SEQUENTIAL OPEN-LOOP SCHEDULING STRATEGIES

P. Nash, R.R. Weber

Cambridge University

ABSTRACT

For certain scheduling problems with pre-emptive processing, a dynamic programming formulation reduces the problem to a sequence of deterministic optimal control problems. Simple necessary and sufficient optimality conditions for these deterministic problems are obtainable from the standard results of optimal control theory, and sometimes lead to analytic solutions. Where this does not happen, then as with many dynamic programming formulations, computational solution is possible in principle, but infeasible in practice. After a survey of this approach to scheduling problems, this paper discusses a simplification of the method which leads to computationally tractable problems which can be expected to yield good, though sub-optimal, scheduling strategies. This new approach is based on the notion of sequential open-loop control, sometimes used in control engineering to solve stochastic control problems by deterministic means, and is not based on dynamic programming.

1. INTRODUCTION

The basic tools of most of optimization theory are optimality conditions obtained by variational methods. A fundamental difficulty with scheduling problems is their combinatorial nature, which usually makes their solution by variational techniques impossible. Thus for most scheduling problems, first- and second-order optimality conditions analogous to those of mathematical programming are unavailable. For one class of problems, namely those in which scheduling is completely pre-emptive, this is not the case. In such problems, processor

M. A. H. Dempster et al. (eds.), Deterministic and Stochastic Scheduling, 385–397.

effort can be regarded as infinitely divisible at each point in time, and the allocation of effort as instantaneously variable. Under these conditions, variational methods can be applied. As a bonus, we find that when we formulate such problems appropriately, they can be approached by completely deterministic methods.

In this paper, we outline a method whereby scheduling problems of this type can be reduced to deterministic optimal control problems. The application of the maximum principle to these problems leads to necessary conditions for optimality of a schedule, while the Hamilton-Jacobi-Bellman equation for the problem gives a corresponding sufficient condition. For a number of cases, these conditions enable one to derive the optimal scheduling strategy. This is most easily done for static single-processor problems, and becomes increasingly difficult as the structure of the problem becomes more complex. Extending results from the static to the dynamic case requires more complicated notation and proofs, and when we examine the parallel-processor case, the arguments needed to produce analytic results become very delicate indeed. The formulation is based on a particular dynamic programming approach, and when analytic results are not obtainable, solution by direct computation is not a practical proposition. Without going into formal proofs in detail, we indicate the features of the problem which make its solution difficult, and these lead us to propose the investigation of what control engineers call sequential open-loop strategies.

To apply an sequential open-loop strategy to a scheduling problem, we compute an allocation of processor effort for all future times, by optimizing an objective functional on the assumption that the chosen allocation will be followed - even to the extent of leaving resources idle - irrespective of the realizations of the job completions. This allocation is followed for the duration of some review period (possibly random), then a new allocation is computed in the same way, after updating all the probability distributions. This new allocation is put into effect during the next review period, and so on.

Such a strategy will be sub-optimal, but it will approximate to optimality in some cases. In particular cases, indeed, such a strategy is optimal if the review times are suitably chosen. Where we can have some hope that such a strategy will perform well, it has considerable advantages: it can be obtained analytically at least as often as the optimal closed-loop strategy, while being relatively easy to compute even if not so obtainable.

The next section describes the reduction of the scheduling problem to a problem in deterministic optimal control, and reviews some of the results that can be obtained from this formulation. The exposition will be brief, as this material has appeared elsewhere

[5], [6], [8]. In the following sections, we examine sequential open-loop schedules for a number of examples and compare them with known optimal strategies.

2. AN OPTIMAL CONTROL FORMULATION

For simplicity, we consider first the single-server problem with no arrivals. At time $t=0$ there are n jobs waiting to be processed. Processing effort is available at a constant rate, which may be taken to be unity. The amount of processing needed to complete job i is a random variable whose distribution is known, and has distribution F_i with density f_i. We define

$$P_i=1-F_i,\ \rho_i=f_i/P_i,\ i=1,2,\dots,n,$$

the survivor functions and completion rate functions respectively. On completion of each job, a reward is received which depends on the job and on the time of its completion. The reward received if job i is completed at time t is $r_i(t)$. (We assume that the functions F_i, f_i, ρ_i and r_i are all well behaved.) We seek to allocate processing effort between jobs over time so as to maximize the expected total reward.

The state of the set of jobs at time t is described by the pair $(\Gamma(t),X(t))$, where $\Gamma(t)$ is a list, possibly empty, of the indices of the jobs that have been completed by time t, and $X(t)=(X_1(t),X_2(t),\dots,X_n(t))$ is a vector whose i'th component is the amount of processing received by job i during $(0,t)$. A scheduling strategy U is a rule which determines, for any state (Γ,X) and time t an allocation $U(\Gamma,X,t)$ of the total processing effort. We use $\Gamma(t|U)$ and $X(t|U)$ to denote the components of the process $(\Gamma(t),X(t))$ when a fixed strategy U is used.

A crucial feature of the controlled semi-Markov process $(\Gamma(t),X(t))$ is that for any fixed strategy U, the evolution of $X(t|U)$ is completely deterministic between jumps in $\Gamma(t|U)$. Any strategy U can therefore be implemented in the following way. Define, for each (Γ,X) and t,

$$u^{\Gamma}(X,t;s)=U(\Gamma,x^{\Gamma}(X,t;s),s),\quad t\leq s<\infty,$$

where $x^{\Gamma}(X,t;s)$ is determined by

$$\dot{x}^{\Gamma}(X,t;s)=u^{\Gamma}(X,t;s),\ t\leq s<\infty,\ x^{\Gamma}(X,t;t)=X.$$

Let $C_0,C_1,\dots,C_n$ denote the initial time and the times of the first, second, etc. job completions. Let $\Gamma_0,\Gamma_1,\dots,\Gamma_n$ denote the lists of completed jobs initially and at times just after the first, second, etc. job completions. Then U is implemented by using the allocation

$$u^{\Gamma_0}(X(C_0),C_0;s)$$

at all times s up to the first job completion, then using the allocation

$$u^{\Gamma_1}(X(C_1),C_1;s)$$

for all times between the first and second completions, then switching to

$$u^{\Gamma_2}(X(t_2)C_2;s),$$

and so on.

This idea allows us to write the following recursive expression for expected total reward obtained using strategy U and starting at time t in state (Γ,X):

$$V^{\Gamma}(X,t|U)=\int_t^{\infty}\Big(\sum_{i\varepsilon\bar{\Gamma}}\Big\{\Big(\prod_{\substack{k\neq i\\ k\varepsilon\bar{\Gamma}}}P_k(x_k^{\Gamma}(X,t;s))\Big)r_i(s)$$

$$+V^{\Gamma,i}(x^{\Gamma}(X,t;s)),s|U)\Big\}f_i(x_i^{\Gamma}(X,t;s))u_i^{\Gamma}(X,t;s)\Big)ds,$$

with the obvious terminal condition that the expected value in states with no uncompleted job is identically zero. This makes it clear that for any initial state, the scheduling problem can be viewed as one of choosing an optimal sequence

$$\{u^{\Gamma_i}(X(C_i),C_i;s):\ i=0,1,2,\dots,n-1\}$$

of deterministic controllers. If we apply the dynamic programming optimality principle to this sequence of choices, we see that U is optimal if and only if each such controller $u^{\Gamma}(X,t;s)$ solves a deterministic optimal control problem

$$P^{\Gamma}(X,t):\ \text{Maximize}\ \int_t^{\infty}\Big(\sum_{i\varepsilon\bar{\Gamma}}h_i^{\Gamma}(x(s),s)u_i(s)\Big)ds$$

$$\text{subject to}\quad \dot{x}(s)=u(s),\ t\leqslant s<\infty,\ x(t)=X,$$

$$\sum_i u_i(s)=1,\ t\leqslant s<\infty,$$

where we define for each Γ,x,s

$$h_i^{\Gamma}(x,s)=\Big\{\Big(\prod_{\substack{k\neq i\\ k\varepsilon\bar{\Gamma}}}P_k(x_k)\Big)r_i(s)+V^{\Gamma,i}(x,s|U)\Big\}f_i(x_i).$$

Not only is this problem deterministic, but it has a particularly simple form, especially in its dynamics. We quote two theorems which show that this simplicity leads to equally simple necessary and sufficient conditions for optimality. Let P1 be the problem

$$P1(X,t): \text{ Maximize } \int_t^\infty \left(\sum_i h_i(x(s),s)u_i(s)\right)ds$$

$$\text{subject to } \dot{x}(s)=u(s),\ t\leqslant s<\infty,\ x(t)=X,$$

$$\sum_i u_i(s)=1,\ t\leqslant s<\infty,\ u(s)\,\varepsilon\,\Omega,\ 0\leqslant s<\infty.$$

Theorem 1. A necessary condition for the optimality of u in P1 is that , for each time t,

$$\sum_i u_i(t)\int_t^\infty \left(\sum_j H_{ij}(x(s),s)u_j(s)\right)ds = \max_{\substack{\omega\varepsilon\Omega \\ \Sigma\omega_i=1}} \left\{\sum_i \omega_i \int_t^\infty \left(\sum_j H_{ij}(x(s),s)u_j(s)\right)ds=0,\right.$$

where

$$H_{ij}(x,s)=\frac{\partial h_j(x,s)}{\partial x_i}-\frac{\partial h_i(x,s)}{\partial x_j}+\frac{\partial}{\partial s}(h_j(x,s)-h_i(x,s)).$$

This condition is just a statement of the maximum principle for P1, after integrating the co-state equations to give the co-state variables in terms of the states and controls. The details are in [5].

Theorem 1 is a necessary condition, and relates to open-loop controllers, that is controllers specified as functions of time for particular sets of initial data. We can ask for a closed-loop or feedback controller, which assigns to each state a control action. The open-loop controllers are then just the implementations of the closed-loop controller on trajectories with particular initial data. By using dynamic programming on P1, we obtain a necessary and sufficient condition for optimality of a feedback controller. Let $V(x,t|u)$ denote the value of the objective functional in P1 starting from state x and time t, using control u.

Theorem 2. A necessary and sufficient condition for the feedback controller $u(x,t)$ to be optimal for P1 is

$$\sum_i u_i(x,t)\left\{\frac{\partial V(x,t|u)}{\partial x_i}+h_i(x,t)\right\} \geq \sum_i \omega_i\left\{\frac{\partial V(x,t|u)}{\partial x_i}+h_i(x,t)\right\}$$

for all $\omega\varepsilon\Omega$ such that $\Sigma\omega_i=1$, for all x and t.

Again, the proof of this result is a straightforward application of the techniques of optimal control theory. The details can be found in [8].

The deterministic problem $P^{\Gamma}(X,t)$ is of the same form as P1, and theorems 1 and 2 may therefore be applied to it to obtain necessary and sufficient optimality conditions. To solve the scheduling problem with the usual initial data - no jobs completed and no processing as yet carried out, starting at time 0 - we need to solve $P(0,0)$. A snag is that the conditions provided by the theorems will involve the derivatives of the optimal value functions V^i for the (n-1)-job scheduling problems arising at the first completion. It is readily shown [5] that these derivatives can be written in terms of the co-states for the (n-1)-job problems. When the optimal scheduling strategy has a structure which is independent of the number of jobs (when it is given by a priority index, for example), it is usually possible to prove optimality by using this as the basis of an inductive argument. When this is not the case, the dependence of the objective in $P(0,0)$ on the functions $\{V^i\}$ means that any computational procedure based on this approach would have to be recursive, and would involve the solution of all $2,3,\ldots,(n-1)$ job sub-problems over a region of the state space of the n job problem. While possible in principle, there is little hope that such a procedure could be practical.

This is a pity, because the formulation permits of great generality. For example, we can deal with parallel processors just by changing the constraints to be satisfied by the deterministic controllers. For the multi-server case, we just add the constraints

$$u_i(s)\leqslant 1/m, \quad i=1,2,\ldots,n.$$

Some results for multi-server problems obtained in this way are described in [8]. Precedence relations between jobs can be modelled by allowing the control constraint set Ω to depend on Γ, so that an optimal controller u^{Γ} for P^{Γ} has to satisfy

$$u^{\Gamma}(t)\in\Omega^{\Gamma}, \quad 0\leqslant t<\infty.$$

This does not affect the form of the deterministic control problems, so that theorems 1 and 2 may still be applied. Queueing problems are included simply by noting that arrivals constitute just another type of random jump, so that the set of times at which deterministic controllers have to be chosen now includes arrival instants as well as completion instants. We thus have to solve a set of problems $P^{\Gamma;\Delta}$, where Δ is the list of jobs which have arrived or been released. Similarly, reneging of customers in a queue is included by adding times at which customers renege to the set of decision times. In fact, almost any effect of this sort, which just adds more jumps to the basic process, can be incorporated in this way, and the objective functional and dynamics in the associated deterministic control problem are always formally the same as those of P1: only the specific definition of the functions h_i changes. Various results involving the determination of optimal strategies can be extended by these means; some of these extensions are described in [6].

In the remainder of this paper, we investigate a possible method of using this formulation to generate scheduling strategies in problems where the full deterministic control problem derived from the optimality principle as above cannot be solved analytically. We note that most of the difficulties arise because of the dependence of the optimality conditions on the lower-order sub-problems, and consider a formulation in which these dependencies do not occur.

3. OPEN-LOOP FORMULATIONS

The basic observation we make is that if we pretend that the evolution of the system will be unobserved, or that at any rate we are unable to act on any observations, then we are faced with the problem of finding an open-loop controller, which is simply an allocation of processor effort for all future times, which will be followed exactly, even if it implies allocating effort to already completed jobs. This is equivalent to controlling the probability dynamics of the process, which is the approach used in [1] and [7], although neither of these papers makes explicit reference to the fact that the control is open-loop.

The approximation involved in open-loop control is obviously a gross one, and the allocations produced would become more and more ridiculous as jobs were completed. A compromise between a full closed-loop solution and the completely open-loop one that has been used in a number of control applications is open-loop feedback or sequential open-loop control [2]. To implement a sequential open loop controller, we solve an open-loop problem at each of a succession of review times $\tau_0, \tau_1, \ldots$, and employ the resulting controller between succesive review times. The review times may be predetermined, or state-dependent(and hence random). Each time we solve for a new open-loop controller, we incorporate up-to-date information about the system state.

For this to be a useful approach to scheduling and allocation problems, a number of criteria must be met. First, we must have some expectation that the scheduling strategies that result are near-optimal, and better than those that might result from the application of one or other of the well-known scheduling heuristics. Secondly, where the resulting open-loop problems are not soluble analytically, they must be computationally tractable. Thirdly, their applicability should be reasonably general.

3.1 Sequential open-loop scheduling

We begin by presenting an open-loop analogue of problem $P^\Gamma(X,t)$.

$$\mathrm{OLP}^\Gamma(X,t):\ \text{Maximize}\ \int_t^\infty \Big(\sum_{i\epsilon\Gamma} r_i(t')f_i(x(s))u_i(s) \Big)ds'$$

$$\text{subject to}\ \ \dot{x}(s)=u(s'),\ t\leqslant s<\infty,\ x(t)=X,$$

$$\sum_i u_i(s)=1,\ \ t\leqslant s<\infty.$$

This is formally the same as P1, so that theorems 1 and 2 apply. Note also that this is a different problem from that of maximizing the reward for the first completion, which would be formulated

$$\mathrm{MFC}^\Gamma(X,t):\ \text{Maximize}\ \int_t^\infty \sum_i \{ \prod_{j\neq i} P_j(x_j(s))\} r_i(s)f_i(x(s))u_i(s)ds$$

$$\text{subject to}\ \ \dot{x}(s)=u(s),\ x(t)=X,$$

$$\sum_i u_i(s)=1,\ \ t\leqslant s<\infty.$$

We could, of course, as well use the latter objective functional as that of OLP. We are just seeking to generate a sequence of open-loop controllers by solving a surrogate optimization problem, and the precise form of that problem is obviously open to choice. The main reason for concentrating on OLP is its extreme simplicity.

Because OLP no longer contains the recursion implicit in P, the proof of analytic results about OLP, where these exist, is in general much simpler than for P. An example is the proof of theorem 4 of this paper. From the computational point of view, OLP is a particularly simple member of a class of standard problems in optimal control. Well-developed algorithms are available for the solution of such problems. We will not go further into this aspect here, but in the rest of this paper we compare the performance of sequential strategies derived from OLP with known optimal solutions.

Consider, for example, the case where

$$r_i(t)=w_i e^{-\alpha t},\ i=1,2,\ldots,n,$$

where the w_i's are positive constants. The solution of P in this case given by a dynamic priority index ν which is a particular form of the Gittins index [4] for multi-armed bandits:

$$\nu_i(x)=\sup_{x'>x} \left\{ \frac{\int_x^{x'} w_i f_i(\zeta)e^{-\alpha\zeta}d\zeta}{\int_x^{x'} P_i(\zeta)e^{-\alpha\zeta}d\zeta} \right\}.$$

For this class of reward functions, OLP is a deterministic, continuous time multi-armed bandit. Its solution is given by a dynamic priority index β, where

$$\beta_i(x)=\sup_{x'>x}\left\{\frac{\int_x^{x'} w_i f_i(\zeta)e^{-\alpha\zeta}d\zeta}{\int_x^{x'} e^{-\alpha\zeta}d\zeta}\right\}.$$

Thus the sequential open-loop controller for P derived from OLP works as follows. For each $\tau\varepsilon(\tau_r,\tau_{r+1})$, compute $\beta_i(X_i(\tau))$ for each job in $\bar{\Gamma}(\tau)$, on the basis of the current data. Over the interval (τ_r,τ_{r+1}), schedule using the dynamic priority index β. This heuristic has the property that, as the intervals $\{(\tau_r,\tau_{r+1})$: $r=1,2,\ldots\}$ tend to zero in length, it tends to a closed-loop controller for P which is based on the priority index γ, where

$$\gamma_i(x)=\beta_i(x)/P_i(x).$$

Theorem 3.

(i) If $\rho_i(x)$ is a monotonic, non-increasing function of x, then $\gamma_i(x)=\nu_i(x)$ for all x.

(ii) If the functions f_i are all identical, with increasing completion rate ρ, and the weights w_i all equal, then ν and γ determine identical scheduling strategies.

Proof.

(i) If ρ_i is a decreasing function, so is f_i. Thus

$$\gamma_i(x)=w_i\rho_i(x)=\nu_i(x).$$

(ii) When ρ is increasing in x, then so is ν. It therefore suffices to show that this is also true of γ. This is obvious for all values of x for which $\alpha(x)=\rho(x)$. Otherwise, suppose that in the definition of $\beta_i(x)$, the supremum on the RHS is achieved by some value $x'=\zeta(x)(>x)$. Then

$$\gamma(x)>\rho(x).$$

Hence, for some $\delta>0$, and all $\xi\varepsilon(x,x+\delta)$,

$$\gamma(\xi)=\sup_{x'>\xi}\left\{\frac{\int_\xi^{x'} w_i f_i(\zeta)e^{-\alpha\zeta}d\zeta}{P_i(\xi)\int_\xi^{x'} e^{-\alpha\zeta}d\zeta}\right\}$$

$$\geqslant\frac{\int_\xi^{\zeta(x)} w_i f_i(\zeta)e^{-\alpha\zeta}d\zeta}{P_i(\xi)\int_x^{\zeta(x)} e^{-\alpha\zeta}d\zeta}$$

$$>\frac{\int_x^{\zeta(x)} w_i f_i(\zeta)e^{-\alpha\zeta}d\zeta}{P_i(x)\int_x^{\zeta(x)} e^{-\alpha\zeta}d\zeta}=\gamma(x).$$

This theorem shows that the sequential open-loop strategy is the same as the optimal strategy in the limit of small review period, for these particular classes of completion time distribution, and a single processor with this reward structure. We call a sequential open-loop strategy with this property asymptotically optimal.

The theorem extends to the problem of minimizing the weighted flowtime, by taking the limit of the discounted case as $\alpha\to 0$. Part (i) of the theorem extends to a more general class of reward functions. Suppose that

$$r_i=w_i r+d_i,\ i=1,2,\ldots,n,$$

where w_i is a positive constant, d_i is a constant and r is a decreasing function of time. Then the optimal controller for this version of OLP is still given by the dynamic priority index whose value is $w_i f_i$. This result can be proved by the straightforward application of theorem 1 to OLP. In the limit of small review periods, this becomes the strategy determined by the priority index $w_i\rho_i$. This can be shown [5] to be optimal for P, again via theorem 1.

We now consider a class of single-processor problems for which a full optimal strategy is hard to find. Suppose that the jobs represent customers in a queue, and that customers will become dissatisfied and leave (renege) at random times if their service is not already complete. This is equivalent to giving each job a randomly distributed due date, after which no reward can accrue from the completion of that job. Some results concerning the optimal strategy for a variant of this problem when the jobs have a common deadline and the processing time distributions are exponential are in [3]. Let G_i denote the distribution of the time that the i'th customer is prepared to wait, with density g_i and survivor function Q_i. Then the open-loop problem is

$$\mathrm{OLR}^{\Gamma}(X,t)\text{: Maximize } \int_t^{\infty}\Big(\sum_{i\in\bar{\Gamma}} r_i(s)Q_i(s)f_i(x(s))u_i(s)\Big)ds$$

$$\text{Subject to } \dot{x}(s)=u(s),\ t\leq s<\infty,\ x(t)=X,$$

$$\sum_i u_i(s)=1,\ t\leq s<\infty.$$

To maximize the weighted number of satisfied customers, we take $r_i\equiv w_i$ for all i. Then OLR and OLP are the same, with Q_i replacing r_i. As before, the optimal control for OLR when the f_i are decreasing is myopic, provided the derivatives of the Q_i are proportional.

Suppose the customers are all equally impatient, the distributions $\{G_i\}$ being identical and negative exponential. Then the results given by theorem 3 for the asymptotic form of the sequential open-loop strategy hold. This provides an example where the sequential open-loop strategy need not be asymptotically optimal. If n=2, and

$$f_i(x)=\lambda_i e^{-\lambda_i x},\ \ i=1,2,$$

then it is optimal to process job 1 first if

$$w_1\lambda_1(1+\frac{\alpha}{\alpha+\lambda_1})\geq w_2\lambda_2(1+\frac{\alpha}{\alpha+\lambda_2})\ .$$

The asymptotic form of the sequential open-loop strategy is to process the jobs in $w_i\lambda_i$ order. These two indices are equivalent when $w_1=w_2$, and asymptotically equivalent for large λ_1, λ_2 for any weights, but one may readily find values of the parameters for which this is not so.

We remark that for the variant of the problem where there is a single deadline, the sequential open-loop strategy is asymptotically optimal if the conditions of either theorem 3(i) or (ii) hold for the completion time distributions. This follows from the formal equivalence of the full problem with problem P, and of the open loop problem with OLP, with r_i replaced in each case by Q, the survivor function of the deadline distribution.

As a final example, we consider a multi-server problem. Suppose we are interested in minimizing the flowtime for the set of n jobs on m identical machines. Then we add to OLP the constraints

$$u_i(s)\leq 1/m,\ \ i=1,2,\ldots,n,\ \ t\leq s<\infty.$$

Suppose that all the jobs have identical completion time distributions, but have possibly received different amounts of processing. Using theorem 2, it can be shown [8] that if the common completion time distribution has monotone completion rate, then the strategy of processing at each time the jobs with currently greatest completion rate minimizes the expected flowtime. If the common completion time density is log-convex or log-concave, then this strategy minimizes the flowtime in distribution. These results are extensions of similar results for the single server problem P, but their proof is much more difficult. In contrast, the corresponding open-loop results extend easily to the multi server case when f is a decreasing function.

Theorem 4. Suppose the common completion rate distribution of the n jobs has monotone completion rate. The the sequential open-loop strategy asymptotically minimizes the expected flowtime.

Proof.

What we actually prove is that f is a priority index for the problem under these conditions; the result proper then follows as in theorem 3. Observe that, under our assumptions, two jobs which have received the same amount of processing are identical. Thus we can solve OLP with a controller which preserves the initial ordering of the values of f_i, $i=1,2,\ldots,n$, which we may as well take to be the same as the lexicographic order. A simple calculation shows that if $i<k$, then

$$\int_t^\infty \Big(\sum_j H_{ij}(x(s),s)u_j(s)\Big)ds \leqslant \int_t^\infty \Big(\sum_j H_{kj}(x(s),s)u_j(s)\Big)ds$$

with equality only if i and k are identical at time t and receive identical treatment thereafter. The result follows by the maximizing property of the optimal u.

4. CONCLUSION

The use of optimal control theory is a powerful analytical tool in the solution of scheduling problems when the processing can be pre-emptive. Its use in computing optimal scheduling strategies where these are not obtainable analytically is not likely to be practicable. If we seek sequential open-loop strategies, by finding solutions to intuitively plausible surrogate problems, we are faced with control problems which have analytical solutions over a somewhat wider range of conditions than the closed-loop counterpart. Where these problems do not yield an analytic solution, numerical solution should not be difficult in practice. The examples considered here give us some grounds for hope that the resulting strategies will be effective. At the same time, we have shown that the optimal strategies for a range of problems can be characterized as the limit of sequential open-loop controllers.

Before the use in practice of open-loop sequential scheduling can be recommended, a number of further points have to be examined. Chiefly, their performance on problems with more difficult structures than those considered here needs to be tested, probably by computer. Algorithms for solving deterministic optimal control problems are numerous, and it would be worth knowing which perform well on problems like OLP.

REFERENCES

[1] Alam, M.: 1969, An application of modern control theory to a time-dependent queueing system for optimal operation, Int. J. Sys. Sci. 10, pp. 693-700.
[2] Aoki, M.: 1967, "Optimization of Stochastic Systems", Academic Press, New York.
[3] Derman, C., Lieberman, G., and Ross, S.M.: 1978, A renewal decision problem, Management. Sci. 24, pp. 554-561.
[4] Gittins, J.C.: 1976, Bandit processes and dynamic allocation indices, J.R. Stat. Soc. B 41, pp. 148-177.
[5] Nash,P. and Gittins, J.C.: 1977, A Hamiltonian approach to stochastic resource allocation, Adv. App. Prob. 9, pp. 55-68.
[6] Nash, P.: 1969, Controlled jump process models for stochastic scheduling, Int. J. Control 29, pp. 1011-1025.
[7] Scott, C.H. and Jefferson, T.R.: 1976, Optimal regulation of service rate for a queue with finite waiting room, J.O.T.A. 20, pp. 245-250.
[8] Weber,R.R.: 1982, Scheduling jobs with unknown processing requirements on parallel machines to minimize makespan and flowtime, J. App. Prob. 19, (to appear).

The basic optimal control theory underlying these papers can be found in

Lee, E.B. and Markus, L: 1968, "Foundations of Optimal Control Theory". Wiley, New York.

The theory of maximum principles for discrete state, continuous-time Markov processes is discussed in, for example,

Rishel, R.: 1974, in "Lecture notes in Economics and Mathematical systems", Vol 107. Springer, Berlin.

Sworder, D.D.: 1969, I.E.E.E. Trans. Autom. Control 14, 9.

The following paper considers the reduction to discrete semi-Markov decision problems of controlled jump process models in general.

Yushkevitch, A.A.: 1981, On reducing a jump-controllable Markov model to a model with discrete time, SIAM J. Prob. Theory and Appl. 25.