

DYNAMIC PROGRAMMING SOLUTION TO
THE STOCHASTIC MULTIPLE LOT
DISPATCHING IN AN FMS

F. Chauny, A. Haurie
P. L'Ecuyer, R. Loulou

G-88-08

Avril 1988

Les textes publiés dans la série des rapports de recherche H.E.C. n'engagent que la responsabilité de leurs auteurs. La publication de ces rapports de recherche bénéficie d'une subvention du Fonds F.C.A.R.

**DYNAMIC PROGRAMMING SOLUTION TO THE STOCHASTIC
MULTIPLE LOT DISPATCHING IN AN FMS***

F. Chauny¹, A. Haurie², P. L'Ecuyer³, R. Loulou⁴

* Research supported by "Actions Structurantes" MESS-Québec, (Grant # 6.1/7.4(28)), NSERC-Canada (Grants # A3358, A4952, A5468), and FCAR-Québec (Grants # 88EQ3528, EQ2831).

¹ GERAD, Ecole des Hautes Etudes Commerciales, Montréal, Qué., Canada.

² GERAD, Ecole des Hautes Etudes Commerciales and Ecole Polytechnique, Montréal, Qué., Canada.

³ GERAD and Département d'informatique, Université Laval, Ste-Foy, Qué., Canada.

⁴ GERAD and Faculty of Management, McGill University, Montréal, Qué., Canada.

Abstract

The production scheduling problem considered in this paper is related to the planning of operations of a flexible manufacturing cell composed of a punch press and a shear that produce parts from metal plates. A stochastic control problem is identified, to optimize the expected discounted cost of the system. The control consists in dispatching n bins to a processor units which can process only one bin at a time. A dynamic programming approach is used, and numerical techniques are applied for the solution of the two-bin case. Numerical results are presented and discussed.

Résumé

Nous considérons dans cet article un problème de production relatif à la planification des opérations d'un atelier flexible. Le problème consiste à choisir parmi n lots lequel sera le prochain à être envoyé en production. La taille des lots augmente selon une demande aléatoire. Deux types de coûts sont pris en compte: un coût fixe associé au temps de réglage de la machine et à la perte de matériel et un coût proportionnel à la durée de l'attente d'un lot. Les temps d'usinage sont déterministes. Une approche basée sur la programmation dynamique est utilisée. Un exemple constitué de deux types de lots est traité.

1. The multiple lot dispatching problem

The production scheduling problem considered in this paper is related to the planning of operations of a flexible manufacturing cell composed of a punch press and a shear that produce parts from metal plates. Three process planning problems for this cell, dealing respectively with the nesting of parts, sequencing of punch operations and clamp positioning, have already been considered in [1], [2] and [3]. The present study deals with the upstream problem, where one has to decide how to build and schedule the lots to be produced by the cell. The orders for parts arrive at random times and are accumulated in lots. A lot consists of a set of parts having the same material requirements (e.g. metal gauge and quality). The part nesting process generates some material loss mostly concentrated on the last plate of metal used to produce a lot. In first approximation this loss can be considered as a fixed loss plus a loss proportional to the number of parts in the lot sent to production. This fixed loss induces an incentive to build large lots before starting a production run. On the other hand delaying the production of an ordered part bears a cost very similar to a holding cost. The problem is to find a good compromise between these two costs.

This production system can be represented schematically as follows: there is a *processor* unit, and n *feeder* units called *bins*. Each bin receives and stores unprocessed parts which then constitute a *production lot*, corresponding to one possible line of production, with specific material requirements and processing cost. There is no possibility to transfer parts from one bin to the other. If the processor is idle, and if a bin is selected for production, then the whole lot currently in the selected bin is instantaneously transferred to the processor unit and the emptied bin can immediately resume the lot building process (i.e. receive new parts). The processing time in the processor depends on the size of the lot sent for production.

This system may equivalently be described as a queue with one server that serves n classes of arrivals, each class modeled by an independent Poisson process. When idle, the server observes the contents of the n queues and may decide to remain idle, or to dispatch one of the n queues. The service time to process one entire queue is dependent upon that queue size. The problem is related to the *clearing system problem* [9], a classical one in the queuing literature, of which the *Shuttle Dispatching Problem* is a special case (see [9]). In this problem, the server is a shuttle that waits for a sufficient number of passengers to board, and then departs, thus becoming unavailable for a certain duration (service time). In this situation there is only one class of customers

(versus n in our case) and service time is independent of queue size. Clearing systems also have applications in inventory control, and in dispatching of multiple vehicles, with or without capacity constraints ([9], [7]). In the present paper we favor a stochastic control formulation of the problem.

For $j \in \{1, \dots, n\}$, we define the *state* of bin j at time t as its anticipated processing time, given the size of the lot currently in the bin. This state is denoted $x_j(t)$. Let P_j be the set of parts which are assigned to bin j . We assume that the arrival process describing the accumulation of parts of type $p_{kj} \in P_j$ is Poisson with intensity λ_{kj} . This assumption is not essential for this model, however it simplifies the transition probability kernel expression for the Markov renewal decision process considered in section 2, and eases the numerical solution. If such a part enters the bin, it increases the state by a (known) amount ξ_{kj} . The arrival processes corresponding to different parts are assumed to be independent.

Two kinds of costs are incurred: (a) a holding cost rate for the parts in the bin, which is considered to be proportional to the bin contents, and (b) a lumped cost, related to the setup of the processor and the material loss, which only depends on the production line (i.e. the selected bin) corresponding to the lot that is sent for production. Therefore, if one observes bin j on the interval $[t_1, t_2]$ for a particular sample value ω , and if the bin has been emptied (i.e. the lot has been sent to production) at a time $t_3(\omega) \in [t_1, t_2]$, then the cost incurred by the system for bin j , on the interval $[t_1, t_2]$, is given by

$$\int_{t_1}^{t_2} e^{-\rho t} w_j x_j(t, \omega) dt + e^{-\rho t_3(\omega)} W_j. \quad (1)$$

In this expression w_j is the unit holding cost rate, and W_j is the setup cost, for bin j . These costs are discounted at rate $\rho > 0$.

A controller can decide, at some specific times that will be defined shortly, either to postpone production or to send the current lot contained in some bin j to be processed (we also say that bin j is emptied into the processor unit). The *action set* is thus defined as $A = \{0, 1, \dots, n\}$ where 0 indicates that nothing is sent to the processor, and j indicates that bin j is emptied. We assume that the controller uses a *policy* to operate the system over an infinite time horizon. In order to define precisely the set of admissible policies, we now give a more complete description of the controlled stochastic process corresponding to this system.

Let $y(t)$ be equal to 0 if the processor is idle at time t , and equal to the remaining processing time of the work in process otherwise. The *state of the system* at time t is thus defined as $s(t) = (y(t), X(t)) = (y(t), x_1(t), \dots, x_n(t))$, and we denote by S the state set. We consider the stochastic process $s(\cdot) = (s(t) : t \geq 0)$ and we assume that the controller intervenes at the jump times of $X(\cdot)$, i.e. at the instants of arrivals. Let $A(s)$ be the set of all *admissible actions* when in state s . If the y component of s is positive (i.e. if the processor is busy) then $A(s) = \{0\}$, otherwise $A(s) = A$. A *policy* will thus be defined as a mapping π which sends any state s into the admissible action set $A(s)$. Under a given policy π , $s(\cdot)$ becomes a semi-Markov process governed by a transition probability kernel defined in the next section. Associated with an initial state s^0 and a policy π is the total expected discounted cost

$$V_\pi(s^0) = E_\pi \left[\sum_{j=1}^n \int_0^\infty e^{-\rho t} w_j x_j(t) dt + \sum_{\ell=0}^\infty e^{-\rho t_\ell} W_{j_\ell} \right], \quad (2)$$

where t_ℓ , for $\ell = 0, \dots, \infty$, denote the random times at which a bin is emptied into the processor, and j_ℓ denotes the bin chosen at time t_ℓ . The problem is to characterize an optimal policy π^* which satisfies, for any $s^0 \in S$,

$$V_{\pi^*}(s^0) = V_*(s^0) \stackrel{\text{def}}{=} \inf_{\pi} V_\pi(s^0). \quad (3)$$

Notice that in this formulation, the controller does not intervene as soon as the processor terminates a job and becomes idle (i.e. when $y(t)$ comes down to zero); it waits until the next arrival. Under this assumption, the expected time between any two successive transitions is bounded away from zero, uniformly over the state space, and all the dynamic programming operators defined in the next section are contracting. This permits one to use a successive approximation algorithm, as suggested in [5], to compute a near optimal policy $\tilde{\pi}$, together with bounds on $\sup_{s \in S} |V_{\tilde{\pi}}(s) - V_*(s)|$.

An alternate formulation would also consider the times at which the processor terminates a job as possible intervention times: the controller would be allowed to empty a bin as soon as $y(t)$ reaches a zero value. One difficulty with this formulation is that the dynamic programming operators (to be defined in the next longer section) are no more *contracting*. This is due to the possibility of having a part arrival at a time τ when the y value is extremely small. In that case, whatever be the policy, the expected discount factor for one stage is not uniformly bounded away from one, and thus the model is neither a *C* or a *LC* model according to the terminology of [5].

2. Solving the Markov renewal decision process

In this section we define the Markov renewal decision process (MRDP) associated with the optimization problem defined above. We will use the dynamic programming operator formalism initially introduced by Denardo [4]. Recall that the controller intervenes at the jump times of $X(\cdot)$, i.e. when a new part p_{kj} arrives in one of the bins. Therefore, the random delay θ between two successive intervention times is exponential, with distribution function

$$F(t) = P[\theta \leq t] = \begin{cases} 0 & \text{if } t < 0, \\ 1 - e^{-\Lambda t} & \text{if } t \geq 0, \end{cases} \quad (4)$$

where

$$\Lambda = \sum_{j=1}^n \sum_{k=1}^{n_j} \lambda_{kj}. \quad (5)$$

Given θ , the holding cost over this transition period is given by

$$R(\theta) = \int_0^\theta e^{-\rho t} \sum_{j=1}^n (w_j x_j(t)) dt = \frac{1 - e^{-\rho\theta}}{\rho} \sum_{j=1}^n (w_j x_j), \quad (6)$$

where x_j is the state of bin j at the beginning of this transition period. Let $X = (x_1, \dots, x_n)$ represent the vector state of all bins. Let \mathcal{V} be the class of bounded functions $V(\cdot) : S \mapsto \mathbb{R}$. We define a norm on \mathcal{V} by $\|V\| = \sup_{s \in S} |V(s)|$. We consider the operators H and T acting on \mathcal{V} , where $H(V) : S \times A \mapsto \mathbb{R}$ and $T(V) : S \mapsto \mathbb{R}$ are the functions defined below.

If $y > 0$, then necessarily $a = 0$ and

$$\begin{aligned} H(V)(s, 0) &= \int_0^\infty \Lambda e^{-\Lambda \tau} \left[R(\tau) \right. \\ &\quad \left. + e^{-\rho \tau} \sum_{j=1}^n \sum_{k=1}^{n_j} \left(\frac{\lambda_{kj}}{\Lambda} \right) V(\max(y - \tau, 0), X + \xi_{kj} e_j) \right] d\tau \\ &= \frac{1}{\Lambda + \rho} \left[\sum_{j=1}^n w_j x_j + e^{-(\Lambda + \rho)y} \sum_{j=1}^n \sum_{k=1}^{n_j} (\lambda_{kj} V(0, X + \xi_{kj} e_j)) \right] \\ &\quad + \int_0^y e^{-(\Lambda + \rho)\tau} \left(\sum_{j=1}^n \sum_{k=1}^{n_j} \lambda_{kj} V(y - \tau, X + \xi_{kj} e_j) \right) d\tau, \end{aligned} \quad (7)$$

where e_j denotes the j -th unit vector.

If $y = 0$ and $a = 0$, then

$$\begin{aligned}
H(V)((0, X), 0) &= \int_0^\infty \Lambda e^{-\Lambda\tau} [R(\tau) \\
&\quad + e^{-\rho\tau} \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\lambda_{kj}}{\Lambda} \right) V(0, X + \xi_{kj} e_j)] d\tau \\
&= \frac{1}{\Lambda + \rho} \sum_{j=1}^n ((w_j x_j + \sum_{k=1}^{n_j} \lambda_{kj} V(0, X + \xi_{kj} e_j))).
\end{aligned} \tag{8}$$

If $y = 0$ and $a = i \neq 0$,

$$H(V)((0, X), i) = W_i + H(V)((x_i, X - x_i e_i), 0). \tag{9}$$

The function $T(V)(s)$ is then defined as

$$T(V)(y, X) = \begin{cases} H(V)(y, X, 0) & \text{if } y > 0, \\ \min_{0 \leq i \leq n} H(V)((y, X), i) & \text{if } y = 0. \end{cases} \tag{10}$$

This stochastic control system is a discounted MRDP model, for which there exists a well developed theory (see [4], [5]). It is in fact a contracting model (type C, according to the terminology of [5]), which fits into the contraction mapping framework introduced by Denardo [4]. The expected discount factor between two intervention times is given by

$$\alpha = \int_0^\infty e^{-\rho t} \Lambda e^{-\Lambda t} dt = \frac{\Lambda}{\Lambda + \rho} \tag{11}$$

and the operators H and T are contracting with factor α . The following properties hold (see [5]):

THEOREM 1.

- (a) $T(V) = V$ iff $V = V_*$, i.e. V_* is the only fixed point of the operator T ;
- (b) $\lim_{n \rightarrow \infty} \|T^n(V) - V_*\| = 0$, where T^n denotes the n -fold composition of the operator T ;
- (c) An optimal policy μ_* is defined by

$$\mu_*(0, X) = \arg \min_{0 \leq i \leq n} H(V_*)((0, X), i) \tag{12}$$

when the minimum is attained.

3. Approximation of the optimal policy

In this section we describe the implementation, for this specific scheduling problem, of the approximation technique proposed in [5] within the general framework of discrete event dynamic programming. One wants to approximate the function $V_* : \mathbb{R}^{n+1} \mapsto \mathbb{R}$ which is a fixed point of the operator T . One uses a *value iteration* algorithm which consists to apply the operator T repeatedly until the fixed point is approximated. Convergence is guaranteed by the contraction property of T (Theorem 1 (b)). Since the state set S is infinite and continuous, one must include in the numerical computation another approximation scheme dealing with the evaluation of $T(V)$ over the whole state set S at each iteration. As suggested in [5], one defines a finite grid over $S = \mathbb{R}^{n+1}$. At each iteration, one computes the value of $T(V)$ at each node, and uses interpolation techniques (e.g. spline or finite element) to evaluate $T(V)(\cdot)$ at points other than the grid nodes, in order to obtain an approximation W of $T(V)$.

In [5], it is shown how to compute bounds for the error norm in the computation of V_* , and bounds on $\|V_{\tilde{\pi}} - V_*\|$, where $\tilde{\pi}$ is the policy retained at the end of the algorithm. The latter norm represents a bound on the expected loss incurred by using policy $\tilde{\mu}$ instead of the optimal policy. The following is proven in [5].

THEOREM 2.

At any iteration, one has:

$$V_* \begin{cases} \geq W - \epsilon^- - (\alpha/(1-\alpha))\|(V - W + \epsilon^-)^+\| \\ \leq W + \epsilon^+ + (\alpha/(1-\alpha))\|(W + \epsilon^+ - V)^+\| \end{cases} \quad (13)$$

where $\alpha = \Lambda/(\Lambda + \rho)$, $f^+(\cdot) = \max\{0, f(\cdot)\}$, and ϵ^+ and ϵ^- are positive constants such that

$$-\epsilon^- \leq T(V) - W \leq \epsilon^+. \quad (14)$$

Furthermore, for any $\epsilon_0 > \epsilon^+$, one can find a policy μ such that

$$T_\mu(V) \leq T(V) + \epsilon_0 - \epsilon^+ \leq W + \epsilon_0, \quad (15)$$

and any such policy satisfies

$$\begin{aligned} V_\mu - V_* &\leq \epsilon^- + \epsilon_0 \\ &+ (\alpha/(1-\alpha)) (\|(V - W + \epsilon^-)^+\| + \|(W - V + \epsilon_0)^+\|). \end{aligned} \quad (16)$$

The values ϵ^- and ϵ^+ express the error made when one replaces $T(V)$ by its approximation W . In practice one can evaluate (or estimate) these bounds by using an extra step of the value iteration method, in which $T(V)$ is computed at a large number of points interior to the grid G , and then taking the largest observed errors as estimates for ϵ^+ and ϵ^- . The same can be done for ϵ_0 . The approach is illustrated below for a two-bin system.

4. A two-bin example

When the scheduling problem involves only two bins, the state set S reduces to \mathbb{R}_+^3 . The grid consists of a set of cells defined by parallelepipeds (some being unbounded). The value iteration algorithm is applied at each node of the grid and the $T(V)$ function is approximated at points other than the grid points by using a trilinear interpolation W . When a cell is unbounded w.r.t. one of the three state variables, the function W is taken as constant in this direction.

The iterations are performed in *batches*. At the end of each batch the bounds described in Eq. (13) are computed, and new points may be added to the grid. One can start with a coarse grid, that contains only a small number of points, and refine the grid by adding new points after every batch. These new points can be chosen, for instance, in such a way that the values (x_1, x_2) at which $|T(V) - W|$ is the largest are incorporated into the grid. In our experiments, the ϵ values were evaluated by a technique based on the *golden section search* method (see [6]), on each parallelepiped. Batches of iterations are performed until the distance between the bounds in Eq. (13) becomes sufficiently small.

As a numerical illustration, let $n_1 = n_2 = 1$, $\lambda_{11} = \lambda_{12} = 0.05$, $\xi_{11} = \xi_{12} = 9.5$, and $\rho = 0.025$. One easily computes $\alpha = (\lambda_{11} + \lambda_{12}) / (\lambda_{11} + \lambda_{12} + \rho) = 0.8$. We performed 5 batches of about 35 iterations each. For the last batch, the values of x_1 and x_2 defining the grid were 0.0, 40.0, 50.0, 90.0, 100.0, 120.0, 130.0, 140.0, 150.0, 156.7, 160.9, 165.0, 172.5, 180.0, 185.0, 190.0, 195.0, 200.0, 206.2, 212.5, 218.7, 225.0, 232.0, 240.0 and 250.0. After the last batch, we had $\|V - W\| < 0.0031$, and we estimated $\epsilon^+ = 68.0$ and $\epsilon^- = 0.0$. The bounds given by theorem 2 are

$$0 \leq V_* - W \leq 340.284,$$

$$0 \leq V_\mu - V_* \leq 340.284,$$

where μ is the retained policy (approximated using Eq. (12)). We obtain the same results for the two bounds, because $\epsilon^- = 0$ and V is always least than W . This corresponds to a relative approximation error of 3%. The computations for this example took about 18 hours of CPU time on a SUN-3/50 with a math. co-processor. Most of this time was spent in the last batch of iterations, where the grid contains around 15000 nodes. The policy μ retained at the end of the algorithm appears in figure 1. This figure also illustrates the resulting policies for examples with the same data except for ρ , for which we used $\rho = 0.01$, and $\rho = 0.04$. One notices that as ρ increases, the region where one decides to postpone production tends to become a square. For small values of ρ , this region has a more complex shape. A common characteristic feature of these sets is the pike in the direction of the bisector $x_1 = x_2$. This suggests that when the two bins are equally full, one should wait a little bit longer before deciding which lot will be sent to production. Figure 2 illustrates the retained policy if the controller intervenes as soon as the processor terminates a job. As we can see, this policy is the same as the one without these possible intervention times.

5. Conclusion

In [8], Schweitzer and Seidman have proposed a different although related stochastic dynamic programming model for the part selection problem in an FMS. Their model uses a discrete state set and average cost criterion. In the present paper we have considered a different dynamic programming model also related to the problem of feeding an FMS. In our formulation, the state set is infinite and continuous and we use a discounted cost criterion. The numerical solution makes use of advanced approximation techniques in Dynamic Programming. This technique could be adapted to a deterministic version of this problem.

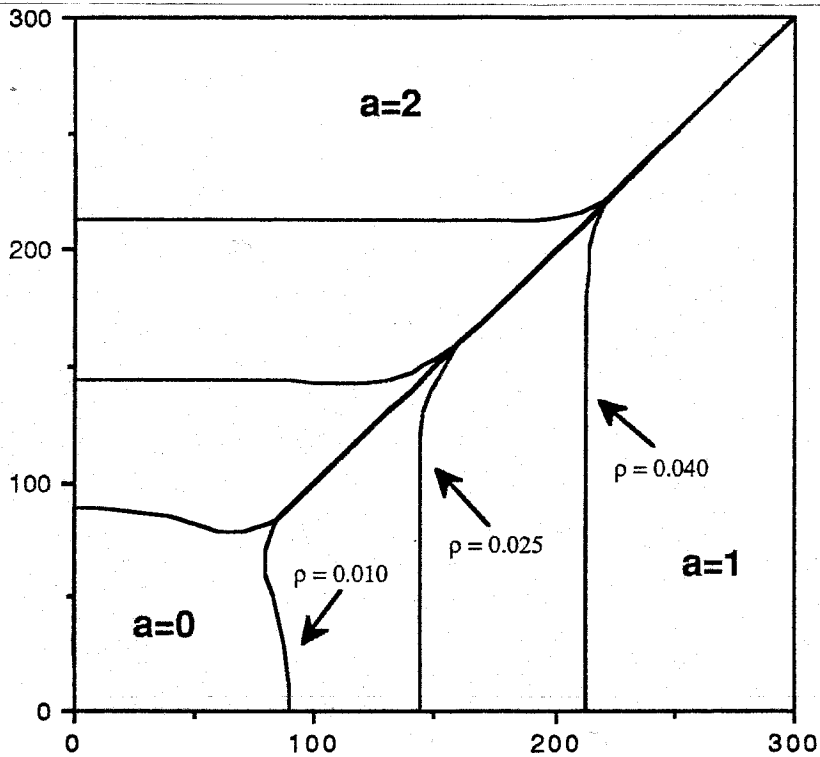


Figure 1: Resulting policies for different values of ρ .

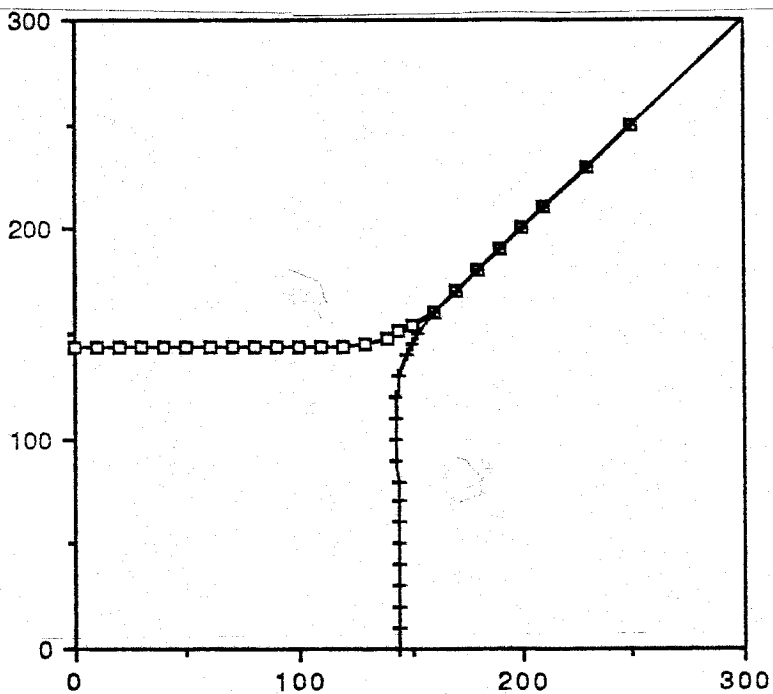


Figure 2: Resulting policy if the controller intervenes as soon as the processor terminates a job.

References

- [1] F. Chauny, R. Loulou, S. Sadones, F. Soumis, "A Two Phase Heuristic for Strip-Packing : Algorithm and Probabilistic Analysis", *OR Letters*, Vol. 6, No. 1, 1987.
- [2] F. Chauny, A. Haurie, R. Loulou, E. Wagneur, "Sequencing Punch Operation in an FMS : a Three Dimensional Spacefilling Curve Approach", *INFOR*, Vol. 25, No. 1, 1987.
- [3] F. Chauny, A. Haurie, R. Loulou, E. Wagneur, "Clamps Positioning and Optimal Holding Positions in an FMS", To appear in *IEEE Transactions on Automatic Control*, 1988.
- [4] E.V. Denardo, "Contraction Mappings in the Theory underlying Dynamic Programming", *SIAM Review*, Vol. 9, pp. 165-177, 1967.
- [5] A. Haurie and P. L'Ecuyer, "Approximation and Bounds in Discrete Event Dynamic Programming", *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 3, pp. 227-235, March 1986.
- [6] D. G. Luenberger, *Linear and Nonlinear Programming*, Second ed., Addison-Wesley, 1984.
- [7] W. B. Powell and P. Humblet, "The Bulk Service Queue with a General Control Strategy: Theoretical Analysis and a New Computational Procedure", *Operations Research*, Vol 34., No. 2., March-April 1986, pp. 267-275.
- [8] P.A. Schweitzer and A. Seidmann, "Part Selection Policy for a Flexible Manufacturing Cell Feeding Several Production Lines", *IIE Transactions*, Vol 12, No. 4, pp355-362, 1984.
- [9] S. Stidham, "Clearing Systems and (s,S) Inventory Systems with Nonlinear Costs and Positive Lead Times", *Operations Research*, Vol. 34, No. 2, March-April 1986, pp. 277-280.