

Change Detection in Heuristic Search

Eyke Hüllermeier

IRIT - Université Paul Sabatier
31062 Toulouse Cedex (France)
eyke@irit.fr

Abstract

The order in which nodes are explored in a (depth-first) iterative deepening search strategy is principally determined by the condition under which a path of the search tree is cut off in each search phase. A corresponding criterion, which has a strong influence on the performance of the overall (heuristic) search procedure, is generally realized in the form of an upper cost bound. In this paper, we develop an effective and computationally efficient termination criterion based on statistical methods of change detection. The criterion is local in the sense that it depends on properties of a path itself, rather than on the comparison with other paths. Loosely speaking, the idea is to take a systematic change in the (heuristic) evaluation of nodes along a search path as an indication of suboptimality. An expected utility criterion which also takes the consequence of the suboptimal search decision on the solution quality into account is proposed as a generalization of this idea.

Introduction

Heuristic search strategies (for single-agent path-finding problems) explore more promising paths of a search tree (or, more generally, a search graph) before less promising ones by evaluating search states, thereby putting the successors of an inner node in some order according to their desirability. A frequently used type of (numeric) evaluation function is of the form $f(\eta) = g(\eta) + h(\eta)$, where $g(\eta)$ denotes the cost of the path from the root (initial state) to the node η and $h(\eta)$ is an estimation of the cost, $h^*(\eta)$, of the shortest (= lowest-cost) path from η to some goal state. The value $h(\eta)$ is generally derived from some “features” of the search state associated with η . Considerable attention has been paid to so-called *admissible* heuristics which underestimate the true cost function h^* , i.e., which satisfy $h(\eta) \leq h^*(\eta)$ for all nodes η . Such heuristics are of interest since they allow for realizing search algorithms which are guaranteed to return an optimal (lowest-cost) solution.

One possibility of deriving admissible heuristics in a systematic way is to use simplified (less constrained) models of the original problem (Pearl 1984). Still, the discovery of good admissible heuristics remains a difficult task, let

alone the automatization of this process (Prieditis 1993). Besides, the computation of relaxation-based heuristic functions might be expensive since it calls for solving new, even though simplified, problems. In this paper, we assume heuristic information which does not provide lower bounds but estimations of cost values in a proper (statistical) sense. The (data-oriented) acquisition of this type of heuristic function seems to be less difficult than the (knowledge-oriented) invention of admissible heuristics. For instance, statistical techniques can be used for approximating the cost function h^* from a set of training data (Hüllermeier 1998). Of course, when basing a (best-first) search strategy on inadmissible heuristics, the (first) solution found might not be optimal. Nevertheless, corresponding evaluation functions are often more accurate, and using them in lieu of admissible ones may yield a considerable improvement in (average) time complexity at the cost of an acceptable (perhaps provably bounded (Harris 1974)) deterioration of solution quality.

It is well-known that (depth-first) iterative deepening (ID) search can overcome the main limitations of the basic breadth-first (BFS) and depth-first (DFS) search strategies, namely the exponential space complexity of the former and the non-optimality and non-completeness of the latter (Korf 1985). The criterion used by ID for cutting off search paths is generally given in the form of a global depth-limit or cost-limit. In this paper, we propose alternative termination criteria based on statistical methods of *change detection*. Loosely speaking, the idea is not to use the evaluation of individual (frontier) nodes itself but the *change* in the node evaluation along a search path in order to detect (and cut off) hardly promising paths. The paper is organized as follows: We begin with explaining the problem of change detection in its general form. Then, some basic algorithms for approaching this problem are briefly reviewed. The application of change detection in the context of heuristic search is discussed afterwards. Finally, a generalized approach to search termination based on an expected utility criterion is proposed. The paper concludes with some remarks.

The Problem of Change Detection

The problem of change detection is understood as the detection of *abrupt changes* in some characteristic properties of a system. Generally, such characteristics are not observed directly (e.g., through corresponding sensors) but can only

be inferred indirectly from the measurements available. The problem of change detection arises, e.g., in pattern recognition (segmentation of signals), quality control, monitoring in biomedicine, or in connection with the detection of faults in technological processes (Basseville & Nikiforov 1993).

Many problems of this type can be stated as one of detecting a change in the *parameters* of a (static or dynamic) stochastic model. One possibility of approaching change detection is hence from the viewpoint of mathematical statistics. Consider, for instance, a (discrete) sequence $(Y_t)_{1 \leq t \leq T}$ of random variables, where Y_t is characterized by the (parametrized) conditional probability (density) $\phi_\theta(\cdot | y_{t-1}, \dots, y_1)$. We suppose that a change is reflected by the change of the parameter vector θ . Thus, assuming that the random variables are independent of each other and that at most one change has occurred, we have

$$\phi(y_t) = \begin{cases} \phi_{\theta_0}(y_t) & \text{if } 1 \leq t < t_0 \\ \phi_{\theta_1}(y_t) & \text{if } t_0 \leq t \leq T \end{cases},$$

where t_0 denotes the change time. The problem of change detection can comprise different tasks: (a) Detecting a change as soon as possible, i.e., deciding at each point of time whether the sequence observed so far contains a change or not. (b) Estimating the change time t_0 , if any. (c) Estimating the (possibly unknown) parameters θ_0 and θ_1 . These problems can be formalized within the framework of mathematical statistics. The first task, for instance, can be considered as a problem of hypotheses testing: The hypothesis H_0 that $\theta(t) = \theta_0$ for $1 \leq t \leq T$, where $\theta(t)$ denotes the parameter at time t , is tested against the alternative, H_1 , that

$$\begin{aligned} \exists 1 < t_0 \leq T : \theta(1) = \dots = \theta(t_0 - 1) = \theta_0 \\ \wedge \theta(t_0) = \dots = \theta(T) = \theta_1. \end{aligned}$$

(This test assumes θ_0 and θ_1 to be known.) Likewise, the second task can be approached as a problem of estimating a (discrete) parameter, namely t_0 . Note that the second and third task can be considered as auxiliary problems, once the first question has been answered in favor of H_1 .

Different performance measures for assessing change detection procedures exist. Even though the importance of a criterion depends on the respective application, the following criteria are relevant in any case: The *expected delay* of a detection is the expected number of time points between t_0 and the time when the change is detected. The *probability of a false detection* is the probability of deciding on H_1 even though H_0 is true. These criteria are irreconcilable, of course: Reducing the probability of a false detection will generally increase the expected delay and vice versa.

Algorithms for Change Detection

In this section, we review one of the basic procedures for change detection, the so-called cumulative sum (CUSUM) algorithm. It is based upon the repeated use of the sequential probability ratio test (SPRT). The CUSUM algorithm was first proposed in (Page 1954).

The Sequential Probability Ratio Test

The SPRT, investigated in depth by WALD (1947), is one of the most important test procedures in sequential statistical

analysis, mainly due to its optimality properties (Wald & Wolfowitz 1948) and its computational efficiency. It is a test for deciding between two simple statistical hypotheses

$$H_0 : \theta = \theta_0, \quad H_1 : \theta = \theta_1 \quad (1)$$

about some (constant) parameter θ . Consider a sequence $(Y_t)_{t \geq 1}$ of independent and identically distributed random variables, and let ϕ_{θ_0} and ϕ_{θ_1} denote the probability (density) function of Y_t valid under H_0 and H_1 , respectively. Moreover, suppose that this sequence has been observed up to time T . A value of the log-likelihood ratio (the logarithm of the likelihood ratio)

$$\mathcal{L}_T = \sum_{t=1}^T \ln(\phi_{\theta_1}(y_t)) - \ln(\phi_{\theta_0}(y_t)) \quad (2)$$

larger (smaller) than 0 can then be interpreted as an indication of the validity of H_1 (H_0). An intuitive decision rule is hence to accept H_0 , if \mathcal{L}_T falls below some threshold $\alpha < 0$, and to reject H_0 (i.e., to decide on H_1), if \mathcal{L}_T exceeds a threshold $\beta > 0$. The SPRT realizes this idea by combining a stopping rule and a terminal decision rule:

```
while  $\alpha < \mathcal{L}_T < \beta$ 
  continue sampling ( $T = T + 1$ )
if  $\mathcal{L}_T \leq \alpha$  then accept  $H_0$ 
if  $\beta \leq \mathcal{L}_T$  then reject  $H_0$ 
```

Observe that the log-likelihood ratio can be written recursively as $\mathcal{L}_T = \mathcal{L}_{T-1} + z_T$, where $z_T = \ln(\phi_{\theta_1}(y_T)) - \ln(\phi_{\theta_0}(y_T))$. Loosely speaking, the SPRT takes observations as long as evidence in favor of either H_0 or H_1 is not convincing enough. The thresholds α and β are directly related to the probabilities of two types of errors, namely of falsely rejecting H_0 and of deciding in favor of H_0 even though H_1 is true. Indeed, α and β can be used for controlling these errors. Of course, a tradeoff between a small probability of an error and the quickness of a decision, i.e., the length of the observed sequence, has to be achieved: The smaller (larger) α (β) is, the smaller is the probability of an error. At the same time, however, the terminal decision will be delayed, i.e., the *exit time*, T_0 , will be larger.

The CUSUM Algorithm

The SPRT assumes that the complete data collected during the test is generated by one and the same model. Thus, it has to be modified in the context of change detection, where the data-generating process is assumed to change in-between.

Again, we are interested in testing the hypotheses (1) repeatedly. Now, however, H_0 is definitely true at the beginning, and the sampling is stopped only if a change seems to have occurred, i.e., if the test statistic exceeds an upper bound. In order to avoid a detection delay due to the fact that (2) decreases as long as H_0 holds true it is hence reasonable to restrict the accumulated evidence in favor of H_0 . This can be achieved by letting α define a lower bound to the test statistic. When choosing $\alpha = 0$ (Lorden 1971), the CUSUM algorithm can be written in its basic form as follows:

```

g0 = 0, T = 0
repeat until gT ≥ β
  T = T + 1
  gT = max{0, gT-1 + ln(φθ1(yT)) - ln(φθ0(yT))}

```

If the CUSUM algorithm has terminated at time T_0 , the maximum likelihood (ML) estimation of the change time t_0 is given by

$$\hat{t}_0 = \arg \max_{1 < t \leq T_0} \sum_{k=t}^{T_0} \ln(\phi_{\theta_1}(y_k)) - \ln(\phi_{\theta_0}(y_k)).$$

Often, not only the change time t_0 is unknown but also the parameters θ_0 and θ_1 . A standard statistical approach, then, is to use the corresponding ML estimations instead. This leads to

$$\begin{aligned} g_T &= \max_{1 < t_0 \leq T} \mathcal{L}_{t_0}(\hat{\theta}_{0,t_0}, \hat{\theta}_{1,t_0}) \\ &= \max_{1 < t_0 \leq T} \sum_{t=t_0}^T \ln(\phi_{\hat{\theta}_{1,t_0}}(y_t)) - \ln(\phi_{\hat{\theta}_{0,t_0}}(y_t)), \end{aligned} \quad (3)$$

where $\hat{\theta}_{0,t_0}$ and $\hat{\theta}_{1,t_0}$ denote, respectively, the ML estimations of θ_0 and θ_1 based on the observations y_1, \dots, y_T and the assumption that t_0 is a change time. $\mathcal{L}_{t_0}(\theta_0, \theta_1)$ is the log-likelihood ratio associated with the hypotheses

$$\begin{aligned} H_0 &: \forall 1 \leq t \leq T : \theta(t) = \theta_0, \\ H_1 &: \theta(t) = \begin{cases} \theta_0 & \text{if } 1 \leq t \leq t_0 - 1 \\ \theta_1 & \text{if } t_0 \leq t \leq T \end{cases}. \end{aligned}$$

Again, the stopping rule is defined by $g_T \geq \beta$, with β being a predefined threshold. That is, the exit time T_0 is defined as the smallest T such that $g_T \geq \beta$. The (ML) estimation of t_0 is then given by the change time $1 < \hat{t}_0 \leq T_0$ for which the maximum in (3) is attained. Observe that g_T , as defined in (3), can no longer be determined by means of a simple recursive formula.

Changes in Heuristic Search

In this section, we shall apply the idea of change detection to heuristic tree-search algorithms. We assume heuristic information to be available in the form of an estimation h of the cost function h^* which assigns to each node η the cost of the (cost-)optimal path from η to a goal state. A value $h(\eta)$ is considered as the realization of a random variable $H(\eta)$ (Hansson & Mayer 1989). We suppose the random variables associated with different search states to be independent of each other, a simplifying assumption commonly made in the probabilistic analysis of search trees (Zhang & Korf 1995). For the sake of simplicity, we also assume $E(H(\eta)) = h^*(\eta)$ and $(H(\eta) - h^*(\eta)) \sim \Phi$ for all nodes η , where E denotes the expected value operator.¹ Statistically speaking, $h(\eta)$ is an unbiased estimation of $h^*(\eta)$. Φ is a distribution with associated probability (density) function ϕ , i.e., $\phi(x)$ is the probability (density) that $H(\eta) - h^*(\eta) = x$.

¹This assumption can easily be relaxed.

The overall cost associated with a node η is given by $f(\eta) = g(\eta) + h(\eta)$, where $g(\eta)$ denotes the cost of the path from the initial state to η . The evaluation function f is used for guiding the search process. According to our assumptions above, $f(\eta)$ can be interpreted as the realization of a random variable $F(\eta)$ such that $(F(\eta) - g(\eta) - h^*(\eta)) \sim \Phi$.

Path Profiles

Consider a search path ρ , i.e., a sequence (η_1, \dots, η_K) of nodes with η_1 being the root of the search tree and η_K the (current) frontier node. We associate sequences

$$\begin{aligned} f(\rho) &= (f(\eta_1), \dots, f(\eta_K)), \\ f^*(\rho) &= (f^*(\eta_1), \dots, f^*(\eta_K)) \end{aligned}$$

with ρ , where $f^*(\eta_k) = g(\eta_k) + h^*(\eta_k)$. $f(\rho)$ is called the *path profile* of ρ . The sequence $f^*(\rho)$ is obviously non-decreasing, i.e.,

$$f^*(\eta_1) \leq f^*(\eta_2) \leq \dots \leq f^*(\eta_K). \quad (4)$$

Suppose that we have not made a suboptimal search decision (node generation) so far, i.e., a cost-optimal solution can still be reached from η_K . Then, all inequalities in (4) are in fact equalities. That is, the random variables $F(\eta_k)$ have the same expectation and, hence, the same distribution. Now, suppose all but the k th decision to be optimal. Then,

$$\begin{aligned} \mu_0 &= f^*(\eta_1) = f^*(\eta_2) = \dots = f^*(\eta_k) \\ &< f^*(\eta_{k+1}) = f^*(\eta_{k+2}) = \dots = f^*(\eta_K) = \mu_1. \end{aligned} \quad (5)$$

That is, the expected value of the random variables $F(\eta_1), \dots, F(\eta_k)$ is μ_0 , whereas the expectation of $F(\eta_{k+1}), \dots, F(\eta_K)$ is μ_1 . More generally, we have the case that some (but at least one) of the inequalities in (4) are strict if not all decisions have been optimal. Subsequently, we assume that at most one suboptimal decision has been made.²

Change Detection and Search Termination

Iterative deepening search is the method of choice for many applications since it combines the merits of both, BFS (completeness, or even optimality) and DFS (linear space complexity). Taking a repeated (partially informed) depth-first search as a point of departure, the ID algorithm is principally determined by the conditions under which search along a path is broken off. The common approach is to use a depth-limit or, more generally, a cost-limit, for each search phase. For instance, the iterative deepening version of the A* algorithm, IDA*, continues search along a path as long as the frontier node η satisfies $f(\eta) \leq c$, with c being the f -cost-limit valid for the respective search phase.

When being interested in finding good solutions, or even optimal ones, one should obviously avoid the exploration of suboptimal search paths. Indeed, this idea is supported by the gradually increased cost-limit in IDA*. When being also interested in minimizing computational effort, it is

²Experimental studies have shown that a generalization of this assumption hardly improves the quality of search decisions.

reasonable, not only to *terminate* search along less promising paths, but also to *continue* search along a path as long as it appears promising, thereby leaving other (perhaps also promising) paths out of account. This idea is actually not supported by the use of an f -cost-limit which brings about a permanent comparison between a (large) set of simultaneously explored paths.³ In fact, “following a search path as long as it might lead to a (near-)optimal solution” principally requires a *local* termination criterion which makes do with (heuristic) information from nodes of an *individual* path. Here, we shall use statistical methods of change detection in order to realize corresponding termination criteria. As will be seen, these methods allow for estimating the quality of a solution eventually found when following a path without referring to other search paths, or other solutions already encountered.

Consider a path $\rho = (\eta_1, \dots, \eta_K)$ with path profile $f(\rho) = (f(\eta_1), \dots, f(\eta_K))$. The parameter of interest, i.e., the change of which has to be detected, is the expected value $E(F(\eta_k))$. Depending on the information available about μ_0 and μ_1 , the expected values before and after a change, different algorithms can be applied. Generally, it must be assumed that neither μ_0 nor μ_1 is known. Consequently, the basic CUSUM algorithm cannot be used directly. Rather, we have to proceed from (3). Suppose $1 < k_0 \leq K$ to be a changepoint, i.e., the generation of η_{k_0} was not optimal. The ML estimations $\hat{\mu}_{0,k_0}$ and $\hat{\mu}_{1,k_0}$ of μ_0 and μ_1 are then given by the mean of the values $f(\eta_1), \dots, f(\eta_{k_0-1})$ and $f(\eta_{k_0}), \dots, f(\eta_K)$, respectively. The test statistic (3) thus becomes

$$\begin{aligned} \mathcal{L}(\rho) &= \max_{1 < k_0 \leq K} \mathcal{L}_{k_0}(\rho) \\ &= \max_{1 < k_0 \leq K} \sum_{k=k_0}^K \ln(\phi_{\hat{\mu}_{1,k_0}}(f(\eta_k))) - \ln(\phi_{\hat{\mu}_{0,k_0}}(f(\eta_k))), \end{aligned}$$

where ϕ_{μ} denotes the probability (density) function of the random variable $X + \mu$ and $X \sim \Phi$. Observe that we should actually use the *constrained* ML estimations of μ_0 and μ_1 , taking into account that $\mu_0 \leq \mu_1$. In fact, the case where $\hat{\mu}_{0,k_0} > \hat{\mu}_{1,k_0}$ indicates an (actually impossible) decrease of the expectation and can hence be ignored. The termination criterion should thus be defined as

$$\mathcal{L}'(\rho) = \max_{1 < k_0 \leq K} \mathcal{L}'_{k_0}(\rho) \geq \beta, \quad (6)$$

where $\mathcal{L}'_{k_0}(\rho) = \mathcal{L}_{k_0}(\rho)$ if $\hat{\mu}_{0,k_0} \leq \hat{\mu}_{1,k_0}$ and 0 otherwise, and $\beta > 0$ is a predefined threshold. $\mathcal{L}'(\rho)$ can be computed in time $O(K)$ by the algorithm in Figure 1.

An Iterative Deepening Algorithm

When making use of the above termination criterion, a search algorithm cuts of the current path and starts a backtracking whenever (6) is satisfied (or a terminal node has been reached). As already mentioned before, the threshold β can be used for controlling two (conflicting) criteria: The

```

function change( $f(\eta_1), \dots, f(\eta_K)$ )
   $m(1) = f(\eta_1)$ 
  for  $k = 2$  to  $K$ 
     $m(k) = m(k-1) + f(\eta_k)$ 
  for  $k_0 = 2$  to  $K$ 
     $\hat{\mu}_{0,k_0} = m(k_0-1)/(k_0-1)$ 
     $\hat{\mu}_{1,k_0} = (m(K) - m(k_0-1))/(K - k_0 + 1)$ 
   $\mathcal{L}' = 0, \hat{k}_0 = 0, s = 0$ 
  for  $k_0 = K$  down to 2
     $s = s + \ln(\phi_{\hat{\mu}_{1,k_0}}(f(\eta_{k_0}))) - \ln(\phi_{\hat{\mu}_{0,k_0}}(f(\eta_{k_0})))$ 
    if  $\hat{\mu}_{0,k_0} \leq \hat{\mu}_{1,k_0}$  and  $\mathcal{L}' < s$  then
       $\mathcal{L}' = s, \hat{k}_0 = k_0$ 
  return  $\mathcal{L}', \hat{k}_0, \hat{\mu}_{0,\hat{k}_0}$ 

```

Figure 1: Pseudo-code for computing the termination criterion (6) and the ML estimations \hat{k}_0 and $\hat{\mu}_0$.

expected delay of a change detection corresponds to the expected number of nodes explored along a path after a suboptimal search decision has been made. Of course, the smaller the threshold β is, the smaller is the detection delay. The probability of a *false detection* corresponds to the probability of cutting off an optimal search path. Keeping this probability small calls for a large threshold β .

Choosing $\beta = \infty$ leads to a pure depth-first search and, hence, excludes the probability of a false detection completely. In general, however, it may happen that an optimal search path is cut off and the corresponding search process terminates before having found a solution. Suppose, for instance, that a change has occurred at $k_0 = 11$. The following table shows the (experimentally determined) probability p of terminating before $k = 11$ and the expected detection delay, d , for different thresholds β if Φ is the standard normal distribution and $\mu_1 = \mu_0 + 2$:

β	p	d	β	p	d
5	0.48	1.51	8	0.28	2.55
6	0.39	1.86	9	0.25	2.85
7	0.33	2.20	10	0.22	3.17

If the search process terminates without a solution, a new search phase with a larger threshold (which might be defined on the basis of the search process so far) has to be started. Based on an increasing sequence $(\beta_i)_{i \geq 0}$ of thresholds we thus obtain an iterative deepening algorithm, IDCD (Iterative Deepening based on Change Detection).

Of course, the first solution found by IDCD is not necessarily an optimal one. In fact, using the kind of heuristic information available does generally not allow for guaranteeing optimality, since the true cost value of a search node might not only be underestimated but also overestimated. It seems hence reasonable to let IDCD return the first solution found. Since the order in which solutions are encountered depends on the order in which (promising) search paths are

³Still, the depth-first component in A* can be strengthened by using a weighted evaluation function (Pohl 1970).

explored, the successors of inner nodes should be arranged according to their f -evaluations. This way, a partially informed backtracking is realized in each search phase.

An interesting question concerns the completeness of IDCD. It can be shown that IDCD terminates and returns a solution with probability 1 under rather general conditions. This result can be proved by making use of a related termination property of the SPRT (Wald 1947). However, the proof becomes non-trivial due to the fact that one has to consider an infinite number of search paths.

It is worth mentioning that the change detection criterion suggests some kind of (probabilistic) backmarking strategy (Gaschnig 1979): Instead of returning to the closest unexpanded ancestor, one may back up to the parent of the most likely changepoint \hat{k}_0 directly, thereby pruning (large) parts of the search tree. Of course, a corresponding algorithm will in general not guarantee completeness.

Utility-Based Search

The principle underlying IDCD is to continue a search path only if it might lead to an optimal solution. Now, suppose that we consider a *comprehensive* value of a computation (Horvitz 1988) including the solution quality as well as the running time, and that we are willing to gain efficiency at the cost of solution quality. In other words, we are interested in finding a “good” solution with reasonable computational effort, rather than finding an optimal one regardless of the running time. In this section, we shall propose a generalization of (6) suitable for supporting search under these assumptions.

Suppose the degree to which a user is satisfied with a solution, x , to depend on some relation between the cost of x and the cost of an optimal solution, say, the difference $\Delta(x)$ between these values. The preferences of the user can then be formalized by means of a (non-increasing) utility function $U : [0, \infty) \rightarrow [0, 1]$, where $U(\Delta(x))$ is the utility of a solution x .

Given the preferences of the user thus defined, a reasonable generalization of the termination criterion (6) is to break off search along a path $\rho = (\eta_1, \dots, \eta_K)$ if the *expected utility* of the best solution reachable from η_K , $x^*(\rho)$, falls below some threshold u_0 , i.e., if

$$E(U(\rho)) = \int_0^\infty U(\Delta) \cdot \pi(\Delta | f(\rho)) d\Delta < u_0. \quad (7)$$

Here $\pi(\Delta | f(\rho))$ denotes the probability (density) that $\Delta(x^*(\rho)) = \Delta$, given the path profile $f(\rho)$.

Suppose prior knowledge concerning Δ to be available in the form of a (prior) probability (density) function π . The posterior $\pi(\cdot | f(\rho))$ is then specified by

$$\pi(\cdot | f(\rho)) \propto \lambda(f(\rho) | \cdot) \times \pi, \quad (8)$$

where $\lambda(f(\rho) | \Delta)$ is the probability of observing $f(\rho)$, given the deviation Δ . If we take π as an uninformative prior,⁴ then $\pi(\cdot | f(\rho))$ is proportional to the likelihood func-

tion $\lambda(f(\rho) | \cdot)$, i.e., $\pi(\cdot | f(\rho)) \propto \lambda(f(\rho) | \cdot)$, where

$$\lambda(f(\rho) | \Delta) = \prod_{k=1}^{k_0-1} \phi_{\mu_0}(f(\eta_k)) \prod_{k=k_0}^K \phi_{\mu_0+\Delta}(f(\eta_k)). \quad (9)$$

The posterior $\pi(\cdot | f(\rho))$ cannot be derived from (8) directly if k_0 and μ_0 in (9) are unknown. In Bayesian analysis, such parameters are called *nuisance parameters*. There are different possibilities of deriving a posterior distribution for the parameters of interest in the presence of nuisance parameters (Basu 1977). One might consider, e.g., the marginal distribution of the full posterior distribution which contains both, the parameters of interest as well as the nuisance parameters. This solution may become computationally expensive, however. A further possibility is to replace the nuisance parameters by their ML estimations. This approach seems reasonable in our situation since we can fall back on the results obtained in connection with the detection of changes: Both, the estimation of the change point, \hat{k}_0 , and of the corresponding expectation, $\hat{\mu}_0$, are computed by the algorithm in Figure 1. A (utility-based) termination criterion can thus be realized as follows:

- Derive \hat{k}_0 and $\hat{\mu}_0$ for the current search path ρ .
- Let $E(U(\rho)) = 1$ if $\hat{k}_0 = 0$. Otherwise, derive $E(U(\rho))$ according to (7), (8) and (9).
- Break off the search path if $E(U(\rho)) < u_0$, with $u_0 \leq 1$ being a predefined threshold.

This procedure reveals that the expected utility criterion can be seen as an extension of the change detection approach: Instead of only looking for a jump in the expected value, the decision whether to terminate search or not also depends on the size of this jump. The other way round, (6) corresponds to the special case of (7) in which the utility function is given by $U = 1_{\{1\}}$.

Observe that $\pi(\cdot | f(\rho))$ is not necessarily 0 on $(-\infty, 0)$ since (9) might be positive (even though small) also for $\Delta < 0$. Instead of incorporating the constraint $\Delta \geq 0$ by means of the usual conditioning (proportional allocation of probability mass) it seems reasonable to take the (actually impossible) event $\Delta < 0$ as evidence for $\Delta = 0$. Formally, this is equivalent to integrating over $(-\infty, \infty)$ in (7) with an extended utility function such that $U(\Delta) = U(0) = 1$ for all $\Delta < 0$.

By making use of (7) in conjunction with a sequence of utility thresholds $(u_i)_{i \geq 0}$ instead of (6) (and the sequence $(\beta_i)_{i \geq 0}$), we obtain an iterative deepening algorithm UBID (Utility-Based Iterative Deepening). Again, the search strategy is not only controlled by (7) but also by $(u_i)_{i \geq 0}$. A threshold $u_i = 0$, for instance, entails a pure depth-first search. The smaller the increments $\Delta u_i = u_i - u_{i-1}$ are defined, the more cautious search paths are explored, i.e., the more UBID resembles an (iterative deepening) best-first search.

Of course, since the expected utility criterion requires the computation of the integral (which is a sum in the case of a discrete distribution Φ) in (7) it is computationally more complex than the change detection criterion. Still, it can be

⁴Suppose $\pi(\cdot | f(\rho))$ to exist under this condition.

realized very efficiently for several special cases. For instance, if Φ is a normal distribution⁵ and U corresponds to an (extended) $\{0, 1\}$ -valued utility function $1_{(-\infty, \Delta_0]}$, the computation of (7) can be realized by simply looking up a value (which depends on $\hat{\mu}_0, \hat{\mu}_1, \hat{k}_0$) in the table of the (cumulative) standard normal distribution (Hüllermeier & Zimmermann 1998). Let us also mention a further possibility of avoiding integration, namely that of basing the termination rule not on the *expected* but on the *most likely* utility $U(\hat{\mu}_1 - \hat{\mu}_0)$.

Concluding Remarks

We have proposed an iterative deepening strategy in which the decision whether to terminate or continue search along a path is based on properties of the sequence of node evaluations along that path. The basic idea is to “search deep” in the first place, due to reasons of efficiency, but only as long as the expected utility of the solution that will eventually be found is acceptable. This (heuristic) search principle has been formalized based on statistical methods of change detection. The corresponding search algorithm is parametrized by means of a utility function and a sequence of utility thresholds.

Taking the idea of a *local* termination rule for granted, the criteria (6) and (7) appear natural in the sense that they *fully* exploit the information which has become available while following a search path. Observe that admissible (but non-monotone) estimations $f(\eta_k)$ along a path can be aggregated by taking their maximum, which is an obvious way of avoiding a loss of information. As opposed to this, a simple combination (such as, e.g., the average) of all evaluations does hardly make sense in our case since the random variables $F(\eta_k)$ might have different expected values.

Due to reasons of space we could not present experimental results in this paper. Still, a successful implementation providing evidence for the efficiency of our approach as well as a comparison with alternative search methods can be found in (Hüllermeier & Zimmermann 1998), where UBID is applied in the context of knowledge-based configuration. The heuristic function used in this implementation is learned from a set of already solved problems by means of linear regression techniques (Hüllermeier 1998).

The possibility of an efficient parallel implementation served as a main motivation for the development of IDCD and UBID. Indeed, these algorithms are particularly suitable for parallel computation methods since (6) and (7) work with information provided by an individual search path alone and, hence, can lead to a substantial reduction of communication costs. Still, one might think of extending these termination rules by including information from other search paths. This way, it would be possible to base a termination decision on both, the relative change of estimated costs and their absolute values. Particularly, global information can be used for deriving better estimations of the cost of an optimal solution (μ_0 in (5)) and, hence, for improving the test statistic (6). In fact, the methods proposed in this paper can be generalized in further directions as well. Prior information about change

points t_0 or deviations Δ , for instance, can be incorporated by means of Bayesian methods or constrained (ML) estimations. Besides, our termination criteria, or variations thereof, might also be useful as stopping rules for selective search in game playing.

Acknowledgements

The author gratefully acknowledges financial support in the form of a TMR research grant funded by the European Commission. Parts of this work have also been supported by the DFG (German Research Association, SFB 376).

References

- Basseville, M., and Nikiforov, I. 1993. *Detection of Abrupt Changes*. Prentice Hall.
- Basu, D. 1977. On the elimination of nuisance parameters. *J. American Stat. Association* 72:355–366.
- Gaschnig, J. 1979. A problem similarity approach to devising heuristics: First results. In *Proc. IJCAI-79*.
- Hansson, O., and Mayer, A. 1989. Heuristic search as evidential reasoning. In *Proc. 5th Workshop on Uncertainty in AI*, 152–161. Morgan Kaufmann.
- Harris, L. 1974. The heuristic search under conditions of error. *Artificial Intelligence* 5(3):217–234.
- Horvitz, E. 1988. Reasoning under varying and uncertain resource constraints. In *Proceedings AAAI-88*.
- Hüllermeier, E., and Zimmermann, C. 1998. A two-phase search method for solving configuration problems. Tech. Rep. tr-rsfb-98-062, Department of Computer Science, University of Paderborn.
- Hüllermeier, E. 1998. Approximating cost functions in resource-based configuration. Tech. Rep. tr-rsfb-98-060, Department of Computer Science, University of Paderborn.
- Korf, R. 1985. Depth-first iterative deepening: An optimal admissible tree search. *Artif. Intell.* 27(1).
- Lorden, G. 1971. Procedures for reacting to a change in distribution. *Annals of Math. Stat.* 42:1897–1908.
- Page, E. 1954. Continuous inspection schemes. *Biometrika* 41:100–115.
- Pearl, J. 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley.
- Pohl, I. 1970. First results on the effect of error in heuristic search. In Meltzer, B., and Michie, D., eds., *Machine Learning* 5. 219–236.
- Prieditis, A. 1993. Machine discovery of effective admissible heuristics. *Machine Learning* 12:117–141.
- Wald, A., and Wolfowitz, J. 1948. Optimum character of the sequential probability ratio test. *Annals of Mathematical Statistics* 19:326–339.
- Wald, A. 1947. *Sequential Analysis*. New York: John Wiley & Sons.
- Zhang, W., and Korf, R. 1995. Performance of linear-space search algorithms. *Artif. Intell.* 79:241–292.

⁵This distribution often applies at least approximately.