The Odds-algorithm based on sequential updating and its performance

F.Thomas Bruss Guy Louchard

April 1, 2008

Outline

Abstract

Let $I_1, I_2, \ldots, I_n$ be independent indicator functions on some probability space $(\Omega, \mathcal{A}, \mathbb{P})$. We suppose that these indicators can be observed sequentially. Further let T be the set of stopping times on $(I_k), k = 1, \ldots, n$ adapted to the increasing filtration $(\mathcal{F}_k)$, where $\mathcal{F}_k = \sigma(I_1, \ldots, I_k)$. The odds-algorithm solves the problem of finding a stopping time $\tau \in T$ which maximizes the probability of stopping on the last $I_k = 1$, if any. To apply the algorithm one needs only the odds for the events $\{I_k = 1\}$, that is $r_k = p_k/(1 - p_k)$, where $p_k = \mathbb{E}(I_k), k = 1, 2, \ldots, n$, or at least a certain number of them. The goal of this work is to offer tractable solutions for the case where the p_k are unknown and must be sequentially estimated.

The motivation is that this case is important for many real word applications of optimal stopping. We study several approaches to incorporate sequential information in the algorithm. Our main result is a new version of the odds-algorithm based on online observation and sequential updating. Questions of speed and performance of the different approaches are studied in detail, and the comparisons are conclusive so that we propose to always use this algorithm to tackle selection problems of this kind.

Introduction

Let $I_1, I_2, \ldots$ be independent indicator functions on some probability space $(\Omega, \mathcal{A}, \mathbb{P})$ with $p_k = \mathbb{E}(I_k)$. Further let

$$q_k = 1 - p_k, r_k = p_k/q_k,$$

that is r_k presents the odds of the event $\{I_k = 1\}$. We may observe the indicators sequentially and may stop on at most one, but only *online*, that is, at the moment of observation. We win if we stop on the *last* $I_k = 1$ (if any) and lose otherwise (including not stopping at all). Formally, let T be the set of non-anticipative stopping rules defined by $T = \{\tau : \{\tau = k\} \in \mathcal{F}_k\}$, where $\mathcal{F}_k$ is the $\sigma-$algebra generated by $I_1, I_2, \ldots, I_k$. The odds-theorem of optimal stopping (Bruss [2]), determines the rule which maximizes the probability of stopping on the last indicator which takes the value one (if any). This solution is conveniently computed via the odds-algorithm described in the following Algorithm.

odds-algorithm

Input: define

$$R_k := r_n + r_{n-1} + \cdots + r_k, k = 1..n,$$
$$Q_k := q_n q_{n-1} \cdots q_k, k = 1..n,$$

and precompute

$$s = s(n) = \begin{cases} 1, & \text{if } R_1 < 1 \\ \sup\{k : R_k \geq 1\}, & \text{otherwise.} \end{cases} \quad (1)$$

Output: optimal stopping rule.
The optimal stopping rule to stop on the last "1" is: stop on the first indicator I_k with $I_k = 1$ and $k \geq s$. If none exists, stop on n and lose.

We say that we "win" if the algorithm stops on the last 1. The optimal win probability (as seen at time $1, 2, \ldots, s-1$) equals $R_s Q_s$. The odds-algorithm is very convenient and allows for many interesting applications as e.g. selection problems for randomly arriving objects, timing problems, buying and selling problems and clinical trials, automated maintenance problems and others. (Bruss [2], [4], Tamaki [13], and Iung et al. [8]). The algorithm can also be adapted to continuous time decision processes with Poisson arrivals (see [2]). Related problems have been studied by Suchwalko and Szajowski [11], Szajowski [12] and Kurushima and Ano [9].

A particular feature of the odds-algorithm is that the number of computational steps to find the optimal rule is (sub)-linear in n. The algorithm is optimal itself, in the sense that clearly no algorithm can exists which would in general yield the rule with less than $\mathcal{O}(n)$ computations. It yields the optimal rule and the optimal win probability at the same time, and is optimal itself.

A related problem to the problem of stopping on the last event $\{I_k = 1\}$ is the problem of stopping with maximum probability on the kth last indicator which equals 1. (see Bruss and Paindaveine [6]). The precise solution is more complicated but a slight modification of the odds-algorithm gives a good approximation. A harder related problem is the problem of stopping on a last specific pattern in an independent sequence of variables taken frome some finite or infinite alphabet as studied by Bruss and Louchard [5]. In these problems, the p_k are supposed to be known.

Unknown odds

The applicability of the above odds-algorithm is somewhat restricted, because in many practical applications, the decision maker would not know beforehand the values p_k, at least not precisely.

The corresponding optimal stopping problem for unknown p_k is now in general much harder. In some cases , the precise solution can be given, and this also within the framework of the odds-algorithm (see Van Lokeren [10]), but these cases are very special. In this work we study the problem in more generality.

Note that we cannot give too much freedom to the randomness of the p_k, because, if we allow, as we typically do, the p_k to be different from each other, they must be still estimable. More precisely, the odds $r_{k+1}, r_{k+2}, \ldots, r_n$ must be estimable from $I_1, I_2, \ldots, I_k$. This means that the number of unknown parameters on which the p_k (and thus the r_k) may depend, must stay very small compared with n. Since n is, in many important applications, as for instance the compassionate use-clinical trial example (see Bruss [4]), itself not large (10 or 15 say) we focus our interest in this work on only one unknown parameter, p say. Hence the p_k are thought of as being deterministic function of one unknown parameter p.

The model $p_k = pf_k$

This is our main model. The parameter p is unknown but the factor f_k is supposed to be known. This is an adequate setting for many problems. In the mentioned clinical-trial example, for instance, p is considered as the unknown success probability for a medical treatment and f_k is a factor (between 0 and 1) which reduces the success probability for the kth patient according to his or her state of health.

The idea is to combine the convenience of the odds-algorithm with the concurrent task of estimating the "future odds" from preceding observations. We will study both the case of a Bayesian setting with a prior for the unknown parameter p as well as the case of a completely unknown p. Both cases are well-motivated. If a new type of practical problem is encountered, one has sometimes so little information that one should not take the risk of introducing a bias by a prior distribution. However, with some confirmed prior information, the Bayesian setting has typically the advantage of leading to more efficient estimators.

Let thus $(f_k), k = 1, \ldots, n$ be a sequence of known real non-negative values. We put

$$p_k = pf_k, p \in [0, 1], pf_k \leq 1.$$

Here it is understood that if we suppose a support $[a, b]$ for the distribution of p other than $[0, 1]$, then the f_k may range between 0 and $1/b$, that is f_k is not necessarily a reducing factor, but may also increase the intrinsic success probability.

Fixed p

Let

$$p_k = pf_k,$$
$$q_k := 1 - pf_k,$$

and

$$r_k := \frac{pf_k}{1 - pf_k},$$

that is r_k is the (unknown) odds for $\{I_k = 1\}$. If $I_k = 1$ we say that a success occurs at time k. Further let

$$I_k(p) := [\![\text{ success occurs at time } k]\!].^1$$

It is easy to see that

$$\mathbb{E}\left[\sum_1^s I_k(p)\right] = p \sum_1^s f_k,$$

$$\mathbb{V}\left[\sum_{1}^{s} I_k(p)\right] = p\sum_{1}^{s} f_k(1-pf_k) = V_1(s),\ \text{say}\ ,$$

where $\mathbb{V}$ denotes the variance. The odds-algorithm gives

$$s^* = \sup\left\{s : \sum_{s}^{n} r_k \geq 1\right\},\ \text{if}\ \left\{s : \sum_{s}^{n} r_k \geq 1\right\} \neq \emptyset,$$
$$1,\ \text{otherwise}$$

We should write $s^*(n)$, but here and in the sequel, we drop the n to simplify the notation (when there is no ambiguity).

Hence s^* is the time index from which onwards it is optimal to stop on the first event $I_k = 1$, and the corresponding optimal win probability equals $R_{s^*} Q_{s^*}$ (see the odds-algorithm). Here, of course, R_s and Q_s are functions of p and $f_1, \ldots, f_s$. We think of the f_k as being fixed and write

$$s^* = \varphi(p),$$

and

$$\psi(s, p) = \prod_s^n q_l \sum_s^n r_l.$$

Hence, the optimal success probability for a given p is given by

$$\psi^*(p) = \psi(s^*, p). \tag{2}$$

Sequential estimation

We use as an estimator of p

$$\hat{p}(s,p) = \frac{\sum_1^s I_k(p)}{\sum_1^s f_k}, \tag{3}$$

and this for two reasons: first, $\hat{p}(s,p)$ is an unbiased estimator of p. Indeed,

$$\mathbb{E}(\hat{p}(s,p)) = \frac{\sum_1^s \mathbb{E}(I_k(p))}{\sum_1^s f_k} = p.$$

$$\mathbb{V}(\hat{p}(s,p)) = V_1(s) \Bigg/ \left[\sum_1^s f_k\right]^2 .$$

Secondly, this estimator is efficient for constant f_k, that is it has the smallest possible variance, as one can readily show using the Fisher-information and Rao-Cramer's bound. We note however that (3) is in general not a maximum likelihood estimator of p, as one can easily check. This is why we offer also an alternative approach later on.

Let us consider the distribution for $\hat{p}$ for index s and parameter p both fixed. We denote it by

$$\hat{P}(\rho|s,p) := \mathbb{P}[\hat{p}(s,p) = \rho].$$

One can see that $\hat{P}(\rho|s,p)$ becomes the Binomial distribution, if the f_k are constant. In the general case, it can be numerically computed by extracting the coefficients from the generating function

$$G_s(z) := \prod_1^s [pf_i z + 1 - pf_i]. \tag{4}$$

We get

$$\mathbb{P}\left[\hat{p}(s,p) = \frac{\nu}{\sum_1^s f_k}\right] = [z^\nu]G_s(z),$$

where $[z^n]f(z)$ denotes the coefficient of z^n in the power expansion of $f(z)$.

The distribution of the number of successes

Let

$$\nu(s) := \sum_{k=1}^{s} I_k = \# \text{ successes up to time } s.$$

We note that $\nu(s)$ follows no well-known distribution unless the f_k are constant. However, we can construct a tractable recurrence relation for the law of $\nu(s)$ from $G_s(z)$ as given by (4). We obtain a recurrence to compute $\{\mathbb{P}(\nu(s) = m\}_{m=0,1,\ldots,s}$, namely

$$\mathbb{P}(\nu(s) = m) = \frac{1}{m} \sum_{k=0}^{m-1} \mathbb{P}(\nu(s) = k)(-1)^{m-1-k} \sum_{j=1}^{s} r_j^{m-k}.$$

with initial condition

$$\mathbb{P}(\nu(s) = 0) = q_1.q_2 \ldots .q_s.$$

The proof is given in the full report, where we also briefly investigate a stopping rule based on sequential maximum likelihood.

Qualitative assessment

Let us now discuss the intrinsic weakness of any approach based on sequential estimation.
If $\hat{p}(s, p)$ is small at the beginning (no events $\{I_k = 1\}$ at the beginning), the stopping threshold s is also small and we could consequently stop too early. It is true that we only can stop on a success, so that $\hat{p}$ jumps up at each such instance. This reduces the risk of under-estimation. However, it does not exclude it.
Similarly, if we wait some time before we compute and use $\hat{p}(s, p)$, and if p is small, we could stop too late.

As an alternative we may decide to use some fixed learning samples and never to stop on the first $s_d - 1$ values, that is, we start the algorithm at $s = s_d$. Here $s_d = 1$ corresponds to the classical algorithm with no delay.

The question of an optimal delay s_d will be analyzed later on.

The odds-algorithm for the stopping threshold s leads to the equation

$$\varphi(\hat{p}(s, p)) \leq s.$$

The threshold computation procedure is given in the following Algorithm.

odds-algorithm with sequential estimation of odds

Input: precompute the optimal delay s_d (if we use a delay)
Output: an optimal stopping threshold s

$s := s_d$;$cont := true$
while $cont$ **do**
  $\nu := \sum_1^s I_k, \hat{p}(s) = \frac{\nu}{\sum_1^s f_k}$
  if $\sum_{s+1}^n r_k(\hat{p}(s)) < 1$ **then**
    $cont := false$
  else
    $s := s + 1$
    if $s = n$ **then**
      $cont := false$
    end if
  end if
end while
return s

Winning probability

s is a random variable with some distribution $\phi(s, p)$, say.
Fix p. For each time s, the possible values of the random variable $\nu := \sum_1^s I_k$ satisfying

$$\varphi\left(\frac{\nu}{\sum_1^s f_k}\right) \leq s$$

are constrained to stay in an interval denoted by

$$[0, \gamma[s]].$$

In order to stop at any case not later than n, we set $\gamma[n] = n$. In the case of delaying, we just put $\gamma[s] = -1, s = 1..s_d - 1$.
ν is represented by a Markov chain. In the following we drop the p parameter to ease the notation. Let

$$\Pi[s, \mu] := \mathbb{P}[\nu = \mu, \text{ no stopping threshold } \textit{before } s].$$

Then,

$$\Pi[1,1] = pf_1, \Pi[1,0] = 1 - pf_1, \phi[1,p] = \sum_{\mu=0}^{\gamma[1]} \Pi[1,\mu]$$

and, for $s \geq 2$,

$$\Pi[s,\mu] = \Pi[s-1,\mu-1]pf_s[\![\mu \neq 0 \wedge \mu - 1 > \gamma[s-1]]\!]$$
$$+ \Pi[s-1,\mu](1-pf_s)[\![\mu > \gamma[s-1]]\!].$$

The stopping threshold probability distribution is now given by

$$\phi(s,p) = \sum_{\mu=0}^{\gamma[s]} \Pi[s,\mu].$$

Finally

$$\mathbb{P}(\text{win}) = \mathbb{P}(\text{ algorithm stops on the last } 1|p) = \sum_{s=1}^{n} \phi(s,p)\psi(s,p) = \Theta(p$$

Choice of f_k and n.

In the examples given in this paper, we use two different choices for the sequence (f_k). One is $f_k = 1$ for all k. This is a natural choice for the case when all I_k are i.i.d. Bernoulli random variables. We could also have used $f_k = C$ for some fixed constant $0 < C \leq 1$. Our second and most frequent choice is $f_k = 1/k$. One reason is that we want to cover the case when all odds are different. Besides this, there is nothing really special about this choice except that it solves a new version of a well-known best-choice problem, that is the secretary problem with *unknown* availability probability. Indeed, suppose that in a sequence of candidates, all are equally likely the best, second best, and so on, that the kth candidate is available, independent of his rank, with probability p. Then this candidate is best so far *and* available with probability p/k. See also Ano et al. [1].

Examples

We usually use the sample size $n = 15$, but again there is nothing special about this choice and most graphs would look similar for n not too small. Clearly small n like $n \leq 6$, say lead to unreliable odds-estimates and hence to bad performance.

The following Figure 1 gives $\Theta(p)$ as a function of p, for $s_d = 1..5$. We have choosen $n = 15$, $f_k = 1/k$ (These parameters will always be used in the sequel). The circle graph gives $\psi^*(p)$, the horizontal line represents $1/e$. The relevance of a comparison with $1/e$ will be explained below.

Figure 1

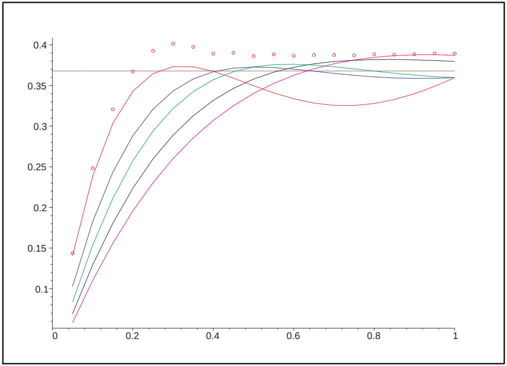

Figure 1: $\Theta(p)$ as a function of p, for $s_d = 1..5$, from red to magenta, $n = 15$, $f_k = 1/k$, $k = 1, \dots, n$, circle : $\psi^*(p)$, horizontal line : $1/e$

Note that $\Theta(p)$ possesses a local maximum and a local minimum for some values of s_d. This can be explained as follows: when p is small, the chance of having ones is small, and hence the total win probability is small for any strategy. Since the estimated odds are very likely to be small as well for small p, the risk of a wrong decision by the odds strategy is also small simply because stopping on the very first 1 (if any) is the best to do. But for growing p this risk increases in the middle range of p so that the total win probability goes somewhat down before getting the full benefit of large success probabilities. The difference between the local maximum and the local minimum is of course also dependent of the choice of the f_k. A more detailed approach is given later on.

There is a good reason why the comparison of the performance of this algorithm with the value $1/e$ is the most adequate one. Indeed, it was shown that if the odds are known and if their sum is at least 1, then $1/e$ is the exact lower bound for the win probability over all such sequences $p_1, p_2, \dots, p_n$. (Bruss [3]). But, moreover, if n becomes large and if $\sum_{k=1}^{n} p_k^2 / \sum_{k=1}^{n} p_k \to 0,$ as $n \to \infty$, then the win probability converges to $1/e$ (Bruss [2]). Hence, in particular if the sum of all odds is at least one, then it suffices that $p_k \to 0$, as $k \to \infty$. We finally observe that, for any value of p, there is an optimal value of s_d. This will be useful later on.

Unknown p according to a distribution $P(p)$

We now suppose that the unknown parameter p follows a distribution $P(p)$, which is unknown to the decision maker. Let $\Theta(p)$ denote, as before, the conditional probability of winning for a given p. The absolute win probability using our algorithm is then given by

$$Pw := \mathbb{P}(\text{ win }) = \int_0^1 P(p)\Theta(p)dp.$$

There is no statistical inference on p other than using the sequential estimator (3). The only focus is the impact of delaying as a function of the distribution $P(p)$. Statistical inference based on a (known) prior distribution of p will be used in the Bayesian approch in Section 7.

Examples.

1) As a first example, we let $P(p)$ be given by a parabola on $[0..1]$, with maximum occuring respectively at $p_m \in [1/16, 1/8, 1/4, 1/2, 3/4, 7/8]$. The parabola is starting at the origin for $p_m = 1/16, 1/2, 3/4, 7/8$, and landing at 0 for $p = 1$ for $p_m = 1/8, 1/4$. For $p_m = 1/8$, for instance, we give, in Figure 2, Pw as function of the delay parameter s_d.

Figure 2

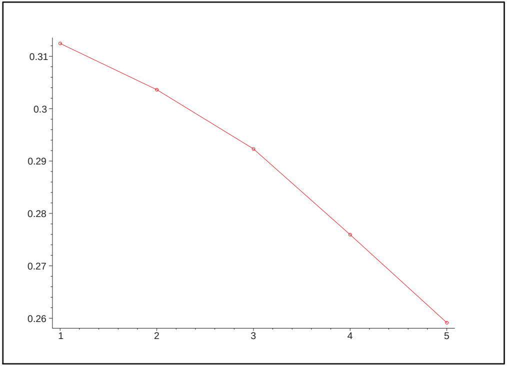

Figure 2: Pw as function of $s_d, f_k = 1/k, p_m = 1/8$

We see that nothing is gained by delaying stopping. The situation is the same for $p_m = 1/16, 1/4$. However, for $p_m \in [1/2, 3/4, 7/8]$, we see that it is better to ignore the first event. We see this, in Figure 3, for $p_m = 7/8$, where Pw is plotted as function of the delay parameter s_d.

Figure 3

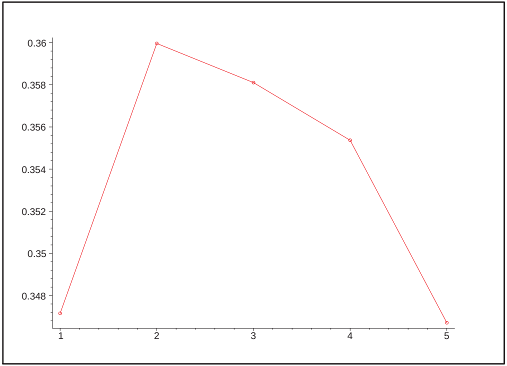

Figure 3: Pw as function of $s_d, p_m = 7/8$

The optimal s_d values, for our six parabolae, are given by $[1, 1, 1, 2, 2, 2]$. We see that these optimal values are rather robust: a minimal information on the shape of $P(p)$ is enough to choose s_d.

2) As an example of large s_d, we have computed Pw with a linear $P(p) = 2p$. This leads to $s_d = 4$.

In the case of sequential updating, we will denote by $Pw(p_m)$ the success probability, without delay, and by $Pw_{opt}(p_m)$ the success probability, with optimal delay, for our six parabola distributions.

If we know p beforehand, we must use $\psi^*(p)$, this leads to

$$Pw^* = \int_0^1 P(p)\psi^*(p)dp.$$

Algorithm cost

The computational cost of the odds-algorithm with sequential updating depends essentially on the computation of $\hat{p}(s)$ and on the instruction: if $\sum_{s+1}^{n} r_k(\hat{p}(s)) < 1$. Assuming for simplicity, that each numerical operation costs 1 unit, we have, at time s, a cost of

$$C(s,p) = \sum_{1}^{s}(n - v + 1) = (n + 1/2)s - s^2/2. \qquad (5)$$

and

$$C'_s(s,p) = n + 1/2 - s \geq 0.$$

The mean cost $M(p)$ is now given by

$$M(p) = \sum_{1}^{n} \phi(s,p)[(n + 1/2)s - s^2/2] = (n + 1/2)\bar{s} - \bar{s^2}/2,$$

$$\bar{s^i} := \sum_{1}^{n} \phi(s,p)s^i.$$

Similarly the second moment is given by

$$M^{(2)}(p) = \sum_{1}^{n} \phi(s,p)[(n+1/2)s - s^2/2]^2$$
$$= \bar{s^4}/4 - (n+1/2)\bar{s^3} + (n+1/2)^2\bar{s^2},$$

and the variance

$$\mathbb{V}(p) = M^{(2)}(p) - M(p)^2.$$

Cost distribution

The cost distribution is itself computed as follows. For fixed p, we have

$$
\begin{aligned}
C(1,p) &= n, \\
C(2,p) &= 2n-1, \\
C(3,p) &= 3n-3, \\
&\vdots \\
C(n,p) &= n(n+1)/2.
\end{aligned}
$$

For each cost C (if this value is possible), the corresponding value of s is given by

$$s = [(2n+1) - \sqrt{(2n+1)^2 - 8C}]/2.$$

This allows, with $\phi(s,p)$, the computation of the cost distribution $H(p)$. Figure 4 gives $H(1/2)$ for our usual parameters. But in this case, only three values of s lead to non-null values of $\phi(s,p)$, which explains the shape of $H(1/2)$.

Figure 4

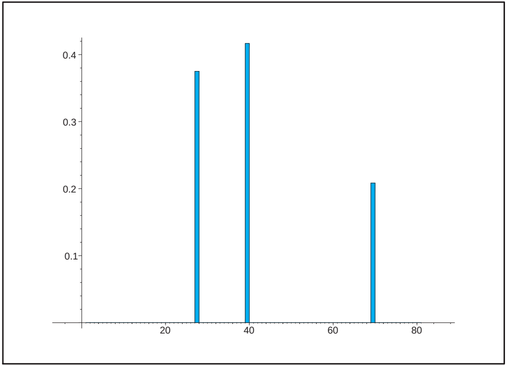

Figure 4: Cost distribution $H(1/2), s_d = 1$

Asymptotic behaviour of cost, $n \longrightarrow \infty$

To study the asymptotic behaviour of the cost, as $n \longrightarrow \infty$, we must distinguish between two cases
i) if $\sum_1^\infty f_k$ converges and $\sum_1^\infty f_k/(1-f_k) > 1$ (otherwize we always stop at $s=1$), we have, for each p, a maximum $s^*(p)$ such that $\sum_{s^*}^\infty r_k(p) \geq 1$, and asymptotically, $\varphi(p)$ is independent of n. $\phi(s,p)$ also becomes independent of n and we have a cost given by (5), which is linear in n. Also, setting

$$\hat{s} := \sup\{j : \sum_j^\infty f_k/(1-f_k) \geq 1\},$$

we have $s^*(p) \leq \hat{s}$ and

$$C(s,p) \leq (n+1/2)\hat{s} - \hat{s}^2/2.$$

ii) if $\sum_1^\infty f_k$ diverges, $\varphi(p)$ is close to n, $\phi(s,p)$ gives a maximum weight in the neighborhood of n, and the cost is now of the order of n^2.
For instance, if $f_k = 1$, the odds-algorithm gives $s \sim n - q/p$ and

$$C(s,p) \sim n^2/2,$$

if $f_k = 1/k$, we have $s \sim ne^{-1/p}$ and

$$C(s,p) \sim [e^{-1/p} - e^{-2/p}/2]n^2.$$

The asymptotic behaviour of $\psi^*(p, n)$, $f_k = 1/k$

We can show that, asymptotically, $\psi^*(p, n)$ possesses a unique maximum at some critical point $p^*(n)$ and, to the right of it, a unique minimum, using a continuous equivalent. The proof is given in the full report.
For $n = 100$, we have constructed a plot of the continuous version of $\psi^*(p, n)$ on the whole range $p \in [0, 1]$, given in Figure 5. This function is continuous, but its derivative is not. We also compare it with the discrete expression for $\psi^*(p, n)$

Figure 5

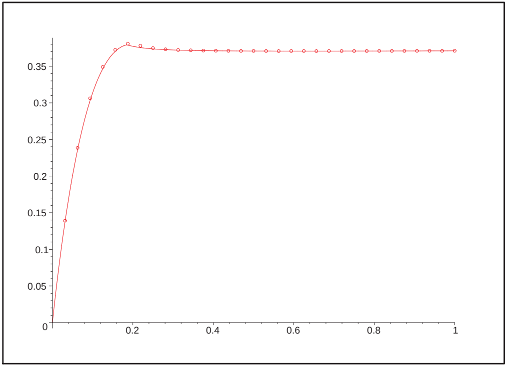

Figure 5: $\psi^*(p, n)$ (continuous version, line) versus $\psi^*(p, n)$ (discrete version, circle), $n = 100$

The fit is quite good, given that we used Euler-Maclaurin in the continuous approach, with one error term, a continuous $s^*(p, n)$ instead of the discrete one , and a not too large value for n. Note that Figure 5 has a similar behaviour as Figure 1 for $\Theta(p)$, in the sequential updating approach. In Figure 1, the difference between maximum and minimum is even more pronounced.

For large n, the difference between exact (continuous) expressions and first order asymptotics (neglecting $\mathcal{O}(1/n)$ errors) become negligible. We confine our interest to the difference between the discrete and the continuous approach.

To give a better view of the minimum, Figure 6 gives, on the right of $p^*(n)$, the same comparison.

Figure 6

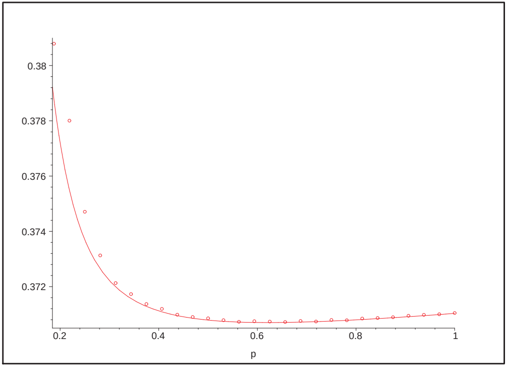

Figure 6: $\psi^*(p, n)$ (continuous version, line) versus $\psi^*(p, n)$ (discrete version, circle), $n = 100, p \geq p^*(n), f_k = 1/k$

Let us note that the optimal s_d values, for our six parabolae, $n = 100$, are given by $[1, 3, 4, 6, 13, 13]$. Again these optimal values are rather robust.
We note that it would be hard to prove existence and unicity of $\min, \max$ in the discrete case as well as for $\Theta(p)$.

Bayesian approach-The theory

We follow in this approach the work of Van Lokeren [10] (Mémoire de DEA, under supervision of F.T. Bruss, unpublished). The problem is as before, that is maximizing the probability of stopping on the last success. Allowing the different success parameters $p_1, \ldots, p_n$ to vary independently of each other leads to an ill-posed problem. Therefore we make the following assumptions.
Let p be a random variable taking values in $[0,1]$ and let $\Psi : [0,1] \times \mathbb{N} \to [0,1]$ be a deterministic (known) function. We assume the success parameter p_k to be given by

$$p_k = \Psi(p, k)$$

Furthermore, we suppose
(i) The conditional law of I_k, given $p = x$, is a Bernoulli law with known success parameter $\Psi(x, k)$.
(ii) The random variables $I_1, I_2, \ldots, I_n$ are conditionally independent, given $p = x$.
The general solution is given in the full report.

The algorithm for the Bayesian approach

The algorithm deals with a vector $a[1..n]$ of bits. We convert this vector into an integer $I = \sum_1^n a[i]2^{i-1}$ with the procedure $I := conv_1(a)$. Similarly, for any I, we compute the corresponding vector a with a procedure $a := conv_2(I)$. Then, according to [10], we compute the two matrices $C[0..n, 0..2^n - 1]$ and $V[1..n, 0..2^n - 1]$ with the following formulae

$C[0,0] := 1$; for i to $2^n - 1$ do $C[0,i] := 0$ od;

for k to n do for l from 0 to $2^n - 1$ do

$a := conv_2(l)$;

and, in general,

$$C[k,l] := \int_0^1 \prod_1^k (xf_i)^{a[i]}[1-(xf_i)]^{1-a[i]} P(x)dx; \text{ od ; od ;}$$

for l from 0 to $2^n - 1$ do $V[1, l] := C[n, l + 2^{n-1}]/C[n-1, l]$; od

for k from $n - 2$ by -1 to 0 do

for l from 0 to $2^k - 1$ do

$A := C[n, l + 2^k]/C[k+1, l + 2^k]$;

$B := V[n - k - 1, l + 2^k]$;

$T := \max(A, B)$;

$$V[n-k, l] := C[k+1, l]/C[k, l]V[n-k-1, l] + C[k+1, l+2^k]/C[k, l]T; \text{ od ; od ;}$$

Finally, the Bayesian optimal value is given by

$$Pw^B = V[n, 0].$$

The practical procedure is given in in the following Algorithm.

Optimal strategy

Input: precompute C, V
Output: an optimal stategy
set $a[k] := I_k, k = 1..n$
Stop at the first I_k for which $I_k = 1$ and, with
$l_k := conv_1(a[1..k]), \frac{C[n,l_k]}{C[k,l_k]} \geq V[n-k, l_k]$
Stop at I_n if the above conditions are not fulfilled for any
$1 \leq k \leq n-1$.

We have computed, with our five parabola distributions, the success probability given by the Bayesian approach: $Pw^B(p_m)$. Figure 7 gives ($n = 15$) $Pw^*(p_m), Pw^B(p_m), Pw(p_m), Pw_{opt}(p_m)$. Pw^* gives naturally the best result. The other ones are comparable, with a slight advantage for Pw^B and Pw_{opt}, but Pw is rather close. Note that for some values of p_m, the value Pw^B is better than Pw_{opt}, but the the opposite is true for other values of p_m.

Figure 7

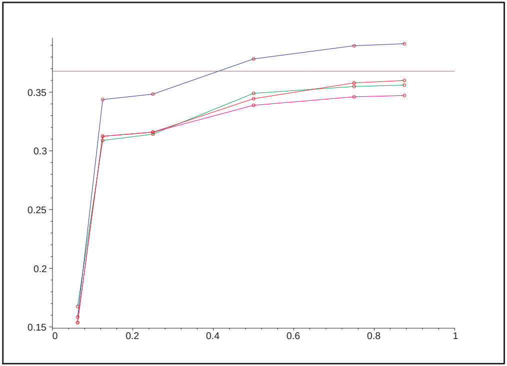

Figure 7: $Pw^*(p_m)$ (blue), $Pw^B(p_m)$ (green), $Pw(p_m)$ (magenta), $Pw_{opt}(p_m)$ (red), $p_m \in [1/16, 1/8, 1/4, 1/2, 3/4, 7/8], n = 15, f_k = 1/k$

Case $f_k = 1$

The case $f_k = 1$ for all $k = 1, 2, \ldots, n$ is the simplest interesting special case. The following Figure 8 gives $\Theta(p)$ as a function of p, for $s_d = 1..14$ and $n = 15$. The circle graph displays $\psi^*(p)$, the horizontal line represents $1/e$. If we compare this graph with Figure 1, we see that the maximum and minimum are more pronounced (at least for small values of s_d).

Figure 8

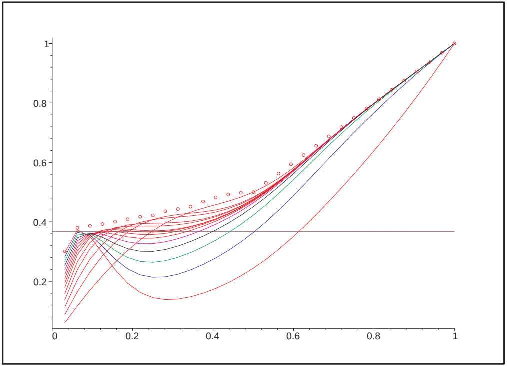

Figure 8: $\Theta(p)$ as a function of p, for $s_d = 1..14$, from red to magenta then red, $n = 15, f_k = 1, k = 1, \ldots, n$, circle : $\psi^*(p)$, horizontal line : $1/e$

The delay analysis shows that, for $p_m = 1/16$, no delay is necessary, but for $p_m = 1/8$ already, we have an optimal $s_d = 10$. The optimal s_d values, for our six parabolae, are given by $[1, 10, 10, 11, 12, 12]$. Again, these optimal values are rather robust: a minimal information on the shape of $P(p)$ is enough to choose s_d.

Figure 9 displays $H(1/2)$ (see Section 5), and Figure 10 displays $Pw^*(p_m), Pw^B(p_m), Pw(p_m), Pw_{opt}(p_m)$. Again, Pw^* gives the best result, but its advantage is less pronounced. Pw^B and Pw_{opt} are rather close to each other. Pw is definitively bad.

Figure 9

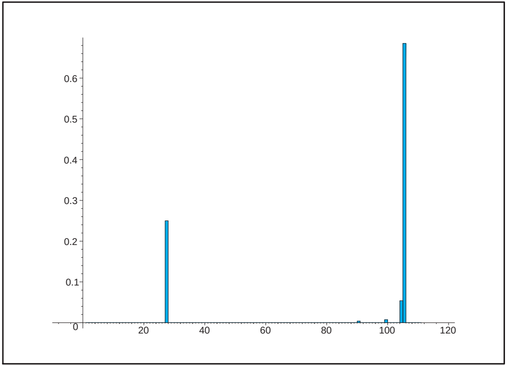

Figure 9: Cost distribution $H(1/2), s_d = 1$

Figure 10

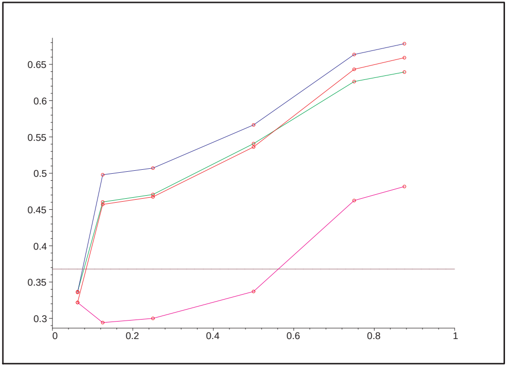

Figure 10: $Pw^*(p_m)$ (blue), $Pw^B(p_m)$ (green), $Pw(p_m)$ (magenta), $Pw_{opt}(p_m)$ (red),
$p_m \in [1/16, 1/8, 1/4, 1/2, 3/4, 7/8]$, $n = 15, f_k = 1$

Of course, we could compute an equivalent continuous analysis of $\psi^*(p, n)$, as we did previously, but we will not pursue this matter in this work.

Conclusion

The solution of the problem of maximizing the probability of stopping on a last success in a sequence of independent indicators has many real-world applications, ranging from best choice problems (secretary problems) over buying-selling strategies up to applications in sequential search and clinical trials. If the odds are known in advance, the odds-algorithm provides this solution in a straightforward way, and this algorithm is, itself, optimal. If the odds are unknown, and must be estimated from preceeding observations, then the optimal rule is not obvious and can be made explicit in special cases only. The objective of this work was to examine the question wether there are good approximations for the optimal rule. We have proposed an algorithm which is based on the odds-algorithm and on a simple unbiased sequential estimator of the successes probabilities $p_k = \mathbb{P}(I_k = 1)$. Although we have no precise estimates by how much it misses optimality, we have established several importants facts.

First it is asymptotically optimal, because as $n \to \infty$, the sequential estimators of the odds will converge, in our model, to the true odds, and we know that for the true odds the odds-algoritm gives the optimal solution.
Secondly, its cost compares well with that of the more complicated decision rule obtained by maximal likelihood estimates. Anyway, the maximal likelihood algorithm should not be better than the optimal algorithm, leading to Pw^*, and we have seen that our algorithm compares favourably with it.

Thirdly, a comparison is given with decision rules based on the Bayesian model. Here again the computational cost is uncomparably higher, but the result is not uniformly better. We can now sumarize our conclusions. Taking all arguments together, we would suggest to always use the odds-algorithm with sequential updating based on the estimator defined in (3). With some additional information we may somewhat improve on this by a slight delay factor s_d as explained before. Note also that this algorithm, working with a number of computations which is at most quadratic in n, stands out from a computational point of view.

K. Ano, M. Tamaki, and M. Hu.
A secretary problem with uncertain employment when the number of offers is restricted.
Journal of the Operation Research Society of Japan, 39:307–315, 1996.

F.T. Bruss.
Sum the odds to one and stop.
Annals of Probability, 28(3):1384–1391, 2000.

F.T. Bruss.
A note on the odds-theorem of optimal stopping.
Annals of Probability, 31(4):1859–1861, 2003.

F.T. Bruss.
The art of a right decision: Why decision makers may want to know the odds-algorithm.
Newsletter of the European Mathematical Society, 62:14–20, 2006.

F.T. Bruss and G. Louchard.
Optimal stopping on patterns in strings generated by independent random variables.
Journal of Applied Probability, 40:49–72, 2003.

F.T. Bruss and D. Paindaveine.
Selecting a sequence of last successes in independent trials.
Journal of Applied Probability, 37:389–399, 2000.

R. L. Graham, D. E. Knuth, and O. Patashnik.
Concrete Mathematics (Second Edition).
Addison Wesley, 1994.

B. Iung, E. Levrat, and E. Thomas.
Odds-algorithm - based opportunistic maintenance task execution for preserving product conditions.
CIRP-Annals, 56(1):13–16, 2007.

A. Kurishima and K. Ano.

A note on the full-information Poisson arrival selection problem.
Journal of Applied Probability, 40(4):1147–1154, 2003.

M. Van Lokeren.
DEA Mémoire en statistique, Université libre de Bruxelles.
2007.

A. Suchwalko and K. Szajowski.
On the Bruss stopping problem with general gain function.
Game Theory and Applications, 9:161–171, 2003.

K. Szajowski.
A game version of the Cowan-Zabczyk-Bruss problem.
Statistics and Probability Letters, 77:1683–1689, 2007.

M. Tamaki.
Optimal stopping on trajectories and the ballot theorem.
Journal of Applied Probability, 38(4):946–959, 2001.