Ranking Hypotheses to Minimize the Search Cost in Probabilistic Inference Models

Peter Damaschke
Department of Computer Science and Engineering
Chalmers University, 41296 Göteborg, Sweden
ptr@cs.chalmers.se

Abstract

Suppose that we are given n mutually exclusive hypotheses, m mutually exclusive possible observations, the conditional probabilities for each of these observations under each hypothesis, and a method to probe each hypothesis whether it is the true one. We consider the problem of efficient searching for the true (target) hypothesis given a particular observation. Our objective is to minimize the expected search cost for a large number of instances, and for the worst-case distribution of targets. More precisely, we wish to rank the hypotheses so that probing them in the chosen order is optimal in this sense. Costs grow monotonic with the number of probes. While it is straightforward to formulate this problem as a linear program, we can solve it in polynomial time only after a certain reformulation: We introduce mn^2 so-called rank variables and arrive at another linear program whose solution can be translated afterwards into an optimal mixed strategy of low description complexity: For each observation, at most n rankings, i.e., permutations of hypotheses, appear with positive probabilities. Dimensionality arguments yield further combinatorial bounds. Possible applications of the optimization goal are discussed.

Keywords: probabilistic inference, searching, ranking, mixed strategy, linear programming, polytopes

1 Problem Statement

We study an optimization problem in probabilistic inference. Assume that among n mutually exclusive hypotheses exactly one is true. We make exactly one observation out of m possible observations which are mutually exclusive as well. For every hypothesis h and every observation D we know $P(D|h)$, defined as the conditional probability to observe D if hypothesis h is true. Note that we mean by D a complete description of what we observe, hence the conditional probabilities satisfy $\sum_D P(D|h) = 1$ for every h. The $P(D|h)$ are known from background knowledge about causal relations, or estimated from statistical material. In contrast to these conditional probabilities, the distribution of targets may be arbitrary and unpredictable.

The problem is, first in a vague formulation, as follows. Given an observation D, a Searcher wants to find the correct hypothesis h, also called the *target hypothesis* or simply the *target*, in a cheap and efficient way. For the moment we assume that every one hypothesis can be tested for being the target by some reliable experiment. (We discuss this matter further in Section 2.) Consider two natural settings:

- Let $g < n$ be a fixed number. For any D, we are allowed to choose g hypotheses, and we would like to have the target in our selected set. One may think of g as an acceptable number of attempts to identify the target, whereas additional tests are undesirable or come with an extra cost.

- A smoother cost assumption is that all verification experiments have unit cost, and their number is not limited other than by n. Then, for any D, we have to rank the hypotheses and test them in the chosen order. We want to find the target early, i.e., minimize the number of tests. In other words, we want to rank the target low.

The target is unknown (otherwise there is nothing to search for), but the $P(D|h)$ and the observed D may hint to it. Note that the selection or ranking need not be deterministic, that is, Searcher may apply a randomized strategy. We are interested in strategies that perform well on average, on a large number of problem instances but for the given fixed table of conditional probabilities. That is, Searcher is presented many independent cases and wants to minimize his total search costs in the long run. (There is not much to say about the actual performance on any single instance, but for a large number of instances, expected costs turn into average actual costs.) Since we assume nothing about the distribution of targets h, it is natural to optimize the expected search cost in the worst case, i.e., maximized over all h. This corresponds also to the usual game-theoretic framework: An adversary is presenting the instances, and a strategy is sought that gives the best guarantee for the objective (here: for the expected costs). We will discuss motivations and applications of this optimization goal in Section 2. Note that the probability space, for each h, is defined by the randomness both in the observations and in the strategy, and expectation refers to this probability space. This will become explicit in the formal description below. For calculations it is more convenient to denote the observations by indices $k = 1, \ldots, m$ and hypotheses by $j = 1, \ldots, n$. We write p_{kj} for $P(D_k|h_j)$.

Search Cost Minimization
Let us be given a sequence of costs $c_1 \leq \ldots \leq c_n$, and for each pair k, j ($1 \leq k \leq m$ and $1 \leq j \leq n$) the conditional probability p_{kj} for observation k if j is the target hypothesis. The task is, for each particular observation k, to rank the n hypotheses so that the expected value of c_r (expected search cost) is minimized, where r is the rank of the target. More specifically, since the target j is not known, we wish to *minimize the expected search cost in the worst case*,

i.e., maximized over all j. Expectation refers to the random rank of j in the strategy responding to the random observation k caused by j.

Before we give a more formal specification of the problem, we show that cost sequences actually capture the two aforementioned cases. In fact, the "g-selection" problem is now specified by $c_1 = \ldots = c_g = 0$ and $c_{g+1} = \ldots = c_n = 1$, which expresses that the first g tests are for free, and further tests cost some one-time fee. Thus we are interested in maximizing the probability to catch the target. The second case with sequential testing is, obviously, equivalent to $c_r = r$. General monotone sequences c_r can model more complex cost structures (see also Section 2). Anyhow, our algorithm for Search Cost Minimization will work for any monotone costs c_r in the same way, thus we study the problem straightaway in this generality.

Now we turn to the formalization. Since our Searcher can decide on a ranking by a randomized strategy, a Searcher's strategy is specified by a set of probabilities $x_{\pi k} \geq 0$ to choose permutation π of hypotheses in the event of observation k, for all π and k. Of course, strategies must satisfy

$$\forall k: \sum_{\pi} x_{\pi k} = 1. \tag{1}$$

The expected cost in case that j is the target amounts to $\sum_k \sum_\pi c_{r(j,\pi)} x_{\pi k} p_{kj}$, where $r(j,\pi)$ denotes the rank of hypothesis j in permutation π. Our objective is therefore

$$\min \max_{j} \sum_{k} \sum_{\pi} c_{r(j,\pi)} x_{\pi k} p_{kj}. \tag{2}$$

This is an optimization problem with linear constraints and a set of linear objective functions the maximum of which shall be minimized. A standard trick transforms any such problem into a linear program (LP): Introduce a new variable u and new constraints expressing that the jth objective is less than or equal to u, and minimize u. However, this straightforward reduction to an LP does not solve Search Cost Minimization in polynomial time, as we get $mn!$ variables $x_{\pi k}$. (Note that the $r(j,\pi)$ are not problem variables, r is just a "meta"-function describing the indices in general form.)

The main technical contribution of this paper is a *polynomial* algorithm for Search Cost Minimization. We still use a reduction to an LP, but on a detour via auxiliary variables. From the optimal solution of the corresponding LP we finally construct solutions to the original problem, i.e., randomized ranking strategies. For each observation they need at most n different rankings with nonzero probabilities. We emphasize that it is a suitable reformulation of the LP which leads to an efficient solution.

The paper is organized as follows. In Section 2 we discuss the model by some examples and review related literature. Then the technical results follow. Section 3 gives the announced algorithm, except the procedure which actually constructs the randomized ranking. This construction is better developed separately, in Section 4. In Section 5 we point out that optimal randomized solutions are still "very much deterministic", which makes them easier to handle in practice. Section 6 concludes the paper with some open questions and directions of further research.

2 Illustrations and Discussion

Let us recall the main issues:

- The $P(D|h)$ are known and fixed, but nothing is assumed about the distribution of hypotheses h. Expressed in game-theoretic terms, targets may be generated by a malicious adversary.
- We are interested in the average search costs that can be guaranteed for a large number of instances, and for any possible h. Observation D is the "contingent" part in each instance, drawn according to the $P(h|D)$, and just used as a hint to the possible targets.
- Rankings can be randomized.

As a first illustration we start with some calculation example that is kept as simple as possible, to make the point. Let be $m = 2$, $n = 3$, and conditional probabilities given by:

$$p_{11} = 1.0,\ p_{12} = 0.5,\ p_{13} = 0.0,$$
$$p_{21} = 0.0,\ p_{22} = 0.5,\ p_{23} = 1.0.$$

Since here only two hypotheses have nonzero p_{kj} for each k, the cost sequence is not relevant, and costs are just proportional to the probability to err with the hypothesis ranked first. Intuitively one might rank the hypotheses by decreasing likelihood, that is, 1-2-3 if $k = 1$, and 3-2-1 if $k = 2$. Then we are always right if 1 or 3 is the target, but also always wrong if 2 is the target, which is bad during times when target 2 abounds. A more balanced and fair strategy ranks hypothesis 1 or 2 first, each with probability 0.5, if $k = 1$, and similarly it ranks hypothesis 2 or 3 first, each with probability 0.5, if $k = 2$. Then we err with probability 0.5, regardless what the true hypothesis is. Expected costs are lower than in the maximum-likelihood strategy already if hypothesis 2 happens to be true in more than half of the cases. But no strategy dominates the other one in terms of costs, rather, that depends on the target frequencies. However, SEARCH COST MINIMIZATION gives the best possible guarantee for any target distribution. This smoothing effect is beneficial in the following more complex scenario.

Suppose that some error-prone transmission or copying mechanism changes each character of an input string S into an output character (identical or different), according to known conditional probabilities that are characteristic for this *noisy channel*. We wish to reconstruct the input S from output string T, with the prior knowledge that S belongs to a certain formal language of "conceivable" or "meaningful" strings.

A natural approach is to rank the possible input characters at every position, depending on the output character there, in such a way that the true input character is always ranked as early as possible. Then the hypothetical input characters can be connected to possible substrings of S, say of length l, which are finally assembled to infer the whole of S, using context information. Let us define the score of a candidate substring of S as some weighted sum of ranks of its symbols. We may keep all meaningful substrings of length l, sorted by ascending scores up to some threshold. Now, if our ranking gives a guarantee on this expected score $E[c]$ for all possible input characters (as our optimization does), and the threshold is chosen slightly above $lE[c]$, then the true substring of S is found *with high probability* among these candidates. In contrast, a simple deterministic ranking (e.g., by likelihood) at each position can systematically rule out certain target strings. (For instance, if we have conditional probabilities as in the above example, and string $222\ldots 2$ belongs to the language, this string would *always* get the worst score, even though it may occur frequently in S.) Since the search space explodes with the ranks, the expected score of the target is the function that should be minimized. A comprehensive study of noisy string reconstruction is beyond the scope of this work. Overall efficiency depends on many factors like density and structure of the language (how many other meaningful substrings are smuggled in as false candidates), the time for testing whether a string belongs to the language, and the search procedures applied. But in any case, local hypothesis ranking is the basis for efficiency.

The noisy channel model is well established in language processing, e.g., for spelling correction (where the units in S and T are words rather than characters). A system proposes, upon detection of putative misprints, possible corrections as ranked lists. Conditional probabilities of various types of editing errors have been modelled, see [5, 19] for details. Bayes' theorem is then applied to guess the most likely original words, ranked by posterior probabilities. The hope is to rank the true hypotheses early in each case. Another module may even test the proposed corrections one by one and pick the target automatically, using context information and a model of the language. However, a certain problem is that prior probabilities of the correct words are needed in Bayes' theorem, albeit word frequencies are very different in various text corpora, which makes estimates of priors rather questionable. In contrast, typos can be assumed to be independent local events with characteristic conditional probabilities. Furthermore, the average performance on a large amount of text is relevant. Altogether this matches the assumptions behind Search Cost Minimization. In practice, reasonably small clusters of (both correct and misspelled) words that are close with respect to edit distance could be represented each by an onw matrix of conditional probabilities $P(D|h)$, since every word gives rise to only a few different misspellings, while all other variants have negligible probabilities.

Actually, our interest in ranking local hypotheses in strings emerged from a bioinformatics project aiming at heuristics for the prediction of protein backbone torsion angles from measured spectroscopic data that are correlated to these torsion angles [3]. There, the input and output "characters" are discretized intervals of values, and probabilities of measured values given a torsion angle are known empirically. Meaningful strings S are sequences of torsion angles that are geometrically and energetically possible. Good local estimates of torsion angles as early hypotheses would speed up the reconstruction of 3D structures of proteins. It can be assumed that correlations between torsion angles and measurements follow some fixed conditional probabilities, while frequencies of torsion angles can be vary a lot between different protein chains S, depending on whether helix, strand, or coil structures abound.

Other application domains one may think of are technical and medical diagnosis. Here, each hypothesis h is a possible failure scenario or an underlying disease, and each observation D is a complete system response or a syndrome, respectively. Again, we may assume that the $P(D|h)$ are stable, coming from causal dependencies that can have probabilistic elements but do not change over time, whereas the frequencies of failures or diseases vary, for previously unknown systematic reasons (defective technical equipment, epidemics, etc.). Therefore we do not assume that the target distribution can be learned from the past. It may appear counterintuitive to compute a ranking of hypotheses, given a particular D, with help of the conditional probabilities of all D (which have not been observed right now). However, note that the goal is still to minimize average examination costs or time for a large number of cases, and no prior or posterior target probabilities exist in this model that could be connected to a single D. (A different discussion is to what extent decisions in medicine can be based on quantitive reasoning at all, other than using it as a first guideline. Every "case" has special circumstances that appear in no model, and experience and intuition are required, too.)

The assumption that hypotheses and observations, respectively, are mutually exclusive and exactly one is present is not a proper restriction. Multiple, structured or overlapping hypotheses or observations can be reformulated as disjoint cases, so that we can work with a single target. (This is also customary in Bayesian inference.) For example, if multiple diseases can occur, some hypotheses may represent possible combinations, besides the hypotheses for single diseases. However, a restriction is that we assume costs that merely depend on the rank of the target.

The c_r are in general not supposed to be exactly proportional to actual search costs or times, rather they can reflect estimated typical costs when the target is at position r in the list. In string reconstruction as proposed above, the c_r just act as weights in the scores and determine the order in which candidate strings are considered. In applications with independent problem instances like medical testing, the optimization problem with $c_r = r$ is most appropriate if tests are made sequentially, and a specific test is available for every disease (in the set of hypotheses) that gives no negative side information excluding other hypotheses. Otherwise, sublinear c_r can account for such extra information that makes subsequent diagnostics cheaper.

Rank-dependent costs can also be used in a natural way to deal with unreliable tests. Suppose that each test gives the wrong answer with some known maximal error probability. False positives are simply handled by repeating a positive test until the desired confidence is reached. More interesting are false negatives. Instead of going through the ranked list only once, we need to specify (algorithmically) a sequence of tests that returns to every rank arbitrarily often, until a positive response is received. To be concrete, let us assume that we get false negatives with probability ϵ, that is, each test on the target fails to say yes with probability ϵ. Then we can formulate the following result. Note that ϵ is previously known, whereas r is not.

Theorem 1 *Given an ordered set of hypotheses and a verification test that wrongly gives a negative answer on the true hypothesis with probability ϵ. Then we can find the target with an expected number of $(1+2\sqrt{\epsilon}+O(\epsilon))r$ tests, if the target has rank r in the sequence.*

Proof. For simplicity, our test sequence will even work for an infinite sequence of hypotheses $r = 1, 2, 3, \dots$ If we have only a finite number n of hypotheses, we may use the tests assigned to $r > n$ on the existing hypotheses instead, which can only improve the performance.

Let q be an integer depending on ϵ that we fix later. Tests are indexed $1, 2, 3, \dots$. Each rank $r \geq 1$ can be uniquely represented as $r = i(q-1)+j$, where $i \geq 0$ and $1 \leq j \leq q-1$. We check hypothesis r with the tests indexed $(iq+j)q^0, (iq+j)q^1, (iq+j)q^2, (iq+j)q^3, \dots$ In fact, every test is assigned to exactly one hypothesis. This is seen as follows. Indices $iq+j$ of the initial tests on all hypotheses are exactly those positive integers not divisible by q. Furthermore, by elementary number theory every positive integer has a unique representation sq^k where s is not a multiple of q.

If the target has rank r, it will be identified after an expected number of tests given by:

$$(1-\epsilon)(iq+j)\sum_{k=0}^{\infty}(\epsilon q)^k = \frac{(1-\epsilon)(r+i)}{1-\epsilon q} \leq \frac{(1-\epsilon)(1+\frac{1}{q-1})r}{1-\epsilon q}.$$

With $q = \lfloor 1/\sqrt{\epsilon} \rfloor$ we get the result. $\diamond$

That is, we have devised a test sequence which, for any fixed error probability and every possible r, detects the target at rank r after $O(r)$ expected tests. By visiting the lower ranks more (less) frequently than in this sequence, we can obtain arbitrary sublinear (superlinear) cost functions c_r instead.

We finish this introductory part with some related literature. The widely used maximum-likelihood inference principle works with a table of conditional probabilities $P(D|h)$. What is different in the present paper is that we want to support the *search* for the true hypothesis rather than proposing a single, “most likely” hypothesis, and that we do not work with priors. (In Bayesian inference, prior probabilities are often chosen by symmetry or simplicity considerations. They might have little to do with the actual target frequencies in a set of instances.) The approach may complement, but not replace, Bayesian inference.

We did earlier work on this topic, but mainly in the framework of competitive analysis of decisions under incomplete information, and for problem settings where the verification of different hypotheses may need different periods of time, see [7, 8]. Mathematically most similar to the present paper is a work [3] about the “orthogonal” problem: select a minimum weight subset of hypotheses that misses the target with a given error probability. Ranking was not considered in [3]. See the editorial [2] for other optimization models for hypothesis selection. In general, treating decision making as optimization problems (e.g., expected utility maximization) is an established point of view.

Searching for hidden objects has been extensively studied in graph theory and computational geometry as well. We mention competitive analysis of the cow-path problem and its variants, against a malicious adversary (worst case) as in [1, 14, 15, 16] or under probabilistic assumptions [12]. There, the object is found when the Searcher, walking along paths, hits the object, that is, Searcher must rank the possible locations he wants to visit, and verify the true location. The cost is the total distance walked. Problems of this type in graphs are addressed, e.g., in [11, 18]. Online construction of hybrid algorithms for search tasks is another motivation of this line of research [16, 13]. The linear ordering (or: binary choice) polytope has been well studied for many years, however in connection with *discrete* optimization problems; see [9] for further hints.

3 Efficient Solution with n Rankings

We return to SEARCH COST MINIMIZATION as specified in Section 1. The key idea leading to polynomial complexity is the following reformulation. Instead of the too many variables $x_{\pi k}$ we introduce mn^2 *rank variables* $z_{rj}^k \geq 0$ for all observations k, ranks r, and hypotheses j, and the following constraints for every fixed k:

$$\forall j : \sum_{r=1}^{n} z_{rj}^k = 1, \tag{3}$$

$$\forall r : \sum_{j=1}^{n} z_{rj}^k = 1. \tag{4}$$

That is, the rank variables form a doubly stochastic matrix for every k. We refer to (3) and (4) as the *rank constraints.* Variable z_{rj}^k shall indicate the probability that the Searcher assigns hypothesis j the rank r if observation k has shown up. Since hypotheses and ranks are mapped one-to-one in a (deterministic) schedule, the rank constraints are obvious. By linearity of expectation, the expected cost of a strategy solely depends on the rank variables. we can now express our objective (2) as:

$$\min \max_j \sum_{k=1}^{m} p_{kj} \sum_{r=1}^{n} c_r z_{rj}^k. \tag{5}$$

Recall that (3)-(5) can be easily rewritten as an LP. As for the description complexity of an optimal strategy, we start with a preliminary result because it has a very short proof.

Proposition 2 *Every instance of* SEARCH COST MINIMIZATION *with arbitrary cost function has an optimal strategy where, for each observation, fewer than n^2 permutations of hypotheses appear with positive probabilities.*

Proof. Solve the LP with rank constraints (3),(4) and objective (5). For every k this gives a matrix of entries z^k_{rj} where all row and column sums are 1. By a classical theorem of Birkhoff [4], every doubly stochastic matrix is a convex linear combination of fewer than n^2 permutation matrices, i.e., such with exactly one 1 in each row and column, and 0's else. This yields the existence of a strategy as claimed. $\diamond$

In the following we show that SEARCH COST MINIMIZATION has optimal solutions with only n rankings (permutations) for each observation, and that such combinatorially simpler optimal strategies are computable in polynomial time. The next idea is to introduce yet another, smaller set of mn variables x^k_j indicating the expected cost if j is the target and k has been observed. We call them the *cost variables.* Note that they are connected to the rank variables by $x^k_j = \sum_{r=1}^n c_r z^k_{rj}$. Still we will need the rank variables z^k_{rj} to do the optimization in polynomial time.

A first lemma states some constraints on the cost variables. Note that numbers $1, \ldots, n$ serve here only as *names* of hypotheses, hence we may permute them arbitrarily, i.e., re-index the hypotheses (to formulate statements about any ordered sequences of hypotheses). Moreover, in the following we omit superscript k whenever we are dealing with one fixed observation k.

Lemma 3 *For any assignment of indices from $\{1, \ldots, n\}$ to the hypotheses, the rank constraints imply the so-called* cost constraints *(6) and (7), where $j = 1, \ldots, n$.*

$$\forall j : \sum_{i=1}^{j} x_i \geq \sum_{i=1}^{j} c_i \tag{6}$$

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} c_i \tag{7}$$

Proof. For any fixed j we have (with the understanding that a sum is zero if the lower limit is greater than the upper limit of the summation index):

$$\sum_{i=1}^{j} x_i = \sum_{i=1}^{j} \sum_{r=1}^{n} c_r z_{ri}$$

$$= \sum_{i=1}^{j} \sum_{r=1}^{j} c_r z_{ri} + \sum_{i=1}^{j} \sum_{r=j+1}^{n} c_r z_{ri}$$

$$\geq \sum_{i=1}^{j} \sum_{r=1}^{j} c_r z_{ri} + c_j \sum_{i=1}^{j} \sum_{r=j+1}^{n} z_{ri}$$

$$= \sum_{i=1}^{j} \sum_{r=1}^{j} c_r z_{ri} + c_j \sum_{i=1}^{j} (1 - \sum_{r=1}^{j} z_{ri})$$

$$= \sum_{r=1}^{j} c_r \sum_{i=1}^{j} z_{ri} + c_j j - c_j \sum_{r=1}^{j} \sum_{i=1}^{j} z_{ri}$$

$$= \sum_{r=1}^{j} c_r \sum_{i=1}^{j} z_{ri} + c_j j - c_j \sum_{r=1}^{j} (1 - \sum_{i=j+1}^{n} z_{ri})$$

$$= \sum_{r=1}^{j} c_r \sum_{i=1}^{j} z_{ri} + c_j \sum_{r=1}^{j} \sum_{i=j+1}^{n} z_{ri}$$

$$\geq \sum_{r=1}^{j} c_r \sum_{i=1}^{j} z_{ri} + \sum_{r=1}^{j} c_r \sum_{i=j+1}^{n} z_{ri}$$

$$= \sum_{r=1}^{j} c_r \sum_{i=1}^{n} z_{ri} = \sum_{r=1}^{j} c_r$$

which gives (6). For $j = n$ observe that the two inequalities in this chain are equations. This yields (7). $\diamond$

The next lemma which relies only on cost constraints is central for our complexity result:

Lemma 4 *For any sequence of costs $c_1 \leq \ldots \leq c_n$ and any sequence $x_1, \ldots, x_n$ satisfying the cost constraints, there exists a polynomial-time computable distribution on the rankings of hypotheses $1, \ldots, n$ such that x_j equals the expected search cost if j is the target, and where at most n rankings appear with positive probabilities.*

We defer the somewhat technical proof and first use Lemma 4 to get the final result of this section:

Theorem 5 *Every instance of* Search Cost Minimization *with arbitrary monotone cost function has a polynomial-time computable optimal strategy where, for each observation, at most n rankings appear with positive probabilities.*

Proof. For the given p_{kj}, solve the LP with rank variables $z_{rj}^k \geq 0$, specified by (3)-(5). Since the z_{rj}^k satisfy the rank constraints, Lemma 3 yields that the x_j^k satisfy the cost constraints (6),(7). Finally use the polynomial-time algorithm delivered by Lemma 4 to construct the m probability distributions on the rankings that realize these x_j^k, and hence the optimal value of the LP (3)-(5). This step can change the rank variables, but not the costs.

To show that this solution from the LP also minimizes $\max_j \sum_{k=1}^{n} p_{kj} x_j^k$ among all strategies (as desired), consider any set of distributions on the rankings, one for each observation k, and let $\hat{x}_j^k$ be the expected cost if j is the target and k has been observed. Let z_{rj}^k be the probability to find hypothesis j at rank r. Clearly, these variables are nonnegative, fulfill the rank constraints, and $\hat{x}_j^k = \sum_{r=1}^n c_r z_{rj}^k$. Hence there exists also a feasible solution to LP (3)-(5) where the objective function value is $\max_j \sum_{k=1}^n p_{kj}\hat{x}_j^k$. But this value cannot be smaller than in the optimal solution to the LP. That means, any other strategy can only be worse. $\diamond$

4 Construction of the Ranking

It remains to prove Lemma 4 that constructs the randomized rankings from the values of cost variables. First we introduce the notation and sketch the plan of the proof, then we fill in the details.

Let $c_1 \leq \ldots \leq c_n$ be the given monotone cost sequence. Let $x_1, \ldots, x_n$ be a sequence satisfying the cost constraints (6),(7). On the real axis we mark points with coordinates x_j, in the following called *cost points*. Next, we place n pebbles indexed $i = 1, \ldots, n$ at points $y_i := c_i$, $i = 1, \ldots, n$. If several x_j or several c_i, respectively, are identical, such cost points are marked with their multiplicities and, similarly, the proper number of pebbles is placed at one point.

Recall that x_j is the expected cost of identifying hypothesis j as the target (if j is actually the target). Pebbles shall represent hypotheses, however, the assignment of hypotheses to pebbles will be decided later. The expected cost of a hypothesis is the coordinate of the pebble representing this hypothesis. Since initially the pebbles' coordinates are the c_i, the initial placement corresponds to a trivial distribution (deterministic strategy) consisting of one ranking with probability 1. In order to make the multiset of expected costs of hypotheses equal to the desired x_j's we will match pebbles and cost points. This is done by moving pebbles to cost points in such a way that the pebbles' coordinates y_i satisfy the cost constraints all the time. (Note that this condition is trivially satisfied in the beginning.) Simultaneously with the moves we will adjust our probability distribution on the rankings in such a way that the coordinate of each pebble still equals the expected cost of the hypothesis represented by that pebble.

We say that a cost point and a pebble are *matched* if the pebble is on that point and we have explicitly declared them as matched. Yet unmatched cost points and pebbles are called *free*. Only when everything is matched we rename the pebbles and eventually assign index (hypothesis) j to the pebble at point x_j. Thus, each hypothesis gets the desired expected cost in the end.

We have to show that these moves can be carried out, respecting the aforementioned invariants, and using only the assumption that the x_j fulfill the cost constraints. For the probability distribution we have to show that at most n different rankings of hypotheseses get positive probabilities. Next we describe one generic step of our procedure in detail.

We take the leftmost free cost point x. If there is a free pebble at x, we match one such pebble with x, and nothing else is changed. In the other case, we take the nearest free pebble s and t to the left and to the right, respectively, of x.

Claim: Pebbles s, t as specified exist.

For the moment suppose that this is true. We start moving the pebbles s and t towards x at equal speed, until at least one arrives at x. Whenever a moving pebble passes a matched pebble, we rename the pebbles so that they always appear in the order $1, \ldots, n$ on the axis. (That is, index s and t of a moving pebble may increase and decrease, respectively, but $s < t$ remains true all the time.) As soon as some pebble arrives at x, we match one with x.

Considering the leftmost j pebbles at any moment, we see that the cost constraints do always hold for their y-coordinates: This was trivially true in the beginning ($y_i = c_i$), and later a pebble is moved to the left only if another pebble with smaller index is moved to the right, so that the prefix sums $\sum_{i=1}^{j} y_i$, which are the left-hand sides of (6), can only increase, and $\sum_{i=1}^{n} y_i$, which is the left-hand side of (7), remains fixed.

In order to adjust the probability distribution on the rankings of hypotheses during a move, recall that the pebbles' positions are the expected costs of the assigned hypotheses, and costs depend monotonically on the ranks. Since $s < t$, there exists some ranking π with positive probability, where the hypothesis assigned to s is to the left of that assigned to t. Define π' as the ranking obtained from π by swapping s and t. While moving the pebbles, we decrease the probability of π and increase that of π' at the same speed, so that the pebbles' coordinates always equal the expected costs. (Since the expected cost of s increases and that of t decreases at the same rate, this is consistent, i.e., the sum of probabilities remains constantly 1.) Whenever we have to rename the pebbles, we also change the assignment of pebbles to hypotheses so that always the same two hypotheses remain assigned to the *moving* pebbles. In fact, this is possible, as the exchange happens only between pebbles with identical coordinates (costs).

We reach one of these two cases: Either (a) some pebble arrives at x as desired, or (b) π vanishes before, i.e., the probability of π goes down to 0. In case (b) there must exist another ranking π with the aforementioned properties, and we continue with the corresponding π'. Note that, in π, the hypothesis assigned to the currently moving s stands to the left of the hypothesis assigned to the currently moving t, and in π' it is the other way round. Thus, a ranking can take on only one role, either π or π'. It follows that case (b) can appear only once for each ranking π that had positive probability before. Hence we will end up in case (a) and match one more cost point x with a pebble. The net effect of the whole procedure to match x is that at most one more ranking than before got positive probability, and the computational complexity is polynomial for every cost point.

Together with the initial ranking, iterating the procedure for all n cost points creates at most $n+1$ rankings with positive probabilities. Actually, their number is at most n, because the last two free pebbles match the last two free cost points simultaneously, which follows from cost constraint (7).

It remains to prove the Claim. Since the sum of coordinates of pebbles and cost points are equal by cost constraint (7), it is impossible that all free pebbles are to the left of the leftmost free cost point x. Now consider the first moment when all free pebbles are to the right of x. Since we have served the cost points from left to right, and we have always moved the nearest free pebbles towards them, only the leftmost free pebble can have been moved so far. Hence, for some j, the leftmost $j-1$ pebbles are matched with cost points, the jth pebble is free, and all pebbles to the right are yet untouched. Due to the last fact, the sum of coordinates of the leftmost j pebbles still equals their initial sum, that is, $\sum_{i=1}^{j} c_i$. On the other hand, since cost constraint (6) holds for j, the j smallest x_i have a sum at least $\sum_{i=1}^{j} c_i$. This contradicts the assumption that the jth pebble has a coordinate greater than x. $\diamond$

5 Further Properties of Optimal Solutions

For the special case of the Search Cost Minimization problem with cost function $c_1 = \ldots = c_g = 0$ and $c_{g+1} = \ldots = c_n = 1$, cost constraints (6)-(7) are equivalent to the simpler constraints $0 \leq x_j^k \leq 1$ and $\sum_{j=1}^{n} x_j^k = n-g$. We refer to this case as

Selecting Hypotheses
Given conditional probabilities p_{kj} for all k, j ($1 \leq k \leq m$ and $1 \leq j \leq n$), a number $g < n$, and a particular observation k, select g hypotheses so that the probability not to select the target is minimized. More specifically, since the target j is not known, we wish to *minimize the failure probability in the worst case*, i.e., maximized over all j.

It is worth noticing that there always exist optimal solutions that are deterministic to a large extent. (Cf. an analogous result for a similar problem with prescribed error probability and unspecified g in [3].) We start with an obvious technical lemma.

Lemma 6 *For any non-negative numbers p, q, r, s that satisfy, without loss of generality, $ps \geq qr$, there exists $x, y \neq 0$ such that $px - ry \geq 0$ and $-qx + sy \geq 0$. In particular, any x, y with $q/s \leq y/x \leq p/r$ will do.* $\diamond$

Theorem 7 *Any instance of* Selecting Hypotheses *has a polynomial-time computable optimal solution where the matrix of costs x_j^k has no 2×2 submatrices of entries that are strictly between 0 and 1.*

Proof. Consider any solution. Since the only constraints are $0 \leq x_j^k \leq 1$ and constant sums $\sum_{j=1}^{n} x_j^k$ for each k, we may add some number x to x_j^k and subtract x from some other $x_{j'}^k$. If, for some other k', some x' is added to $x_{j'}^{k'}$ and taken from $x_j^{k'}$, then the probability to have hypothesis j and j', respectively, in the selected set is changed by $p_{kj}x - p_{k'j}x'$ and $-p_{kj'}x + p_{k'j'}x'$, respectively. If both changes are nonnegative, an optimal solution remains optimal after this modification. Such $x, x' \neq 0$ do always exist and are easy to find, using Lemma 6. Hence we can change at least one of $x_j^k, x_j^{k'}, x_{j'}^k, x_{j'}^{k'}$ into 0 or 1. This way we can compute, in polynomial time, an optimal solution with the claimed property. $\diamond$

The lack of 2×2 submatrices with "truly probabilistic" entries is as such an interesting structural property of the solution matrix. From Theorem 7 and known combinatorial bounds in extremal graph theory [6, 10, 17] we may now conclude that at most $\min(m\sqrt{n}, n\sqrt{m}) + m + n$ variables have values other than 0,1. However, a more basic argument gives even stronger bounds:

Theorem 8 *Any instance of* SELECTING HYPOTHESES *has a polynomial-time computable optimal solution where at most $m+n$ variables x_j^k are strictly between 0 and 1. In case $n=2$ there exist at most two such variables.*
Any instance of SEARCH COST MINIMIZATION, *with arbitrary cost function, has a polynomial-time computable optimal solution where at most $(2m+1)n$ variables z_{rj}^k are strictly between 0 and 1.*

Proof. Any instance of SELECTING HYPOTHESES can be written as an LP with one extra variable u for the objective function, n constraints expressing that u is at least the failure probability for each target, constraints $0 \leq x_j^k \leq 1$ describing a hypercube, and m nontrivial constraints $\sum_{j=1}^n x_j^k = n - g$. Some optimal solution to the LP is a vertex of the polyhedron of feasible solutions. Since the number of binding constraints in a vertex is at least the dimension of the space (number of variables), and at most $m+n$ constraints other than $0 \leq x_j^k$ and $x_j^k \leq 1$ can be binding, at least $mn - m - n$ hypercube constraints are binding. Moreover, since only one such constraints of each pair can be binding, at least $mn - m - n$ different variables are 0 or 1. In case $n = 2$ we apply the same reasonoing, but before we rewrite the LP and replace each x_2^k with $1 - x_1^k$, hence only two constraints remain. As for the general SEARCH COST MINIMIZATION problem, recall that we have n constraints involving the objective variable u, and $2n$ rank constraints for each observation $k = 1, \ldots, m$. $\diamond$

6 Open Questions

A. Specific open questions:
Applying the "binding constraints argument" to other LP formulations we get further combinatorial bounds, as in the previous section. Theorem 5 gave at most n rankings for each of the m observations. From the LP with m constraints (1) and the n constraints expressing that u is at least the failure probability for each target, we get that at most $m+n$ rankings in total (for all m observations) appear with positive probabilities in some optimal strategy. However, since this LP has $mn!$ variables for the probabilities of rankings, it is not clear whether we can still compute such a strategy in polynomial time.

According to the proof of Lemma 4, only unmatched (i.e., distinct) c_i and x_j create rankings with positive probabilities. Also remember that $x_j = \sum_{r=1}^n c_r z_{rj}$. Together with considerations as in Theorem 8, this may lead to fewer rankings in polynomial time.

Anyway, we always have optimal strategies with a quite low combinatorial complexity: Most hypotheses or rankings are chosen with probability 0 or 1, and the ranks of most hypotheses are decided deterministically. A valuable aspect of this very deterministic structure of optimal solutions is some robustness against changes in the conditional probabilities p_{kj} which are in practice only estimated and not accurately known.

Are our combinatorial bounds optimal? E.g., is there an example where n rankings are needed in Theorem 5? This may also depend on the cost sequence.

How difficult is it to compute an optimal strategy which uses also the smallest possible number of rankings for the given instance?

Our polynomial-size linear program for Search Cost Minimization has a special structure. Is there an algorithm for this class of linear programs (e.g., based on network flows) that runs faster than a generic linear programming algorithm ?

As for probing ranked hypotheses with unreliable tests, is the competitive ratio $1 + 2\sqrt{\epsilon} + O(\epsilon)$ in Theorem 1 already optimal?

B. More general and far-reaching questions:

The problem formulation in Section 1 does, of course, not cover every application, and some extensions would be natural. For search problems with multiple targets we have to generalize the linear program if we want accurate cost assumptions. Another more generalization would be able to assign different costs to different hypotheses rather than ranks. Also, some information about likely and unlikely hypotheses might be incorporated, still without using fixed prior probabilities. However, the challenge for every extended model is to find polynomial-time algorithms for constructing optimal solutions.

Strategies for the recovery of noisy strings could be studied systematically for various important types of formal languages, and the approach to spelling correction could be further developed and tested.

Acknowledgement

I would like to thank the anonymous referees for detailed comments and discussion that helped to sort out several aspects of the model.

References

[1] R.A. Baeza-Yates, J.C. Culberson, G.J.E. Rawlins. Searching in the plane, *Information and Computation* 106 (1993), 234-252

[2] K.P. Bennett, E. Parrado-Hernndez. The interplay of optimization and machine learning research, *J. of Machine Learning Research* 7 (2006), 1265-1281 (editorial of Special Topic on Machine Learning and Large-Scale Optimization)

[3] A. Bergkvist, P. Damaschke, M. Lüthi. Linear programs for hypotheses selection in probabilistic inference models, *J. of Machine Learning Research* 7 (2006), 1339-1355

[4] G. Birkhoff. Three observations on linear algebra, *Univ. Nac. Tucumn. Rev. Ser. A* 5 (1946), 147-151

[5] E. Brill, R.C. Moore. An improved error model for noisy channel spelling correction, *38th Annual Meeting of the Association for Comp. Linguistics ACL 2000*, 286-293

[6] D. de Caen, L.A. Székely. The maximum size of 4- and 6-cycle free bipartite graphs on m, n vertices, *Colloq. Math. Soc. Janos Bolyai* 60, North-Holland 1992, 135-142

[7] P. Damaschke. Scheduling search procedures, *J. of Scheduling* 7 (2004), 349-364

[8] P. Damaschke. Scheduling search procedures: The wheel of fortune, *J. of Scheduling* 9 (2006), 545-557

[9] J.P. Doignon, S. Fiorini, G. Joret. Facets of the linear ordering polytope: a unification for the fence family through weighted graphs, *J. of Math. Psychology* 50 (2006), 251-262

[10] P. Erdös, R.J. Faudree, C.C. Rousseau, R.H. Schelp. The number of cycle lengths in graphs of given minimum degree and girth, *Discrete Math.* 200 (1999), 55-60

[11] S. Gal. On the optimality of a simple strategy for searching graphs, *Int. J. of Game Theory* 29 (2001), 533-542

[12] M.Y. Kao, M.L. Littman. Algorithms for informed cows, *AAAI-97 Workshop on On-Line Search*, 1997

[13] M.Y. Kao, Y. Ma, M. Sipser, Y. Yin, Optimal constructions of hybrid algorithms, *J. of Algorithms 29* (1998), 142-164

[14] M.Y. Kao, J.H. Reif, S.R. Tate. Searching an unknown environment: an optimal randomized algorithm for the cow-path problem, *Information and Computation* 131 (1996), 63-79

[15] A. López-Ortiz and S. Schuierer. The ultimate strategy to search on m rays?, *Theoret. Computer Science* 261 (2001), 267-295

[16] A. López-Ortiz and S. Schuierer. On-line parallel heuristics, processor scheduling and robot searching under the competitive framework, *Theoret. Computer Science* 310 (2004), 527-537

[17] A. Naor, J. Verstraete. A note on bipartite graphs without $2k$-cycles, *Probability, Combinatorics and Computing* 14 (2005), 845-849

[18] B. v. Stengel, R. Werchner. Complexity of searching an immobile hider in a graph, *Discrete Applied Math.* 78 (1997), 235-249

[19] K. Toutanova, R.C. Moore. Pronunciation modeling for improved spelling correction, *40th Annual Meeting of the Association for Comp. Linguistics ACL 2002*, 144-151