Faculty Working Papers

MARKOV CHAIN DESIGN PROBLEMS

Richard V. Evans, Professor of Business Administration

#474

College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

FACULTY WORKING PAPERS

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

March 15, 1978

MARKOV CHAIN DESIGN PROBLEMS

Richard V. Evans, Professor of Business Administration

#474

Summary:

Markov chain design problems are defined. A gradient algorithm is presented. Numerical results for an example are discussed. A convergence result is proved.

This research was supported, in part, by a 1977 Summer Grant from the Investors in Business Education.

A discrete state Markov design problem has the form

$$\text{Max } (S(\mu), V(\mu))$$

subject to

$$\text{(a)} \quad S(\mu)(I - \beta T(\mu)) = P_0$$

or

$$\text{(b)} \quad S(\mu)(I - T(\mu)) = 0, \ |S(\mu)| = 1$$

and $\mu \in R$

where μ is a design parameter (possibly vector), $V(\mu)$ and P_0 are m dimensional vectors, $T(\mu)$, an m x m transition matrix and β a parameter between 0 and 1. The brackets in the objective function symbolize the scalar product. The vector norm is the sum of the absolute values of the coordinates. The first constraint is equivalent to

$$S(\mu) = \sum_{n=0}^{\infty} \beta^n P_0 T^n(\mu)$$

or $S(\mu)$ is the discounted conditional expected number of visits to the states of the system. The second constraint assumes that

$$\lim_{n \to \infty} P_0 T^n(\mu) = S(\mu)$$

exists and defines the behavior of the system in a typical period. These discrete time discrete state problems are important on their own and as approximations for continuous state and time questions.

The classical example of this type and the problem on which the algorithm of this paper has been used extensively is the design of a finite state

single server queue. The service rate is the single design parameter. This problem can be solved by formula for simple functions $V(\mu)$ but is a problem for numerical analysis for general $V(\mu)$ such as

$$V(\mu) = (V_0(\mu), \ldots, V_m(\mu))$$

$$V_i(\mu) = \mu g \delta(i) - h(i) - c(\mu)$$

$$\delta(i) = \begin{cases} 0 & i = 0 \\ 1 & i > 0 \end{cases}$$

$\mu g \delta(i)$ = the expected revenue from a completed customer

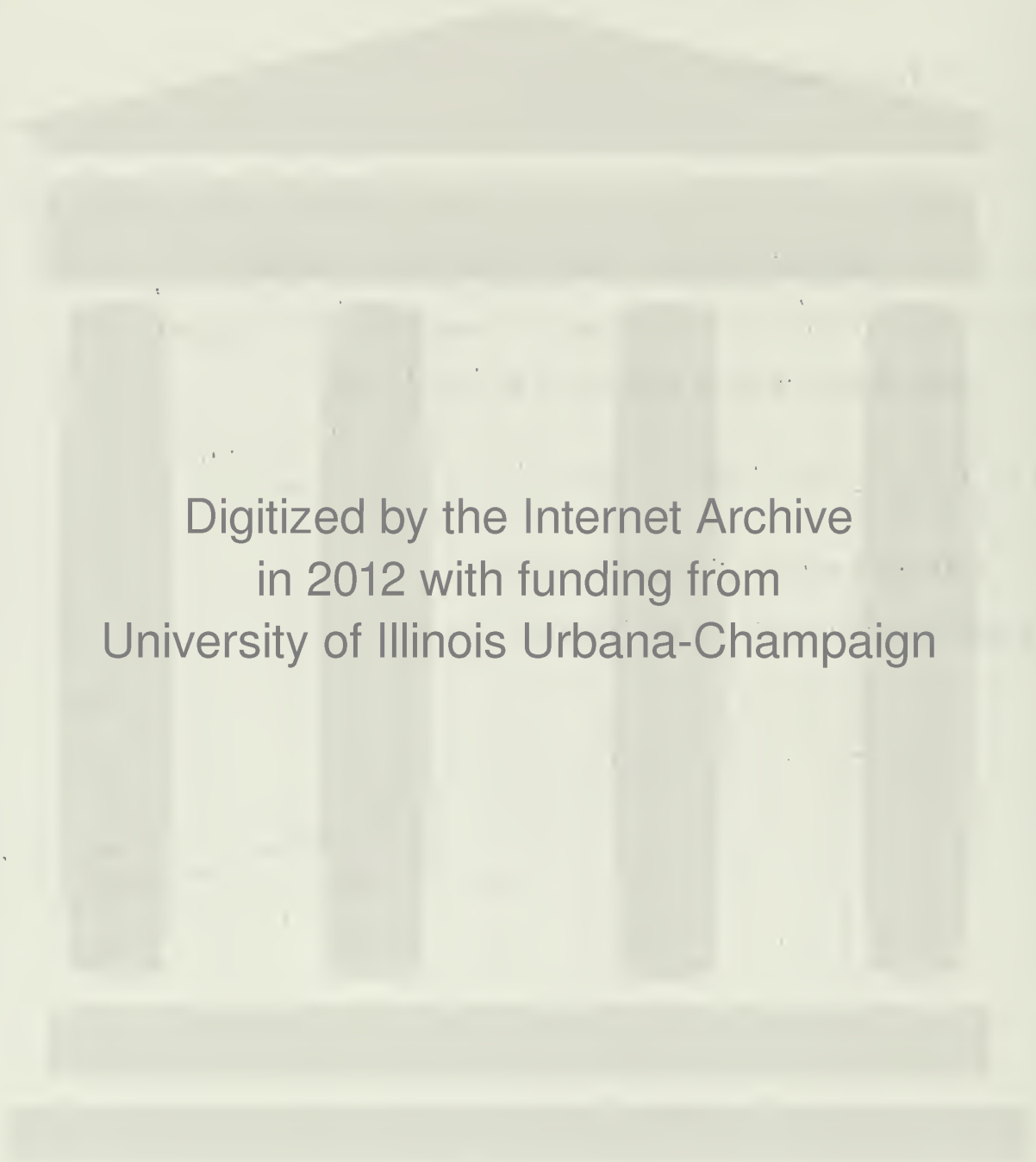
$h(i) = a_1 i + a_2 i^2 + a_3 i^3$ = the cost of congestion in a period which begins with i customers in the system

$c(\mu) = \sum_{i=1}^{6} c_i \mu^i$ = the cost of service rate μ for a period.

This is assumed a convex function.

The transition operator is the familiar matrix

$$T(\mu) = \begin{pmatrix} 1-\lambda & \lambda & 0 & 0 & \ldots & 0 & 0 \\ \mu & 1-\lambda-\mu & \lambda & 0 & \ldots & 0 & 0 \\ & & & & & & \\ 0 & 0 & 0 & 0 & & 1-\lambda-\mu & \lambda \\ 0 & 0 & 0 & 0 & \ldots & \mu & 1-\mu \end{pmatrix}$$



In real terms this problem represents which machine or computer to buy. It is quite different from the control problem of how to adjust the service rate at times at which congestion is observed. In spite of this natural distinction between design and control, there is a relationship. If the optimal control is a stationary policy, the control problem becomes one of choosing a vector $\mu = (\mu_0,\ldots,\mu_m)$ of service rates. The formal optimization problem is the design problem modified so that μ is a vector. Although control problems often are design problems, the reverse is not very frequent. Design parameters often affect costs and transition probabilities of several states. This is equivalent to a constraint on the control problem that the same decision must be selected for several states. In the single server queue example, the service rate must be the same in all states. Dynamic programming algorithms cannot be used when the decision in one state must be the same as that made for another state, for generally this means that the necessary vector maximization is impossible.

The example provides another distinction between design and control problems. The problem as stated contains a cost of using service rate μ but no cost of changing the service rate. The simplest control problem assumes that there are two possible service rates μ_1 and μ_2 and costs $c_{1,2}$ and $c_{2,1}$ for changing from μ_1 to μ_2 and μ_2 to μ_1 respectively. The natural Markov chain uses two dimensional states N,i where N is the number in the system and i is the server which is in use. If the service rate is a continuous variable say $0 \leq \mu \leq 1 - \lambda$ this Markov process becomes an infinite state process which is obviously more complicated than the simple chain describing the number in the system for a single fixed value of the service rate.

The example problem has more structure than most design problems. The design parameter is one dimensional and this suggests using special optimization algorithms. In addition, the transition matrix $T(\mu)$ has enough structure that the constraint equations can be simplified analytically. Although most design problems probably have some useable regularities in $T(\mu)$, discovering how to take advantage of them can take a long time. Perhaps fascination with these equations is the reason why few queueing theory results have been developed to the point that practitioners actually use them to find optimal designs. This investigation adopted the constraint that the computations should be applicable to a variety of design problems without major analytic work. This approach is at best only partially successful. Optimization codes can be constructed which find local optima, but convergence to a single global optimum can only be guaranteed by additional assumptions about the specific problem. This paper contains some general ideas for gradient search inside the set of feasible design parameter values. The presumably simple example is embarassing because it is not possible to guarantee that a local maximum is the global maximum. Nevertheless, all numerical calculations found global optima in about the number of iterations required to compute two values of $S(\mu)$ iteratively for μ values close to the optimum.

Gradient

In continuous parameter nonlinear optimization, the fundamental procedure is a gradient search. Even if gradient procedures are not the best, they provide a standard to which other algorithms can be compared. A gradient search in some bounded region R of the parameter space consists of a routine which produces moves in the gradient direction for points μ strictly inside R and

some special procedures for moves when the current μ value is at or close to the boundary of R and the gradient points outside of R. This paper considers moves inside R and used an obvious modification at the boundary for the one dimensional example in which $0 \leq \mu \leq 1 - \lambda$.

The basic gradient search move for maximizing a function for the vector μ involves computing $\nabla f(\mu)$. The move is from μ to $\mu^* = \mu + \gamma \nabla f(\mu)$ where γ is arbitrary. If $f(\mu^*) > f(\mu)$ the move is a success and either a larger value of γ is tried or μ^* replaces μ and the process repeats. If $f(\mu^*) < f(\mu)$ the move is a failure and a new μ^* is calculated for a smaller value of γ. Much of the calculation is spent computing values of f and ∇f for the sequence of μ values.

Since the extension of the calculation from the derivative with respect to a real variable μ to the partial derivatives with respect to coordinates of a k dimensional vector μ is merely a k-fold repetition of the one dimensional calculation, the discussion will assume μ is a one dimensional variable. In this case, formally for the Markov design problem

$$\frac{d}{d\mu} (S(\mu),V(\mu)) = (S'(\mu),V(\mu)) + (S(\mu),V'(\mu))$$

and either

$$S(\mu)(I - \beta T(\mu)) = P_0$$

$$S'(\mu)(I - \beta T(\mu) - \beta S(\mu)T'(\mu) = 0$$

or

$$S(\mu)\ (I - T(\mu)) = 0 \quad |S(\mu)| = 1$$

$$S'(\mu)\ (I - T(\mu)) - S(\mu)T'(\mu) = 0,\ (S'(\mu),e) = 0$$

where e is a vector of 1's. Both S' and V' are vectors of derivatives with respect to μ and T' is a matrix of derivatives with respect to μ. Thus some form of Gauss elimination can be used to solve for $S(\mu)$ and $S'(\mu)$. The advantage of this is that the calculation time depends on the number of states but is independent of the μ value. Accuracy of the calculation does vary with μ. If $T(\mu)$ has special structure as in the example, this can often be exploited, but it can also and often more easily reduce the computing time to multiply a vector times $T(\mu)$.

Iterative calculations provide a major alternative to elimination. In contrast to Gauss elimination, these procedures provide controllable accuracy. The two aspects of this control are the starting point and the number of iterations used. Thus if $S(\mu)$ and $S'(\mu)$ are used as the starting values in iterative calculations to find $S(\mu^*)$ and $S'(\mu^*)$ the greater $|\mu - \mu^*|$ generally the larger the number of iterations to compute $S(\mu^*)$ and $S'(\mu^*)$ to some prescribed accuracy. This raises the very interesting question of how accurately should $S(\mu)$ and $S'(\mu)$ be computed. Just as it is impossible to prove and generally not true that gradient direction search is the best search, there is at present no demonstrably best accuracy. It is reasonable to expect this is not constant with greater accuracy required the closer one is to identifying a local optimum. If the step size of each move is decreased as the optimum is approached because the constant γ is decreased and or the gradient of the objective function decreases, then the accuracy of any fixed number of iterations in the calculation of successive values of $S(\mu)$ and $S'(\mu)$ should improve. Although decreasing step sizes is characteristic of search procedures a precise description of the accuracy improvement is not available.

The iterative process for computing S and objective function values are

(a) $S_i = P_0 + \beta\, S_{i-1} T(\mu_i) \qquad S_0 = P_0$

$f_i = (S_i, V(\mu_i))$

(b) $S_i = S_{i-1}\, T(\mu_i) \qquad S_0 = P_0$

$f_i = (S_i, V(\mu_i))$

For μ_i constant, the convergence of these processes is well studied. The first converges geometrically because $0 \leq \beta < 1$ with any measure of error bounded by $K\beta^n$ for some suitable K. The second is guaranteed to converge only under additional assumptions. For most Management Science applications it is possible to assume $T^k(\mu_i)$ is strictly positive for some k. This means that $T(\mu)$ has one eigenvalue of absolute value 1 with a 1 dimensional eigensubspace. In this case (b) converges geometrically. From now on assume $T^k(\mu)$ is strictly positive for some k, all μ in R. The dual processes are

(a') $W_i = V(\mu_i) + \beta T(\mu_i)\, W_{i-1} \qquad W_0 = 0$

(b') $W_i = T(\mu)\, W_{i-1} \qquad W_0 = V(\mu)$

The convergence of a' for constant μ_i is a standard dynamic programming in Markov chains result for the degenerate case of 1 action per state. The positive operator assumption guarantees convergence of b' for fixed μ. This process is not well adapted to the current problem since it does not permit μ to be varied easily. At least formally the a' process does not change as much as $(S_i, V(\mu_i))$ when μ_i changes. As sums for $i \geq 1$

$$W_i = V(\mu_i) + \sum_{j=1}^{i} \beta^i \left(\prod_{k=0}^{j-1} T(\mu_{i-k})\right) V(\mu_{i-j})$$

$$(S_i, V(\mu_i)) = V(\mu_i) + \sum_{j=1}^{i} \beta^j \left(\prod_{k=0}^{j-1} T(\mu_{i-k})\right) V(\mu_i)$$

The second expression uses $V(\mu_i)$ in each term of the sum while the first uses the sequence $V(\mu_{i-k})$. Thus in algorithms for Markov chain design problems, the primal probability process seems to be the iteration to use as opposed to the dual conditional expected value process of dynamic programming.

For the vectors of derivatives, there are also iterative schemes

a*) $S_i' = \beta\,(S_{i-1}' T(\mu_i) + S_i\, T'(\mu_i))$

b*) $S_i' = S_{i-1}'\, T(\mu_i) + S_i T'(\mu_i)$

In the hope of improved accuracy S_i is used rather than the anticipated S_{i-1}. Combined with S_i, these iterations produce an approximation of the derivative of the objective function.

$$f_i' = (S_i', V(\mu_i)) + (S_i, V'(\mu_i))$$

For partial derivatives of a vector design parameter each coordinate variable gives rise to an iterative scheme of this form. The approximate gradient of the objective function is the vector of the partial derivative approximations. For one dimensional μ this is just the one number f_i'.

Using the iterative calculation of the objective function value and the gradient within the iterative gradient search produces the iterative process

$$\mu_i = \mu_{i-1} \qquad \text{for } i_j < i \leq i_{j+1} \qquad i_{j+1} - i_j = N_j$$

$$\mu_{i_{j+1}} = \mu_{i_{j+1}-1} + \gamma_j f'_{i_{j+1}-1}$$

and either

(a) $S_i = P_0 + \beta S_{i-1} T(\mu_i)$

$$S'_i = \beta (S'_{i-1} T(\mu_i) + S_i T'(\mu_i))$$

$$f'_i = (S'_{i-1}, V(\mu_i)) + (S_i, V'(\mu_i))$$

$$f_i = (S_i, V(\mu_i))$$

or (b) $S_i = S_{i-1} T(\mu_i)$

$$S'_i = S'_{i-1} T(\mu_i) + S_i T(\mu_i)$$

$$f'_i = (S'_i, V(\mu_i)) + (S_i, V'(\mu_i))$$

$$f_i = (S_i, V(\mu_i))$$

The parameters γ_j and N_j must be supplied as numbers or as rules for their computation. In keeping with the previous description of gradient search, it is necessary to continue to try a value of γ_j and iterate N_j times to compute $f_{i_{j+1}-1}$ to compare it with f_{i_j-1}. Only if $f_{i_{j+1}-1} > f_{i_j-1}$ is the trial value of γ_j adopted, otherwise a smaller trial value is selected and the N_j iterations repeated. This means a greater amount of storage than the iterative process requires if γ_j is not found by a trial and error process. Reducing the computing time by computing only S_i until an acceptable value of γ_j is found means either storing these values or recomputing them in the calculation of S'_i. This problem can be avoided by using $S_{i_{j+1}-1}$ in the S'_i iterati

This may produce a better approximation for $S'(\mu_{i_{j+1}})$. The alternative is to make $\gamma_{j+1} < \gamma_j$ but not change γ_j when $f_{i_{j+1}-1}$ is less than f_{i_j-1}.

Especially if the failure to improve the objective function value is used to decrease the next γ value, it is reasonable to consider using a small value of γ. Since the step size depends on both γ and the gradient, more complete control of the step size can be achieved by using γ times the normalized gradient, i.e.

$$\mu_{i_{j+1}} = \mu_{i_j} + \gamma_j \frac{1}{|f'_{i_{j+1}-1}|} f'_{i_{j+1}-1}$$

There is another interesting characteristic of this iterative scheme. Generally, the larger γ_j the greater N_j will have to be to achieve some fixed level of accuracy for the objective function and gradient. Compared to many non linear programming problems, this calculation of objective function and gradient is relatively expensive. This perhaps suggests constantly using new information as it is developed. Thus, small γ and small N may well provide the most efficient results.

Results for the Example

Rather extensive numerical calculations were performed for the example especially for the discounted expected profit objective function. For this the algorithm used the general iterative process specialize to $N_j \equiv 1$ for and j and

$$S_0 = P_0$$

$$S_0' = 0$$

$$\mu_{n+1} = \begin{cases} \min(\mu_n + \gamma_n, 1-\lambda) & \text{if } f_n' > 0 \\ \max(\mu_n - \gamma_n, 0) & \text{if } f_n' < 0 \end{cases}$$

$$\gamma_{n+1} = \begin{cases} 1/2\ \gamma_n & \text{if } \mu_{n+1} = \mu_{n-1} \\ \gamma_n & \text{otherwise} \end{cases}$$

Stop if f_n' and γ_n are approximately 0 and $S_n - S_{n-1}$ and $S_n' - S_{n-1}'$ are approximately 0. In all cases, the calculation converged in approximately the number of iterations required to compute the objective function at the optimum μ to the required accuracy. This is probably about as fast as this type of algorithm can be expected to operate. For a large number of cases the objective function was computed over the entire range of μ values. The maximum found by this enumeration agreed with that found by the gradient search to the accuracy used in both calculations.

In many cases, the optimum was at the endpoint and it was necessary to adjust the cost parameters to force the optimum into the interior of the interval $0 \leq \mu \leq 1-\lambda$. Even when the optimum was not at an extreme point the objective function tended to be relatively flat close to the optimum and the objective function value at $1-\lambda$ was not greatly below the optimum value. This suggests that the heuristic of, when it is desireable to be in a service business, buy the best possible equipment, may be a good management principle. Unfortunately, no simple description of when this is optimal has been discovered.

Convergence

The numerical work stopped only when successive μ_n, S_n, and S'_n differences were small. This is more testing than necessary because S_n and S'_n converge under reasonably general conditions when the sequence μ_n converges. The proof of this requires only a modest generalization of the results for constant operator processes.

For the discounted process assume that $T(\mu)$ and $T'(\mu)$ are both uniformly continuous and bounded for $\mu \in R$. The successive differences of S_n are

$$S_n - S_{n-1} = \beta S_{n-1} T(\mu_n) - \beta S_{n-2} T(\mu_{n-1})$$

For any ε there is δ such that for n large enough that $|\mu_n - \mu_{n-1}| < \delta$ then $|T(\mu_n) - T(\mu_{n-1})| < \varepsilon$. Moreover, since S_n is the conditional expected number of visits to the various states even for a non constant operator process, the entries in S_n remain less than $1/(1-\beta)$.

Since $\beta < 1$ either

$$|S_n - S_{n-1}| < |S_{n-1} - S_{n-2}|$$

or

$$|S_n - S_{n-1}| < 2\beta K\varepsilon$$

Since ε goes to 0 as n goes to infinity, either condition guarantees that $|S_n - S_{n-1}|$ converges to 0.

For the derivatives

$$|S'_n - S'_{n-1}| \leq \beta\, |S'_{n-1}\, T(\mu_n) - S'_{n-2} T(\mu_{n-1}) + S_n T'(\mu_n) - S_{n-1} T'(\mu_{n-1})|$$

$$\leq \beta\{|S'_{n-1} - S'_{n-2}| + |S'_{n-2}||T(\mu_n) - T(\mu_{n-1})| + |S_n - S_{n-1}||T'(\mu_n)|$$

$$+ |S_n||T'(\mu_n) - T'(\mu_{n-1})|\}$$

Again either $|S'_n - S'_{n-1}| < |S'_{n-1} - S'_{n-2}|$ or it is less than a quantity which becomes arbitrarily small providing $|S'_n|$ is bounded. Since

$$|S_n| \leq 1 + \beta |S_{n-1}|$$

$$|S'_n| \leq \beta |S'_{n-1}| + \beta |S_n||T'(\mu_n)|$$

and $T'(\mu_n)$ was assumed uniformly bounded, there is a constant K such that by induction

$$|S_n| \leq 1/1-\beta$$

$$|S'_n| \leq K \sum_{i=1}^{n} i\, \beta^i = K\, 1/(1-\beta)^2$$

As in the constant operator case, the limiting expected number in the system is more delicate than the discounted expected number in the system. Assume that for each μ, $T^k(\mu)$ is strictly positive for some k. Let the sequence μ_n converge to μ_∞ with transition matrix $T(\mu_\infty)$ in such a way that both

$$\sum_{n=N}^{\infty} \sup_{j \geq n} |T(\mu_j) - T(\mu_\infty)| < g(N)$$

and

$$\sum_{n=N}^{\infty} \sup_{j \geq n} |T'(\mu_j) - T'(\mu_\infty)| < g(N)$$

where $g(N)$ goes to 0 as N goes to infinity. From the recursive expression

$$S_n = S_0 T^n(\mu_\infty) + \sum_{i=0}^{N} S_i (T(\mu_{i+1}) - T(\mu_\infty)) T(\mu_\infty)^{n-i-1}$$
$$+ \sum_{i=N+1}^{n-1} S_i (T(\mu_{i+1}) - T(\mu_\infty)) T^{n-i-1}(\mu_\infty)$$

Now let n go to infinity. From positive operator theory first term converges geometrically to the fixed point of $T(\mu_\infty)$. In addition since S_i is a probability vector, $S_i(T(\mu_{i+1}) - T(\mu_\infty))$ is in the eigen-subspace associated with eigenvalues other than 1. Thus each term in the second sum converges to 0 geometrically and so does the sum. The final term is a vector with norm not exceeding $g(N)$. Letting N go to infinity, the entire right hand side becomes the fixed point of $T(\mu_\infty)$. Thus S_n converges to S_∞ where $S_\infty = S_\infty T(\mu_\infty)$ and $|S_\infty| = 1$.

For the derivative an alternative to the recursive definition of S_n' is

$$S_n' = \sum_{i=1}^{N} (S_i T'(\mu_i) \prod_{j=i+1}^{n} T(\mu_j)) + \sum_{i=N+1}^{n-1} S_i T'(\mu_i) \prod_{j=i+1}^{n} T(\mu_j) + S_n T'(\mu_n)$$

The first term converges to a zero vector. The convergence of S_n and any S_0 implies that the matrix product converges to a matrix of identical rows. Since $T(\mu)$ is a transition matrix for all $\mu \in R$, $T'(\mu_i)$ must have row sums of 0. This implies that $T'(\mu_i)$ times the matrix product must become 0. Thus for all N the first term is 0. Since S_i converges there are δ_i converging to 0 for which $|S_i - S_\infty| < \delta_i$. Since $\prod_{j=i+1}^{n} T(\mu_j)$ and $T^{n-i-1}(\mu_\infty)$ converge to the same operator there must be some finite n for which the norm of their difference has its maximum value and thus some constant ρ such that

$$\left| \prod_{j=i+1}^{n-1} T(\mu_j) - T^{n-i-2}(\mu_\infty) \right| < \rho \sup_{j \geq i} |T(\mu_j) - T(\mu_\infty)|$$

Thus

$$\left| \sum_{i=N+1}^{n-1} S_i T'(\mu_i) \prod_{j=i+1}^{n} T(\mu_j) + S_n T'(\mu_n) - \sum_{i=N+1}^{n-1} S_\infty T'(\mu_\infty) T^{n-i-1}(\mu_\infty) \right| \leq$$

$$\sum_{i=N+1}^{n} (\delta_i |T'(\mu_i)| + \rho \sup_{j \geq i} |T(\mu_j) - T(\mu_\infty)| + |T'(\mu_i) - T'(\mu_\infty)|)$$

Since the bound converges to 0, the iterative process converges to the same limit when μ_j varies as when it is always the limiting value μ_∞ and the initial vector is the fixed point of $T(\mu_\infty)$. Moreover, reordering the terms of the second sum the common limit is

$$\sum_{i=0}^{\infty} S_\infty T'(\mu_\infty) T^i(\mu_\infty)$$

which is S'_∞ [1].

At best these convergence results suggest that the algorithm cannot stop except at a local optimum. They also guarantee that if the last value of N_j must be large and the gradient equal to the origin before stopping can occur, then the error in the approximations of the objective function and gradient will be small. If one combines this with an assumption or better yet a proof that a gradient search will solve the problem, then convergence can be guaranteed. The numerical work seems to support the contention that this approach will increase the cost of the search unnecessarily. Even this simple example runs into theoretical difficulties in an attempt to provide an algorithm which must work. The difficulty comes in guaranteeing that the contribution to the objective function from the congestion penalty h(i) is concave or at least unimodal. This is definitely not concave for all convex h(i) and values of μ, β and m for the discounted objective function.

References

1) Evans, R. V., "Programming Problems and Changes in the Stable Behavior of a Class of Markov Chains," J. Appl. Prob. 8, 543-550.

2) Derman, C., Finite State Markovian Decision Processes, Academic Press, N.Y., 1970.

3) Karlin, S., A First Course in Stochastic Processes, Academic Press, N.Y.., 1966.

4) Kato, T., Perturbation Theory for Linear Operators, Springer Verlag, N.Y., 1966.

5) Kumin, H., The Design of Markovian Congestion Systems, Technical Memorandum No. 115, Operations Research Department, Case Western Reserve University, 1968.

6) Morse, P. M., Queues, Inventories and Maintenance, Wiley, N.Y., 1958.

7) Schweitzer, P. J., "Perturbation Theory and Finite Markov Chains," J. Appl. Prob. 5, 401-413.