\sum_{j=1}^n (Y_{j1} - b_{j1})$. Generalizing the 95% service level to λ percent, we obtain the constraint

$$\lambda \left\{ \sum_{j=1}^n (Y_{j1} - b_{j1}) + B \right\} \leq B. \quad (9)$$

In addition to the speed of work constraint represented by Equation (9), management may wish to constrain the *pattern of flow* of work completed over the course of the day. By this we mean that we are given χ_t as the desired fraction of the day's work to be completed by the end of period t , for $t = 1, 2, \dots, T - 1$. This type of constraint is given by

$$Y_{n+1,t+1}/B \geq \chi_t, \quad t = 1, 2, \dots, T - 1. \quad (10)$$

(Recall that $Y_{n+1,t+1}$ is the cumulative work completed by time t , work processed during period T of one day is "credited" to period 1 of the next day and $Y_{n+1,T} = B$.)

2.8. Side Constraints

A variety of side constraints can be included in the model. For example, we can require that a given percentage α of workers must be full time workers (i.e., those working shifts of length eight hours or more). This constraint can be expressed as

$$(1 - \alpha) \left\{ \sum_{k=1}^K \sum_{(j_1j_2) \in a_k} \sum_{h \in H, h \geq 8} \sum_{\tau \in ST} X_{k,(j_1j_2),h,\tau} \right\} - \alpha \left\{ \sum_{k=1}^K \sum_{(j_1j_2) \in a_k} \sum_{h \in H} \sum_{\tau \in ST} X_{k,(j_1j_2),h,\tau} \right\} \geq 0. \quad (11)$$

For another example suppose the number of workers who switch between different stations is at most γ percent. This constraint can be written as

$$(1 - \gamma) \left\{ \sum_{k=1}^K \sum_{(jj) \in a_k} \sum_{h \in H} \sum_{\tau \in ST} X_{k,(jj),h,\tau} \right\} - \gamma \left\{ \sum_{k=1}^K \sum_{(j_1j_2) \in a_k} \sum_{h \in H} \sum_{\tau \in ST} X_{k,(j_1j_2),h,\tau} \right\} \leq 0. \quad (12)$$

The model can include many other side constraints.

The optimization problem is to determine how many workers to assign to each station during each period so as to minimize the total cost of the system subject to all the given constraints.

2.9. Size of the LP

The size of the LP model in terms of number of constraints is mainly determined by the number of workstations in the system and the time scale used, whereas the number of variables is dependent upon the degree of flexibility and heterogeneity of the workforce being modeled.

There are $3nT$ constraints describing the system dynamics, nT constraints each for the buffer capacity and worker space capacity limitations, T constraints governing the speed and pattern of the workflow, and a constraint each for the limitations on the amount of job switching and part time work allowed. More constraints can be added to define different time windows for productivity levels. Therefore the total number of constraints is approximately $(5n + 1)T + 2$. For example in a 6-station system with T set at 48 to represent half-hour periods the number of constraints will be 1,490.

The number of variables relating to the flow of work is $2nT$, i.e. the variables Y_{jt} and O_{jt} . The number of labor variables $X_{k,(j_1j_2),h,\tau}$ is $|\mathbf{H}| |\mathbf{ST}| \sum_k |A_k|$. For example in a workforce that had 10 different types of workers, each qualified to perform each of the 6 different tasks in the system, the sum $\sum_k |A_k|$ would be 360. If there were 4 shift lengths and 6 start times, the total number of labor variables would be 8,640. In the 6 station system modeled in §4 of this paper, with 48 half-hour periods, there are 576 flow variables for a total of 9,216 variables. It is easy to see that the variety and flexibility in the workforce is what most strongly determines the number of variables in the problem.

3. A Simple Feasibility Test

Before preparing the detailed data base for linear programming execution, one might wish to check that a feasible solution exists. While we do not know how to do this for the general case, we have developed a simple check for the case in which neither buffer capacity constraints nor staffing limitations at the workstations are invoked. In such a case, it would be possible (with sufficiently high staffing levels) to push work through the system with no period-to-period inventory remaining in buffers; that is, each unit of work presented to station j at the beginning of period t would be processed at that station during period t . In that case, work proceeds through the system as directed by the Markov chain having transition probability matrix $P = (p_{ij})$. For this Markov chain, we wish to see if the "λ constraint" [Eq. (9)] is met, that is if a sufficiently large fraction of the daily work can exit the system on the day presented. Factors working against completing work by end-of-day include late-in-the-day arrivals of exogenous work and/or feedback (i.e., cycling) in the workstation network.

For the situation described above, let

$f_j(t)$ = flow of work (in units of work) through station j during period t .

Define the $n + 1$ -vector $\mathbf{F}(t) = (f_j(t))$, $j = 1, 2, \dots, n + 1$. Define the $n + 1$ -vector of exogenous inputs at period t , $\mathbf{b}(t) = (b_j(t))$. Then, at the end of each day, just before the end of period T , there exists a quantity of end-of-day work at state j equal to $O_{jT} = \pi_j$, $j = 1, 2, \dots, n$. For $j = n + 1$, we know that $\pi_{n+1} = B$. The end-of-day work is propagated into the next day according to the transition probabilities (p_{ji}). For instance, $\pi_2 p_{21}$ is the quantity of work propagated from station 2 to station 1 at the start of the next day.

Define the $n + 1$ vector of end-of-day work, $\boldsymbol{\pi} = (\pi_j)$. Following a recursion through several periods, we obtain,

$$\mathbf{F}(1) = \mathbf{b}(1) + \boldsymbol{\pi}P,$$

$$\mathbf{F}(2) = \mathbf{b}(2) + \mathbf{F}(1)P = \mathbf{b}(2) + \mathbf{b}(1)P + \boldsymbol{\pi}P^2,$$

$$\mathbf{F}(3) = \mathbf{b}(3) + \mathbf{F}(2)P = \mathbf{b}(3) + \mathbf{b}(2)P + \mathbf{b}(1)P^2 + \boldsymbol{\pi}P^3,$$

or, in general,

$$\mathbf{F}(t) = \sum_{k=0}^{t-1} \mathbf{b}(t-k)P^k + \boldsymbol{\pi}P^t.$$

Exhibit 3 Six Station Service Factory

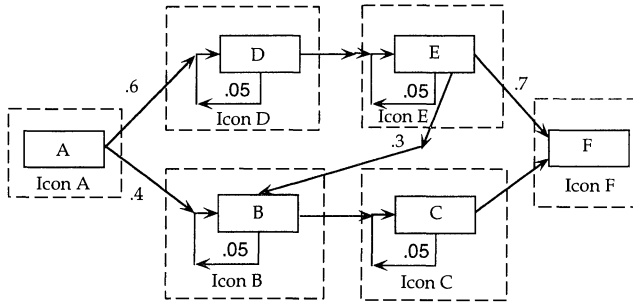


Exhibit 4: The Six Allowed Worker Types for Illustrative Example

Worker Type	Qualification	Productivity*	Salary**
1	A, E	60, 80	30
2	B, F	80, 80	38
3	A, C	80, 80	36
4	A, D	60, 60	30
5	E, F	80, 80	34
6	D, E	80, 80	36

*Units of work processed per hour

**Dollars per hour

We can write $F(T) = (F^*(T), B)[\pi = (\pi^*, B)]$, where $F^*(T)[\pi^*]$ is an n -vector corresponding to the n (real) workstations and the scalar B is the total amount of work accumulated in the trap state up to and including period T . The $(n + 1)$ by $(n + 1)$ transition probability matrix P can also be partitioned, with P^* corresponding to the n by n square sub matrix corresponding to P with the $n + 1$ st row and column removed. P^* is a substochastic matrix (having row sums less than or equal to one, with at least one row sum strictly less than one). But, for states $j = 1, 2, \dots, n$, due to the time-cyclic nature of the process, we have

$$F^*(T) = \pi^* = \sum_{k=0}^{T-1} b(T-k)P^{*k} + \pi^*P^{*T}$$

or, solving for π^* ,

$$\pi^* = \left[\sum_{k=0}^{T-1} b(T-k)P^{*k} \right] [I - P^{*T}]^{-1}. \quad (13)$$

Equation (13) is guaranteed to have a solution since, due to P^* being sub stochastic, the determinant $|I - \mu P^{*T}|$ has no solutions for $\mu = 1$.

At time T the total amount of work in the system that will be held over for the next day is $\sum_{i=1}^n \pi_i^*(1 - p_{i,n+1})$. Following Equation (9), we must have

$$\begin{aligned} & \sum_{i=1}^n \pi_i^*(1 - p_{i,n+1}) \\ & \leq (1 - \lambda) \left[B + \sum_{i=1}^n \pi_i^*(1 - p_{i,n+1}) \right] \quad \text{or,} \\ & \sum_{i=1}^n \pi_i^*(1 - p_{i,n+1}) \leq \{(1 - \lambda)/\lambda\}B. \end{aligned} \quad (14)$$

If Equation (14) is not satisfied, then the problem has no feasible solution. Since Equation (13) can be solved in $O(Tn^3)$ time, its use with Equation (14) could save a much larger amount of computer time (and analyst's detailed data preparation time).

4. Illustrative Computational Results

The model presented in §2 was programmed in object-oriented form as a "problem generator" to provide ease of operation for the operations research/management science analyst. The "output file" from the program, invisible to the user, is designed in a form that is compatible with the widely available commercial LP solver, CPLEX™. All executions of the model have been done using CPLEX as a "black box" LP solver.

To illustrate several features of the model, we present numerical results based on a service factory having six

Exhibit 5 Time-of-Day Exogenous Input Work Profile

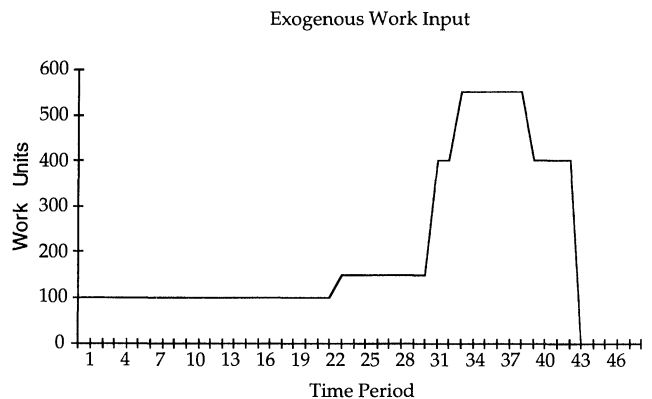
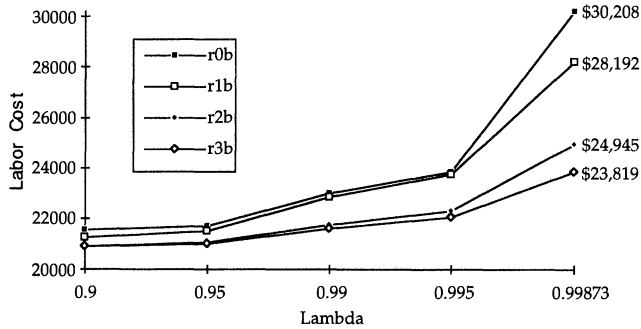


Exhibit 6 Results of the First Set of Executions of the Model



stations arranged in the network depicted in object-oriented form in Exhibit 3. Each lettered rectangular icon represents a workstation, and nonzero interstation workflows are depicted by directed arcs. The work day is divided into $T = 48$ 30-minute time intervals, with interval 1 starting at midnight. Work only enters the system at station A and exits at station F. At each of the stations B, C, D, and E there is a 5 percent defect-related rework rate. At all stations there is an inventory storage limit of 550 units of work, which is also the maximum one period exogenous work input to the system. The (probabilistic) routing of the work occurs according to the transition probabilities shown adjacent to the respective directed arcs in Exhibit 3, all other interworkstation transitions occur with probability 0.

We assume that the potential labor pool for staffing the above system comprises the worker types shown in Exhibit 4, e.g., a type 1 worker is qualified at station A

with a productivity of up to 60 units of work per hour; and at station E with a productivity of up to 80 units of work per hour. We further assume that the first and last half hour of a worker's shift are performed at half the maximum productivity rate, and that during the half hour mid-shift break the productivity is zero. These assumptions allow for set-up and set-down times at the beginning and end of the shift, respectively, and for a midshift meal break. (While we could express all model parameters in terms of our notations of $\$2$, we choose not to do this in order to maximize intuitive understanding of the model.)

Exogenous work arrives according to the time-of-day exogenous work profile shown in Exhibit 5. The main characteristic of this schedule is that a high percentage of the total exogenous work input for a day arrives during a relatively short time span late in the day. Such a pattern is typical, for instance, of large MPC's of the United States Postal Service.

We operate the model so as to minimize total labor costs under several different operating policies. An operating policy will be defined by four decisions:

1. Is part time work allowed?
2. Is intrashift job switching allowed?
3. Is backlogging of work at workstations allowed?
4. How much residual inventory is left in the system at day's end?

A *full time* shift is 8.5 hours, comprising 8.0 hours of work with a one half hour (meal)break after the first four work hours. When part time work is allowed we restrict it to at most 20 percent of the workforce, with

Exhibit 7

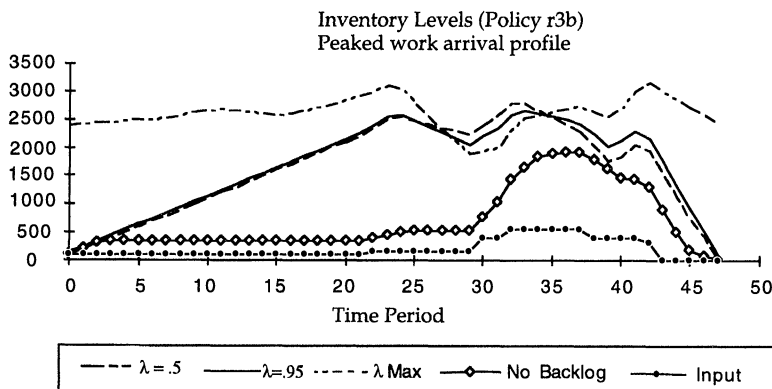


Exhibit 8: Optimal Objective Function Values, Six Start Times, Backlogging

lambda	max final inv.	r0b	r1b	r2b	r3b
0.5	9,400	19,804	19,151	19,058	18,989
0.8	2,350	19,804	19,151	19,058	18,989
0.9	1,044	19,804	19,151	19,058	18,989
0.99	95	20,502	20,019	19,561	19,455
0.995	47	20,877	20,300	19,745	19,555
0.99873	12	21,322	20,790	20,090	19,768

part time shifts being defined as 4 hour and 6.5 hour shifts. [$\alpha = 0.8$ in Equation (11).] Shifts of length 6.5 hours have a one half hour break, whereas shifts of length 4.0 hours will not. In model runs in which job switching is allowed we restrict it to at most 30 percent of the workforce. [$\gamma = 0.3$ in Equation (12).] Switches occurring in mid shift result in a half hour break for *all* job switching employees full time or part time. This means, for example, that a type 2 worker who works a four hour shift that is split between stations B and F will spend the first two hours at station B, one half hour switching over to station F, and then two more hours working at station F; so, even though she does not officially get a 30 minute meal break, there is a 30 minute break in the shift after 2 hours, and the entire amount of time spent in the factory is 4.5 hours. We assume that shifts may begin at four different *times* during the day: 0, 6, 12, and 16 hours (i.e., time periods 1, 13, 25 and 33, respectively).

We label eight different policies in terms of possible combinations of the first three decisions and then perform runs parameterized on the allowed residual inventory level. The properties of the eight policies are defined in the following:

Policy	Part Time	Jobswitching	Backlogging
r0	No	No	No
r0b	No	No	Yes
r1	No	Yes	No
r1b	No	Yes	Yes
r2	Yes	No	No
r2b	Yes	No	Yes
r3	Yes	Yes	No
r3b	Yes	Yes	Yes

We minimize labor costs for the different policies with different values for λ .

For this problem when we perform the feasibility test of §3 we compute that the inventory at the end of the day is 11.95 units of work; this is equivalent to $\lambda = 0.99873$. This means there will always be at least 11.95 units of work carried over from one day to the next. For this example the feasibility check computations are very simple. When P^* is raised to successively higher powers, it quickly reduces to the zero matrix. Therefore, the inverse of $[I - P^{*T}]$ is the identity matrix, and we do not have to compute many terms of the summation in Equation (13).

The following numerical results demonstrate how adding flexibility to the flow of work and to the use of the workforce give large benefits in labor cost reductions. Furthermore we see that as the required service level becomes more demanding, the more flexible systems adapt in a less expensive way than the relatively inflexible ones. The model run results for the four cases of backlogging are shown in Exhibit 6. When the policies with no backlogging are used, the system performs at the highest service rate which is equivalent to setting λ to 0.99873. The labor costs associated with the various policies are: r0: \$49,947 r1: \$46,731 r2: \$39,650 r3: \$36,850, as opposed to the following (dramatically reduced) costs associated with the corresponding backlogging policies with λ equal to 0.99873: r0b: \$30,208 r1b: \$28,192 r2b: \$24,945 r3b: \$23,819.

In Exhibit 6 we can see that when backlogging is allowed the highest cost is always with the policy r0b, i.e. when no part time or job switching is allowed. When this policy is applied with the 0.99873 service level the cost is \$30,208 which is 82% of the least expensive policy

Exhibit 9: Optimal Objective Function Values, Four Start Times, Smoother Inputs, Backlogging

lambda	max final inv.	r0b	r1b	r2b	r3b
0.5	9,400	21,618	21,465	20,952	20,914
0.8	2,350	21,618	21,465	20,952	20,914
0.9	1,044	21,618	21,465	20,952	20,914
0.95	495	21,618	21,466	20,952	20,917
0.99	95	22,186	22,048	21,359	21,305
0.995	47	22,663	22,535	21,475	21,425
0.99872	12	23,746	23,365	22,034	21,806
0.99918	7.7	25,924	24,782	22,381	22,093

with no backlogging. Thus we see the extreme importance of backlogging, an option that allows smoother workflows and less paid lost time (i.e., time during which workers are paid but not productively working).

Exhibit 7 shows the effect that different settings for λ (0.5, 0.99, 0.99873) have upon the inventory levels over the course of the day. In addition we have plotted the inventory level for the no backlogging policies and the exogenous "peaked" input work profile of Exhibit 5.

Next we increase the number of allowable start times to six evenly spaced throughout the day, commencing at times 0, 4, 8, 12, 16, and 20 hours. As expected this leads to lower labor costs. With no backlogging, costs are: r0: \$40,100 r1: \$39,751 r2: \$33,815 r3: \$32,291.

The costs with backlogging are shown in Exhibit 8.

We next flatten the profile of the exogenous work input into the system. The results follow the same pattern as before; but, as expected, the more even input workflow is easier to cope with and results in lower labor costs when backlogging is allowed. However, when no backlogging is permitted, the flatter demand requires

more staffing throughout the day and thus increases labor costs. When the original *four* start times are used the costs for the no backlogging policies are: r0: \$50,119 r1: \$46,344 r2: \$39,342 r3: \$36,418.

The corresponding costs with backlogging are shown in Exhibit 9.

When the *six* start times are used, the benefits over four start times for the no backlogging policies are sharper than when we used a peaked input profile. The costs with no backlogging are: r0: \$33,00 r1: \$32,771 r2: \$28,951 r3: \$28,291.

The corresponding costs with backlogging are shown in Exhibit 10.

We plot the inventory levels with four start times for the smoother input (see Exhibit 11). The maximum λ in this case is 0.99918.

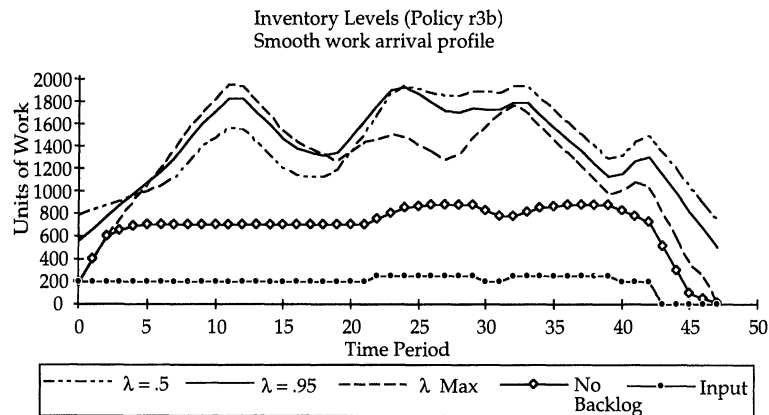
5. Summary

We have presented and executed an object-oriented linear programming (OOLP) model of a high volume fac-

Exhibit 10: Optimal Objective Function Values, Six Start Times, Smoother Inputs, Backlogging

lambda	max final inv.	r0b	r1b	r2b	r3b
0.5	9,400	21,129	21,021	20,797	20,759
0.9	1,044	21,129	21,021	20,797	20,759
0.95	495	21,129	21,021	20,797	20,770
0.99	95	21,557	21,498	21,065	21,038
0.995	47	22,028	21,975	21,179	21,140
0.99918	7.71	24,222	23,930	21,866	21,715

Exhibit 11



tory. We believe the key results are (1) model-relevant for certain firms operating such factories, providing a generic modeling structure in which to analyze a wide variety of operations; (2) policy relevant for managers and workers, since the results demonstrate the cost advantages of cross-trained, flexible workers and inventorying of work-in-progress; (3) product-relevant for operations researchers and management scientists, since a large and perhaps foreboding LP model can be developed and used with ease via the bundling of functions within an object-oriented computer software implementation. Within the realm of linear systems (and hence linear programming), there is little reason to believe that more sophisticated "icons" could not be developed to represent the LTI behavior of systems more complex than those discussed here. Going to nonlinear programming, even more complex system components could be modeled in this way, still providing ease of use by the nontechnical manager or planner.¹

¹ The authors thank the National Science Foundation (Grant SES 911962) and NSERC for support of this work. We also thank Rob Freund and Steve Graves for helpful comments on an earlier draft.

References

- Baker, Kenneth R., "An Optimal Procedure for Allocating Manpower with Cyclic Requirements," *AIIE Trans*, 5, 2 (1973), 119–126.
- , "Workforce Allocation in Cyclical Scheduling Problems: A Survey," *Oper. Res. Quarterly*, 27, 1 (1976), 155–167.
- Bartholdi, John J., III, "Cyclical Scheduling Via Integer Programs with Circular Ones," *Oper. Res.*, 28, 5 (1980).
- Bazaraa, Mokhtar S., *Linear Programming and Network Flows*, John Wiley and Sons, New York, 1990.
- Bechtold, S. E. and L. W. Jacobs, "Implicit Modeling of Flexible Break Assignments in Optimal Shift Scheduling," *Management Sci.*, 36, 11 (1990), 1339–1351.
- Berman, Oded and Richard C. Larson, "An LP Model for Workforce and Workflow Scheduling," paper presented at Symposium: *The Service Productivity & Quality Challenge*, Fishman-Davidson Center for the Study of the Service Sector, Univ. of Pennsylvania, Wharton School, Philadelphia, PA, 1992.
- Bixby, Robert, "Very Large-Scale Linear Programming: A Case Study in Combining Interior Point and Simplex Methods," *Oper. Res.*, 40, 5 (1992), 885–897.
- Cahn, M. F., R. C. Larson, and O. Berman, "A Linear Programming Model to Analyze a Flexible USPS Workforce," Operations Research Society of America/The Institute of Management Sciences (Joint National Meeting), San Francisco, CA, November 1992.
- Edie, L. C., "Traffic Delays at Toll Booths," *Oper. Res.*, 2, 2 (1954), 107–138.
- Emmons, H. and R. N. Burns, "Off-Day Scheduling with Hierarchical Worker Categories," *Oper. Res.*, 39, 3 (1991), 484–495.
- Geoffrion, A. M., "An Introduction to Structured Modeling," *Management Science*, 33, 5 (1987), 547–588.
- , "The Formal Aspects of Structural Modeling," *Oper. Res.*, 37, 1 (1989), 30–51.
- , "The SML Language for Structural Modeling: Levels 1 and 2," *Oper. Res.*, 40, 1 (1992), 38–57.
- , "The SML Language for Structural Modeling: Levels 3 and 4," *Oper. Res.*, 40, 1 (1992), 58–75.
- Glover, Fred, "The General Employee Scheduling Problem: An Integration of Management Science and Artificial Intelligence," *Computers and Oper. Res.*, 13, 4 (1986), 563–573.
- Henderson, W. B., "Heuristic Methods for Telephone Operator Shift Scheduling: An Experimental Analysis," *Management Sci.*, 22 (1976), 1372–1380.
- , "Determining Optimal Shift Schedules for Telephone Traffic Exchange Operators," *Decision Sci.*, 10 (1977), 126–135.

- Howard, R. A., *Dynamic Probabilistic Systems*, Wiley, New York, 1971.
- Jackson, J. R., "Networks of Waiting Lines," *Oper. Res.*, 5 (1957), 518–521.
- Mabert, V. A., "The Detail Scheduling of a Part-Time Work Force: A Case Study of Teller Staffing," *Decision Sci.*, 8 (1977), 109–120.
- Maier-Roth, C., "Cyclic Scheduling and Allocation of Nursing Staff," *Socio-Economic Planning Sci.*, 7 (1973), 471–487.
- Morris, James and Michael Showalter, "Simple Approaches to Shift, Days-Off and Tour Scheduling Problems," *Management Sci.*, 29, 8 (1983), 942–950.
- Murty, Katta G., *Linear Programming*, John Wiley and Sons, New York, 1983.
- Segal, M., "The Operator-Scheduling Problem: A Network Flow Approach," *Oper. Res.*, 22, 4 (1974), 808–823.
- Shapiro, J. F., *Mathematical Programming Structures and Algorithms*, John Wiley and Sons, New York, 1979.
- Sittler, R. W., "Systems Analysis of Discrete Markov Processes," *I.R.E. Trans. Circuit Theory*, CT-3, 1 (1956), p. 257.
- United States Dept. of Labor, "Monthly Labor Review August 1992," 115, 8 (1992), p. 74.
- United States Government, "Budget of United States Government Fiscal Year 1993," U.S. Gov. Printing Office, Washington D.C., A1-1020, 1992.
- Warner, D., "A Mathematical Programming Model for Scheduling Nursing Personnel in Hospitals," *Management Sci.*, 19 (1972), 411–422.

Accepted by Stephen C. Graves; received February 18, 1994. This paper has been with the authors 7 months for 2 revisions.