MATHEMATICS OF OPERATIONS RESEARCH
Vol. 32, No. 3, August 2007, pp. 589–593
ISSN 0364-765X | EISSN 1526-5471 | 07 | 3203 | 0589

DOI 10.1287/moor.1070.0256

Stochastic Search in a Forest Revisited

Jay Sethuraman
IEOR Department, Columbia University, New York, New York 10027,
jay@ieor.columbia.edu, http://www.columbia.edu/~js1353

John N. Tsitsiklis
EECS Department, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139,
jnt@mit.edu, http://web.mit.edu/jnt/www/home.html

We consider a generalization of the model of stochastic search in an out-forest, introduced and studied by E. V. Denardo, U. G. Rothblum, L. Van der Heyden. 2004. Index policies for stochastic search in a forest with an application to R&D project management. *Math. Oper. Res.* **29**(1) 162–181. We provide a simpler proof of the optimality of index-based policies.

Key words: stochastic search; multiarmed bandit; Gittins index; project management; stochastic scheduling; dynamic programming
MSC2000 subject classification: Primary: 90B40, 90B36; secondary: 90B15
OR/MS subject classification: Primary: project management, dynamic programming, scheduling
History: Received November 12, 2005; revised October 23, 2006, and January 4, 2007.

1. Introduction. Motivated by the issue of investing in a research and development project, Denardo et al. [1] introduced and studied a stochastic search problem in an out-forest, which we will be referring to as the DRV model. In particular, they established the optimality of "index" policies for either linear or exponential utility functions.

While their main result is simple and is reminiscent of similar results on multiarmed bandit problems, their proof is not. In fact, the authors note that standard lines of analysis in the bandit literature do not seem to yield their results. In this paper, we show that a short proof is possible for a suitable *generalization* of the DRV model, using the approach of Tsitsiklis [2] for the classical multiarmed bandit problem. The reason for introducing a more general model is precisely that it enables the simpler proof.

We remark that both the proof in Denardo et al. [1] and our proof are inductive in nature. The fundamental distinction between the two approaches is that the proof in Denardo et al. [1] proceeds by iteratively eliminating leaf edges, whereas the proof we present iteratively identifies and eliminates a "highest priority" edge. Our approach results in an exponential growth in problem size, so a direct implementation of the algorithm we propose is not efficient (although, for the special case of the DRV model, this can be remedied—see the end of §4). In contrast, Denardo et al. design an algorithm for their model that finds an optimal policy in $O(m^2)$ time when the given out-forest has m edges.

The rest of the paper is organized as follows. Section 2 presents the model and the problem formulation; §3 provides the proof of the main result; §4 discusses indexability and computational issues; and §5 deals with various extensions.

2. The model. Let $G = (N, E)$ be an out-forest, that is, a directed acyclic graph in which every node has an in-degree of zero or one. For each edge $e \in E$, $A(e)$ denotes the (possibly empty) set of *immediate successors* of edge e. More precisely, if $e = (i, j)$, then $A(e)$ contains all edges of the form (j, k) for some k. We say that an edge e' is a *successor* of e (and that e is a predecessor of e') if there is a sequence of edges $e = e_1, e_2, \ldots, e_k = e'$ such that each e_{i+1} is a successor of e_i. An edge e is called a *leaf edge* if $A(e)$ is empty.

The search model takes the form of a sequential decision process, whereby at each stage an edge is selected and attempted. To define the model, we describe the state of the process, the set of available actions, the transition mechanism, and the associated rewards.

The *states* of the sequential decision process are subsets S of E, with the property that none of the edges in S is a predecessor of another edge in S; the edges in S are said to be *available*. In addition, there is a special termination state, denoted by T. If the state S is the empty set, the process moves to the termination state at the next step. If a nonempty and nonterminal state ($S \neq \varnothing, T$) is reached, the decision maker must select and attempt an edge $e \in S$, resulting in a random immediate reward, whose expected value is R_e. (Note that R_e is allowed to be negative.) If edge e is attempted, the process either moves to the terminal state T (with probability π_e) or it "succeeds" and a random set of immediate successors of e becomes "available;" for each set X of immediate successors of e, we use $p_e(X)$ to denote the probability that the set X is generated. In particular,

$$\pi_e + \sum_{X \subseteq A(e)} p_e(X) = 1.$$

Formally, given a current state $S \neq \varnothing, T$, and given that e was attempted, the probability $P(S' \mid S, e)$ of transitioning to a next state S' is given by

$$P(S' \mid S, e) = \begin{cases} \pi_e, & \text{if } S' = T, \\ p_e(X), & \text{if } X \subseteq A(e) \text{ and } S' = (S \cup X)\backslash\{e\}. \end{cases}$$

The outcomes at different edges are mutually independent events. (However, the immediate random reward is allowed to be dependent on the set of immediate successors that become available.) The goal of the decision maker is to maximize the expected total reward earned until termination. (Note that voluntary termination on the part of the decision maker can be modeled by adding an independent edge e with $R_e = 0$, $\pi_e = 1$.)

The initial state of this decision process consists of all the edges of E that have no predecessors. It is straightforward to verify that the decision maker reaches state S (with $S \neq \varnothing, T$) only if he has *successfully* attempted all predecessors of the edges in S, but has not attempted any edge in S.

The decision maker is allowed to use general, possibly randomized, history-dependent policies. However, standard results from the theory of Markov decision processes imply the existence of an optimal policy within the class of deterministic and stationary policies, to be referred to as DS policies. (That is, the decision at each stage is just a function of the current state.) Formally, a DS policy is a function ψ from the state space into E, where each nonempty and nonterminal state S is mapped into an edge $\psi(S) \in S$. A DS policy ψ is *optimal for state* S if it maximizes the total expected reward when S is the initial state. A policy is called *optimal* if it is optimal *jointly* for all states. A policy ψ is called a *priority* policy if there is an ordering on the edges such that for each state S, $\psi(S)$ is the first edge in S according to this ordering. Note that priority policies are automatically DS policies. The main result in this paper establishes the existence of an optimal priority policy.

2.1. Relation to the DRV model. The model we introduced is a generalization of the DRV model considered by Denardo et al. [1]. More specifically, the *basic* DRV model is the following special case:

(a) For every edge e, $p_e(X) = 0$, if $X \neq \varnothing, A(e)$;
(b) For all edges, $R_e = \pi_e r_e - c_e$, where $c_e > 0$; and
(c) For all nonleaf edges, $\pi_e = 0$, and so r_e is irrelevant for these edges.

For the basic model, they prove the existence of an optimal priority policy and show how it can be computed in $O(m^2)$ time, when the out-forest has m edges. They exploit the special cost structure to show that it is optimal to conduct a series of "depth first" searches of paths to leaf edges. Finally, they observe that the structural results (including the existence of an optimal priority policy) hold for the more general probabilistic model in which condition (a) above is dropped.

The motivation for the basic DRV model is as follows: Consider a research and development project, whose various activities are represented by the edges of the out-forest. Each activity can be attempted at a certain cost, and the outcome is a success with a certain probability. If an activity is attempted successfully, all of its immediate successors become eligible to be attempted. The overall project succeeds—and yields a reward—if a leaf edge (an edge with no successors) is attempted successfully. It is evident that for the overall project to succeed, there must be a path from a stem edge (an edge with no predecessors) to a leaf, all of whose edges are attempted successfully. The objective is to find an investment strategy that maximizes expected utility. For instance, the various leaf edges may represent different technologies for accomplishing a certain outcome, and the stem-leaf paths may represent the sequence of tasks that must be undertaken successfully to build the corresponding technology.

3. Main result. The proof technique in Tsitsiklis [2] essentially involves two steps: First, identify by inspection or by some elementary computation a highest-priority edge; and second, find a reduced problem in which this edge is eliminated. The first step identifies an edge e^* with the property that whenever edge e^* is *available*, there is at least one optimal policy that attempts it. Once such an edge e^* is identified, the second step eliminates that edge using the following reasoning: If attempting edge e causes e^* to become available, the decision maker will necessarily attempt e^* next; therefore, one can contract edge e^* and update the parameters associated with edge e in a manner that captures this two-step attempt. Note that because the given graph is an out-forest, there is *at most* one edge e that can cause e^* to become available, and so e^* can be safely eliminated. The resulting problem has one fewer edge, to which the same argument can be inductively applied. Given the graph G and the coefficients R_e, π_e, we define

$$\gamma_G(e) = \begin{cases} R_e/\pi_e, & \text{if } \pi_e \neq 0, \\ -\infty, & \text{if } \pi_e = 0 \text{ and } R_e < 0, \\ +\infty, & \text{if } \pi_e = 0 \text{ and } R_e \geq 0. \end{cases}$$

Let

$$e^* \in \arg\max_{e\in E} \gamma_G(e),$$

and $\gamma^*(e^*) = \gamma_G(e^*)$.

Lemma 3.1. *There is an optimal DS policy ψ, with $\psi(S) = e^*$ for every state S that contains e^*.*

Proof. For any DS policy ψ, we say that a state S is *exceptional* if $e^* \in S$, but $\psi(S) \neq e^*$. In this terminology, the lemma asserts the existence of an optimal DS policy that has *no* exceptional states.

Let ψ^* be an optimal DS policy with the smallest number of exceptional states. If ψ^* has no exceptional states, we are done. Suppose ψ^* has at least one exceptional state. Then, there exists an exceptional state S^* such that: (i) ψ^* attempts $e \in S^*$, with $e \neq e^*$; and (ii) if the attempt at e does not result in termination, ψ^* attempts e^*. To see this, consider the *state-transition graph*, whose nodes are the exceptional states of ψ^*, and whose edges are the pairs (S, S') of exceptional states such that if ψ^* reaches S, then it is possible to reach S' in the next step. This is a directed acyclic graph, so it must have a node S^* with out-degree zero. Such a state S^* has the properties claimed above.

Now consider the following alternative policy ψ. If the initial state is not S^*, then ψ selects the same action as ψ^* at every time step. If, on the other hand, the initial state is S^*, then: (i) ψ attempts e^* at the first time step; (ii) if the attempt at e^* does not result in termination, ψ attempts e in the following step; (iii) ψ agrees with ψ^* after the first two steps. (Note that ψ is not, in general, a DS policy; it is history dependent, because the action at the second time step may be affected by the initial state.) As a result of this "local interchange," the increase in the expected total reward when S^* is the initial state is

$$R_{e^*} + (1-\pi_{e^*})R_e - R_e - (1-\pi_e)R_{e^*}. \tag{1}$$

Because ψ^* is an optimal policy, this expression is less than or equal to zero; by the definition of e^*, however, this expression is nonnegative. Thus, the net change in expected total reward as a result of this interchange is zero. It follows that the action of attempting e^* at state S^* is an optimal action. Therefore, the DS policy $\widehat{\psi}$ that satisfies $\widehat{\psi}(S^*) = e^*$, and $\widehat{\psi}(S) = \psi^*(S)$ for $S \neq S^*$, is also optimal. We have thus constructed a new DS policy $\widehat{\psi}$, which is optimal and has one fewer exceptional state, contradicting the definition of ψ^*. Therefore, ψ^* must be an optimal policy with no exceptional states. □

We are now ready for the main result.

Theorem 3.1. *There is an optimal policy that is a priority policy.*

Proof. Our proof is by induction on the number of edges in the given out-forest. If the given out-forest has only one edge, the result is trivially true. Suppose the theorem holds for all out-forests with fewer than m edges. Let us now consider an out-forest G with m edges, and let e^* be an edge for which $\gamma_G(e)$ is largest. Let $\Psi(e^*)$ be the class of DS policies that attempt e^* whenever it is available. By Lemma 3.1, there is an optimal policy within the class $\Psi(e^*)$. We argue next that the problem of finding an optimal policy within the class $\Psi(e^*)$ can itself be formulated as a search problem in an out-forest involving the remaining $m-1$ edges.

To define this reduced problem, we consider two possibilities, depending on whether or not e^* has an immediate predecessor. If e^* has no predecessor, then the reduced problem is equivalent to the out-forest $(N, E\backslash\{e^*\})$. In that case, the following priority policy is optimal: First attempt e^*, and then (in the absence of termination) follow an optimal policy for the out-forest $(N, E\backslash\{e^*\})$; the latter policy can be taken to be a priority policy by the induction hypothesis.

Suppose now that e is the immediate predecessor of edge e^*. If a policy in $\Psi(e^*)$ attempts e, the process does not terminate, and e^* becomes available; it will then immediately attempt e^*. By viewing this "automatic" attempt of e^* as part of a single composite step initiated with the attempt of e, we obtain a reduced but equivalent model, in which e^* is eliminated, and which involves an out-forest with $m-1$ edges. We now specify the details of the reduced model.

The new set of edges is simply $\bar{E} = E\backslash\{e^*\}$. The new termination probability $\bar{\pi}_e$ when attempting e needs to include the probability that e^* becomes available and its attempt results in termination. Thus,

$$\bar{\pi}_e = \pi_e + q_{e^*}\pi_{e^*}, \tag{2}$$

where

$$q_{e^*} = \sum_{X\subseteq A(e):\, e^*\in X} p_e(X)$$

is the probability that when e is attempted, the attempt is successful and edge e^* becomes available. Similarly, the new expected reward $\bar{R}_e$ needs to include the expected reward from the possible subsequent attempt of e^*:

$$\bar{R}_e = R_e + q_{e^*} R_{e^*}. \tag{3}$$

The set $\bar{A}(e)$ of immediate successors of e in the reduced model will be the set of all edges that may become available once the composite step is carried out. Thus,

$$\bar{A}(e) = \big(A(e)\backslash\{e^*\}\big) \cup A(e^*).$$

It remains to specify the probabilities with which different subsets of $\bar{A}(e)$ become available. Consider a typical subset of $\bar{A}(e)$, of the form $X \cup Y$, where $X \subseteq A(e)\backslash\{e^*\}$ and $Y \subseteq A(e^*)$. The probability that the set of newly available edges at the end of the composite step equals $X \cup Y$ is given by

$$\bar{p}_e(X \cup Y) = \begin{cases} p_e(X) + p_e(X \cup \{e^*\}) p_{e^*}(\varnothing), & \text{if } Y = \varnothing, \\ p_e(X \cup \{e^*\}) p_{e^*}(Y), & \text{if } Y \neq \varnothing. \end{cases}$$

For the case where $Y = \varnothing$, the two terms in the formula above correspond to the cases where e^* did or did not become available when e was attempted. Note that the mutual independence of the sets generated when attempting different edges in the reduced problem follows from the corresponding assumption for the original problem.

There is a one-to-one correspondence between policies in Ψ^* and DS policies for the reduced problem. Furthermore, because of the definition of the reduced problem, corresponding policies have the same expected total reward. Consider an optimal policy for the reduced problem that is a priority policy. (Such a policy exists because the reduced problem corresponds to an out-forest with $m-1$ edges, and the induction hypothesis applies.) This priority policy on the reduced problem, together with giving top priority to e^*, defines a priority policy for the original problem that is optimal. □

4. Indices and computation.

The proof of Theorem 3.1 suggests a recursive algorithm for determining an optimal priority policy by a repeated application of the following two steps: (i) identifying a highest-priority edge e^* from the problem data (ties can be broken arbitrarily, e.g., lexicographically); and (ii) deleting e^*, and updating the coefficients R_e and π_e as in Equations (2)–(3), to obtain a smaller problem.

We make some observations on the structure of the algorithm.

(a) With the above algorithm, every edge will eventually be eliminated. Let e_k be the kth edge to be eliminated by the algorithm. We define the *index*, $\gamma^*(e_k)$, of edge e_k to be the value of R_{e_k}/π_{e_k} computed by the algorithm at the beginning of the kth iteration, that is, the iteration at which edge e_k is eliminated. From Equations (2)–(3), and the fact that R_{e^*}/π_{e^*} is maximal, we see that $\bar{R}_e/\bar{\pi}_e \le R_{e^*}/\pi_{e^*}$, where we use the same conventions as in the definition of $\gamma_G(e)$, when a denominator is zero. It follows that indices are generated in nonincreasing order, that is, $\gamma^*(e_{k+1}) \le \gamma^*(e_k)$ for every k. In particular, edges with a higher index value get higher priority.

(b) A further property, which is apparent from the structure of the reduction, is that $\gamma^*(e)$ is completely determined by the data associated with e and its successors in the original out-forest. In particular, if the forest consists of several independent trees, the index computation can be carried out separately at each tree. This is in the spirit of indexability results for classical multiarmed bandit problems, where the index of a state of a particular bandit can be calculated independent of the data associated with the other bandits.

The algorithm above will in general run in exponential time because of the multiplicative increase in the number of positive probability subsets to be considered, and we suspect that this is unavoidable. For an example, consider a tree consisting of a path $e_k, e_{k-1}, \ldots, e_1$, together with additional leaf edges $e'_{k-1}, \ldots, e'_1$, arranged so that each edge e_{i+1} has an immediate successor e_i that belongs to the path, and another immediate successor e'_i that is a leaf edge. Suppose that $p_{e_{i+1}}(\{e_i\}) > 0$ and $p_{e_{i+1}}(\{e_i, e'_i\}) > 0$. Suppose furthermore that the edges $e_1, \ldots, e_{k-1}$ are eliminated first. In the reduced graph, after the first $k-1$ iterations, all of the edges e'_i will be immediate successors of e_k, and every subset of this set of successors will have positive probability. An example of such an exponential increase is possible even for the DRV model.

In some cases, an efficient algorithm becomes possible by bypassing the computation of the probabilities $p_e(X)$ for the reduced problems. We only need to be able to efficiently compute R_e and π_e for a reduced problem. Indeed, for a reduced problem in which edges $e_1, \ldots, e_k$ have been eliminated, R_e is the expected cost of a policy for the original problem that starts with edge e, continues by choosing each time the highest-priority available edge within the set $\{e_1, \ldots, e_k\}$, and terminates voluntarily once no such edge is available (even if other edges are available). It turns out that this interpretation leads to an efficient algorithm for computing R_e (and similarly, π_e) for the basic DRV model. We do not provide any further details because this approach essentially recovers the efficient algorithm given in Denardo et al. [1].

5. Extensions.

We end by noting that the same approach applies to the variants of the basic model described in Denardo et al. [1, §5, p. 171]. We briefly discuss the necessary changes for the cases of risk-averse and risk-seeking utility functions; the other variants can be handled in a straightforward way.

Consider the case of a risk-seeking utility function, where the utility of a reward x is $e^{\lambda x}$, and λ is a positive constant. We denote by R_e the expected utility resulting from a single attempt at edge e. We note that $R_e > 0$, and that utility maximization is equivalent to maximizing the expected value of the product of the single-step utilities R_e of the attempted edges. Voluntary termination is modeled by an independent edge e with $\pi_e = 1$ and $R_e = 1$. Lemma 3.1 and Theorem 3.1 are valid with the following modifications. We define $\gamma_G(e)$ by

$$\gamma_G(e) = \begin{cases} \dfrac{\pi_e R_e}{1-(1-\pi_e)R_e}, & \text{if } 1-(1-\pi_e)R_e \neq 0, \\ -\infty, & \text{if } 1-(1-\pi_e)R_e = 0. \end{cases}$$

Let E^- be the set of edges e with $1-(1-\pi_e)R_e \leq 0$. If $E^- \neq \varnothing$, we let

$$e^* \in \arg\max_{e\in E^-} \gamma_G(e),$$

otherwise,

$$e^* \in \arg\max_{e\in E} \gamma_G(e).$$

With this choice of e^*, Lemma 3.1 remains valid. The only change in the proof is that expression (1) now becomes

$$\pi_{e^*}R_{e^*} + (1-\pi_{e^*})\pi_e R_{e^*} R_e - \pi_e R_e - (1-\pi_e)\pi_{e^*} R_{e^*} R_e. \tag{4}$$

With our definition of e^*, the above expression is guaranteed to be nonnegative. Theorem 3.1 and its proof remain valid, with $\bar{\pi}$ as before and with

$$\bar{R}_e = (1-q_{e^*})R_e + q_{e^*} R_e R_e^*.$$

Consider now the case of a risk-averse utility function, where the utility of a reward x is $-e^{-\lambda x}$, and λ is a positive constant. We define R_e to be the negative of the utility resulting from a single attempt at edge e. We note that $R_e > 0$ and that utility maximization is equivalent to *minimizing* the expected value of the product of the single-step disutilities R_e of the attempted edges. The definition of the $\gamma_G(e)$ and the rest of the argument is the same as in the risk-seeking case. The only change is that we now define E^+ as the set of edges e with $1-(1-\pi_e)R_e > 0$. If $E^+ \neq \varnothing$, we let

$$e^* \in \arg\min_{e\in E^+} \gamma_G(e),$$

otherwise,

$$e^* \in \arg\min_{e\in E} \gamma_G(e).$$

With this definition, the expression (4) is guaranteed to be nonpositive.

Acknowledgments. The authors are grateful to Uri Rothblum for many insightful discussions about the problem. In particular, he pointed out a flaw in an earlier proof, and suggested the "probabilistic" model discussed here. The authors thank Eric Denardo and Uri Rothblum for detailed comments and suggestions on an earlier version of this paper. This research was supported by the National Science Foundation under grants DMI-0093981, ECS-0312921, and ECS-0426453, and by an IBM Faculty award.

References

[1] Denardo, E. V., U. G. Rothblum, L. Van der Heyden. 2004. Index policies for stochastic search in a forest with an application to R&D project management. *Math. Oper. Res.* **29**(1) 162–181.

[2] Tsitsiklis, J. N. 1994. A short proof of the Gittins index theorem. *Ann. Appl. Probab.* **4**(1) 194–199.