ON THE SAMPLE COMPLEXITY OF FINDING GOOD SEARCH STRATEGIES

Pekka Orponen*
University of Helsinki
Department of Computer Science
SF-00510 Helsinki, Finland

Russell Greiner†
University of Toronto
Department of Computer Science
Toronto, Ont. M5S 1A4, Canada

ABSTRACT

A satisficing search problem consists of a set of probabilistic experiments to be performed in some order, without repetitions, until a satisfying configuration of successes and failures has been reached. The cost of performing the experiments depends on the order chosen. Earlier work has concentrated on finding optimal search strategies in special cases of this model, such as search trees and and-or graphs, when the cost function and the success probabilities for the experiments are given. In contrast, we study the complexity of "learning" an approximately optimal search strategy when some of the success probabilities are not known at the outset. Working in the fully general model, we show that if n is the number of unknown probabilities, and C is the maximum cost of performing all the experiments, then

$$2(\frac{nC}{\epsilon})^2 \ln \frac{2n}{\delta}$$

trials of each undetermined experiment are sufficient to identify, with confidence $1-\delta$, a search strategy whose cost is within ϵ of the optimal.

INTRODUCTION

Consider the following situation (from [3]): A sequence of tests are to be performed to decide whether a product specimen is satisfactory. There is a cost associated with each test, and the tests are related by precedence constraints so that certain tests can only be performed after the results of others are known. Let us assume for the moment that the probabilities of a specimen passing each test are independent of each other and known at the outset. It is desired to find a testing sequence that obeys the precedence constraints and minimizes the expected cost before a decision to either accept or reject the specimen can be made.

This is a simple example of a *satisficing search problem* (term due to Simon and Kadane [6]), where the goal is to find a single satisfactory configuration of events (in this case, an informative combination of test results) at minimum expected cost. Other examples of such problems include, e.g, screening employment candidates for a position [3], competing for prizes at a quiz show [3], mining for gold buried in Spanish treasure chests [6], and performing inference in simple expert systems [7]. Garey gave in [3] an efficient algorithm for finding optimal search strategies (in the

*Work supported by the Academy of Finland. E-mail: orponen@cs.helsinki.fi.
†Work supported by an Operating Grant from the National Science and Engineering Research Council of Canada. E-mail: greiner@ai.toronto.edu.

sense of minimal expected cost) for satisficing search problems when the precedence constraints can be represented as a tree. The more general problem of search in directed acyclic graphs was later studied by Simon and Kadane [6]. They, however, did not explicitly suggest any efficient algorithm for finding optimal search strategies, and in fact the problem of finding optimal strategies in this case is NP-hard [4]. Also satisficing searches in and-or graphs would be of interest; unfortunately, here again it can be shown that finding optimal strategies is NP-hard, even when all the success probabilities are 1 [8]. Still, partial results can be obtained for restricted classes of strategies: for instance, Natarajan has presented an efficient algorithm for finding optimal *depth-first* search strategies for and-or trees [5].

All these algorithms take as input a description of the search graph, together with a listing of the costs of the experiments and their success probabilities. In most real-life situations, of course, the actual success probabilities of the experiments are not known and must be estimated, typically by sampling. A natural question then arises: just how many samples are needed to guarantee, with a high level of confidence, that a strategy based on the resulting estimates really is close to optimal? A moment's reflection on any of the proposed algorithms shows that they are very sensitive to estimation errors: small changes in the success probabilities may lead to drastically different strategies.

However, as we shall show below, even though the structure of the optimal strategy is very sensitive to the actual values of the probabilities, its *cost* is not. Given a search graph G with n unknown experiment success probabilities, we can think of the optimal search cost of G as a function of the unknown probabilities. We show below how to bound the partial derivatives of this function, leading to the following result: for any values of parameters $\epsilon, \delta > 0$, if at least

$$N(G;\epsilon,\delta) = 2(\frac{nC}{\epsilon})^2 \ln \frac{2n}{\delta}$$

trials of each undetermined experiment are performed, where C is the maximum cost of performing all the experiments, then the probability estimates obtained are good enough to guarantee, with probability $1-\delta$, that the cost of an estimated optimal strategy differs from the cost of the actual optimal strategy by at most an additive term of ϵ. In terms of pac-learnability theory [9], we can say that $N(G;\epsilon,\delta)$ samples of each experiment are sufficient to "learn" a good search strategy.

A very special case of our problem was studied by Barnett in [1]; namely the robustness of the Garey–Simon–Kadane algorithm against errors in probability estimates when there are two independent experiments (and, consequently, only two alternative search strategies).

BASIC NOTIONS: SEARCH STRUCTURES AND STRATEGIES

A *satisficing search structure* is a four-tuple $I = \langle E, p, R, c\rangle$, where

- $E = \langle e_1, \ldots, e_n\rangle$ is a set of *experiments*;
- $p = \langle p_1, \ldots, p_n\rangle$ is a vector of mutually independent *success probabilities* for the experiments;
- R is a *result* function mapping subsets of $E \times \{+,-\}$, such that for no experiment e, both $\langle e,+\rangle$ and $\langle e,-\rangle$ appear in the set, to the set $\{\mathcal{S},\mathcal{F},\mathcal{U}\}$ (for Success, Failure, Undecided);
- c is a *cost* function mapping ordered subsets of E (i.e., finite sequences without repetition) to nonnegative reals, so that for all sequences σ and experiments e, where σ does not contain e, $c(\sigma \cdot e) \geq c(\sigma)$.

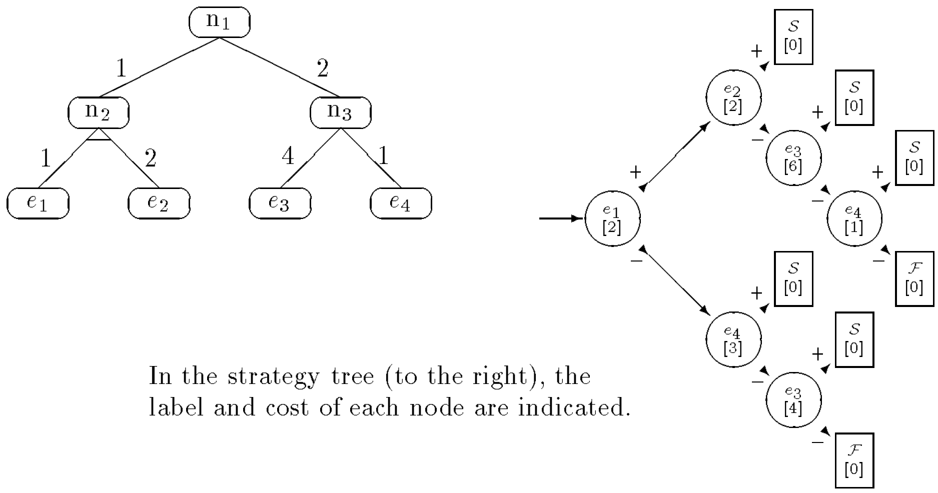

Figure 1: An And-Or Tree and an Associated Strategy Tree

The notion of a satisficing search structure generalizes that of a weighted and-or search graph. As an illustration, consider the and-or tree on the left in Figure 1. We can represent this as a search structure by having just four experiments $e_1, \ldots, e_4$, corresponding to the leaves of the tree, and letting $p_1, \ldots, p_4$ be the success probabilities of the respective search subgoals. (In general, we could model also the possibility of a search terminating at an internal vertex, cf. [3, 6, 7].) Some representative values of the appropriate result and cost functions in this case are $R(\{\langle e_1, +\rangle\}) = \mathcal{U}$, $R(\{\langle e_1, +\rangle, \langle e_2, +\rangle\}) = \mathcal{S}$, $R(\{\langle e_1, -\rangle, \langle e_3, -\rangle, \langle e_4, -\rangle\}) = \mathcal{F}$; and $c(\langle e_1\rangle) = 2$, $c(\langle e_1, e_2\rangle) = 4$, $c(\langle e_1, e_4, e_3\rangle) = 9$.

A *strategy* for a search structure $I = \langle E, p, R, c\rangle$ is a binary tree s whose internal nodes are labelled by experiments in E, and terminal nodes by $\mathcal{S}$ or $\mathcal{F}$. The label of a node u is denoted by $l(u)$, and the two immediate descendants of an internal node u are denoted by u^+ and u^-. No label $e \in E$ is repeated on any path $\pi = \langle u_1, \ldots, u_k, t\rangle$ from the root to a leaf t in s, and the label of the terminal node is $l(t) = R(L^\pi)$, where $\langle e, +\rangle \in L^\pi$ ($\langle e, -\rangle \in L^\pi$) if and only if for some node $u \in \pi$, $l(u) = e$ and $u^+ \in \pi$ (resp. $u^- \in \pi$). One possible strategy for our example and-or tree is indicated on the right in Figure 1.

Based on the cost function of I, we can associate a cost to each node u in s as follows. Let us first define the cost of a path $\pi = \langle u_1, \ldots, u_k\rangle$ in s as $c(\pi) = c(\langle l(u_1), \ldots, l(u_k)\rangle)$ if u_k is not a leaf, and $c(\pi) = c(\langle l(u_1), \ldots, l(u_{k-1})\rangle)$ otherwise; as a special case, $c(\langle\rangle) = 0$. Let $\pi = \langle u_1, \ldots, u_k\rangle$ be the unique path from the root of s to the predecessor u_k of u, and define $c(u) = c(\pi \cdot u) - c(\pi)$. The *probability* of a path $\pi = \langle u_1, \ldots, u_k\rangle$ is defined as

$$p(\pi) = \prod_{i=1}^{k-1} p^{\pm}_{u_i}, \qquad \text{where } p^{\pm}_{u_i} = \begin{cases} p_j, & \text{if } l(u_i) = e_j \text{ and } u_{i+1} = u_i^+, \\ 1 - p_j & \text{if } l(u_i) = e_j \text{ and } u_{i+1} = u_i^-. \end{cases}$$

The (expected) *cost* of a strategy s is defined as $c(s) = \sum_\pi p(\pi)c(\pi)$, where the sum is over all complete paths from the root to a leaf in s.

Given a class of strategies $\mathcal{S}$ and a search structure I, a strategy $s \in \mathcal{S}$ is *optimal for I within* $\mathcal{S}$ if s is a strategy for I, and for any other $s' \in \mathcal{S}$ that is a strategy for I, $c(s') \geq c(s)$.

Since all the structures we are dealing with are finite, optimal strategies can always be found by exhaustive search (assuming that the class $\mathcal{S}$ is decidable). Of course, exhaustive search is in general impractical, and if we are dealing with search structures with concise encodings, such as and-or graphs, it is not even guaranteed that the optimal strategies will have useful polynomial-size representations. As pointed out in the introduction, however, in many interesting special cases optimal strategies have simple representations and can be determined in polynomial time.

For the remainder of this paper, we assume that some class of search structures $\mathcal{I}$ and a class of strategies $\mathcal{S}$ of interest have been fixed. Moreover, we assume the existence of an algorithm $s^*(I)$ that is capable of mapping any structure $I \in \mathcal{I}$ to an optimal strategy for I within $\mathcal{S}$.

THE COMPLEXITY OF FINDING A GOOD STRATEGY

We consider the situation where repeated searches are to be performed over the same structure I, but some of the success probabilities p_i are not known at the outset. By a slight abuse of notation, we let I denote also the corresponding partially specified search structure, and let $I(p)$ denote this structure completed with a probability vector p. The cost of any strategy s for the partially specified I will now be determined only relative to the missing probabilities; we denote by $c(s; p)$ the cost of a strategy s when the missing values are given by a vector p.

Our question of interest is: given a partially specified search structure I, and constants ϵ, δ, how much sampling effort is needed to obtain a strategy for I that with confidence $1 - \delta$ has a cost within an error of ϵ from optimal? We obtain the following bound:

Theorem 1 *Let $I = I(r) = \langle E, p^0 \cdot r, R, c\rangle$ be a partially specified search structure, where the vector p^0 contains the known probability values, and the parameters $r = \langle r_1, \ldots, r_n\rangle$ stand for the unknown probabilities. Assume that, in a particular situation, the correct values for the parameters r are given by $p = \langle p_1, \ldots, p_n\rangle$. Let $\epsilon, \delta > 0$ be given constants. Denoting $C = \max\{c(\pi) | \pi$ is a permutation of $E\}$, let $\hat{p} = \langle \hat{p}_1, \ldots, \hat{p}_n\rangle$ be an estimate for p, where each $\hat{p}_i$ is obtained as the relative frequency of successes in*

$$N \geq N(I; \epsilon, \delta) = 2(\frac{nC}{\epsilon})^2 \ln \frac{2n}{\delta}$$

trials of experiment e_i. Then

$$\Pr[\, c(s^*(I(\hat{p})); p) \ \leq \ c(s^*(I(p)); p) + \epsilon \,] \ \geq \ 1 - \delta.$$

(In other words, an estimated optimal strategy based on the probabilities $\hat{p}$ is, with probability $1 - \delta$, only an additive cost of ϵ worse in the actual environment than the truly optimal strategy.)

Proof. Given probability vectors p, p', let us for brevity denote $c_p(p') = c(s^*(I(p)); p')$. To restate the theorem in this notation: we are claiming that if $N \geq N(I; \epsilon, \delta)$, then $\Pr[c_{\hat{p}}(p) - c_p(p) \leq \epsilon] \geq 1 - \delta$. Note that since $s^*(I(p))$ is an optimal strategy for probabilities p, it is always the case that $c_{\hat{p}}(p) \geq c_p(p)$. We split the proof into two lemmas:

Lemma 1 *If $N \geq N(I; \epsilon, \delta)$, then $\Pr[$ for all $i = 1, \ldots, n : |\hat{p}_i - p_i| \leq \frac{\epsilon}{2nC}] \geq 1 - \delta$.*

Lemma 2 *If it is the case that for all $i = 1, \ldots, n : |\hat{p}_i - p_i| \leq \frac{\epsilon}{2nC}$, then $c_{\hat{p}}(p) - c_p(p) \leq \epsilon$.*

Proof of Lemma 1. This is a straightforward application of the Chernoff bound for the probability in the tail of a binomial distribution. A simple form of the bound (see, e.g. [2, p. 12]) states that if N experiments are performed to obtain an estimate $\hat{p}_i$ for a probability p_i, and $\lambda \geq 0$, then $\Pr[|\hat{p}_i - p_i| > \lambda] \leq 2e^{-2N\lambda^2}$. We apply this with $N = N(I;\epsilon,\delta)$, $\lambda = \frac{\epsilon}{2nC}$ to obtain the bound

$$\Pr[|\hat{p}_i - p_i| > \frac{\epsilon}{2nC}] \quad \leq \quad 2\exp\left(-2[2(\frac{nC}{\epsilon})^2 \ln\frac{2n}{\delta}](\frac{\epsilon}{2nC})^2\right) \quad = \quad \frac{\delta}{n}.$$

Hence if $N \geq N(I;\epsilon,\delta)$, then

$$\begin{aligned} &\Pr[\text{ for all } i = 1,\ldots,n : |\hat{p}_i - p_i| \leq \frac{\epsilon}{2nC}] \\ = \;& 1 - \Pr[\text{ for some } i = 1,\ldots,n : |\hat{p}_i - p_i| > \frac{\epsilon}{2nC}] \\ \geq \;& 1 - n\cdot\frac{\delta}{n} \quad = \quad 1-\delta. \end{aligned}$$

Proof of Lemma 2. Since $c_{\hat{p}}(p) - c_p(p) \leq |c_{\hat{p}}(p) - c_{\hat{p}}(\hat{p})| + |c_{\hat{p}}(\hat{p}) - c_p(p)|$, we can again split the proof in two parts:

Lemma 2.1 *If the condition of Lemma 2 holds, then* $|c_{\hat{p}}(p) - c_{\hat{p}}(\hat{p})| \leq \frac{\epsilon}{2}$.

Lemma 2.2 *If the condition of Lemma 2 holds, then* $|c_{\hat{p}}(\hat{p}) - c_p(p)| \leq \frac{\epsilon}{2}$.

Proof of Lemma 2.1. Observe that for any strategy s for I, the cost function $c(s;r)$ is everywhere differentiable with respect to the variables $r = \langle r_1,\ldots,r_n\rangle$. We shall compute the partial derivatives $\frac{\partial c}{\partial r_i}$.

Let us first introduce some notation. Given any node u_j in s, let π_j denote the path leading from the root of s to u_j. The set of terminal nodes below u_j is denoted by $t(u_j)$. If $u_l \in t(u_j)$, let π_{jl} denote the path from u_j to u_l, and if $u_l \in t(u_j^+)$, let π_{jl}^+ denote the path from u_j^+ to u_l. Let $c(s_j^+)$ denote the "expected cost of the $+$-subtree of u_j":

$$c(s_j^+) = c(u_j^+) + \sum_{u_l \in t(u_j^+)} p(\pi_{jl}^+)c(\pi_{jl}^+),$$

Analogous definitions hold for $u_l \in t(u_j^-)$, $p(\pi_{jl}^-)$, and $c(s_j^-)$.

Recall that in s, any experiment label can occur at most once on each path from the root to a leaf. By adding zero-cost nodes to the paths we may, without changing the function $c(s;p)$, assume that in fact every label occurs exactly once on each path. Let $u_1,\ldots,u_k$ be the nodes in the (modified) tree s with label e_i. We may partition the paths in s according to which of the nodes u_j they pass through, and use this representation to obtain a very simple formula for the influence of variable r_i on the function $c(s;r)$, as follows:

$$\begin{aligned} c(s;p) &= \sum_{\pi} p(\pi)c(\pi) \\ &= \sum_{j=1}^{k} \sum_{u_l \in t(u_j)} p(\pi_l)c(\pi_l) \\ &= \sum_{j=1}^{k} \left(\sum_{u_l \in t(u_j^+)} p(\pi_l)c(\pi_l) + \sum_{u_l \in t(u_j^-)} p(\pi_l)c(\pi_l) \right) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{k} \left(\sum_{u_l \in t(u_j^+)} p(\pi_j) r_i p(\pi_{jl}^+)(c(\pi_j) + c(u_j^+) + c(\pi_{jl}^+)) + \sum_{u_l \in t(u_j^-)} p(\pi_j)(1-r_i) p(\pi_{jl}^-)(c(\pi_j) + c(u_j^-) + c(\pi_{jl}^-)) \right) \\
&= \sum_{j=1}^{k} p(\pi_j) \left(r_i \sum_{u_l \in t(u_j^+)} p(\pi_{jl}^+)(c(\pi_j) + c(u_j^+) + c(\pi_{jl}^+)) + (1-r_i) \sum_{u_l \in t(u_j^-)} p(\pi_{jl}^-)(c(\pi_j) + c(u_j^-) + c(\pi_{jl}^-)) \right) \\
&= \sum_{j=1}^{k} p(\pi_j) \left[r_i \left(c(\pi_j) + c(u_j^+) + \sum_{u_l \in t(u_j^+)} p(\pi_{jl}^+) c(\pi_{jl}^+) \right) + (1-r_i) \left(c(\pi_j) + c(u_j^-) + \sum_{u_l \in t(u_j^-)} p(\pi_{jl}^-) c(\pi_{jl}^-) \right) \right] \\
&= \sum_{j=1}^{k} p(\pi_j) \left[r_i(c(\pi_j) + c(s_j^+)) + (1-r_i)(c(\pi_j) + c(s_j^-)) \right] \\
&= \sum_{j=1}^{k} p(\pi_j)[c(\pi_j) + r_i c(s_j^+) + (1-r_i) c(s_j^-)].
\end{aligned}
$$

From this representation we immediately obtain the partial derivatives:

$$
\frac{\partial c}{\partial r_i} \quad = \quad \sum_{j=1}^{k} p(\pi_j)(c(s_j^+) - c(s_j^-)).
$$

Moreover, since

$$
\begin{aligned}
c(s_j^+) &= c(u_j^+) + \sum_{u_l \in t(u_j^+)} p(\pi_{jl}^+) c(\pi_{jl}^+) \\
&\leq c(u_j^+) + (\max_{u_l \in t(u_j^+)} c(\pi_{jl}^+)) \sum_{u_l \in t(u_j^+)} p(\pi_{jl}^+) \\
&= \max_{u_l \in t(u_j^+)} c(\pi_{jl}) \quad \leq \quad \max_{u_l \in t(u_j^+)} c(\pi_l) \quad \leq \quad C
\end{aligned}
$$

(and similarly for $c(s_j^-)$), we obtain the bound

$$
|\frac{\partial c}{\partial r_i}| \quad = \quad |\sum_{j=1}^{k} p(\pi_j)(c(s_j^+) - c(s_j^-))| \quad \leq \quad C|\sum_{j=1}^{k} p(\pi_j)| \quad = \quad C.
$$

Consider then the strategy $\hat{s} = s^*(I(\hat{p}))$, and the values of the associated cost function $c_{\hat{p}} = c(\hat{s}; r)$ for vectors p and $\hat{p}$. By the mean-value theorem, there is a vector $r = \hat{p} + \theta(p - \hat{p})$, $0 \leq \theta \leq 1$, such that

$$
|c_{\hat{p}}(p) - c_{\hat{p}}(\hat{p})| \quad = \quad |\sum_{i=1}^{n} \frac{\partial c_{\hat{p}}(r)}{\partial r_i}(p_i - \hat{p}_i)| \quad \leq \quad \sum_{i=1}^{n} C|p_i - \hat{p}_i| \quad \leq \quad nC\frac{\epsilon}{2nC} = \frac{\epsilon}{2}.
$$

Proof of Lemma 2.2. Let us denote $c^*(r) = c_r(r) = \min_{s \in \mathcal{S}} c(s; r)$. As a minimum of finitely many continuous functions, c^* is continuous. In particular, the projections of c^* on each variable r_i are piecewise linear functions, differentiable everywhere except at finitely many points. Whenever the partial derivative $\frac{\partial c^*(r)}{\partial r_i}$ exists, it is the case that

$$
|\frac{\partial c^*(r)}{\partial r_i}| \quad \leq \quad \max_{s \in \mathcal{S}} |\frac{\partial c(s; r)}{\partial r_i}| \quad \leq \quad C.
$$

Given vectors $p, \hat{p}$, let us denote $p^{(i)} = \langle \hat{p}_1, \ldots, \hat{p}_i, p_{i+1}, \ldots, p_n \rangle$, and as special cases, $p^{(0)} = p$ and $p^{(n)} = \hat{p}$. Then

$$|c^*(\hat{p}) - c^*(p)| \quad \leq \quad \sum_{i=1}^{n} |c^*(p^{(i)}) - c^*(p^{(i-1)})| \quad \leq \quad \sum_{i=1}^{n} C|\hat{p}_i - p_i| \quad \leq \quad nC\frac{\epsilon}{2nC} = \frac{\epsilon}{2}.$$

This concludes the proof of Lemma 2.1, and hence also of the Theorem. □

CONCLUSION

We have proved a low-order polynomial bound on the number of samples needed to determine approximately optimal search strategies in a very general model of satisficing search problems. This bound can be used to obtain search strategy learning methods for any class of structures $\mathcal{I}$ conforming to the model, and any class of strategies $\mathcal{S}$, provided only that there is an algorithm s^* capable of mapping any pair consisting of a partially specified structure $I \in \mathcal{I}$ and a success probability vector p to a strategy $s^*(I(p))$ that is optimal for the completed structure $I(p)$ within $\mathcal{S}$.

References

[1] J. A. Barnett: How much is control knowledge worth?: A primitive example. *Artif. Intell. 22* (1984), 77–89.

[2] B. Bollobás: *Random Graphs.* Academic Press (1985).

[3] M. R. Garey: Optimal task sequencing with precedence constraints. *Discr. Math. 4* (1973), 37–56.

[4] R. Greiner: Finding the optimal derivation strategy in a redundant knowledge base. *Artif. Intell.* (to appear).

[5] K. S. Natarajan: Optimizing depth-first search of AND-OR trees. Research report RC-11842, IBM T. J. Watson Research Center (Jan. 1986).

[6] H. A. Simon and J. B. Kadane: Optimal problem-solving search: All-or-none solutions. *Artif. Intell. 6* (1975), 235–247.

[7] D. E. Smith: Controlling backward inference. *Artif. Intell. 39* (1989), 145–208.

[8] S. Sahni: Computationally related problems. *SIAM J. Comput. 3* (1974), 262–279.

[9] L. G. Valiant: A theory of the learnable. *Comm. ACM 27* (1984), 1134–1142.