Simplified policy iteration for skip-free Markov decision processes

E.J. Collins
Department of Mathematics,
University of Bristol,
University Walk,
Bristol BS8 1TW, UK.

Abstract

We describe and analyse a new simplified policy iteration type algorithm for finite average cost Markov decision processes that are skip-free in the negative direction. We show that the algorithm is guaranteed to converge after a finite number of iterations, but the computational effort required for each iteration step is comparable with that for value iteration. We show that the analysis can be easily extended to solve continuous time models, discounted cost models and communicating models, and provides new insights into the formulation of the constraints in the linear programming approach to skip-free models. We also introduce and motivate a new class of models for multidimensional control problems which we call *skip-free Markov decision processes on trees* and show that the algorithm naturally extends to this wider class of models.

KEYWORDS: Markov decision processes on trees; multidimensional Markov decision processes; x-revised problems

AMS 2000 SUBJECT CLASSIFICATION: PRIMARY 90C40; SECONDARY 90C39

1 Introduction

Markov decision processes (MDPs) provide a class of stochastic optimisation models that have found wide applicability to problems in Operation Research. The standard methods for computing optimal policy are based on value iteration, policy iteration and linear programming algorithms. Each approach has its advantages and disadvantages. In particular, each step in value iteration is relatively computationally inexpensive but the value function may take some time to converge and the algorithm provides no direct check that it has computed the optimal value function and an optimal policy. Conversely, each step in policy iteration may be computationally expensive but the algorithm can be proved to converge in a finite number of steps, confirms when it has converged and automatically identifies the optimal value function and an optimal policy on exit.

Here we focus on models with special structure, in that they are *skip-free in the negative direction* (Keilson 1965, p.10) or *skip-free to the left* (Stidham & Weber 1989); i.e. whatever the action taken, the process cannot pass from one state to a 'lower' state without passing through all the intervening states. Such skip-free models arise naturally in many areas of OR. The most obvious examples are the control of discrete time random walks and continuous time birth and death processes (Serfozo 1981) such as queueing control problems with single unit arrivals and departures (see, for example, Stidham & Weber (1989) and references therein). In these basic one-dimensional models, the state space S is (a subset of) the integer lattice and transitions are only possible to the next higher or lower integer state. However there are several other standard OR models that fall within the wider one-dimensional skip-free framework including examples from the areas of inventory control (Miller 1981) and reliability and maintenance (Derman 1970, Thomas 1982).

Previous treatments of controlled skip-free processes have considered only the one-dimentional formulation. For processes with the 'skip-free to the left' property, work has focussed on qualitative properties, in particular the existence of monotone optimal policies for models with appropriately structured cost functions (Stidham & Weber 1989, Stidham & Weber 1999). Conversely, work on processes with the corresponding 'skip-free to the right' property has concentrated on analysis of an approximating bisection method for countable state space models (Wijngaard & Stidham 1986, Wijngaard & Stidham 2000).

One way of characterising the essential features of a finite skip-free model is in terms of the following properties: (i) there is a single distinguished state, say 0; (ii) for any other state i there is a unique shortest path from i to 0; (iii) from each state $i \neq 0$ the process can only make transitions to either the adjacent state in the unique path from 0 to i, or to some state j for which i lies in the unique shortest path from 0 to j. Thus the model is skip-free if and only if the state space can be identified with the graph of a finite tree, rooted at 0, with each state corresponding to a unique node in the tree.

In this setting, the one-dimensional skip-free model above, with state space $S = \{0, 1, \ldots, M\}$, corresponds to the simplest case where each interior node is connected to just two adjacent nodes and the tree reduces to a single linearly ordered branch connecting the root node 0 to the terminal (or leaf) node M. However, the analysis extends easily to cases where the state space has a richer, possibly multidimensional, structure. Here, the analogue of the simple birth and death process is a *tree process* (Keilson 1979), in which transitions are only possible to states corresponding to adjacent nodes in the tree. Examples of genuinely skip-free models with multi-dimensional state spaces arise in simple multi-class queueing systems with batch arrivals (Yeung & Sengupta 1994, He 2000, and references therein), but such treatments have focussed mainly on describing the behaviour of the process for fixed parameter settings.

In this paper we consider finite state MDPs that are skip-free in the negative direction. For the standard recurrent average cost skip-free model, our main contribution is a new simplified policy iteration algorithm in which the computational effort required for each iteration step is comparable with that for value iteration, but which is guaranteed to converge after a finite number of iterations and which automatically identifies the optimal value function and an optimal policy on exit. In the more general setting, our contribution is what appears to be the first development and analysis of multidimensional MDP models on trees, the extension of the simplified policy iteration algorithm to skip-free MDP models on trees, and a corresponding proof of the convergence properties of the extended algorithm. In both cases, the analysis can be extended to continuous time models, discounted cost models and communicating models, and provides new insights into the formulation of the constraints in the linear programming approach to skip-free models.

The remaining sections are organized as follows. In Section 2, we describe the standard discrete time skip-free model with finite state space $S = \{0, 1, \ldots, M\}$. We identify the appropriate average cost optimality equations for recurrent models, develop an interpretation in terms of a corresponding 'x-revised' problem, and present and prove convergence for the new policy iteration algorithm. In Section 3, we show that, with simple modifications, results for the new policy iteration algorithm can be extended to continuous time average cost models, discounted cost models and communicating models on S, and that they lead to an alternative set of constraints in LP formulations of the average cost problem. Finally, in Section 4, we introduce and illustrate a new class of multidimensional MDP models, and show how the optimality equations, the new policy iteration algorithm and the convergence results can all be extended to the skip-free average cost multidimensional setting, together with its continuous-time, discounted, communicating and LP variations.

2 The skip-free MDP model

Consider a discrete time Markov decision process (MDP) with finite state space $S = \{0, 1, 2, \ldots, M\}$ over an infinite time horizon $t \in \{0, 1, 2, \ldots\}$. Associated with each state $i \in S$ is a non-empty finite set of possible actions; since S is finite, we assume without loss of generality that the set of actions A is the same for each i. If action $a \in A$ is chosen when the process is in state $X_t = i$ at time t, then the process incurs an immediate cost $c_i(a)$ and the next state is $X_{t+1} = j$ with probability $p_{ij}(a)$.

When S is a subset of the integer lattice, we say the MDP model is *skip-free in the negative direction* (Keilson 1965, Stidham & Weber 1989) if $p_{ij}(a) = 0$ for all $j < i - 1$ and $a \in A$, i.e. the process cannot move from each state i to a state with index $j < i$ without passing through all the intermediate states. To avoid degeneracy, we assume that $p_{00}(a) < 1$ for $a \in A$ and that for each $i \in \{1, \ldots, M\}$, $p_{ii-1}(a) > 0$ for at least one $a \in A$. A similar definition applies to MDP models that are *skip-free in the positive direction* or *skip-free to the right* (Wijngaard & Stidham 1986, Wijngaard & Stidham 2000); on the finite integer lattice the models are interchangeable, in that each can be converted to the other by an appropriate relabelling of the states.

Obvious examples include the many applications where the process can be modelled as a controlled random walk or (in continuous time) a controlled birth and death process (Serfozo 1981), such as arrival and service rate control for $M/M/1$ queues with finite buffers. Simple discrete time examples with non-degenerate skip-free transitions include (i) inventory control with single-item demands, where the state i is the stock level, the action a is the amount ordered, and transitions are only possible to states $j = i - 1$ (demand and no re-order), $j = i + a$ (demand plus re-order), or $j = i + a$; (ii) maintenance/replacement problems where the state i is the performance level of a machine (0 = broken and must be replaced, M = newly replaced), where the state deteriorates by at most one level each time period, and where the maintenance action a determines the probability the state will improve to, say, $j = i + k$, and may include deterministic transitions to state M under a replacement action. Simple continuous time examples with non-degenerate skip-free transitions include control of finite $M/M/1$ queues with batch arrivals (Stidham & Weber 1989), and perhaps less obvious examples such as control of $M/E_K/1$ queues (Stidham & Weber 1989). In the latter case, each service is composed of K exponential stages, and a state $i = rK + s$ denotes a situation where there are r jobs currently waiting in the buffer and the job currently in service has s stages left to complete. Thus transitions are only possible to states $j = i - 1$ (the next service stage is completed) or $j = (r + 1)K + s$ (a new job arrives to the buffer). More generally, the model applies to control of those $M/PH/1$ queues, for which the phase-type (PH) service distribution is itself skip-free, in that transitions from stage/phase s are only possible to stages/phase $K, K - 1, \ldots, s, s - 1$.

For finite skip-free MDP models on the integer lattice, it is often more convenient to define the upper tail probabilities

$$\bar{p}_{ij}(a) \equiv P(X_{t+1} \geq j \,|\, X_t = i, A_t = a) = \sum_{s=j}^{M} p_{is}(a)$$

and to assume that the model is specified in terms of the parameters

$$p_{ii-1}(a),\ 1 \leq i \leq M, a \in A \qquad \bar{p}_{ij}(a),\ 0 \leq i < j \leq M, a \in A.$$

We will see that it is easier to represent quantities and perform calculations in terms of the $\bar{p}_{ij}(a)$, rather than use the standard (and equivalent) representation in terms of the transition probabilities $p_{ij}(a),\ i, j \in S,\ a \in A$.

A policy π is a sequence of (possibly history dependent and randomised) rules for choosing the action at each given time point t. A *deterministic* decision rule corresponds to a function $d : S \to A$ and specifies taking action $a = d(i)$ when the process is in state i. A *stationary deterministic* policy is one which always uses same the deterministic decision rule at each time point t. Where the meaning is clear from the context, we use the same notation d for both the decision rule and the corresponding stationary deterministic policy.

We say an MDP model is *recurrent* if the transition matrix corresponding to every stationary deterministic policy consists of a single recurrent class. In particular, this implies that, for all $a \in A$, $p_{ii-1}(a) > 0$ for $i = 1, \ldots, M$ and $p_{ii}(a) < 1$ for all $i \in S$. We say an MDP model is *communicating* if, for every pair of states i and j in S, j is reachable from i under some (stationary deterministic) policy d; i.e. there exists a policy d, with corresponding transition matrix P_d, and an integer $n \geq 0$, such that $P_d(X_n = j | X_0 = i) > 0$. In contrast to recurrent models, communicating models allow there to be i and a with $p_{ii-1}(a) = 0$ and /or $p_{ii}(a) = 1$.

The expected average cost incurred by a policy π with initial state i is given by $g_\pi(i) = \limsup_{n\to\infty} \frac{1}{n}\ E_\pi \left(\sum_{t=0}^{n-1} C_{X_t}(a_t) | X_0 = i\right)$, where X_t is the state at time t and a_t is the action chosen at time t under π. Similarly, for a given discount factor $0 < \beta < 1$, the total expected discounted cost incurred by a policy π with initial state i is given by $V_\pi^\beta(i) = E_\pi \left(\sum_{t=0}^{\infty} \beta^n C_{X_t}(a_t) | X_0 = i\right)$.

For simplicity of presentation, we restrict attention in the remainder of this section to determining a policy which, for each initial state, minimises the expected average cost in a given finite recurrent discrete time average cost skip-free MDP model. We defer to Section 3 the extension of these results to continuous time, discounted cost and communicating models.

2.1 Skip-free average cost optimality equations

For finite recurrent models, the solution to the expected average cost problem can be characterised by the corresponding *average cost optimality equations* (Puterman 1994, §8.4)

$$h_i = \min_{a \in A} \{ \ c_i(a) - g + \sum_{j=0}^{M} p_{ij}(a) h_j \ \} \qquad\qquad i \in S \qquad (1)$$

in that: (i) there exist real numbers g^* and $h_i^*, i \in S$ satisfying the optimality equations; (ii) the optimal average cost is the same for each initial state and is given by g^*; (iii) the optimality equations uniquely determine g^* and determine the h_i^* up to an arbitrary additive constant; (iv) the stationary deterministic policy d^* is an average cost optimal policy, where, for each $i \in S$, $d^*(i)$ is an action achieving $\min_a \{ \ c_i(a) + \sum_{j=0}^{M} p_{ij}(a) h_j^* \ \}$.

For simplicity of presentation, we now assume throughout that the actions have been labelled in some strictly ordered fashion and that, when a minimum over actions is required, the corresponding action is uniquely defined by taking the minimising action to be the one with the lowest valued label in the case of ties.

It follows from (iv) above that there is an optimal policy in the class of stationary deterministic policies. We therefore restrict attention from now on to stationary deterministic policies, writing ‘policy’ as a shorthand for ‘stationary deterministic policy’.

For each $i, j \in S$, we can interpret $h_i^* - h_j^*$ as the asymptotic relative difference in the total cost that results from starting the process in state i rather than state j, under the stationary deterministic policy d^*. Thus the quantities $h_i^* - h_j^*$ are uniquely defined, but the quantities $h_i^*, i \in S$ are defined only up to an arbitrary additive constant. We focus on the particular solution normalised by setting $h_0^* = 0$ and refer to the corresponding h_i^* as the normalised relative costs under an optimal policy.

In general, the optimality equations (1) cannot be solved directly. Instead an optimal policy in the class of stationary deterministic policies is usually found by methods based on value iteration, policy iteration or linear programming, or combinations of these approaches (Puterman 1994). For skip-free models, however, $p_{ij}(a) = 0$ for $j < i - 1$ and equations (1) take the simpler form

$$h_i = \min_{a} \{ \ c_i(a) - g + \sum_{j=i-1}^{M} p_{ij}(a) h_j \ \} \qquad\qquad i = M, \ldots, 0. \qquad (2)$$

Now $c_i(a) - g + \sum_{j=i-1}^{M} p_{ij}(a) h_j \geq h_i$ if and only if $c_i(a) - g + \sum_{j=i+1}^{M} p_{ij}(a)(h_j - h_i) \geq (h_i - h_{i-1}) p_{ii-1}(a)$, with appropriate modifications for $i = 0$ and M, and equality in one expression implies equality in the other. Also $\sum_{j=i+1}^{M} p_{ij}(a)(h_j - h_i) = \sum_{j=i+1}^{M} p_{ij}(a) \sum_{k=i+1}^{j} (h_k - h_{k-1}) = \sum_{k=i+1}^{M} (h_k - h_{k-1}) \sum_{j=k}^{M} p_{ij}(a) = \sum_{k=i+1}^{M} (h_k - h_{k-1}) \bar{p}_{ik}(a)$. Thus, writing y_i for $h_i^* - h_{i-1}^*, \ i = 1, \ldots, M$, and using the normalisation $h_0^* = 0$, we see that for skip-free models the optimality equations are equivalent to the set of equations

$$y_M = \min_a \{ \ (c_M(a) - x)/p_{MM-1}(a) \ \} \tag{3a}$$

$$y_i = \min_a \{ \ (c_i(a) - x + \sum_{k=i+1}^{M} \bar{p}_{ik}(a) y_k)/p_{ii-1}(a) \ \} \qquad i = M-1, \dots, 1 \tag{3b}$$

$$0 = \min_a \{ \ c_0(a) - x + \sum_{k=1}^{M} \bar{p}_{ik}(a) y_k \ \} \tag{3c}$$

in that (i) these equations also have unique solutions x and $y_1, \dots, y_M$; (ii) the optimal average cost is $g^* = x$ and the normalised relative costs under an optimal policy are $h_i^* = y_1 + \cdots + y_i, \ i = 1, \dots, M$; (iii) an optimal policy is given by d^*, where $d^*(i)$ is any action minimising the rhs of the ith equation.

In the optimality equations (3), the value of y_M depends only on x, and in each subsequent equation the value of y_i depends only on the values of y_k for $k > i$. Thus, if the value of x was known, it would be easy to compute the y_i in turn for $y_M, \dots, y_1$ and to determine the corresponding policy, defined below, which takes the optimal action in each state $i = 0, \dots, M$.

Definition 1 *For fixed x, let $a_0, a_1, \dots, a_M$ be the actions minimising the rhs in equations (3) and let $y_1, \dots, y_M$ be the corresponding y values. Define the 'optimality equation' policy d_{oe} to be the policy for which $d_{\text{oe}}(i) = a_i, \ i = 0, 1, \dots, M$. For ease of reference write a_{oe} for $d_{\text{oe}}(0)$, so*

$$a_{\text{oe}} \equiv \text{argmin}_a \{ \ c_0(a) - x + \sum_{k=1}^{M} \bar{p}_{ik}(a) y_k \ \}$$

This observation motivates an iterative approach to finding an average cost optimal policy – choose an initial value for the average cost x, compute the updated 'optimality equation' policy d_{oe} for that x and compute its average cost, set x equal to this new value and iterate until convergence. The principle underlying this iterative approach idea is not new. Low (1974) used a similar solution method but his results were restricted to a specific birth and death model. Other treatments of skip-free models (Wijngaard & Stidham 1986, Stidham & Weber 1989, Stidham & Weber 1999, Wijngaard & Stidham 2000) have used iterative methods to search for a good approximation for the average cost x, based on the value of current and previous approximations, or used the form of the optimality equations to derive qualitative properties of the solution, in particular monotonicity of optimal policies, but neither approach explicitly identified the simple policy improvement algorithm described here.

In contrast, we prove directly that that the policy computed at each stage, using iterations based on d_{oe}, either provides a strict improvement in the average cost or has converged to an optimal policy. Moreover, we develop in the next section an alternative viewpoint that offers more insight into the problem and that helps identify other, possibly better, ways of performing the policy updating process.

2.2 Policy improvement

In this section we will identify and compare three slightly different ways of improving a current policy, and use insights from the corresponding x-revised problem and renewal-reward theory to show that, in each case, the updated policy either provides a strict improvement in the average cost or the process has converged to an optimal policy.

We start our analysis of the average cost model by defining a related problem that we will call the *x-revised first return problem*. The model for this problem has the same state space S, the same action space A and the same transition probabilities $\{p_{ij}(a)\}$ as the average cost model, but the immediate costs are revised downward by the fixed amount x, so $c_i(a)$ is revised to $c_i(a) - x$. The objective for this new problem is to find a policy that minimises the expected x-revised cost until first return to state 0, where, for a process starting with $X_0 = 0$, we define the first return epoch to state 0 to be the smallest value $\tau > 0$ such that $X_{\tau-1} \neq 0$ and $X_\tau = 0$. The MDP is assumed recurrent under any stationary deterministic policy, so τ is well defined and almost surely finite.

Since the process is Markov and skip-free in the negative direction, it follows that a policy minimises the expected x-revised cost until first return to state 0 if and only if it also minimises the expected x-revised total cost until first passage to state 0 for each starting state $i \neq 0$ and hence, for each state $i = 1, \ldots, M$, minimises the expected cost until first passage from i to to $i - 1$. This latter problem has been called the x-revised first passage problem (Stidham & Weber 1989). For each fixed x, let $a_1, \ldots, a_M$ be actions minimising the rhs in equations (3a) and (3b) above and $y_1, \ldots, y_M$ be the corresponding y values. Then the policy d that takes action $d(i) = a_i$ in state i is optimal for the x-revised first passage problem and the minimal expected cost until first passage from i to $i - 1$ is given by y_i (Stidham & Weber 1989). It follows that the policy that uses actions a_i in state $i = 1, \ldots, M$ has the property that for each state $i = 1, \ldots, M$ it also minimises the expected total x-revised cost until first passage to state 0 and that the minimum expected x-revised total cost until first passage to state 0, starting in state $i > 0$, is given by $y_i + y_{i-1} + \cdots + y_1$.

Now consider a process that starts in state 0. Under a policy that specifies action a in state 0, the expected time until the process first leaves state 0 is $1/(1 - p_{00}(a))$ and during that time it incurs x-revised costs at rate $c_0(a) - x$ per unit time. Conditional on leaving state 0, the first transition is to state j with probability $p_{0j}(a)/(1 - p_{00}(a))$. From above, the minimum additional expected total cost until the process next re-enters state 0 is $y_j + y_{j-1} + \cdots + y_1$, and this minimum expected cost is achieved by the policy that takes actions a_i in states $i = 1, \ldots, M$. Thus, if a policy takes action a in state 0, the minimum expected x-revised cost from leaving state 0 until first return to state 0 is $\sum_{j=1}^{M} p_{0j}(a)(y_j + y_{j-1} + \cdots + y_1)/(1 - p_{00}(a)) = \sum_{j=1}^{M} p_{0j}(a)(\sum_{k=1}^{j} y_k)/(1 - p_{00}(a)) = \sum_{k=1}^{M} y_k(\sum_{j=k}^{M} p_{0j}(a))/(1 - p_{00}(a)) = \sum_{k=1}^{M} y_k \bar{p}_{0k}(a)/(1 - p_{00}(a))$. It follows that the optimal action in state 0 is one that minimises the quantity $(c_0(a) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a) y_k)/(1 - p_{00}(a))$. We summarise this analysis in the following definition and lemma.

Definition 2 *For fixed x, let $a_1, \ldots, a_M$ be actions minimising the rhs in equations (3a) and (3b) and let $y_1, \ldots, y_M$ be the corresponding y values. We define the 'first return' policy d_{fr} to be the policy for which $d_{\text{fr}}(i) = a_i,\ i = 1, \ldots, M$, and $d_{\text{fr}}(0) = a_{\text{fr}}$, where*

$$a_{\text{fr}} \equiv \text{argmin}_a \{\ (c_0(a) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a) y_k)/(1 - p_{00}(a))\ \}.$$

Lemma 3 *For given x, d_{fr} is an optimal policy for the x-revised first return problem and the expected x-revised first return cost under d_{fr} is*

$$(c_0(a_{\text{fr}}) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_{\text{fr}}) y_k)/(1 - p_{00}(a_{\text{fr}})).$$

We now show that, if x corresponds to the average cost under some policy d, then the average cost under d_{fr} is no greater than $x = g(d)$. For each fixed policy d on $S = \{0, 1, \ldots, M\}$, write $\tau(d)$ for the expected first return epoch for a process that starts in 0, $C(d)$ for the expected first return cost under d, $H(d)$ for the expected x-revised first return cost under d, and $g(d)$ for the expected average cost under d. To relate these quantities, we view the average cost problem from a renewal-reward perspective. Since state 0 is recurrent under any stationary deterministic policy d, it follows (Ross 1970, p.160) that

$$g(d) = C(d)/\tau(d). \tag{4}$$

In terms of $\tau(d)$ and $C(d)$, the expected x-revised cost under d until first return to state 0 is given by

$$H(d) = C(d) - x\tau(d) \tag{5}$$

since costs are adjusted downwards by an amount x for a time period with expected length $\tau(d)$. Moreover, from this and (4), we have

$$g(d) = x + H(d)/\tau(d). \tag{6}$$

This enables us to compare d_{fr} to a current policy d through the following lemma.

Lemma 4 *Let x correspond to the average cost under some given policy d and let d_{fr} be an optimal policy for the x-revised first return problem. Then:*
(i) the average cost under d_{fr} is no greater than the average cost under d,
(ii) if the average cost under d_{fr} is the same as the average cost under d then d_{fr} is an optimal policy for the average cost problem.

Proof (i) For the fixed x, d_{fr} is by definition an optimal policy for the x-revised first return problem. Thus $H(d_{\text{fr}}) \leq H(d)$, and from (5) this implies $C(d_{\text{fr}}) - x\tau(d_{\text{fr}}) \leq C(d) - x\tau(d)$. Because x corresponds to the average cost under d, then, from (4), $x = g(d) = C(d)/\tau(d)$ so $C(d) - x\tau(d) = 0$. Thus, $H(d_{\text{fr}}) = C(d_{\text{fr}}) - x\tau(d_{\text{fr}}) \leq 0$ and $g(d_{\text{fr}}) = C(d_{\text{fr}})/\tau(d_{\text{fr}}) \leq x = g(d)$.
(ii) If $g(d_{\text{fr}}) = g(d)$, then from above $H(d_{\text{fr}}) = H(d) = 0$. But, from Lemma 3, $H(d_{\text{fr}}) = (c_0(a_{\text{fr}}) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_{\text{fr}})y_k)/(1 - p_{00}(a_{\text{fr}}))$, where $p_{00}(a_{\text{fr}}) < 1$. It follows that $H(d_{\text{fr}}) = 0 \implies (c_0(a_{\text{fr}}) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_{\text{fr}})y_k) = 0$. Thus, when $g(d_{\text{fr}}) = g(d)$, the values $x = g(d_{\text{fr}})$ and $y_1, \ldots, y_M$ satisfy the optimality equations (3a-3c) and d_{fr} is a decision rule corresponding to the actions minmising the rhs of each equation. It follows that d_{fr} is an optimal average cost policy, the optimal average cost is $g^* = g(d_{\text{fr}}) = g(d)$ and the normalised relative costs under the optimal policy are $h_j^* = y_j + \cdots + y_1$, $j = 1, \ldots, M$. □

For any fixed policy d, a similar argument to that preceding Definition 2 can be used to derive the expected x-revised first return cost $H(d)$, the expected first return cost $C(d)$ and the expected first return epoch $\tau(d)$. Say d specifies action b_j in states $j = 0, \ldots, M$. For these fixed b_j, let $w_M = (c_M(b_M) - x)/p_{MM-1}(b_M)$ and let $w_j = (c_j(b_j) - x + \sum_{k=j+1}^{M} \bar{p}_{jk}(b_j)w_k)/p_{jj-1}(b_j)$ for $j = M-1, \ldots, 1$. Then, by considering the possible actions in state j to be restricted to just b_j, it follows that $w_j + w_{j-1} + \cdots + w_1$ can be interpreted as the expected x-revised total cost under d until first passage back to state 0 following the transition to state j. Hence the expected x-revised first return cost is

$$H(d) = (c_0(b_0) - x + \sum_{k=1}^{M} \bar{p}_{0k}(b_0)w_k)/(1 - p_{00}(b_0)). \tag{7}$$

Moreover, from (5), $C(d)$ corresponds to the expected x-revised first return cost for the special case $x = 0$, so $C(d)$ can be computed in exactly the same way, but taking $x = 0$ and making appropriate adjustments in the equations for $w_1, \ldots, w_M$.

For the expected first return epoch under d, write $t_0 = \tau(d) > 0$ and write $t_i > 0$ for the expected first passage time t_i from i to $i-1$. Interpret t_0 as the expected 0-revised first return cost under d for a model with immediate costs $c_i(a) = 1$ for all states and actions (and with $x = 0$), with a similar interpretation for the t_i. Then, as with the w_i, the t_i can be computed as recursively using the following equations:

$$\begin{aligned} t_M &= 1/p_{MM-1}(b_M) \\ t_i &= (1 + \sum_{k=i+1}^{M} \bar{p}_{ik}(b_i)t_k)/p_{ii-1}(b_i) \qquad\qquad i = M-1, \ldots, 1 \\ t_0 &= (1 + \sum_{k=1}^{M} \bar{p}_{0k}(b_0)t_k)/(1 - p_{00}(b_0)) \end{aligned} \tag{8}$$

If the policy d specifies actions b_j and has expected x-revised first passage costs w_j and times t_j, then, from (7) and (8), $H(d) = (c_0(b_0) - x + \sum_{k=1}^{M} \bar{p}_{0k}(b_0)w_k)/(1 - p_{00}(b_0))$ and and $\tau(d) = (1+\sum_{k=1}^{M} \bar{p}_{0k}(b_0)t_k)/(1-p_{00}(b_0))$, so $H(d)/\tau(d) = (c_0(b_0)-x+\sum_{k=1}^{M} \bar{p}_{0k}(b_0)w_k)/(1+\sum_{k=1}^{M} \bar{p}_{0k}(b_0)t_k)$. However, $g(d) = x + H(d)/\tau(d)$ from (6). Since x is fixed, this motivates updating a current policy with the policy d_{mi} defined below and characterised in the following lemma.

Definition 5 *For fixed x, let $a_1, \ldots, a_M$ be actions minimising the rhs in equations (3a) and (3b) and let $y_1, \ldots, y_M$ be the corresponding y values. We define the 'minimising' policy d_{mi} to be the policy for which $d_{\text{mi}}(i) = a_i, \ i = 1, \ldots, M$, and $d_{\text{mi}}(0) = a_{\text{mi}}$, where*

$$a_{\text{mi}} \equiv \text{argmin}_a \{ \ (c_0(a) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a)y_k)/(1 + \sum_{k=1}^{M} \bar{p}_{0k}(a_0)t_k) \ \}.$$

Lemma 6 *For given x, d_{mi} minimises the average cost $g(d)$ over all policies d that take action a_i in states $1, \ldots, M$.*

For fixed x, the policies $d_{\text{oe}}, d_{\text{fr}}$ and d_{mi} specify the same actions in states $i = 1, \ldots, M$, but in general they may all specify different actions in state 0. Nevertheless, the following lemma establishes that all three policies exhibit the same qualitative behaviour relative to the fixed value x.

Lemma 7 *For fixed x, the values of $g(d_{\text{oe}}) - x$, $g(d_{\text{fr}}) - x$ and $g(d_{\text{mi}}) - x$ are either all positive, all negative or all zero. If x corresponds to the average cost under a given policy d then either all three policies strictly improve on d or all three policies provide an optimal average cost policy.*

Proof For fixed x and any policy d, $g(d) - x = H(d)/\tau(d)$ from (6) and $\tau(d)$ is positive, so $g(d) - x$ has the same sign as $H(d)$. Since all three policies take actions a_i in states $i = 1, \ldots, M$, expression (7) gives their expected x-revised first return costs as $H(d) = (c_0(a_0) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_0)y_k)/(1 - p_{00}(a_0))$, where a_0 is the action they specify in state 0 and where $p_{00}(a_0) < 1$ by the assumptions of the skip-free model.

Now $H(d_{\text{oe}}) < 0 \implies (c_0(a_{\text{oe}}) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_{\text{oe}})y_k)/(1 - p_{00}(a_{\text{oe}})) < 0 \implies (c_0(a_{\text{fr}}) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_{\text{fr}})y_k)/(1 - p_{00}(a_{\text{fr}})) < 0$ (as a_{fr} minimises this quantity over choice of a) $\implies H(d_{\text{fr}}) < 0$. Conversely $H(d_{\text{fr}}) < 0 \implies (c_0(a_{\text{fr}}) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_{\text{fr}})y_k)/(1 - p_{00}(a_{\text{fr}})) < 0 \implies (c_0(a_{\text{fr}}) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_{\text{fr}})y_k) < 0 \implies (c_0(a_{\text{oe}}) - x + \sum_{k=1}^{M} \bar{p}_{0k}(a_{\text{oe}})y_k) < 0$ (as a_{oe} minimises this quantity over choice of a) $\implies H(d_{\text{oe}}) < 0$. A similar argument utilising the definition of a_{mi} and the positivity of $(1 + \sum_{k=1}^{M} \bar{p}_{0k}(a_0)t_k)$ shows that $H(d_{\text{oe}}) < 0 \iff H(d_{\text{mi}}) < 0$. Exactly similar arguments then show that $H(d_{\text{oe}}) = 0 \iff H(d_{\text{fr}}) = 0 \iff H(d_{\text{mi}}) = 0$, and that $H(d_{\text{oe}}) > 0 \iff H(d_{\text{fr}}) > 0 \iff H(d_{\text{mi}}) > 0$. The second part of the lemma then follows from Lemma 4. □

2.3 Policy iteration algorithm

We now return to where we left off at the end of Section 2.1. For a given current policy d with average cost x, we know from Lemma 7 that updating the policy using any of the three policies $d_{\text{oe}}, d_{\text{fr}}$ and d_{mi} will in each case result in either an improved policy with strictly smaller average cost, or will confirm both d and the updated policy as being average cost optimal.

The update using d_{mi} has the property that, for each current policy d, it generates an improved policy with average cost at least as small as the other two policies. This does not immediately guarantee that improvements using d_{mi} converge faster than improvements using either d_{oe} or d_{fr}. After one iteration, each policy may take us to a different starting point for the next iteration, and our results do not allow us to compare the policies from these different starting points – indeed it might be that the larger the improvement from the first iteration, the smaller the improvement resulting from the second iteration, as the average cost is now closer to the optimal value. Our experience has been that the number of iterations taken by all three methods was often the same. Where one was fastest, it was always d_{mi}, but there were some parameter settings where d_{oe} was observed to be faster than d_{fr} and others for which the order was reversed.

Informed by this discussion, we define the following PIA (policy iteration algorithm) based on d_{mi} and summarise its properties in the accompanying theorem. Note that other ways of initialising the algorithm are possible – one alternative being to set $g_0 = \max_{i,a} c_i(a)$.

PIA (Policy iteration algorithm)

1. Initialisation:

Choose an arbitrary initial policy d_0. Perform a single iteration of step 2 below, with $x = 0$ and with a_i restricted to the single value $d_0(i)$, $i = 0, 1, \ldots, M$. Compute the average cost g_0 under this policy by setting $g_0 = u_0$.

2. Iteration:

Set $x = g_n$.

- For $i = M$ compute:

$$\begin{array}{rcl} a_M & = & \text{argmin}_a\{\ (c_M(a) - x)/p_{MM-1}(a)\ \} \\ y_M & = & (c_M(a_M) - x)/p_{MM-1}(a_M) \\ t_M & = & 1/p_{MM-1}(a_M) \end{array}$$

- For $i = M-1, \ldots, 1$ compute:

$$\begin{array}{rcl} a_i & = & \text{argmin}_a\{\ (c_i(a) - x + \sum_{k=i+1}^M \bar{p}_{ik}(a) y_k)/p_{ii-1}(a)\ \} \\ y_i & = & (c_i(a_i) - x + \sum_{k=i+1}^M \bar{p}_{ik}(a_i) y_k)/p_{ii-1}(a_i) \\ t_i & = & (1 + \sum_{k=i+1}^M \bar{p}_{ik}(a_i) t_k)/p_{ii-1}(a_i) \end{array}$$

- For $i = 0$ compute:

$$\begin{array}{rcl} a_0 & = & \text{argmin}_a\{\ (c_0(a) - x + \sum_{k=1}^M \bar{p}_{0k}(a) y_k)/(1 + \sum_{k=1}^M \bar{p}_{0k}(a) t_k)\ \} \\ u_0 & = & (c_0(a_0) - x + \sum_{k=1}^M \bar{p}_{0k}(a_0) y_k)/(1 + \sum_{k=1}^M \bar{p}_{0k}(a_0) t_k) \end{array}$$

Set $d_{n+1}(i) = a_i$ for $i = 0, \ldots, M$ and set $g_{n+1} = g_n + u_0$.

3. Termination:
If $u_0 < 0$ then return to step 2.

If $u_0 = 0$ then stop and return d_{n+1} as an optimal policy, g_{n+1} as the optimal average cost, and $h_i = y_1 + \cdots + y_i,\ i = 1, \ldots M$ as the corresponding normalised relative costs. □

Theorem 8 *Consider the PIA above applied to a finite recurrent discrete time average cost skip-free MDP model with state space $S = \{0, 1, 2, \ldots, M\}$. Then:*
(i) At each iteration of the PIA either $g_{n+1} < g_n$ and d_{n+1} is a strict improvement on d_n, or $g_{n+1} = g_n$ and d_{n+1} is an optimal average cost policy.
(ii) The PIA converges after a finite number of iterations.

Proof (i) Let d_n be be the policy identified at iteration n, with expected average cost $g(d_n)$. At iteration $n+1$ the PIA computes g_{n+1} and d_{n+1}, where $d_{n+1} = d_{\text{mi}}$ for the fixed value $x = g_n = g(d_n)$. Initially $g_0 = g(d_0)$ by construction, and $g_n = g(d_n)$ implies $g_{n+1} \equiv g_n + H(d_{n+1})/\tau(d_{n+1}) = g(d_{n+1})$ from (6). Thus, by induction, $g_{n+1} = g(d_{n+1})$. Finally, from Lemma 7, either $g(d_{n+1}) = g(d_{\text{mi}}) < x = g(d_n)$ or $g(d_{n+1}) = g(d_{\text{mi}}) = x = g(d_n)$ and d_{n+1} is an optimal average cost policy.

(ii) Since the set of possible stationary deterministic decision rules is finite, and each iteration prior to convergence leads to a strict improvement and hence a strictly different decision rule, the process must converge after a finite number of steps. □

The computational requirement for each iteration in step 2 of the PIA is similar to that of the corresponding step in value iteration. The comparison depends on the relative density of non-zero elements in P and $\bar{P}$ where, for a fixed policy d with transition probabilities $p_{ij} \equiv p_{ij}(d(i))$, we write P for the matrix with elements p_{ij} and write $\bar{P}$ for the same matrix but with the values $\bar{p}_{ij}$ replacing p_{ij} for elements above the diagonal. Iterations for the PIA and for value iteration will require roughly the same effort for a random walk model, where P is at its sparsest and $P = \bar{P}$, and in the most dense case when all the elements above the diagonal in P are positive and P and $\bar{P}$ have zeros in the same positions. For cases in between, $\bar{P}$ will always be more densely filled than P, an extreme example being when all states have positive probability of making transitions to the largest state (c.f. Section 3.2), in which case P may be quite sparse but $\bar{P}$ will be fully dense above the diagonal. Moreover, the algorithm differs from standard policy iteration, in that it computes relative costs under a policy (d_{n+1}) that does not correspond to the average cost ($g(d_n)$) under consideration; only at convergence do the relative costs and average cost correspond to the same (optimal) policy.

3 Variations on the standard model

In this section, we show that, with simple modifications, the results in Section 2 for the new policy iteration algorithm can be extended to continuous time models, discounted cost models and communicating models, and that they lead to an alternative formulation for the constraints in LP treatments of the average cost problem.

3.1 Continuous time models

Consider a continuous time Markov decision process (CTMDP) with finite state space $S = \{0, 1, 2, \ldots, M\}$ and finite action space A. The analysis in Section 2 easily extends to this continuous time setting. We assume that when the current action is a and the process is in state $X_t = i$, the process incurs costs at rate $c_i(a)$ and makes transitions to state $j \in S$ at rate $q_{ij}(a)$ (where transitions back to the same state are allowed). For infinite horizon problems, under either an average cost or a discounted cost criterion, we can restrict attention to stationary policies and to models in which decisions are made only at transition epochs (Puterman 1994, p.560). For simplicity of presentation we again restrict attention to recurrent models and defer treatment of unichain and communicating models to Section 3.3. As for MDPs, we say a CTMDP is skip-free in the negative direction if the process cannot move from each state i to a state $j < i$ without passing through all the intermediate states, i.e. $q_{ij}(a) = 0$ for all $j < i - 1$ and $a \in A$.

To apply the PIA, we first convert the model to an equivalent uniformised model (Lippman 1975) with rate $\Lambda = \max_{i \in S\, a \in A} \sum_{j \in S} q_{ij}(a)$. In this model, when the current action is a and the process is in state i, transitions back to state i occur at rate $\Lambda - \sum_{j \neq i} q_{ij}(a)$ while transitions to state $j \neq i$ occur at rate $q_{ij}(a)$, so that overall transitions occur at uniform rate Λ. Next we construct a discrete time problem with the same state and action space, and with transition probabilities and immediate costs given by:

$$\begin{aligned} p'_{ij}(a) &= q_{ij}(a)/\Lambda, & c'_i(a) &= \Lambda c_i(a), & i \neq j &= 0,1,\dots,M, \quad a \in A \\ p'_{ii}(a) &= 1 - \sum_{j \neq i} q_{ij}(a)/\Lambda, & & & i &= 0,1,\dots,M. \quad a \in A \end{aligned}$$

If the original CTMDP is recurrent and skip-free, then the discretised model is recurrent and skip-free and can be solved using the PIA.

Finally, the optimal policy d^* and the optimal average cost g^* for the uniformised continuous time problem are the same as the corresponding quantities d' and g' for the discrete time problem, and the normalised relative costs for the uniformised problem are given in terms of those for the discrete problem by $h^*_i = h'_i/\Lambda,\ i = 0,1,\dots,M$ (Puterman 1994, §11.5).

3.2 Discounted cost models

Although our treatment has concentrated on average cost problems, the new policy iteration algorithm can also be applied to find an optimal stationary deterministic policy and the corresponding optimal value function for skip-free discounted cost models. Consider an MDP model that is skip-free in the negative direction, with state space $S = \{0,1,\dots,M\}$, finite action space A, transition probabilities $p_{ij}(a)$, immediate costs $c_i(a)$ and discount factor β.

Following Derman (1970, p.31), we construct an average cost MDP with modified state space $\{0,1,\dots,M,M+1\}$ and modified transition probabilities and immediate costs given by:

$$\begin{aligned} p'_{ij}(a) &= \beta p_{ij}(a), & c'_i(a) &= c_i(a), & i,j &= 0,1,\dots,M, & a &\in A \\ p'_{M+1\,M}(a) &= \beta, & c_{M+1}(a) &= 0, & & & a &\in A \\ p'_{i\,M+1}(a) &= 1-\beta, & & & i &= 0,1,\dots,M+1, & a &\in A \end{aligned}$$

In the spirit of similar models (Low 1974, Wijngaard & Stidham 1986), we note that this new average cost MDP inherits from the original model the property of being skip-free in the negative direction.

Let g' and h'_i, $i = 0,\dots,M+1$ be the optimal average cost and the corresponding relative costs for the new average cost problem, normalised by setting $h'_0 = 0$. From above, g' and h'_i, $i = 1,\dots,M+1$, are the unique solutions to the optimality equations (1), and any set of actions achieving the minimum on the rhs defines an optimal policy. In terms of the original parameters, these equations take the form

$$\begin{aligned} h'_{M+1} &= -g' + \beta h'_M + (1-\beta)h'_{M+1} \\ h'_i &= \min_a \{\, c_i(a) - g' + \beta \sum_{j=0}^{M} p_{ij}(a)h'_j + (1-\beta)h'_{M+1} \,\} \qquad i = 0,\dots,M \end{aligned}$$

Now set $v_j = h'_j - h'_{M+1} + g'/(1-\beta),\ j = 0, \ldots, M$. Then rewriting the equations for $h_0, \ldots, h_M$ in terms of $v_0, \ldots, v_M$, we see that the v_i satisfy the equations

$$v_i = \min_a \{ \, c_i(a) + \beta \sum_{j=0}^{M} p_{ij}(a) v_j \, \} \qquad\qquad i = 0, \ldots, M.$$

Thus the v_j satisfy the optimality equations for the discounted cost problem, and so represent the unique optimal β discounted cost function (Puterman 1994, p.148).

Finally, let x' and $y'_0, \ldots, y'_{M+1}$ be solutions to the policy iteration algorithm applied to the new skip-free average cost problem. Then $g' = x'$ and $h'_j = y'_j + \cdots + y'_1$, $j = 1, \ldots, M+1$. Thus the optimal value function for the discounted problem is given explicitly in terms of the output of the policy iteration algorithm by

$$v_j = x'/(1-\beta) - (y'_{j+1} + \cdots + y'_{M+1}) \qquad\qquad j = 0, \ldots, M$$

and a policy which is optimal for the modified average cost problem is also optimal for the original discounted cost problem.

3.3 Communicating models

So far we have assumed the MDP model is recurrent. There are natural applications for which this assumption excludes sensible policies, such as policies that are recurrent only on a strict subset of S. Simple examples include: maintenance/replacement problems where a policy might specify replacing an item when the state reached some lower level $K > 0$ with a item of level $L < M$; inventory problems where a policy might reorder when the stock reached some lower level $K > 0$ and/or reorder up to level $L < M$; queueing control problems where a policy might turn the server off when the queue size reached some lower level $K > 0$ and/or might refuse to admit new entrants when the queue size reached level $L < M$. In each case, determining optimal values for K and L might be part of the problem. In this section we extend our result to the wider class of communicating MDP models, to enable us to address examples like these.

We say an MDP model is communicating if, for every pair of states i and j in S, j is reachable from i under some (stationary deterministic) policy d; i.e. there exists a policy d, with corresponding transition matrix P_d, and an integer $n \geq 0$, such that $P_d(X_n = j | X_0 = i) > 0$. We say that d is unichain if it decomposes S into a single recurrent class plus a (possibly empty) set of transient states; if there is more than one recurrent class we say d is multichain. Let d be a multichain policy and, for each k, let g_k denote the average cost under d starting in a state in E_k, and let E_m be a recurrent set with smallest average cost, say g_m. Because the model is skip-free, E_m must consist of a sequence of consecutive states $K_m, \ldots, L_m$; again, because the model is skip-free, the action in each each state j greater than L_m can be changed if necessary so that E_m is reachable from j; finally, because the model is communicating, the action in each state j less than K_m can be changed if necessary so that E_m is reachable from j. Denote by d' the new policy created by changing actions in this way, if necessary, but leaving the actions in E_m unchanged. Then d' is unichain by construction, and the average cost starting in each state $j \in S$ is g_m, which is no greater than the average cost starting in j under d. Thus, for average cost skip-free communicating models, nothing is lost by restricting attention to unichain policies.

In contrast to recurrent models, communicating models allow there to be i and a with $p_{ii}(a) = 1$ and/or $p_{ii-1}(a) = 0$. For each $r = 0, 1, \ldots, M$, let U_r be the (possibly empty) set of unichain policies d for which $p_{rr-1}(d(r)) = 0$ but $p_{ii-1}(d(i)) > 0$ for $i = r+1, \ldots, M$ (where we take $p_{ii-1}(a) \equiv 0$ for all a for $i = 0$). Every unichain policy must be in U_r for some r. Partition the possible actions for each state $i \in S$ into $B_i = \{a \in A : p_{ii-1}(a) > 0\}$ and its complement $\bar{B}_i = \{a \in A : p_{ii-1}(a) = 0\}$, where $\bar{B}_i$ may be empty but B_i is non-empty by the assumptions of the skip free model in Section 2. Then for a unichain policy $d \in U_r$, we have that $d(i) \in B_i,\ i = r+1, \ldots, M$; that state r is recurrent and $d(r) \in \bar{B}_r$ by definition; and that states $i < r$ are transient.

Thus the minimum average cost over policies in U_r is the same as the minimum average cost for a modified skip-free MDP model Π_r with the same transition probabilities and immediate costs but with reduced state space $S_r = \{r, \ldots, M\}$ and with state-dependent action spaces $A_i = B_i$ for $i = r+1, \ldots, M$ and $A_r = \bar{B}_r$. In this notation, the model of Section 2 corresponds to Π_0 and state r plays the same role as the recurrent distinguished state in Π_r that state 0 plays in Π_0. If we compare the result of applying the PIA to Π_r with the result of applying it to Π_0, we see that, for the same current value of x, the algorithm computes the same values of y_i, t_i, and a_i in states $i = M, M-1, \ldots, r+1$. However, in state r, the PIA applied to Π_r computes quantities appropriate to the distinguished state, say a^r and u^r, where

$$\begin{aligned} a^r &= \operatorname{argmin}_{a \in \bar{B}_r} \{\ (c_r(a) - x + \textstyle\sum_{k=r+1}^{M} \bar{p}_{rk}(a) y_k)/(1 + \textstyle\sum_{k=r+1}^{M} \bar{p}_{rk}(a) t_k)\ \} \\ u^r &= (c_r(a^r) - x + \textstyle\sum_{k=r+1}^{M} \bar{p}_{rk}(a^r) y_k)/(1 + \textstyle\sum_{k=r+1}^{M} \bar{p}_{rk}(a^r) t_k) \end{aligned}$$

and computes an updated 'minimising' policy d^r_{n+1} with average cost g^r_{n+1},where

$$\begin{aligned} d^r_{n+1}(r) &= a^r;\ \ d^r_{n+1}(i) = a_i,\ i = r+1, \ldots, M, \quad \text{and} \\ g^r_{n+1} &= \qquad\qquad\qquad x + u^r. \end{aligned}$$

This motivates the following modified PIA. First, it includes these extra computations for each state r, so that, in a single iteration, it simultaneously computes the optimal policy d^r_{n+1} and its average cost g^r_{n+1} for each S_r. Secondly, at the end of the $n-1$th iteration it sets $x = g_n = \min_r g^r_n$, and sets d_n to be the corresponding policy, where ties are broken by choosing the d^r_n with the smallest index r. Say the minimum average cost at this stage is achieved by a policy with index $r = K$ Then, by the properties of the PIA applied to Π_K, at the end of the next iteration either (i) $g^K_{n+1} < g^K_n = x$, in which case $g_{n+1} = \min_r g^r_{n+1} < x = g_n$; or (ii) $u^K_{n+1} = 0$ and $g^K_{n+1} = g^K_n = x = \min_r g^r_{n+1}$, so $g_{n+1} = g_n$ and $d_{n+1} = d^K_{n+1}$ is an optimal average cost policy for starting states $i = K, \ldots, M$. In this case, because the model is communicating, it is possible (Puterman 1994, p.351) to modify the actions chosen by the policy in the, now transient, states $0, \ldots, K-1$ so that the modified d_{n+1} satisfies the optimality equations for all states $0, \ldots, M$ and is an average cost optimal policy. We summarise this discussion in the following theorem.

Theorem 9 *Consider the PIA modified as above applied to a finite communicating discrete time average cost skip-free MDP model with state space $S = \{0, 1, 2, \ldots, M\}$. Then:*
(i) At each iteration of the PIA either $g_{n+1} < g_n$ and d_{n+1} is a strict improvement on d_n, or $g_{n+1} = g_n$ and for some K the policy satisfies the optimality equations for states $K, \ldots, M$.
(ii) The modified PIA converges after a finite number of iterations.

Finally, note that it is easy to check if a skip-free model is communicating. An assumption of the (non-degenerate) skip-free model was that each state $i < M$ was reachable from $i + 1$. It follows that a skip-free MDP with state space $S = \{0, 1, \ldots, M\}$ is communicating if and only if M is reachable from 0 under at least one stationary deterministic policy d. Let $N_0 = 0$, let N_1 be the index of the maximum state j for which $p_{0j}(a) > 0$ for some $a \in A$, and for $m = 1, 2, \ldots$ let N_{m+1} be the index of the maximum state j for which $p_{ij}(a) > 0$ for some $0 \leq i \leq N_m$ and $a \in A$. As the state space is finite, the sequence $\{N_m\}$ terminates, say with state N. Since the model is skip-free, N is the largest state that is reachable by all states below it, and the model is communicating if and only if $N = M$.

3.4 Linear programming formulation

Finite recurrent average cost MDP models can also be solved using a standard linear programming approach (Puterman 1994, §8.8) in which the form of the primal LP given below follows directly from the standard optimality equations (1).

Standard primal: Maximize g subject to

$$g + h_i - \sum_{j \in S} p_{ij}(a) h_j \leq c_i(a), \qquad i \in S,\ a \in A$$

In a similar way, the new skip-free optimality equations (3) give rise to the following reformulated primal LP for recurrent average cost skip-free models, where the form of the constraints genuinely differs from those of the standard primal applied to a model with $p_{ij}(a) = 0$ for $j < i-1$.

Skip-free primal: Maximize x subject to

$$\begin{aligned} x + p_{MM-1}(a) y_M &\le c_M(a) && a \in A \\ x + p_{ii-1}(a) y_i - \sum_{k=i+1}^{M} \bar{p}_{ik}(a) y_k &\le c_i(a) && i = 1, \dots, M-1,\ a \in A \\ x - \sum_{k=1}^{M} \bar{p}_{ik}(a) y_k &\le c_0(a) && a \in A \end{aligned}$$

Corresponding to the standard primal LP is a standard dual LP, given as follows:

Standard dual: Minimize $\sum_{i\in S} \sum_{a\in A} z_{ia} c_i(a)$ subject to

$$\begin{aligned} \sum_{a\in A} z_{ia} - \sum_{j} \sum_{a\in A} z_{ja} p_{ji}(a) &= 0, && i \in S \\ \sum_{i\in S} \sum_{a\in A} z_{ia} &= 1 \\ z_{ia} &\ge 0, && i \in S,\ a \in A \end{aligned}$$

Although the dual LP can be derived directly by duality arguments from the primal, it also has the following intuitive explanation. Interpret z_{ia} as the stationary probability of being in state i and taking action a under a given policy. Then $\sum_{a\in A} z_{ia}$ represents the stationary probability of being in state i under the policy and $\sum_{i\in S} \sum_{a\in A} z_{ia} c_i(a)$ represents the corresponding average cost. The stationary probabilities $\{\xi_i\}$ for an irreducible finite Markov chain with transition probabilities $\{p_{ij}\}$ are the unique positive number that satisfy the full balance equations $\xi_i = \sum_j \xi_j p_{ji}$ and the normalisation equation $\sum_i \xi_i = 1$. Thus the standard dual LP for an average cost MDP can be interpreted as the minimisation of the average cost $\sum_{i\in S} \sum_{a\in A} z_{ia} c_i(a)$ subject to the constraints that the $\{z_{ia}\}$ correspond to probabilities satisfying the appropriate full balance and normalisation equations.

As in the standard case, the following skip-free dual LP can be derived from the skip-free primal LP by direct duality arguments.

Skip-free dual: Minimize $\sum_{i\in S} \sum_{a\in A} z_{ia} c_i(a)$ subject to

$$\begin{aligned} \sum_{a\in A} z_{ia} p_{ii-1}(a) - \sum_{j=0}^{i-1} \sum_{a\in A} z_{ja} \bar{p}_{ji}(a) &= 0, && i = 1, \dots, M \qquad (9) \\ \sum_{i\in S} \sum_{a\in A} z_{ia} &= 1 \\ z_{ia} &\ge 0, && i = 0, \dots, M;\ a \in A \end{aligned}$$

Even though the constraints (9) do not immediately appear to be in a suitable form, even allowing for the skip-free nature of the process, this skip-free dual LP is capable of the same simple intuitive interpretation as the standard dual. To see this, consider a skip-free process with state space $S = \{0, 1, \ldots, M\}$ and transition probabilities $\{p_{ij}\}$. Let E be any subset of S and set $E^c = S \setminus E$. In equilibrium the flow between E and E^c must balance, so any solution $\{\xi_i\}$ to the full balance equations satisfies the equations $\sum_{j \in E} \sum_{k \in E^c} \xi_j p_{jk} = \sum_{k \in E^c} \sum_{j \in E} \xi_k p_{kj}$. Taking $E = \{i, \ldots, M\}$ in turn for $i = 1, \ldots, M$, using the skip-free nature of the process and writing $\bar{p}_{ij}$ for $\sum_{s=j}^{M} p_{is}$, these set balance equations reduce to

$$\xi_i p_{ii-1} = \sum_{j=0}^{i-1} \xi_j \bar{p}_{ji} \qquad i = 1, \ldots, M. \tag{10}$$

Since each ξ_i can be solved for recursively in terms of ξ_0, equations (10), together with the normalisation equation, uniquely determine the stationary distribution and it is precisely these modified balance equations that generate the constraints (9). Although outside the focus of our results, we note that the upper Hessenberg form of the constraint matrix corresponding to (9) may lead to simplifications in the linear program (Reid 1982).

4 Multidimensional skip-free models

In this section we show how the skip-free MDP model and the PIA introduced in Section 2 can be generalised from the integer lattice to models with a richer, possibly multidimensional, state space structure, illustrating our approach with examples that arise naturally in the control of some multi-class queueing systems. Recall from Section 1 that a finite skip-free model has the following properties: (i) there is a single distinguished state, say 0; (ii) for any other state i there is a unique shortest path from i to 0; (iii) from each state $i \neq 0$ the process can only make transitions to either the adjacent state in the unique path from 0 to i, or to some state j for which i lies in the unique shortest path from 0 to j. Thus the state space of a finite skip-free model can be identified with the graph of a finite tree, rooted at 0, with each state corresponding to a unique node in the tree.

So far we have dealt only with the case where, for each k, there was exactly one state for which the shortest path to state 0 had length k. Thus there was a 1–1 mapping of the states to the integers $\{0, 1, \ldots, M\}$ such that the distinguished state mapped to 0 and the state for which the shortest path had length k mapped to k. We now extend consideration to models that satisfy the same essential features, but where, for each k, there may be more than one state for which the shortest path has length k. In this case, rather than mapping to the integer lattice, there is a fixed tree $\mathcal{T}$ (in the graph theoretic sense) such that each state corresponds to a unique node of the tree, with the distinguished state mapping to the root node. It may help to visualise movement between states in terms of the corresponding movement between nodes on the tree.

We start by describing two continuous time examples that exemplify the type of model we have in mind. After going on to develop a more formal treatment of the model and deriving the corresponding optimality equations, we set out the modified form of the PIA appropriate for these multi-dimensional tree-form models. We show that the algorithm has the same convergence properties as the simpler PIA derived earlier, and that the approach can be extended from the recurrent discrete time average cost model to continuous time models, discounted models and communicating models in a similar way to the extensions described in Section 3.

4.1 Example: Pre-emptive multi-class queueing system

Consider (He 2000, Yeung & Sengupta 1994) a single server multi-class queueing system with exponential interarrival and service time distributions. Assume there are K customer classes and that the system has finite capacity J, including the job (if any) in service, and that jobs that arrive when the system is full are lost. Assume the service discipline is pre-emptive, so that a job that arrives when the system is not full enters service immediately and the job currently in service at that point return to the head of the buffer. When a job completes service, the server next serves the job at the head of the buffer. The memoryless property of the exponential distribution implies that the remaining service time of a job that resumes service has the same distribution as the original service time, independent of the any service received up to the point of resumption.

The state of the system can be fully described by the multidimensional state vector $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_J)$ where κ_1 denotes the class of the job currently in service, κ_j denotes the class of the job waiting for service in place $j,\ j = 2, \ldots, J$, and $\kappa_j = 0$ if there are less than j jobs in the queue so the jth place is empty.

A simple service-rate control model might be that that class k jobs arrive singly according to class dependent arrival rate λ_k, and complete service at class and action dependent service rate $\mu_k(a)$. The possible transitions under this model are:

(i) the arrival of a class k job ($k = 1, \ldots, K$) to a partially full system (with $\kappa_J = 0$); this corresponds to the transition $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_J) \rightarrow \boldsymbol{\kappa}' = (\kappa'_1, \ldots, \kappa'_J)$ where $\kappa'_1 = k$ and $\kappa'_j = \kappa_{j-1},\ j = 2, \ldots, J$, and occurs at rate $\lambda_k(\boldsymbol{\kappa}, a) = \lambda_k$,

(ii) the completion of the job currently in service; this corresponds to the transition $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_J) \rightarrow \boldsymbol{\kappa}' = (\kappa'_1, \ldots, \kappa'_J)$ where $\kappa'_j = \kappa_{j+1},\ j = 1, \ldots, J-1$ and $\kappa_J = 0$, and occurs at rate $\mu(\boldsymbol{\kappa}, a) = \mu_{\kappa_1}(a)$.

Figure 1 illustrates the tree corresponding to the state space for a system with $K = 2$ job classes and with capacity $J = 3$. From the figure, we see that at any given node a service completion moves the state to the adjacent 'lower' node and an arrival moves the state to one of the two adjacent ‘higher’ nodes, depending on the class of the arriving job.

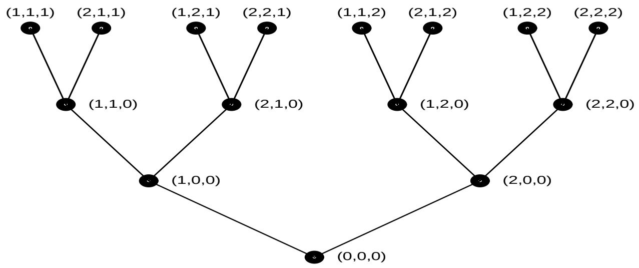

Figure 1: The tree corresponding to the state space for a pre-emptive multi-class queueing system with $K = 2$ job classes and capacity $J = 3$. The head of the queue (the job in service) is on the left.

The model can be easily extended to allow for more general forms of class, state and action dependent arrival and service rates. It can also be generalised to allow for batch arrivals with given (state and action dependent) batch size, type and order distributions. For example, one model might be that, if a batch of size r with job types $\omega_1, \dots, \omega_r$ in that order arrived at a system with $s \leq r$ free places, then the last $r - s$ jobs in the batch would be lost and the remaining jobs would pre-emptively join the system, so the arrival would correspond to the transition $\boldsymbol{\kappa} = (\kappa_1, \dots, \kappa_J) \to \boldsymbol{\kappa}' = (\kappa'_1, \dots, \kappa'_J)$ where $\kappa'_1 = \omega_1, \dots, \kappa'_s = \omega_s$, and $\kappa'_j = \kappa_{j-s}, \; j = s + 1, \dots, J$.

In terms of the tree representation, such batch arrivals move the system from a given node to one of the ‘higher’ nodes in the sub-tree rooted at that given node. For example, the arrival of a batch of size 2 with classes $(2, 1)$ in that order to a system with a single job of class 1 in figure 1 would move the state from node $(1, 0, 0)$ to the node $(2, 1, 1)$ in the sub-tree rooted at $(1, 0, 0)$.

4.2 Example: Pre-emptive multi-class priority queueing system

As a second example, consider again a single server multi-class queueing system with exponential interarrival and service time distributions, with K customer classes and with finite capacity (total buffer size) J, where jobs that arrive when the system is full are lost.

Now, however, assume that classes served pre-emptively in order of priority, with higher numbered classes having higher priority; that within each class jobs are served in order of arrival; and that an arriving job of class k is lost if the job already in service has class $r > k$. More precisely, if a class k job arrives and the job in service also has class k, then the arriving job joins the buffer at the tail of the class k jobs but ahead of any jobs of lower class, whereas if the job in service has class $r < k$, the arriving job pre-empts the job in service and enters service immediately while the pre-empted job returns to the head of the buffer. When a job completes service, the server next serves the job at the head of the buffer. The priority discipline described above implies that the jobs in the buffer are ordered in non-increasing order of class, and that the job at the head of the buffer has class no higher than the job currently in service.

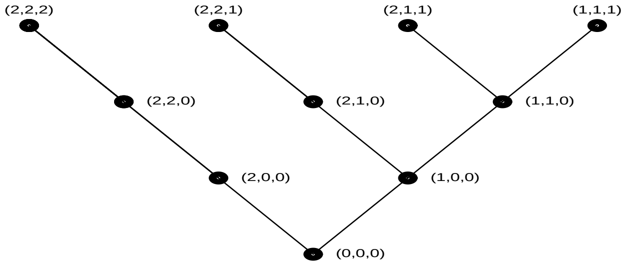

Figure 2: The tree corresponding to the state space for a pre-emptive multi-class priority queueing system with $K = 2$ job classes and capacity $J = 3$. The head of the queue (the job in service) is on the left.

Again, the state of the system can be fully described by a multidimensional state vector $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_J)$ where κ_1 denotes the class of job currently in service and κ_j denotes the class of the job waiting for service in place j, $j = 2, \ldots, J$, and the model can allow for possibly general forms of class, state and action dependent arrival and service rates $\lambda_k(\boldsymbol{\kappa}, a)$ and $\mu(\boldsymbol{\kappa}, a)$. Extensions are possible which allow batch arrivals with given batch size and type distribution.

Figure 2 illustrates the tree corresponding to the state space for such a system, again with $K = 2$ job classes and with capacity $J = 3$. Jobs of class 1 can only enter the system if there are no class 2 jobs already present. From the figure, we see that at any given node a service completion again moves the state to the adjacent 'lower' node and an arrival again moves the state to one of the two adjacent 'higher' nodes, depending on the class of the arriving job. For example, if a class 1 job arrives to a system with a single class 1 job present, it joins the buffer and the state moves from from node $(1, 0, 0)$ to node $(1, 1, 0)$; if a class 2 job arrives in the same situation, it enters service and the pre-empted class 1 job goes to the head of the buffer, so the state moves from from node $(1, 0, 0)$ to node $(2, 1, 0)$;

4.3 Skip-free MDP models on trees

To formalise the examples above, we start by considering a finite rooted tree $\mathcal{T}$ with $N+1$ nodes labelled $0, 1, 2, \ldots, N$, with root node 0, and with a given edge set. The tree structure implies that for each pair of nodes i and j there is a unique minimal path (set of edges) in the tree that connects i and j. Thus the nodes in the tree can be partitioned into level sets $L_0 = \{0\}, L_1, \ldots, L_M$ such that, for $m = 0, \ldots, M-1$, $i \in L_{m+1}$ if and only if the minimal path from 0 to i passes through exactly m intermediate nodes. For adjacent nodes $i \in L_m$ and $j \in L_{m+1}$, we say i is the parent of j and j is a child of i if the minimal path from 0 to j passes through i. More generally, for $i \in L_m$ and $j \in L_r$, $r > m$, we say j is a descendant of i if the minimal path from 0 to j passes through i. Each node $j \neq 0$ has a unique parent. We write $\rho(j)$ for the parent of j, we write $\mathcal{D}(j)$ for the set of descendants of j, and we write $\mathcal{T}(j) \subset \mathcal{T}$ for (the nodes of the) sub-tree rooted at j, so $\mathcal{T}(j) = \{j\} \cup \mathcal{D}(j)$. A state with no descendants is said to be a terminal state, so all states in the highest level L_M are terminal states. For simplicity of presentation we will assume that these are the only terminal states; the analysis easily extends to cases where intermediate levels L_m can also contain some terminal states.

Now consider a finite MDP with state space S and action space A. Assume we can construct a rooted tree $\mathcal{T}$ such that (i) the states in S correspond to the nodes of $\mathcal{T}$, and (ii) for every state $i \in S$ and action $a \in A$, the only possible transitions from state i under action a are either to its parent state $\rho(i)$ or to a state in the subtree $\mathcal{T}(i)$ rooted at i, with appropriate modifications for state 0 which has no parent and for terminal nodes which have only a parent and no descendants. We will say that such an MDP is *skip-free (in the negative direction) on the tree* $\mathcal{T}$.

Generalising the idea of a simple random walk (or simple birth and death process), Keilson (1979, p.28) defined a *tree process* to be a Markov process for which the states in S correspond to the nodes of a tree $\mathcal{T}$ in which states i and j were adjacent nodes if and only if $p_{ij} > 0$. These models have many applications; in particular, any reversible process for which there is a unique path between any two states is a tree process. We focus on models where the same tree structure is assumed to hold under all stationary deterministic policies; the skip-free formulation then extends the range of application by relaxing the restriction that 'upward' transitions to descendants are always to adjacent nodes.

As with the simpler models in Section 2, it is convenient define the upper tail probabilities

$$\bar{p}_{ij}(a) = P(X_{t+1} \in \mathcal{T}(j) | X_t = i, A_t = a),$$

corresponding to the probability that the next transition from state i under action a is to a state in the subtree rooted at j, and to assume that the model is specified in terms of the parameters

$$p_{i\rho(i)}(a),\ i \in S,\ a \in A; \qquad \bar{p}_{ij}(a),\ i \in S, j \in \mathcal{D}(i), a \in A$$

rather than the standard (and equivalent) representation in terms of the transition probabilities $p_{ij}(a)$, $i, j \in S$, $a \in A$.

4.4 Optimality equations

As in Section 2.1, the optimal average cost g^* for finite recurrent models, and the corresponding normalised relative costs $h_i^*, i \in S$, are the unique solutions to the optimality equations (1) which in this setting become

$$h_i = \min_{a \in A} \{ \ c_i(a) - g + \sum_{j \in \mathcal{D}(i)} p_{ij}(a) h_j + p_{ii}(a) h_i + p_{i\rho(i)}(a) h_{\rho(i)} \ \} \qquad i \in S \tag{11}$$

and an optimal stationary deterministic policy is given by d^*, where $d^*(i)$ is any action minimising the rhs of the equation corresponding to state i.

As in the integer lattice case, with appropriate modifications for the root node 0 and for terminal nodes, simple rearrangement shows that $c_i(a) - g + \sum_{j \in \mathcal{D}(i)} p_{ij}(a) h_j + p_{ii}(a) h_i + p_{i\rho(i)}(a) h_{\rho(i)} \geq h_i$ if and only if $c_i(a) - g + \sum_{j \in \mathcal{D}(i)} p_{ij}(a)(h_j - h_i) \geq p_{i\rho(i)}(a)(h_i - h_{\rho(i)})$, and that equality in one expression implies equality in the other.

Now for each $i \neq 0 \in S$ let $y_i = h_i - h_{\rho(i)}$. Then for each $j \in \mathcal{D}(i)$, there is a unique minimal path in the tree connecting i and j. Say the path passes through $s-1$ intermediate states and takes the form $i = r_0 \to r_1 \to \cdots \to r_s = j$. Let $\Delta(i, j)$ denote the states following i in the path to j, so $\Delta(i, j) = \{r_1, \ldots, r_s\}$. For each $k = 1, \ldots, s$, r_{k-1} is the parent of r_k so that $r_{k-1} = \rho(r_k)$, and hence $h_j - h_i = h_{r_s} - h_{r_0} = \sum_{k=1}^s h_{r_k} - h_{r_{k-1}} = \sum_{k=1}^s h_{r_k} - h_{\rho(r_k)} = \sum_{k=1}^s y_{r_k} = \sum_{r \in \Delta(i,j)} y_r$. However, if j is a descendant of i and $r \neq j$ is in the path connecting i and j, then r is a descendant of i and j is in the subtree rooted at r, and vice versa. Thus for fixed i and a we have that $\sum_{j \in \mathcal{D}(i)} p_{ij}(a)(h_j - h_i) = \sum_{j \in \mathcal{D}(i)} \sum_{r \in \Delta(i,j)} p_{ij}(a) y_r = \sum_{r \in \mathcal{D}(i)} \sum_{j \in \mathcal{T}(r)} p_{ij}(a) y_r = \sum_{r \in \mathcal{D}(i)} \bar{p}_{ir}(a) y_r$.

Taking account of the modifications for the root state $i = 0$ and the terminal states $i \in L_M$, and the fact that $i \in L_m \implies \mathcal{D}(i) \subset L_{m+1} \cup \cdots \cup L_M$, it follows that the optimality equations are equivalent to the equations

$$y_i = \min_a \{ \ (c_i(a) - x)/p_{i\rho(i)}(a) \ \} \qquad i \in L_M \tag{12a}$$

$$y_i = \min_a \{ \ (c_i(a) - x + \sum_{k \in \mathcal{D}(i)} \bar{p}_{ik}(a) y_k)/p_{i\rho(i)}(a) \ \} \qquad i \in L_{M-1}, \ldots, L_1 \tag{12b}$$

$$0 = \min_a \{ \ c_0(a) - x + \sum_{k \in S_0} \bar{p}_{0k}(a) y_k \ \} \tag{12c}$$

in that these equations also have unique solutions x and y_i, $i \in S$, with $x = g^*$ and $y_i = h_i^* - h_{\rho(i)}^*$, $i \in S$, and an optimal stationary deterministic policy is given by d^*, where $d^*(i)$ is any action minimising the rhs of the corresponding equation for y_i.

4.5 General algorithm

The results in Section2.2 translate directly into this multidimensional setting, giving the following generalised form of the PIA and a corresponding characterisation of its convergence properties. The proof of Theorem 10 below then exactly mirrors that of Theorem 8.

Policy Iteration Algorithm

1. Initialisation:

Choose an arbitrary initial policy d_0. Perform a single iteration of step 2 below, with $x = 0$ and with a_i restricted to the single value $d_0(i)$, $i \in S$. Compute the average cost g_0 under this policy by setting $g_0 = u_0$.

2. Iteration:

Set $x = g_n$.

• For $i \in L_M$ compute:

$$\begin{aligned} a_i &= \operatorname{argmin}_a\{\ (c_i(a) - x)/p_{i\rho(i)}(a)\ \} \\ y_i &= (c_i(a_i) - x)/p_{i\rho(i)}(a_i) \\ t_i &= 1/p_{i\rho(i)}(a_i) \end{aligned}$$

• For $i \in L_r$, $r = M-1, \ldots, 1$ compute:

$$\begin{aligned} a_i &= \operatorname{argmin}_a\{\ (c_i(a) - x + \textstyle\sum_{k\in\mathcal{D}(i)} \bar{p}_{ik}(a)y_k)/p_{i\rho(i)}(a)\ \} \\ y_i &= (c_i(a_i) - x + \textstyle\sum_{k\in\mathcal{D}(i)} \bar{p}_{ik}(a_i)y_k)/p_{i\rho(i)}(a_i) \\ t_i &= (1 + \textstyle\sum_{k\in\mathcal{D}(i)} \bar{p}_{ik}(a_i))/p_{i\rho(i)}(a_i) \end{aligned}$$

• For $j = 0$ compute:

$$\begin{aligned} a_0 &= \operatorname{argmin}_a \min_a\{\ (c_0(a) - x + \textstyle\sum_{k\in\mathcal{D}(0)} \bar{p}_{0k}(a)y_k)/(1 + \textstyle\sum_{k\in\mathcal{D}(0)} \bar{p}_{0k}(a_0)t_k)\ \} \\ u_0 &= (c_0(a_0) - x + \textstyle\sum_{k\in\mathcal{D}(0)} \bar{p}_{0k}(a_0)y_k)/(1 - p_{00}(a_0)) \\ t_0 &= (1 + \textstyle\sum_{k\in\mathcal{D}(0)} \bar{p}_{0k}(a_0)t_k)/(1 - p_{00}(a_0)) \end{aligned}$$

Set $d_{n+1}(i) = a_i$ for $i = 0, \ldots, M$ and set $g_{n+1} = g_n + u_0$.

3. Termination:

If $u_0 < 0$ then return to step 2.

If $u_0 = 0$ then stop and return d_{n+1} as an optimal policy, g_{n+1} as the optimal average cost, and $h_i = y_1 + \cdots + y_i,\ i = 1, \ldots M$ as the corresponding normalised relative costs. □

Theorem 10 *Consider the PIA above applied to a finite recurrent skip-free average cost MDP model on a tree. Then:*

(i) At each iteration of the PIA either $g_{n+1} < g_n$ and d_{n+1} is a strict improvement on d_n, or $g_{n+1} = g_n$ and d_{n+1} is an optimal average cost policy.

(ii) The PIA converges after a finite number of iterations.

Remarks

The PIA and the convergence results can be extended in the obvious way to models where intermediate levels L_m can also contain some terminal state i; in step 2 the PIA computes a_i using an equation of the form corresponding to a state in L_M (derived from (12a) rather than (12b). They also extend to continuous time, discounted cost and communicating average cost skip-free MDP models on trees, in a similar way to the extensions for the standard model in Section 3.

The extension to continuous time models is straightforward. For discounted cost models, the changes in Section 3.2 required the addition of a single extra state and appropriate changes to the transition probabilities. Since the state space there corresponded to a single linear branch, the extra state could be added to the previous terminal node without violating the requirements of the skip-free model. Skip-free MDP models on trees require the addition of an extra state for each terminal state (node) to preserve the skip-free property. This extra state now becomes the terminal node in that branch. Transitions from this extra state are to the corresponding previous terminal node, with probability β, or back to itself, with probability $1-\beta$. Transition probabilities from non-terminal states are modified as in Section 3.2, by setting $p'_{ij}(a) = \beta p_{ij}(a)$ if j is a non-terminal node of the modified tree and by assigning the remaining transition probability $1 - \beta$ to the newly added terminal nodes of the modified sub-tree $\mathcal{T}(i)$ rooted at i. The precise assignment may be chosen arbitrarily – for example, each new terminal node in the modified sub-tree may be chosen with equal probability – as long as the total probability sums to $1 - \beta$.

For communicating models, the idea again is that for each state i the PIA is modified so that in passing it solves the corresponding sub-problem Π_i with state space $\mathcal{T}(i)$ and with state i as the distinguished state, and then computes the optimal updated average cost and policy by minimising over the costs and policies for each of the sub-problems.

The algorithm also leads to a similar alternative formulation for the constraints in the LP method of solution. The constraints for the primal LP follows directly from the optimality equations (12). For the dual LP, for each i let $\Gamma(i)$ denote the set of states $j \neq i$ in the unique minimal path in the tree connecting 0 and i, so if the path takes the form $0 = r_0 \to r_1 \to \cdots \to r_{s-2} \to r_{s-1} = \rho(i) \to r_s = i$, then $\Gamma(i) = \{0, r_1, r_2, \ldots, r_{s-2}, \rho(i)\}$. Then the constraints for the dual LP follow from a similar intuitive argument to that in Section 3.4 but with the set balance equations (10) now taking the form

$$\xi_i p_{i\rho(i)} = \sum_{j\in\Gamma(i)} \sum_{k\in\mathcal{T}(i)} \xi_j \bar{p}_{jk} \qquad\qquad i \neq 0,\ i \in S.$$

References

Derman, C. (1970), *Finite State Markovian Decision Processes*, Academic Press, New York.

He, Q.-M. (2000), 'Quasi-birth-and-death Markov processes with a tree structure and the $MMAP[K]/PH[K]/N/LCFS$ non-preemptive queue', *European Journal of Operational Research* **120**, 641–656.

Keilson, J. (1965), *Green's Function Methods in Probability Theory*, Griffin, London.

Keilson, J. (1979), *Markov Chain Models – Rarity and Exponentiality*, Springer-Verlag, New York.

Lippman, S. (1975), 'Applying a new device in the optimization of exponential queuing systems', *Operations Research* **23**, 687–709.

Low, D. W. (1974), 'Optimal dynamic pricing policies for an $M/M/s$ queue', *Operations Research* **22**, 545–561.

Miller, B. L. (1981), 'Countable-state average-cost regenerative stopping problems', *Journal of Applied Probability* **18**, 361–377.

Puterman, M. L. (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, New York.

Reid, J. K. (1982), 'A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases', *Mathematical Programming* **24**, 55–69.

Ross, S. M. (1970), *Applied Probability Models with Optimization Applications*, Dover Publications, Inc.

Serfozo, R. (1981), 'Optimal control of random walks, birth and death processes, and queues', *Advances in Applied Probability* **13**, 61–83.

Stidham, Jr., S. & Weber, R. R. (1989), 'Monotonic and insensitive optimal policies for control of queues with undiscounted costs', *Operations Research* **87**, 611–625.

Stidham, Jr., S. & Weber, R. R. (1999), Monotone optimal policies for left-skip-free Markov decision processes, *in* J. Shanthikumar & U. Sumita, eds, 'Contributions in Applied Probability and Stochastic Processes (in honor of Julian Keilson)', Kluwer Academic Publishers, Boston, pp. 191–202.

Thomas, L. C. (1982), 'The Wijngaard-Stidham bisection method and replacement models', *IEEE Transactions on Reliability* **R-31**, 482–484.

Wijngaard, J. & Stidham, Jr., S. (1986), 'Forward recursion for Markov decision processes with skip-free-to-the-right transitions, Part I: Theory and algorithms', *Mathematics of Operations Research* **11**, 295–208.

Wijngaard, J. & Stidham, Jr., S. (2000), 'Forward recursion for Markov decision processes with skip-free-to-the-right transitions, Part II: Non-standard applications', *Statistica Neerlandica* **54**, 160–174.

Yeung, R. W. & Sengupta, B. (1994), 'Matrix product-form solutions for Markov chains with tree structure', *Advances in Applied Probability* **26**, 965–987.