Scheduling Workforce and Workflow in a Service Factory

O. Berman, R. Larson, E. Pinker

OR 287-94 February 1994

Scheduling Workforce and Workflow in a Service Factory

Oded Berman, University of Toronto

Richard C. Larson, Edieal Pinker
Operations Research Center,
Massachusetts Institute of Technology

Do not cite or quote without authors' permission

February 18, 1994

Scheduling Workforce and Workflow in a Service Factory

Oded Berman, University of Toronto

Richard C. Larson, Edieal Pinker
Operations Research Center,
Massachusetts Institute of Technology

ABSTRACT

We define a service factory to be a network of service-related-workstations, at which assigned workers process work-in-progress that flows through the workstations. Examples of service factory work include mail processing and sorting, check processing and telephoned order processing. Exogenous work may enter the factory at any workstation according to any time-of-day profile. Work-in-progress flows though 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 midshift. 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, contractural 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.

Key words: Linear programming, scheduling, service, factory, object-oriented, queueing, queueing networks, Markov chains

List of Exhibits

		Page
Exhibit 1	Glossary	9
Exhibit 2.	Schematic Diagram of Generic Workstation	17
Exhibit 3	Summary of LP Model	24
Exhibit 4	Six Station Service Factory	29
Exhibit 5	The Six Allowed Worker Types for Illustrative Example	30
Exhibit 6	Time-of-Day Exogenous Input Work Profile	31
Exhibit 7	Results of the First Set of Executions of the Model	34
Exhibit 8	Inventory Levels (policy r3b)	35
Exhibit 9:	Optimal Objective Function Values, Six Start Times, Backlogging	35
Exhibit 10:	Optimal Objective Function Values, Four Start Times, Smoother Inputs, Backlogging	36
Exhibit 11:	Optimal Objective Function Values, Six Start Times, Smoother Inputs, Backlogging	36
Exhibit 12:	Inventory Levels (policy r3b), smoother inputs	37

The "decade of the services industries" is a popular label for the 1990's. With about three quarters of the jobs and two thirds of the GNP of the U.S.A. associated with services, productivity in services is of paramount concern. Other industrialized countries, in western Europe and Japan for instance, are focusing efforts on productivity improvements in their services industries as well.

One important component of the services industries in these countries is the "service factory," which we define to be a facility that processes and sorts papers, parcels, files, transactions and other usually paper-based or computer-based entities through a network of processing stages toward some definition of completion. A service factory is distinguished from a manufacturing factory in that no product is assembled in a service factory; rather, a service or sequence of services is provided to the entities passing through the service factory.

An example of a service factory is a mail processing center (MPC) of the United States Postal Service (USPS). The inputs to an MPC are both outbound and inbound mail requiring 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 dependant; the fully sorted mail must be on the loading docks prior to pre-specified "dispatch deadlines."

Other examples of service 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, and in many different governmental offices dealing with the processing of various types of applications.

For simplicity in this paper we shall use the nomenclature of the MPC to motivate, illustrate and crystallize the concepts involved. An MPC-focused earlier version of the model was instrumental in assisting the second author's expert testimony before a five member labor/management arbitration panel. 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 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, the 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, Berman 1992). That decision is currently saving the USPS hundreds of millions of dollars per year.

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

We wish to design the operation of the service factory to minimize variable costs while meeting constraints of various types. The largest source of variable cost is labor, and scheduling the workforce is the focus of our effort. But, as we shall see, there are other important controllable quantities, too, particularly the workflow through the service factory. We shall want to deal with both workforce and workflow scheduling simultaneously.

A workforce schedule is an assignment of workers to tasks and times of work. A workflow schedule indicates the amount of work processed at each workstation, by time, and the amount temporarily held in buffers. In simple situations, in which workflow scheduling is not an option, a good workforce schedule will match labor force levels as closely as possible to empirically derived work demand. For the labor intensive service industries that face a varying work demand throughout the day, generating a good workforce schedule can be a complicated task. It is not surprising that this problem has been considered for such service industries as nursing [Warner 1972], bank tellers [Mabert 1977], telephone operators [Henderson 1977], and fast food servers [Glover 1986].

In the operations research literature the approaches to the labor scheduling problem have been shaped 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 time-varying demand. The demand or required work level is assumed to have been determined empirically by methods such as that in [Edie 1954], and the objective is to minimize the amount of labor used while providing an acceptable level of service. Various 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 analytic solutions could be derived [Baker 1976], but 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 were 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 who is 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 non-homogeneous labor pool of limited size with great flexibility in assignment of days off, employee activity preferences, breaks, part time work, etc. Although very complex this formulation too is part of the bank teller paradigm, since its underlying assumption is that work performed by one employee has no influence on the rate at which work arrives to another employee who performs a different task.

On the other side of the research spectrum are studies that focus primarily upon workflow. [Graves 1986a] formulates a stochastic model of workflow through a job shop that is very similar in structure to the way workflow is modelled here. He develops a discrete-time, continuous flow model for the movement of work in the system. His model is different from the one here mainly in terms of the randomness in exogenous work input and simple noise in work transfers from station to station. He does not explicitly model the workforce but instead generates simplified expectations of labor needs that are suggested by the workflows. In [Graves, *et. al.* 1986b] an LP planning model for a mental health care system is developed that optimizes the flow of patients through a health care system according to certain measures of social welfare. This model does not include the assignment of resources to points of the system as decision variables.

The existing research that has linked labor with workflow has focused on worker dispatching rules in stochastic job-shop type work environments. Much of this research is surveyed in [Trevelen 1989] and is characterized by simulation studies of the performance of various dispatching policies when not all of the equipment in a shop is fully staffed. These studies do not consider how to generate schedules for individual workers and do not take into account flexibility in start times or shift length.

Our model combines the bank teller and the factory workflow paradigms. The service 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 service factory, with different customers perhaps taking different routes. Any customer may enter the service factory exogenously at any station and exit the factory at any station. Exogenous 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, we will assume that the various customer flows are large enough so that all the integer variables may be accurately approximated as continuous ("relaxed") variables. By allowing the queue of customers, i.e., work-in-process, 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 midshift. 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 yet satisfy constraints such as required time windows for customers to emerge "fully serviced" from the network.

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 intractibility or conceptual complexity.

The linear program derived from the modeling analysis is an *object oriented LP (OOLP)*, due to the building block nature of the model; the resulting model can be thought of a set of inter-related 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. Our belief is that the ultimate user of this model will see no equations in its generation or use, only icons on a computer screen that - with a mouse click - can be "opened" to input or change parameters related to each icon. The ideas are compatible with many of the suggestions of Geoffrion toward building a "language for structured modeling." (Geoffrion 1987, 89, 92, 92)

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

Glossary: Exhibit 1

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 arrives exogenously to station j and is presented there at the beginning of time period t, $t = 1, 2, \ldots, T$.

B = total daily exogenous work input, or $B \equiv \sum_{j=1}^{n} \sum_{t=1}^{T} b_{jt}$

p_{ij} = fraction of jobs processed at station i that are routed next to station j, $i,j = 1,\ldots,n$.

$\boldsymbol{H}$ = the set of all allowed shift lengths

$\boldsymbol{ST}$ = the set of all allowed starting times for shifts

M_{ki} = 1 (0) if worker type k can (cannot) perform work at station i.

(j_1, j_2) = a pair of workstations, j_1 and j_2, representing a worker's workstation assignments for the first and second half of her shift, respectively

K = number of worker types

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; ; $k = 1, \ldots, K$; $(j_1, j_2) \in A_k$; $h \in \boldsymbol{H}$; $\tau \in \boldsymbol{ST}$.

$C_{k,(j_1,j_2),h,\tau}$ =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; ; $k = 1, \ldots, K$; $(j_1, j_2) \in A_k$; $h \in \boldsymbol{H}$; $\tau \in \boldsymbol{ST}$.

$\beta_{k,(j_1,j_2),h,\tau,t}$ = 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; ; $k = 1, \ldots, K$; $(j_1, j_2) \in A_k$; $h \in \boldsymbol{H}$; $\tau \in \boldsymbol{ST}$., $t=\tau,\tau+1,\ldots,\tau+h-1$.

I_{jt} = total quantity of new work presented to station j at the start of period t

$R_{j,t-1}$ = 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

2. Formulation of the Problem

2.1. Overview

The service factory is arranged in a general way that resembles a manufacturing factory. The service 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. Each workstation has a finite capacity buffer that can be used to inventory locally work waiting to be processed next by that particular 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 of work (sometimes called "jobs") that arrives exogenously to station j and is presented there at the beginning of time period t, $t = 1, 2, \ldots, T$. 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 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 flow of some work from one station to another. It is convenient to add to the network one additional (dummy station) node $n + 1$ which "collects" all the final output from the service factory.

We denote the fraction of work output that flows from station i to station j by p_{ij}, $i,j=1,2,...,n+1$, where $\sum_{j=1}^{n+1} p_{ij}=1$, $i=1,2,\ldots,n+1$ 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 there 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 so that we can treat all flows as deterministic quantities, the route of any individual job can be considered to be 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. Feasibilty 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 only one recurrent state, the "trap" state n+1; all states other than the trap state are transient states. This implies that eventually, with probability one, each piece of work will reach trap state n+1, meaning leaving the service factory. The *path* taken by any individual job is independent of the values of the decision variables, assuming a feasible solution. The *time* that a unit of work is resident in the system is, of course, 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. An analysis of the validilty 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 MPC's, for instance, typical coefficients of variation are less than 0.05, implying that the deterministic assumption is a reasonable one.

One can forgo completely the stochastic interpretation of the model and view the entire model as deterministic. In that case, the "one step Markov transition probabilities" become deterministic "branching ratios," indicating the relative fractions of work routed to each respective subsequent workstation. In fact, the branching ratios could add to a value greater than one, reflecting the possibility that in some service factories a unit of work at a workstation can be "split up" into two or more pieces that are subsequently processed separately along different paths, perhaps to be reunited later "down the line" in the factory. While such a generalization is not conceptually difficult, for simplicity in this paper, we do not explicitly consider work splitting and reuniting, and we require each set of branching ratios, or transition probabilities, to sum to one.

2.3. *The Decision Variables*

The objective is to assign workers by time and task so as 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 the number of time periods in the shift and the time period during which the shift is started, respectively. Let H be the set of all allowed shift lengths (e.g., $H = \{4, 6, 8, 12\}$) and ST be the set of all allowed starting times for shifts (e.g., $ST = \{4, 8, 12, 16, 20, 24\}$).

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 K be the number of worker types. Let M be a matrix with rows corresponding to worker types and columns corresponding to stations, where

$$M_{kj} = \begin{cases} 1 & \text{if worker type } k \text{ is trained on station } j \\ 0 & \text{otherwise} \end{cases}$$

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

> $\beta_{k,(j_1, j_2), h, \tau,t}$ = 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; ; $k = 1, \ldots, K$; $(j_1, j_2) \in A_k$; $h \in H$; $\tau \in ST.$, $t=\tau,\tau+1,\ldots,\tau+h-1$.

This coefficient is equal to zero if M_{kj_1} or M_{kj_2} is zero, i.e., if the worker is not qualified to work at station j_1 or j_2, and is usually positive otherwise; for any particular period t the coefficient may be zero or a small number, reflecting a scheduled meal break, time in transit, etc. When the worker is working at full productivity, we assume $\beta_{k,(j_1, j_2), h, \tau,t} \gg 1$, i.e., that a qualified worker at a station can process many pieces of work per time period. For instance, a worker assigned to an mechanized letter sorting machine in an MPC can process more than 3000 letters per hour. This assumption of processing many pieces of work per unit time allows us to model work movements as continuous flows, not as discrete units or pieces as might be found in some job shop models. The flow assumption is critical to our model.

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:

$X_{k,(j_1,j_2),h,\tau}$ = 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; ; $k = 1, \ldots, K$; $(j_1, j_2) \in A_k$; $h \in H$; $\tau \in ST$.

$C_{k,(j_1,j_2),h,\tau}$ = 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; ; $k = 1, \ldots, K$; $(j_1, j_2) \in A_k$; $h \in H$; $\tau \in ST$.

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

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

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. The model to be developed below will reveal the constraint set against which this minimization is to be performed.

A second set of decision variables, which are implicit, deals with workflow. That is, we assume that there are (finite capacity) buffers at each workstation that can be used to inventory work so that not all work that arrives to a workstation at the start of time period t needs to be processed during period t. Some work, in addition to jobs that require rework due to defects, 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. This progression is determined by a set of linear relationships that specify the operation of each station and its interaction with each other station. Those relationships provide inequality and equality constraints for the model. We develop these relationships within the context of generic workstation j, depicted in Exhibit 2.

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 service 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. This total quantity is the number of jobs in the buffer at station *j* at the beginning of period *t* and represents the maximum amount of potential work that could be processed at station *j during* period *t*.

The *new work* arriving at station *j* at the start of period *t* is the sum of the exogenous work presented to station *j* and the sum of all the work delivered from other workstations, excluding station *j*. Defining

I_{jt} = total quantity of new work presented to station *j* at the start of period *t*,

we have

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

where

$O_{i(t-1)}$ = 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 must be careful to note that the output $O_{i(t-1)}$ represents the total number of units of work processed at station *i* during period *t*-1, including work that has to be retained at station *i* due to defects. The quantity $p_{ij}O_{i(t-1)}$ represents the number of units of work that flow from station *i* to station *j* (virtually instantaneously) at the end of period *t*-1.

The *residual work* remaining at station *j* from period *t* - 1 is the sum of (1) the number of jobs processed during period *t* - 1 at station *i* that must be reworked at station *j* due to defects and (2) 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. Define

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

$R_{j,t-1}$ = total work remaining at station *j* from period *t* - 1.

Y_{jt} is the maximum potential quantity of work that can be processed at station *j* during period *t*. The actual quantity of work will of course depend on the number and types of workers assigned to station *j* during period *t*. If the number of workers assigned is sufficiently large so that their capacity to perform work during the period equals or exceeds the work in the buffer, then the buffer will be depleted entirely during period *t*; otherwise there will remain some quantity of unworked jobs in the buffer for the next period.

Recognizing that the buffer content at the start of a period is the sum of the new work delivered at the start of that period and residual work remaining from the previous period, we can write

$$\begin{aligned} Y_{jt} &= I_{jt} + R_{j,t-1}, \qquad t = 2, 3, \ldots, T \\ Y_{jt} &= I_{jt} + R_{j,T}, \qquad t = 1 \end{aligned} \tag{2}$$

where the residual work is the sum of defect-related (re)work and work remaining in the buffer, i.e.,

$$R_{jt} = p_{jj}O_{jt} + [Y_{jt} - O_{jt}]\,. \tag{3}$$

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

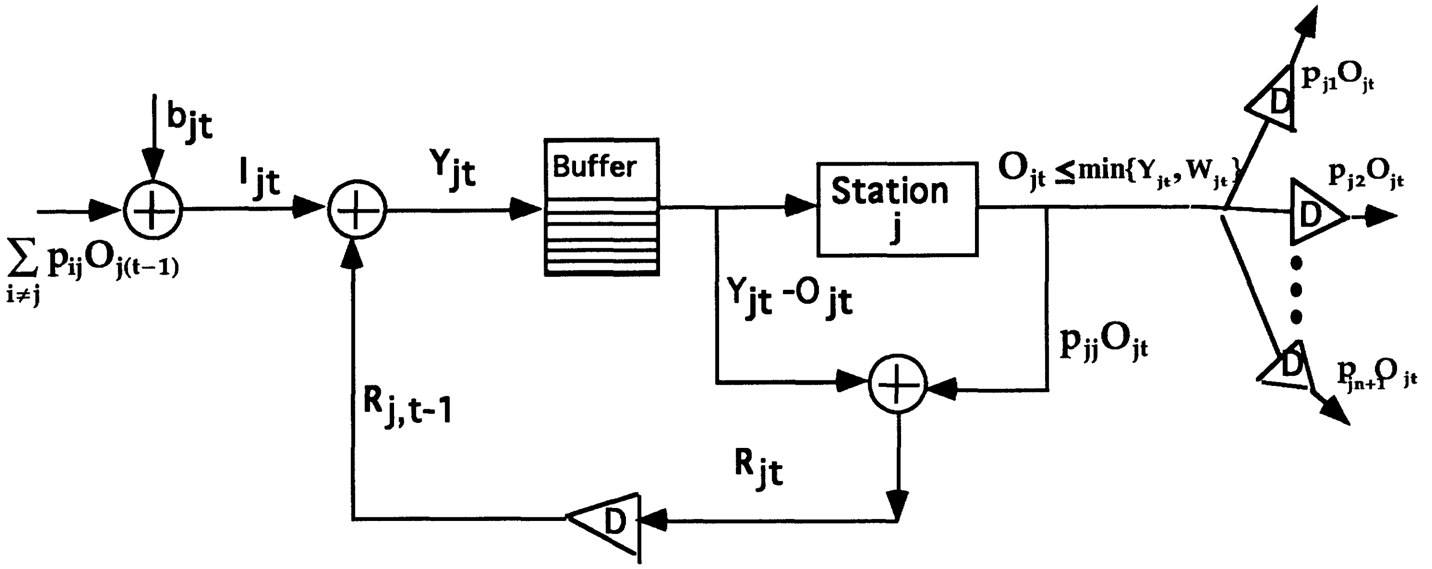

Exhibit 2. Schematic Diagram of Generic Workstation

We now develop expressions for the amount of output from workstation *j* during a given time period. In many instances, this output will be either the maximum amount that the currently deployed workers can produce or the amount in the buffer, whichever is smaller. However, there may be buffer capacity constraints "downstream" from workstation *j*, constraints that would be violated if the maximum possible output were to be produced at station *j*. In such circumstances, the downstream buffer capacity constraints back up or block production "upstream" and temporarily curtail production there. If the buffer capacity constraints do not limit production, minimization of the objective function will force the maximum possible output from station *j*.

The modeling to accomplish the desired behavior is rather simple. If we define

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

and recalling that

O_{jt} = "output" of station *j* during period *t*

= number of jobs worked by assigned personnel at station *j* during period *t* ,

then we can write

$$O_{jt} = \mathrm{MIN}\{Y_{jt}, W_{jt}\}, \tag{4}$$

implying the inequalities,

$$\begin{aligned} O_{jt} &\le Y_{jt} && \text{(a)} \\ O_{jt} &\le W_{jt}. && \text{(b)} \end{aligned} \tag{5}$$

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

$$\begin{aligned} W_{jt} = \sum_{k \in M_j} \sum_{h \in H} \{ \sum_{\tau \in F_{th}} \beta_{k,(j,j),h,\tau,t} X_{k,(j,j),h,\tau} + \sum_{\substack{n_2 \in W_k \\ n_2 \neq j}} \sum_{\tau \in G_{th}} \beta_{k,(j,n_2),h,\tau,t} X_{k,(j,n_2),h,\tau} + \\ \sum_{\substack{n_1 \in W_k \\ n_1 \neq j}} \sum_{\tau \in Q_{th}} \beta_{k,(n_1,j),h,\tau,t} X_{k,(n_1,j),h,\tau} \} \end{aligned} \tag{6}$$

where

M_j is the set of qualified worker types for station *j*

W_k is the set of stations for which worker types *k* is qualified

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

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

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

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, . . . , T, and is forwarded to node n+1 at the *start* of periods 2, 3, . . . , T, 1, respectively; work completed at the end of period T (i.e., at the end of the day), since it is not passed to the collection buffer until *time* $T^+ = 0^+$, is counted as the first completed work for the *next* day. In terms of defined variables, we note that

$Y_{n+1,t}$ = total number of units of work at node n+1 at the start of period t

= total work completed up to and including the start of period t,

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

$$Y_{n+1,t} = \sum_{i=1}^{n} p_{i,n+1}O_{i(t-1)} + Y_{n+1,(t-1)}\,,\ t=2,\ldots,T \tag{7}$$

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 service factory must include constraints on capacities of certain physical facilities to accomodate workers and/or work.

With regard to workers, in most service 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

$$\sum_{k\in M_j}\sum_{h\in H}\{\sum_{\tau\in F_{th}} X_{k,(j,j),h,\tau}+\sum_{\substack{n_2\in W_k\\ n_2\neq j}}\sum_{\tau\in G_{th}} X_{k,(j,n_2),h,\tau}+\sum_{\substack{n_1\in W_k\\ n_1\neq j}}\sum_{\tau\in Q_{th}} X_{k,(n_1,j),h,\tau}\}\leq w_{jt}, \qquad (8)$$

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,

$$Y_{jt}\leq \gamma_{jt}, j=1,\ldots,n; t=1,\ldots,T. \qquad (9)$$

These constraints can easily be added to the workstation macro.

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 service factory (i.e., exiting via node n+1) is equal in magnitude to the amount of work *presented* exogenously 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 $Y_{n+1,T} = B$, i.e., the total output of the system during a day must equal the total input, as measured in units of work.

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 (at the end of period T) cannot contain as work-in-process inventory more than 5% of the total input for the day where, again, input is the sum of exogenous inputs and work left over from the previous day.

Without a speed-of-work constraint, unfinished work would tend to fill up the buffers, thereby greatly smoothing the workflow as experienced by assigned workers, who themselves would tend to be the least expensive alternative for accomplishing the required tasks. An increasingly tight speed-of-work constraint tends to propagate through the factory the temporal variability of the workflow as originally presented to the factory and to delimit the types and costs of workers available to do the work; all this, in turn, increases the total cost of operating the system. A tight speed-of-work constraint implies low in-process inventories; a loose speed-of-work constraint implies large in-process inventories. In fact, one can prove a Little's Law of queueing for this model: time average in-process inventory = (daily arrival rate of work) * (average time spent in the service factory by a random unit of work).

Returning to the model formulation, the work left over from the previous day can be considered to be divided into two components: (1) the work remaining in buffers; (2) the work remaining in the system being transported virtually instantaneously at the end of period T from one node i to another node j; $i, j = 1, 2, \ldots, n$. The quantity of work remaining at station j at the end of period T is the residual work R_{jT}; all of this work will reside in the buffer at station j at the start of the next day. The total quantity of work leaving node j to the set of all other internal nodes i (i=1,..., n; $i \neq j$) at the end of period T can be written as $O_{jT}(1 - p_{jj} - p_{j,n+1})$; all of this work will reside in other buffers i (i=1,..., n; $i{\neq}j$) at the start of the next day.

Generalizing the 95% service level to λ percent, we obtain the constraint

$$\sum_{j=1}^{n} (R_{jT} + O_{jT}(1 - p_{jj} - p_{j,n+1})\,) \leq (1-\lambda)\{\, B + \sum_{j=1}^{n} R_{jT} + O_{jT}(1 - p_{jj} - p_{j,n+1})\}$$

or, equivalently,

$$\sum_{j=1}^{n} (R_{jT} + O_{jT}\{1 - p_{jj} - p_{j,n+1}\}\,) \leq \{\frac{1-\lambda}{\lambda}\}B. \qquad (10)$$

In addition to the speed of work constraint represented by Eq.(10), management may wish to constrain the *pattern of flow* of work completed over the course of the day. By this we mean that "X percent of the day's work should be completed by the end of period t." If we define χ_t as the desired fraction of the day's work to be completed by the end of period t, and recognizing the unit counting delay associated with node n+1, then this type of constraint is given by

$$\{Y_{n+1,t+1}/B\} \geq \chi_t\,, \qquad t = 1, 2, \ldots, T{-}1. \qquad (11)$$

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

2.8. Managerial Constraints

A variety of managerial "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 managerial constraint can be expressed as

$$(1-\alpha)\{\sum_{k=1}^{K}\sum_{(j_1,j_2)\in A_k}\sum_{h\in H,h\geq 8}\sum_{\tau\in ST}X_{k,(j_1,j_2),h,\tau}\}-\alpha\{\sum_{k=1}^{K}\sum_{(j_1,j_2)\in A_k}\sum_{\substack{h\in H\\ h<8}}\sum_{\tau\in ST}X_{k,(j_1,j_2),h,\tau}\}\geq 0 \tag{12}$$

For another example suppose the number of workers who are allowed to switch between different stations can be at most ω percent of all workers scheduled. This managerial constraint can be written,

$$(1-\omega)\{\sum_{k=1}^{K}\sum_{\substack{(j_1,j_2)\in A_k\\ (j_1\neq j_2)}}\sum_{h\in H}\sum_{\tau\in ST}X_{k,(j_1,j_2),h,\tau}\}-\omega\{\sum_{k=1}^{K}\sum_{(j,j)\in A_k}\sum_{h\in H}\sum_{\tau\in ST}X_{k,(j,j),h,\tau}\}\leq 0 \tag{13}$$

.

The model can also include many other managerial side constraints.

The overall 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 physical, managerial and performance constraints.

For reader convenience a summary of the LP model is given in Exhibit 3.

Exhibit 3: Summary of LP Model

Objective Function: Minimize Cost of Workers:

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

Constraints:

1. System Dynamics

1.a. Buffer level=new work + residual work:

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

$= I_{jt} + R_{j,t\text{-}1}$, where $I_{jt} = b_{jt} + \sum_{i\neq j} p_{ij} O_{i(t-1)}$, and $R_{j,t-1} = p_{jj} O_{j(t-1)} + [Y_{j(t-1)} - O_{j(t-1)}]$.

$$Y_{jt} = b_{jt} + \sum_{j=1}^{n} p_{ij} O_{i(t-1)} + [Y_{j(t-1)} - O_{j(t-1)}] \text{ , } j=1,\dots,n;\ t=2,\dots,T$$

$$Y_{j1} = b_{j1} + \sum_{j=1}^{n} p_{ij} O_{iT} + [Y_{jT} - O_{jT}] \text{ , } j=1,\dots,n$$

1.b. Output of a Station: $O_{jt} \leq Y_{jt}$ and $O_{jt} \leq W_{jt}$, where

$$W_{jt} = \sum_{k\in M_j} \sum_{h\in H} \{ \sum_{\tau\in F_{th}} \beta_{k,(j,j),h,\tau,t} X_{k,(j,j),h,\tau} + \sum_{\substack{n_2\in W_k \\ n_2\neq j}} \sum_{\tau\in G_{th}} \beta_{k,(j,n_2),h,\tau,t} X_{k,(j,n_2),h,\tau} +$$
$$\sum_{\substack{n_1\in W_k \\ n_1\neq j}} \sum_{\tau\in Q_{th}} \beta_{k,(n_1,j),h,\tau,t} X_{k,(n_1,j),h,\tau} \}$$

1.c. Completed Work

$Y_{n+1,t}$ = total work completed up to and including period *t*,

$Y_{n+1,t} = \sum_{j=i}^{n} p_{i,n+1} O_{i(t-1)} + Y_{n+1,(t-1)}$, $t=2,\dots,T$, where $Y_{n+1,1} = \sum_{i=1}^{n} p_{i,n+1} O_{iT}$.

2. Resource Capacity Limitations

2.a. Maximum number of workers at a station

$$\sum_{k\in M_j}\sum_{h\in H}\{\sum_{\tau\in F_{th}} X_{k,(j,j),h,\tau}+\sum_{\substack{n_2\in W_k\\ n_2\neq j}}\sum_{\tau\in G_{th}} X_{k,(j,n_2),h,\tau}+\sum_{\substack{n_1\in W_k\\ n_1\neq j}}\sum_{\tau\in Q_{th}} X_{k,(n_1,j),h,\tau}\}\leq w_{jt},$$

2.b. Buffer Capacity

$$Y_{jt}\leq \gamma_{jt}, j=1,\ldots,n; t=1,\ldots,T.$$

3. Speed and Pattern of Work Flow Requirement

3.a. At least λ percent of each day's work must be done that day

$$\sum_{j=1}^{n} R_{jT}\leq\{\frac{1-\lambda}{\lambda}\}\sum_{j=1}^{n}\sum_{t=1}^{T} b_{jt}.$$

3.b. Percentage of day's work that must be completed by a given time.

$$\{Y_{n+1,t}/B\}\geq\chi_t$$

4. Side Constraints

The fraction of workers who are full time must equal or exceed α:

$$(1-\alpha)\{\sum_{k=1}^{K}\sum_{(j_1,j_2)\in A_k}\sum_{h\in H,h\geq 8}\sum_{\tau\in ST} X_{k,(j_1,j_2),h,\tau}\}-\alpha\{\sum_{k=1}^{K}\sum_{(j_1,j_2)\in A_k}\sum_{\substack{h\in H\\ h<8}}\sum_{\tau\in ST} X_{k,(j_1,j_2),h,\tau}\}\geq 0$$

The fraction of workers who may switch jobs at midshift cannot exceed ω:

$$(1-\omega)\{\sum_{k=1}^{K}\sum_{\substack{(j_1,j_2)\in A_k\\ (j_1\neq j_2)}}\sum_{h\in H}\sum_{\tau\in ST} X_{k,(j_1,j_2),h,\tau}\}-\omega\{\sum_{k=1}^{K}\sum_{(j,j)\in A_k}\sum_{h\in H}\sum_{\tau\in ST} X_{k,(j,j),h,\tau}\}\leq 0$$

5. Nonnegativity: All subscripted variables *X, O, Y, I, R* are nonnegative.

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 modelled.

There are $3nT$ constraints describing the system dynamics, nT constraints each for the buffer capacity and worker space capacity limitations, 2 constraints governing the speed and pattern of the workflow, and a constraint each for the limitations on the amount of jobswitching 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 $5nT$+4. For example in an 8 station system with T set at 48 to represent half-hour periods the number of constraints will be 1,924.

The number of variables relating to the flow of work are $2nT$, i.e. the variables Y_{jt} and O_{jt}. The number of labor variables $X_{k,(j1,j2),h,\tau}$ is $|\mathbf{H}|\,|\mathbf{ST}|\sum_k |A_k|$. For example in a workforce that has 10 different types of workers, each qualified to perform each of the 8 different tasks in the system the sum $\sum_k |A_k|$ would be 640. If there were 4 shift lengths and 6 start times the total number of labor variables would be 15,360. In the 8 station system modelled in Section 4 of this paper, in which a 24-hour-day is divided into 48 half-hour periods, there are 768 flow variables for a total of 16,128 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. The numbers of variables and constraints we are speaking of for realistic problems are not large, and the corresponding LP's can be solved "in minutes" on a modern workstation computer.

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. That is, we check to see if there exists a feasible solution for the case in which buffer capacities and staffing limits at the workstations are both infinite. 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.(10)] 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 (1) a large service factory with many workstations in succession; (2) late-in-the-day arrivals of exogenous work; and/or (3) 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 $\boldsymbol{F(t)} = (f_j(t)\ ), j=1, 2, \ldots, n+1$ Define the $n+1$-vector of exogenous inputs at period t, $\boldsymbol{b(t)} = (b_j(t))$, where $b_{n+1}(t) \equiv 0$. 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 $\pi_j \equiv O_{jT}, j = 1, 2, \ldots, n$. 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_{ij}). 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; $\pi_2 p_{22}$ is the amount of (re)work due to defects that remains at station 2; and $\pi_2 p_{2,n+1}$ is the amount of work leaving the system from station 2, to be counted as completed work during period 1 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,

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

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

$$F(3) = b(3) + F(2)\,P = b(3) + b(2)\,P + b(1)\,P^2 + \pi\ P^3,$$

or, in general,

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

We can write $F(T) = (F^*(T), B)$ $[\pi = (\pi^*, \mathrm{B})]$, where $F^*(T)$ $[\pi^*]$ is an n-vector corresponding to the n (real) workstations and the scaler 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 probabilty matrix P can also be partitioned, with P^* corresponding to the n by n square submatrix corresponding to P with the n+1st 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 $j=1, 2, \ldots, n$, due to the time-cyclic nature of the process, we have

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

or, solving for π^*,

$$\pi^* = [\sum_{k=0}^{T-1} b(T-k)\,P^{*k}]\,[I - P^{*T}]^{-1} \quad . \qquad (14)$$

Eq.(14) is guaranteed to have a solution since, due to P^* being substochastic, the matrix $[I - P^{*T}]$ is invertible.

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 the derivation of Eq.(10), we must have

$$\sum_{i=1}^{n} \pi_i^*(1 - p_{i,n+1}) \le (1 - \lambda)\,[\,B + \sum_{i=1}^{n} \pi_i^*(1 - p_{i,n+1})\,]$$

or,

$$\sum_{i=1}^{n} \pi_i^*(1 - p_{i,n+1}) \leq \{(1-\lambda)/\lambda\}B \qquad (15)$$

If Eq.(15) is not satisfied, then the problem has no feasible solution. Since Eq.(14) can be solved in $O(Tn^3)$ time, its use with Eq.(15) could save a much larger amount of computer time (and analyst's detailed data preparation time).

4. Illustrative Computational Results

The model presented in Section 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 stations arranged in the network depicted in object-oriented form in Exhibit 4; each lettered rectangular icon represents a workstation and nonzero inter-station workflows are depicted by directed arcs.

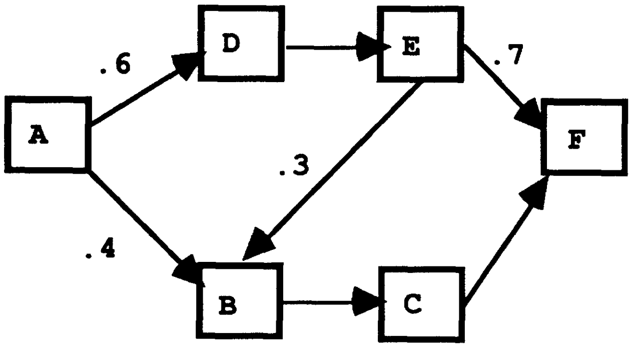

Exhibit 4: Six Station Service Factory

The work day is divided into $T = 48$ 30-minute time intervals, with interval 1 starting at midnight. Exogenous work enters the system only at station A and exits only at station F. At each of the stations B,C,D, and E there is a 5 percent defect-related rework rate for work processed there; this implies that $p_{BB} = p_{CC} = p_{DD} = p_{EE} = 0.05$. Hence, contrary to the appearance of the icon-oriented model in Exhibit 4, the model does include cycling or feedback, but such feedback (being local to each workstation) is not shown explicitly in Exhibit 4. 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 following transition probabilities:

$p_{AD} = 0.6$, $p_{AB} = 0.4$, $p_{DE} = 1$,$p_{BC} = 1$, $p_{CF} = 1$,$p_{EB} = 0.\ 3$, and $p_{EF} = 0.7$, shown adjacent to the respective directed arcs in Exhibit 4, with all other inter-workstation transitions occurring with probability 0.

We will assume that the labor pool potentially available for staffing the above system comprises the worker types shown in Exhibit 5. To illustrate interpretation of Exhibit 5, a type 1 worker is qualified at station A with a productivity of 60 units of work per hour and at station E with a productivity of 80 units of work per hour. The above productivity rates are maximum rates; in the numerical results that follow we 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 mid-shift meal break. [While we could express all model parameters in terms of our notations of Section 2, we choose not to do this in order to maximize intuitive understanding of the model. The ultimate model user will not see any Greek letters, sigma's for summations or other technical notations while using the model; we describe the model's setup and use here in a style compatible with object-oriented model usage.]

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

Exhibit 5: The Six Allowed Worker Types for Illustrative Example

* Units of work processed per hour

** Dollars per hour

Work is input into the system at station A according to the time-of-day exogenous work profile shown in Exhibit 6. 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 timespan 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 jobswitching allowed?
3. Is backlogging of work at workstations allowed?
4. How much residual inventory is left in the system at day's end?

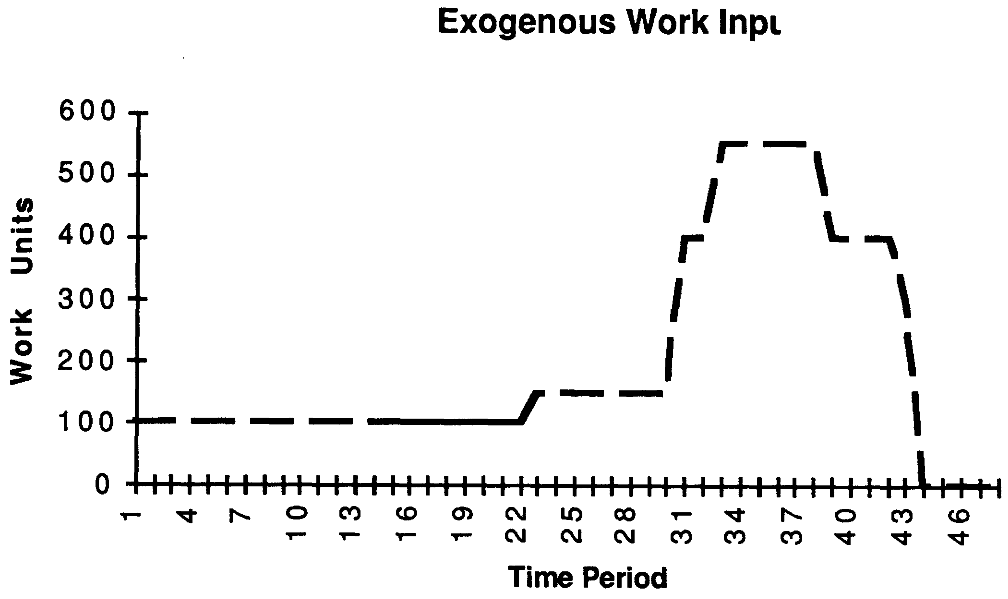

Exhibit 6: Time-of-Day Exogenous Input Work Profile

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 will restrict it to at most 20 percent of the workforce, with part time shifts being defined as 4 hour and 6.5 hour shifts. [This corresponds to $\alpha = 0.8$ in Eq. (12).] Shifts of length 6.5 hours will have a one half hour break, whereas shifts of length 4.0 hours will not. In model runs in which jobswitching is allowed we will restrict it to at most 30 percent of the workforce. [This corresponds to ω= 0.3 in Eq.(13).] Switches occuring in midshift result in a half hour break for *all* jobswitching 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 service 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., at the beginnings of time periods 1, 13, 25 and 33, respectively).

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

r0 - no backlogging, no jobswitching and no part time.
r0b - backlogging, no jobswtiching and no part time.
r1 - no backlogging, jobswitching and no part time.
r1b - backlogging, jobswitching and no part time.
r2 - no backlogging, no jobswitching and part time.
r2b - backlogging, no jobswitching and part time.
r3 - no backlogging, jobswitching and part time.
r3b - backlogging, jobswitching and part time.

We recall that λ represents the percent of all work brought to the factory (both exogenously and left over from the previous day) that must be output by the end of the day. We minimize labor costs for the different policies with different values for λ. It is important to observe that when no backlogging is allowed, i.e. all work at each station is processed during the time period that it arrives to the station, the system will perform as if λ were set to the highest service level that is feasible. That is, the constraint expressed in Eq.(10) is not relevant to the case of no backlogging, but the feasibility test methodology of Sec. 3 is relevant.

For this probem when we perform the feasibility test of Sec. 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 that given the exogenous work input profile of Exhibit 6 and the six station system layout of Exhibit 4, there will always be at least 11.95 units of work carried over from one day to the next. In the case we are examining here the computations involved in the feasibility check are very simple. It is easy to see that in our case when the matrix P^* is raised to successively higher powers it quickly reduces to the zero matrix. Therefore we do not have to compute the inverse of $[I-P^{*T}]$ since it is effectively the identity matrix and we do not have to compute many terms of the summation in Eq. (14).

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 7. 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 7 we can see that when backlogging *is* allowed the highest cost is always with the policy r0b, i.e. when no part time or jobswitching is allowed. When this policy is applied with the .99873 service level the cost is $30,208 which is 82% of the least expensive policy 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).

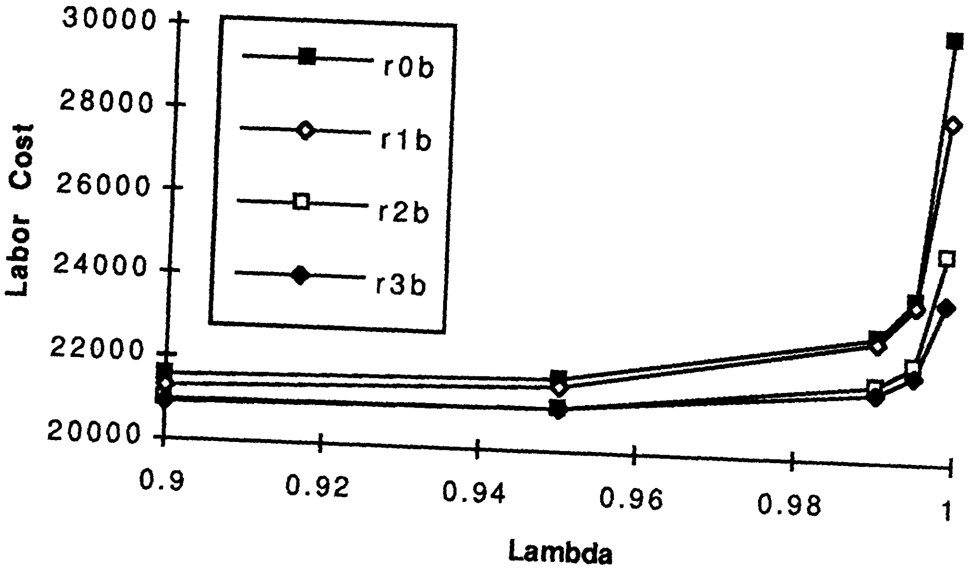

Exhibit 7: Results of the First Set of Executions of the Model

We can see the effect that different settings for λ have upon the inventory levels over the course of the day in Exhibit 8. In this exhibit we have plotted the inventory level during each time period of the day for three different values of λ, 0.5, 0.99, and the maximum feasible λ, of 0.99873 when the policy r3b is used. In addition we have plotted the inventory level for the no backlogging policies which perform according to the maximum lambda of 0.99873, and the peaked input profile.

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. the increase in the number of different start times leads to lower labor costs. The costs when no backlogging is allowed are: r0: $40,100 r1: $39,751 r2: $33,815 r3:$32,291. The objective function values for the backlogging policies are shown in Exhibit 9.

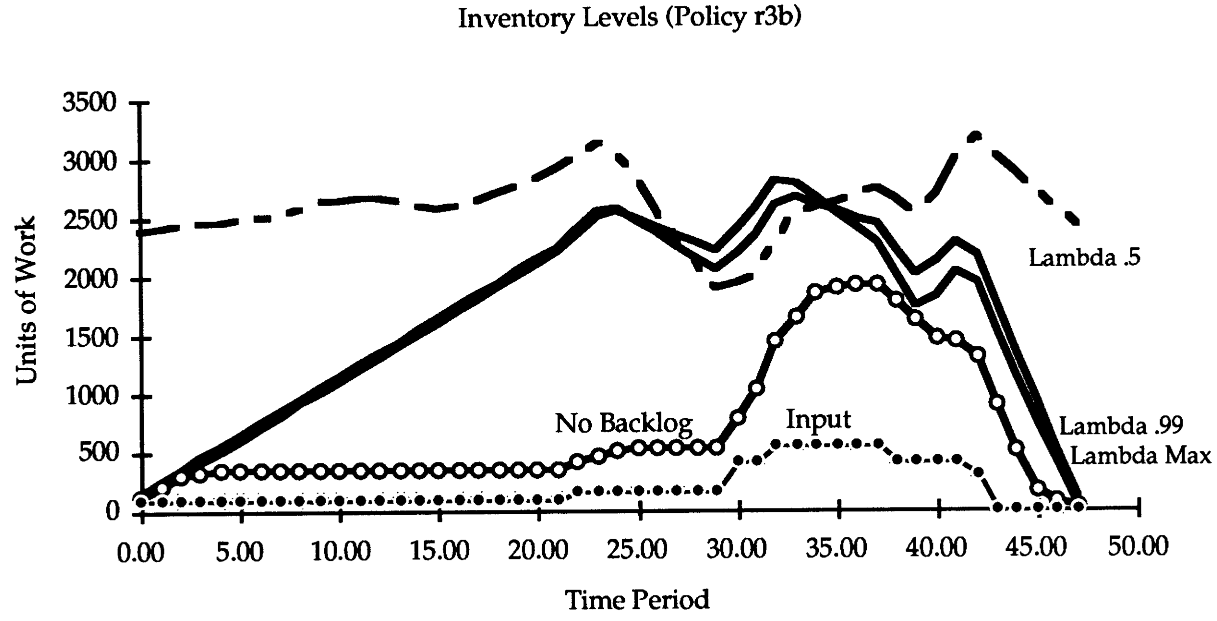

Exhibit 8

lambda	max final inv.	r0b	r1b	r2b	r3b
0.5	9400				
0.8	2350	19804	19151	19058	18989
0.9	1044	19804	19151	19058	18989
0.99	95	20502	20019	19561	19455
0.995	47	20877	20300	19745	19555
0.99873	12	21322	20790	20090	19768

Exhibit 9: Optimal Objective Function Values, Six Start Times, Backlogging

We next perform the same set of runs of the model flattening 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 objective values for the backlogging policies are shown in Exhibit 10.

lambda	max final inv.	r0b	r1b	r2b	r3b
0.5	9400	21618	21465	20952	20914
0.8	2350	21618	21465	20952	20914
0.9	1044	21618	21465	20952	20914
0.95	495	21618	21466	20952	20917
0.99	95	22186	22048	21359	21305
0.995	47	22663	22535	21475	21425
0.99872	12	23746	23365	22034	21806
0.99918	7.7	25924	24782	22381	22093

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

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 for the no backlogging policies are: r0:$33,00 r1: $32,771 r2: $28,951 r3: $28,291. The corresponding objective values for the backlogging policies are shown in Exhibit 11.

lambda	max final inv.	r0b	r1b	r2b	r3b
0.5	9400	21129	21021	20797	20759
0.9	1044	21129	21021	20797	20759
0.95	495	21129	21021	20797	20770
0.99	95	21557	21498	21065	21038
0.995	47	22028	21975	21179	21140
0.99918	7.71	24222	23930	21866	21715

Exhibit 11: Optimal Objective Function Values, Six Start Times,
Smoother Inputs, Backlogging

We can again plot the inventory levels as we did for the peaked input with four start times for the smoother input (see Exhibit 12). 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 service factory. We believe the key results are (1) model-relevant for services firms, providing a generic modeling structure in which to analyze a wide variety of services 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-process; (3) product-relevant for operations researchers and management scientists, since a large and perhaps foreboding LP model can be developed and used with

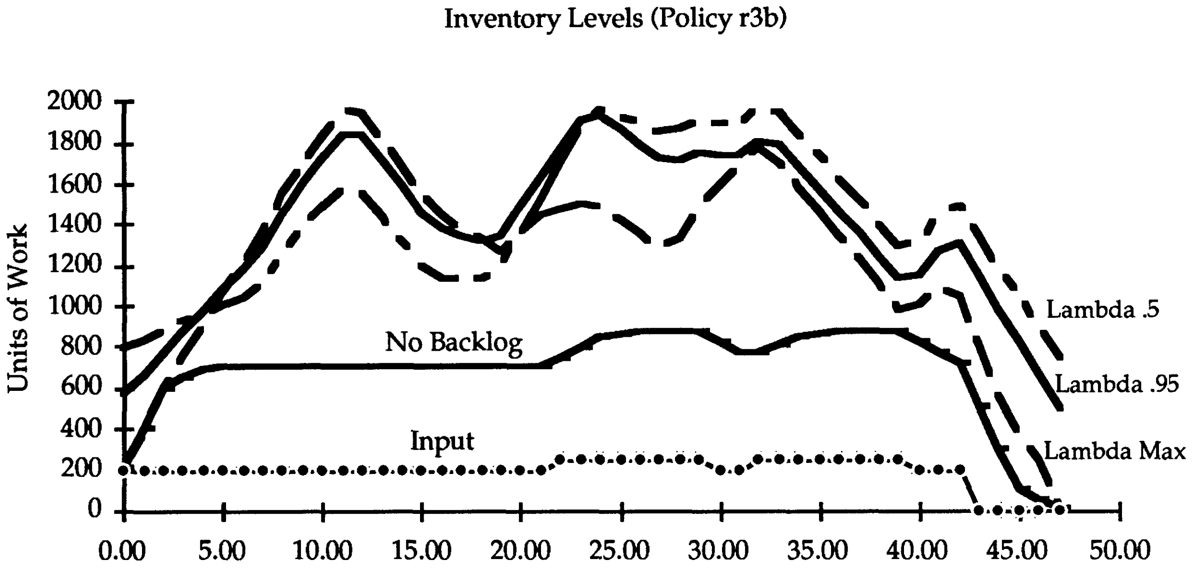

Exhibit 12

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.

Acknowlegements

The authors thank the National Science Foundation (Grant SES 9119962) 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. 1976, "Workforce allocation in cyclical scheduling problems: A survey," *Operational Research Quarterly*, **27** (1), 155-167.

Bartholdi, John J., III. 1980, "Cyclical Scheduling via integer programs with circular ones," *Operations Research*, 28 (5), 1074-1085.

Bazaraa, Mokhtar S. 1990, *Linear Programming and Network Flows*, John Wiley and Sons, New York.

Bechtold, S.E. and L.W. Jacobs. 1990 "Implicit Modeling of Flexible Break Assignments in Optimal Shift Scheduling," *Management Science*, **36** (11), 1339-1351.

Berman, Oded and Richard C. Larson, 1992, "An LP Model for Workforce and Workflow Scheduling," Paper presented at Symposium: *The Service Productivity & Quality Challenge*, The Fishman-Davidson Center for the Study of the Service Sector, University of Pennsylvania, Wharton School.

Bixby, Robert, 1992, "Very Large-Scale Linear Programming: A Case Study in Combining Interior Point and Simplex Methods, *Operations Research*, **40** (5).

Edie, L. C. 1954, "Traffic Delays at Toll Booths," *Operations Research*, **2** (2), 107-138.

Cahn, M.F., R.C. Larson and O. Berman, 1992, "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, California, November 1992.

Emmons, H. and R.N. Burns. 1991, "Off-Day Scheduling with Hiearchical Worker Categories." *Operations Research*, **39** (3), 484-495.

Geoffrion, A. M., 1987, "An Introduction to Structured Modeling," *Management Science*, **33** (5), 547-588.

Geoffrion, A. M., 1989, "The Formal Aspects of Structural Modeling," *Operations Research*, **37** (1), 30-51.

Geoffrion, A. M., 1992, "The SML Language for Structural Modeling: Levels 1 and 2," *Operations Research* **40** (1), 38-57.

Geoffrion, A. M., 1992, "The SML Language for Structural Modeling: Levels 3 and 4," *Operations Research* **40** (1), 58-75.

Glover, Fred 1986, "The general employee scheduling problem: An integration of management science and artificial intelligence," *Computers and Operations Research*, **13** (4), 563-573.

Graves, S. 1986a, "A Tactical Planning Model for a Job Shop," *Operations Research*, **34**, 522-533.

Graves, S., Leff, H., and Dada, M. 1986b, "An LP Planning Model for a Mental Health Community Support System," *Management Science*, **32**, 139-155.

Henderson, W.B. 1976,"Heuristic Methods for Telephone Operator Shift Scheduling: An experimental Analysis,"*Management Science*, **22**, 1372-1380.

Henderson, W.B. 1977, "Determining Optimal Shift Schedules for Telephone Traffic Exchange Operators,"*Decision Science*, **10**, 126-135.

Howard, R. A., 1971, *Dynamic Probabilistic Systems*, Wiley, New York.

Jackson, J. R., 1957, "Networks of Waiting Lines," *Operations Research*, **5**, 518-521.

Mabert, V.A. 1977, "The Detail Scheduling of a Part-Time Work Force: A Case Study of Teller Staffing," *Decision Science*, **8**, 109-120.

Maier-Roth, C. 1973, "Cyclic Scheduling and Allocation of Nursing Staff," *Socio-Economic Planning Ser.*, **7**, 471-487.

Morris, James, and Showalter, Michael. 1983, "Simple approaches to shift, days-off and tour scheduling problems," *Management Science*, **29** (8), 942-950.

Murty, Katta G. 1983, *Linear Programming*, John Wiley and Sons, New York.

Segal M. 1974," The Operator -Scheduling problem: a network flow approach,"*Operations Research*, **22** (4), 808-823.

Shapiro J. F. 1979, *Mathematical Programming Structures and Algorithms*, John Wiley and Sons, New York.

Sittler, R. W., 1956, "Systems Analysis of Discrete Markov Processes," *I.R.E. Transactions of Circuit Theory*, **CT-3**, 1, 257.

Trevelen, M. 1989, "A Review of the Dual Resource Constrained System Research," *IIE Transactions*, **21**, 279-287.

United States Dept. of Labor, 1992, "Monthly Labor Review August 1992," **115** (8), 74.

United States Government, 1992, "Budget of United States Government Fiscal Year 1993," U.S. Gov. Printing Office, Washington D.C., A1-1020.

Warner, D. 1972, "A Mathematical Programming Model for Scheduling Nursing Personnel in Hospitals,"*Management Science*, **19**, 411-422.