Make-or-break: chasing risky goals or settling for safe rewards

Pantelis P. Analytis[1*], **Charley M. Wu**[2], **Alexandros Gelastopoulos**[3]

[1]Danish Institute of Advanced Study, University of Southern Denmark
[2]Center for Adaptive Rationality, Max Planck Institute for Human Development, Berlin
[3]Department of Mathematics and Statistics, Boston University

April 26, 2019

1 Expected rewards of make-or-break tasks are a sigmoid function of time

Recall that the expected reward from investing time t_m in the make-or-break task is

$$\mathbb{E}[r_m(t_m)] = B \cdot \left(1 - \Phi\left(\frac{\Delta - \lambda_m t_m}{\sigma_m \sqrt{t_m}}\right)\right). \tag{1}$$

We are now going to show that this is a sigmoid function for $t_m > 0$, meaning that $\mathbb{E}[r_m(t_m)]$ converges to a number $c \in \mathbb{R}$ as $t_m \to \infty$ and that it has a unique inflection point, with its second derivative being positive on the left and negative on the right of this point. The first statement is straightforward, because as $t_m \to \infty$, $\frac{\Delta - \lambda_m t_m}{\sigma_m \sqrt{t_m}} \to -\infty$, so $\mathbb{E}(r_m(t_m)) \to B$.

For the second statement, we write

$$\mathbb{E}[r_m(t_m)] = B \cdot \left(1 - \Phi\left(\frac{\Delta}{\sigma_m} t_m^{-\frac{1}{2}} - \frac{\lambda_m}{\sigma_m} t_m^{\frac{1}{2}}\right)\right) = B \cdot \left(1 - \Phi\left(a t_m^{-\frac{1}{2}} - b t_m^{\frac{1}{2}}\right)\right), \tag{2}$$

where $a = \frac{\Delta}{\lambda_m}$ and $b = \frac{\lambda_m}{\sigma_m}$. Therefore, it is enough to show that the function

$$f(t) = \Phi\left(a t^{-\frac{1}{2}} - b t^{\frac{1}{2}}\right), \;\; a, b > 0 \tag{3}$$

has a unique inflection point for $t > 0$, with its second derivative passing from a negative to a positive value. We have

$$f'(t) = -\frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{2} \cdot \left(a t^{-\frac{1}{2}} - b t^{\frac{1}{2}}\right)^2} \cdot \left(a t^{-\frac{3}{2}} + b t^{-\frac{1}{2}}\right) \tag{4}$$

*Corresponding author. Inquiries related to the paper can be addressed to pantelispa@gmail.com, Danish Institute of Advanced Study, University of Southern Denmark, Campusvej 55, Odense, 5200, Denmark.

and

$$f''(t) = \frac{1}{4\sqrt{2\pi}\cdot t^{\frac{7}{2}}}\cdot e^{-\frac{1}{2}\cdot\left(at^{-\frac{1}{2}}-bt^{\frac{1}{2}}\right)^2}\cdot\left(b^3t^3+\left(ab^2+b\right)t^2+\left(3a-a^2b\right)t-a^3\right) \tag{5}$$

Because the first two factors in the expression for $f''(t)$ are always positive, its sign is determined by the third degree polynomial

$$z(t) = b^3t^3+\left(ab^2+b\right)t^2+\left(3a-a^2b\right)t-a^3. \tag{6}$$

It is therefore enough to show that $z(t)$ has exactly one root for $t>0$ and its sign switches from negative to positive as t crosses that root from left to right.

By further noticing that $z(0)<0$ and $z(t)\to\infty$ as $t\to\infty$, it is sufficient to show that $z(t)$ has at most one local extremum for $t>0$ (which will have to be a local minimum), or that $z'(t)$ has at most one root for $t>0$. We calculate

$$z'(t) = 3b^3t^2+2(ab^2+b)t+3a-a^2b. \tag{7}$$

Even if this equation has two roots, their sum has to be equal to $-\frac{2(ab^2+b)}{3b^3}$, so that at least one has to be negative. This concludes the proof that $\mathbb{E}[r_m(t_m)]$ is a sigmoid function.

2 Optimal time allocation policy for the one-shot problem

In this section we study how the optimal policy for the one-shot allocation task varies with some of the parameters of the problem. We begin by studying some general properties of the total expected reward function, which is given by

$$h(t_m,t_s) = \mathbb{E}[r_m(t_m)]+\mathbb{E}[r_s(t_s)], \tag{8}$$

where t_m is the time invested in the make-or-break task and t_s is the time invested in the safe-reward task. If the total available time is T, then we may write $t_s = T-t_m$, hence the above simplifies to

$$h(t_m) = \mathbb{E}[r_m(t_m)]+\mathbb{E}[r_s(T-t_m)]. \tag{9}$$

The optimal amount of time invested in the make-or-break task is then given by

$$t_{opt} = \underset{0\le t_m\le T}{\arg\max}\{h(t_m)\} \tag{10}$$

and the expected reward for this optimal strategy is $h(t_{opt})$. In case the maximum is attained at more than one point, we interpret arg max to be the first of these points.[1]

Substituting the equations for the expected rewards into Equation 9, we obtain

$$h(t_m) = B\cdot\left(1-\Phi\left(\frac{\Delta-\lambda_m t_m}{\sigma_m\sqrt{t_m}}\right)\right)+v\cdot\lambda_s\cdot(T-t_m), \tag{11}$$

[1] Continuity of $h(t_m)$ guarantees that a "first" such point exists.

for $t_m \in (0,T]$ and $h(0) = v \cdot \lambda_s \cdot T$. For the first derivative we have

$$h'(t_m) = \frac{B}{2\sqrt{2\pi}\cdot\sigma_m} \cdot \left(\lambda_m \cdot t_m^{-\frac{1}{2}} + \Delta \cdot t_m^{-\frac{3}{2}}\right) \cdot e^{-\frac{(\Delta-\lambda_m t_m)^2}{2\sigma_m^2 t_m}} - v \cdot \lambda_s, \tag{12}$$

for $t_m \in (0,T]$ and $h'(0) = -v \cdot \lambda_s$. For the second derivative, we have $h''(t_m) = [\mathbb{E}(r_m(t_m))]'$, which we showed in the previous section has a single root for $t_m > 0$.

Although we are only interested in times $t_m \leq T$, in studying how t_{opt} varies with the parameters T, λ_m, and B, it will be useful to consider the function $h(t_m)$ for any $t_m \geq 0$. We will need the following lemmas:

Lemma 2.1. *The function $h(t_m)$ either has no local extrema on $(0,\infty)$, or it has a unique local minimum at t_{min} and a unique local maximum at t_{max}, with $t_{min} < t_{max}$. In the latter case, its derivative $h'(t_m)$ is strictly positive inside the interval (t_{min}, t_{max}) and strictly negative outside.*

Proof: Note that $h'(0) < 0$. Therefore, if $h(t_m)$ must have a local minimum before any local maximum in $(0,\infty)$. Also, recall that $h''(t_m)$ has at most one root. Therefore, by the Mean Value Theorem (MVT), $h'(t_m)$ cannot have more than two roots. We conclude that $h(t_m)$, if it has any local extrema, has exactly one local minimum at t_{min} and one local maximum at t_{max}, with $t_{min} < t_{max}$, and that $h'(t_m)$ is positive on the interval (t_{min}, t_{max}) and negative on $(0, t_{min})$ and (t_{max}, ∞). □

Lemma 2.2. *When the local extrema of $h(t_m)$ exist, their positions t_{min} and t_{max} vary continuously with the parameters B, σ_m, λ_m, λ_s, v, Δ, and are independent of T.*

Proof: Independence from T follows from the fact that $h'(t_m)$ is independent of T. For the other statement, note that by the MVT, the unique root of $h''(t_m)$ must occur in the interval (t_{min}, t_{max}). This implies that $h''(t_{min}) > 0$ and $h''(t_{max}) < 0$. Therefore, by continuity of $h''(t_m)$ and of the (partial) derivatives of h' with respect to any of the parameters, we conclude that both local extrema positions t_{min} and t_{max} are stable, in the sense that they persist for small changes of the parameters and vary continuously with them. □

Lemma 2.3. *i) If $h(t_m) \geq h(0)$ for some $t_m \in (0,\infty)$, then $h(t_m)$ has a unique local maximum at $t_{max} \in (0,\infty)$, which is also global.*

ii) If $t_{opt} \neq 0$, then the condition in (i) holds and, moreover, $t_{opt} = \min\{T, t_{max}\}$.

Proof:

i) Since $h'(t_m) < 0$ for large t_m, $h(t_m)$ attains a global maximum in $[0,\infty)$. Even if that happens to be at 0, then by assumption it must also be attained somewhere in $(0,\infty)$. This global maximum in $(0,\infty)$ will also be a local maximum, which by Lemma 2.1 is unique.

ii) If $t_{opt} \neq 0$, then by definition there exists some $t_m \in (0,T]$, such that $h(t_m) > h(0)$. From part (i), there exists a unique local maximum $t_{max} \in (0,\infty)$. Clearly, if $t_{max} \leq T$, then $t_{opt} = t_{max}$. If on the other hand $t_{max} > T$, then since $h(t_m)$'s unique local maximum is at t_{max}, it has no local maximum in $(0,T)$, so its maximum in the interval $[0,T]$ is attained either at 0 or at T, which means that $t_{opt} = 0$ or $t_{opt} = T$. By assumption, the first possibility is excluded, therefore $t_{opt} = T$. This concludes the proof that $t_{opt} = \min\{T, t_{max}\}$. □

2.1 Varying the total available time T

Suppose first that we vary the total available time T, while all other parameters are kept fixed. We want to look at how t_{opt} varies. We distinguish two cases.

First suppose that $h(0) \geq h(t_m)$ for all $t_m \in [0,\infty)$. Note from Equation 11 that this condition is independent of T. If it is true, then investing zero time in the make-or-break task is always preferable to investing non-zero time in it. In other words, $t_{opt} = 0$ no matter what the value of T, so this case is trivial.

For the other case, we have the following proposition.

Proposition 2.4. *Suppose that $h(t_m) > h(0)$ for some $t_m > 0$. Then,*

$$t_{opt} = \begin{cases} 0, & \text{if } T \leq T^* \\ T, & \text{if } T^* < T \leq t_{max} \\ t_{max}, & \text{if } T > t_{max} \end{cases}, \tag{13}$$

where t_{max} is the unique global maximum of $h(t_m)$ and T^* is given by

$$T^* = \inf\{t_m \in (0,\infty) : h(t_m) > h(0)\}, \tag{14}$$

and satisfies $T^* < t_{max}$.

Proof: Let T^* be as in Equation 14. By continuity, $h(T^*) = h(0)$, so $t_{opt} = 0$ if and only if $T \leq T^*$. For $T > T^*$, by Lemma 2.1 we have $t_{opt} = \min\{T, t_{max}\}$. Finally, note that since $h(t_{max}) > h(0)$, we have $T^* < t_{max}$. □

2.2 Varying the skill level λ_m

We now study how t_{opt} changes as we vary λ_m, assuming that all other parameters remain constant. In this section, we use the notation $h(t_m;\lambda_m)$ and $t_{opt}(\lambda_m)$ in order to emphasize the dependence of these quantities on λ_m. For $t_m = 0$, $h(0;\lambda_m)$ does not depend on λ_m, so we will write just $h(0)$.

To simplify the presentation of the result, we will assume that when the skill level for the make-or-break task is 0, then the optimal strategy is always to invest all available time into the safe-reward task.[2] Mathematically this means that $h(t_m;0) < h(0)$ for all $t_m > 0$.

We show that when λ_m is smaller than some value λ_m^*, then the optimal policy is to allocate all time to the safe-reward task, that is $t_{opt}(\lambda_m) = 0$. For $\lambda_m > \lambda_m^*$, the optimal policy is to put at least some time into the make-or-break task. Moreover, the transition at λ_m^* is discontinuous, with t_{opt} jumping from 0 to $t^* > 0$. As λ_m increases further, $t_{opt}(\lambda_m)$ will change continuously (but never increase).

The value of t^* can be either T, in which case at λ_m^* there will be a transition from allocating all time to the safe-reward task to allocating all time to the make-or-break task, or it may be smaller than T, in which case the transition will be from allocating all time to the safe-reward task to allocating time to both tasks. In

[2] This does not have to be the case, especially if the reward threshold Δ is low. If it is not true, then we can show that $t_{opt}(\lambda_m)$ is always positive and varies continuously with λ_m.

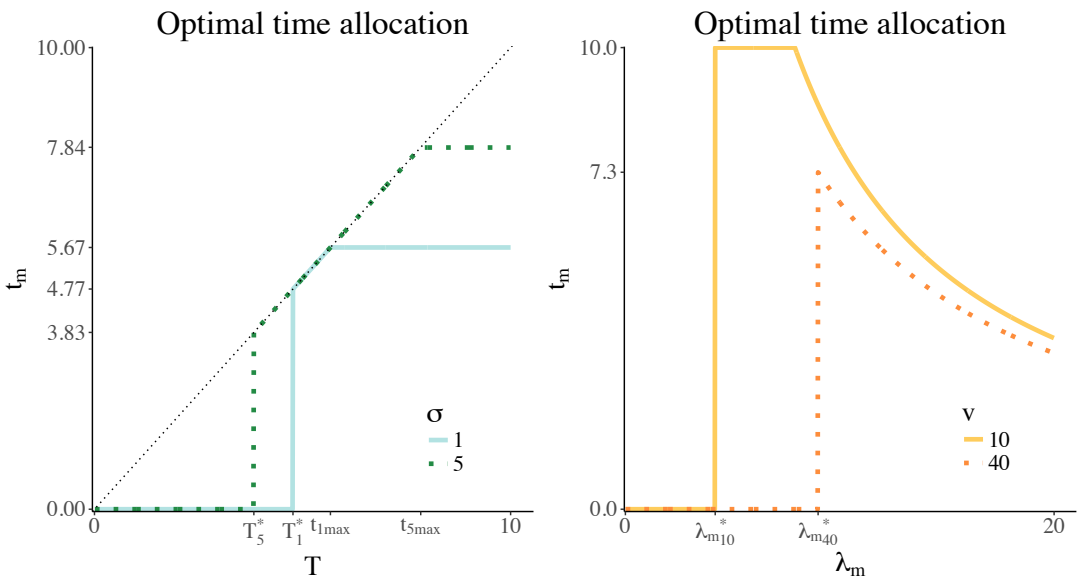

Figure 1: **Left:** The optimal amount of time allocated to the make-or-break task as a function of the total available time $T \in [0,10]$. At very low amounts of total available time (i.e., a tight deadline), the agent allocates all their time to the safe alternative. At T^*, the agent puts all their effort into the make-or-break task. As the total available time increases, the agent continues to invest all their time in the make-or-break task to increase the chances of achieving it, until the point t_{max}, at which the benefits from improving the chances of success are offset by the opportunity cost. Beyond that point, the agent should invest all available time in the safe task, which entails that the time allocation problem has an internal solution. We illustrate these time allocation patterns for low and intermediate uncertainty ($\sigma = 1$ vs. $\sigma = 5$) and denote time allocated to the make-or-break task by a subscript. **Right:** The optimal amount of time allocated to the make-or-break task as a function of the skill level in the task $\lambda_m \in [0,20]$. At skill level λ_m^*, the agent should discontinuously change their strategy from allocating all their time to the safe-reward task to allocating all or at least some of their time to the make-or-break task. When the opportunity cost of time is relatively low ($v = 10$), the agent should invest all their time in the make-or-break task; when the opportunity cost ($v = 40$) is relatively high, some of their time. As the skill level continues to increase, the proportion of time the agent should optimally allocate to the make-or-break task decreases. Consistently with the main text, in both panels we use the default parameter values of $T = 10$, $\sigma_m = \sigma_s = 5$, $B = 1000$, $\Delta = 50$, $v = 10$, $\lambda_m = 10$. We default λ_s to 3 for illustrative purposes and to ensure consistency with Figure 4 in the main text.

the following proposition, we prove all of the above and give some technical conditions that tell us whether $t^* = T$ or $t^* < T$.

Proposition 2.5. *Suppose that $h(t_m;0) < h(0)$ for all $t_m \in (0,T]$. We have the following:*

i)

$$t_{opt}(\lambda_m) = \begin{cases} 0, & \text{if } \lambda_m \leq \lambda_m^* \\ \min\{T, t_{max}(\lambda_m)\}, & \text{if } \lambda_m > \lambda_m^* \end{cases}, \tag{15}$$

where

$$\lambda_m^* = \min\{\lambda_m > 0 : \exists t_m \in (0,T], h(t_m;\lambda_m) = h(0)\}. \tag{16}$$

ii) For $\lambda_m > \lambda_m^*$, $t_{opt}(\lambda_m)$ is a continuous function of λ_m, and

$$\lim_{\lambda_m \searrow \lambda_m^*} t_{opt}(\lambda_m) = t^* \tag{17}$$

where $t^* = \min\{T, t_{max}(\lambda_m^*)\}$. *Moreover,* $t^* = T$ *if and only if* $h'(T;\lambda_m^*) \geq 0$.

Proof:

i) For any $t_m \in (0,T]$, we have that $h(t_m;0) < h(0)$, $h(t_m;\lambda)$ is strictly increasing in λ_m and $\lim_{\lambda_m \to \infty} h(t_m;\lambda_m) > h(0)$. Therefore, the equation $h(t_m;\lambda_m) = h(0)$ has a unique solution. We define $\overline{\lambda}_m(t_m)$ to be this solution and note that $h(t_m;\lambda_m) > h(0)$ if and only if $\lambda_m > \overline{\lambda}_m(t_m)$.

Because the first partial derivatives of $h(t_m;\lambda_m)$ with respect to t_m and λ_m are both continuous and the one with respect to λ_m is non-zero, the Implicit Function Theorem implies that the solution $\overline{\lambda}_m(t_m)$ of $h(t_m;\lambda_m) = h(0)$ is a continuously differentiable function of t_m. Therefore, $\overline{\lambda}_m(t_m)$ attains a minimum in every interval of the form $[c,T]$, with $c > 0$. We will show that it also attains a minimum in $(0,T]$.

Note that for any given $\lambda_m > 0$, $h'(0;\lambda_m) < 0$. Therefore, by continuity of $h'(t_m;\lambda_m)$, there exists some $\varepsilon = \varepsilon(\lambda_m) > 0$, such that $h(t_m;\lambda_m) < h(0;\lambda_m) = h(0)$ for any $t_m \in (0,\varepsilon)$. In particular, there exists some $\varepsilon > 0$, such that $h(t_m;\overline{\lambda}_m(T)) < h(0)$, hence also $\overline{\lambda}_m(t_m) > \overline{\lambda}_m(T)$, for any $t_m < \varepsilon$. This proves our claim that $\overline{\lambda}_m(t_m)$ attains a minimum in $(0,T]$.

We define

$$\lambda_m^* = \min_{t_m \in (0,T]} \{\overline{\lambda}_m(t_m)\} = \min\{\lambda_m > 0 : \exists t_m \in (0,T], h(t_m;\lambda_m) = h(0)\}. \tag{18}$$

Recalling that $h(0) \geq h(t_m;\lambda_m)$ if and only if $\lambda_m \leq \overline{\lambda}_m(t_m)$, we obtain

$$t_{opt}(\lambda_m) = 0 \Leftrightarrow h(0) \geq \sup_{0 < t_m \leq T} h(t_m;\lambda_m) \Leftrightarrow \lambda_m \leq \min_{0 < t_m \leq T} \overline{\lambda}_m(t_m) = \lambda_m^*. \tag{19}$$

Combining this with Lemma 2.3, the result follows.

ii) Continuity of $t_{opt}(\lambda_m)$ for $\lambda_m > \lambda_m^*$ follows from part (i) and Lemma 2.2.

From the definition of λ_m^*, it follows that there exists some $t_m \in (0,T]$ such that $h(t_m;\lambda_m^*) = h(0)$. By Lemma 2.3, $h(t_m;\lambda_m)$ has a unique local maximum $t_{max}(\lambda_m^*)$, and by Lemma 2.2,

$$\lim_{\lambda_m \to \lambda_m^*} t_{max}(\lambda_m) = t_{max}(\lambda_m^*). \tag{20}$$

Combining this with part (i), we obtain

$$\lim_{\lambda_m \searrow \lambda_m^*} t_{opt}(\lambda_m) = \min\left\{\lim_{\lambda_m \searrow \lambda_m^*} t_{max}(\lambda_m), T\right\} = \min\{t_{max}(\lambda_m^*), T\} > 0, \tag{21}$$

because both T and $t_{max}(\lambda_m^*)$ are greater than zero.

Finally, note that by Lemma 2.1 we have that $h'(T;\lambda_m^*) \geq 0$ if and only if $t_{min}(\lambda_m^*) \leq T \leq t_{max}(\lambda_m^*)$. But $T < t_{min}(\lambda_m^*)$ is impossible anyway, because then $h(t_m;\lambda_m^*)$ would be strictly decreasing in $[0,T]$, contradicting the definition of λ_m^*. Therefore, $h'(T;\lambda_m^*) \geq 0$ if and only if $T \leq t_{max}(\lambda_m^*)$. Combining this with Equation 21, we get that $h'(T;\lambda_m^*) \geq 0$ if and only if $\lim_{\lambda_m \searrow \lambda_m^*} t_{opt}(\lambda_m) = T$.

□

2.3 Varying the reward B

We now study how t_{opt} changes as we vary B, assuming that all other parameters remain constant. In this section, we use the notation $h(t_m;B)$ and $t_{opt}(B)$ in order to emphasize the dependence of these quantities on B. The results and the proof are very similar as for λ_m, with one exception: here, the condition $h(t_m;0) < h(0)$ for all $t_m \in (0,T]$ is automatically satisfied, as can be seen directly from Equation 11. We therefore have the following proposition.

Proposition 2.6. *We have the following:*

i.

$$t_{opt}(B) = \begin{cases} 0, & \text{if } B \leq B^* \\ \min\{T, t_{max}(B)\}, & \text{if } B > B^* \end{cases}, \tag{22}$$

where

$$B^* = \min\{B > 0 : \exists t_m \in (0,T], h(t_m;B) = h(0)\}. \tag{23}$$

ii. For $B > B^*$, $t_{opt}(B)$ is a continuous function of B, and

$$\lim_{B \searrow B^*} t_{opt}(B) = t^* \in (0,T] \tag{24}$$

where $t^* = \min\{T, t_{max}(B^*)\}$. Moreover, $t^* = T$ if and only if $h'(T;B^*) \geq 0$.

As when varying λ_m, we see that as B increases, there is a discontinuous jump in the optimal amount invested in the make-or-break task at B^*, from 0 to t^*. The transition can be either to investing all of the available time in the make-or-break task (if $t^* = T$) or to investing time in both tasks (if $t^* < T$).

The proof is completely analogous to the case for λ_m, so we omit it.

3 Switching tasks more than once in the dynamic allocation problem does not provide any benefit

In this section, we show that allowing agents to switch tasks more than once in the dynamic allocation problem does not lead to an improvement in the expected reward of the optimal policy. More precisely, we show that by restricting ourselves to time allocation policies that either never switch tasks or start with the make-or-break task and switch only once, we can get equally high expected rewards as with unrestricted time allocation policies. Thus, given that an optimal policy exists, there will also exist an optimal policy with the specified properties (never switch or start from make-or-break and switch only once). But first, we need a rigorous definition of what a time allocation policy is. We use a rather general definition that requires the satisfaction of only a few intuitive properties.

This section relies on the theory of stochastic processes. A stochastic process is a random function of time. We also refer to the concepts of stopping time and filtration. We give a brief, intuitive description of these concepts and refer the interested reader to Bass (2011) and Karatzas and Shreve (2012) for more details.

A filtration is a technical way to describe the information known up to any specific point in time. We say that a stochastic process is "adapted to a filtration" if its value at any point in time relies only on the information known by that time, with respect to the filtration used.

A stopping time is a specific type of stochastic process which, as its name suggests, often describes the time that another process is stopped or, from another point of view, the time that an event occurs. Saying that the stopping time is adapted to the filtration means that whether or not the event occurs by some point in time follows from the information known so far, with respect to the filtration used. In our case, the event will be switching from one task to the other. Thus, the decision of whether to switch should strictly rely on information about the past performance.

We now proceed with the definition of a time allocation policy. In what follows we use superscripts m and s, instead of subscripts, to distinguish between the make-or-break and the safe-reward task.

Definition 3.1. *A time allocation policy (for the dynamic allocation problem) is a pair of stochastic processes* (τ^m, τ^s)*, defined on* $[0,T]$*, with the following properties:*

- $\tau_t^m + \tau_t^s = t$*, for all* $t \in [0,T]$
- $0 \leq \tau_{t_2}^m - \tau_{t_1}^m \leq t_2 - t_1$ *and* $0 \leq \tau_{t_2}^s - \tau_{t_1}^s \leq t_2 - t_1$*, for any* $t_1, t_2 \in [0,T]$*,* $t_1 \leq t_2$
- *For each* $t \in [0,T]$*,* τ_t^m *and* τ_t^s *are stopping times adapted to the filtration generated by* W^m*.*

We interpret τ_t^m and τ_t^s as the time devoted to the make-or-break task and safe-reward task, respectively, up to time t. The first condition says that the time devoted to both tasks together up to time t, is t. The second condition says that, inside an interval of time $[t_1, t_2]$, the time devoted to either task should be between 0 and $t_2 - t_1$. And the last condition makes sure that decisions on how much time to allocate to each task are based on the observed performance for the make-or-break task so far. Note that we do not allow τ_t^m and τ_t^s to depend on W^s, because the past performance in the safe-reward task has no effect on the future rewards.

Next we want to distinguish time allocation policies that switch tasks at most once and only from the make-or-break task to the safe-reward task. We call these *simple* time allocation policies. More precisely, we have the following definition.

Definition 3.2. *A time allocation policy* (τ^m, τ^s) *is simple if there exists some stopping time* ρ *adapted to the filtration generated by* W^m*, such that* $\tau_t^m = \min\{t, \rho\}$ *for each* t*.*

Intuitively, the above relation says that the time devoted to the make-or-break task increases linearly with time, until some point, where it stops increasing and takes the value ρ. This terminal value should depend only on the observed performance in the make-or-break task; this is the content of requiring ρ to be adapted to the filtration generated by W^m.

By using a time allocation policy (τ^m, τ^s), the reward from the safe-reward task is $v \cdot \lambda_s \cdot \tau_T^s$ and the reward from the make-or-break task is B, if $\tau_T^m \geq \Delta$, and 0 otherwise. These quantities are random, because τ^m and τ^s are themselves random. We denote probability and expectation with respect to a time allocation policy τ^m, τ^s by $\mathbb{P}^{\tau^m, \tau^s}$ and $\mathbb{E}^{\tau^m, \tau^s}$, respectively. Hence, the total expected reward associated with the time

allocation policy τ^m, τ^s is

$$\mathbb{E}^{\tau^m,\tau^s}\left[v \cdot \lambda_s \cdot \tau_T^s + B \cdot \mathbf{1}_{\tau_T^m \geq \Delta}\right] = v \cdot \lambda_s \cdot \mathbb{E}^{\tau^m,\tau^s}\left[\tau_T^s\right] + B \cdot \mathbb{P}^{\tau^m,\tau^s}\left[\tau_T^m \geq \Delta\right], \tag{25}$$

where $\mathbf{1}_{\tau_T^m \geq \Delta}$ denotes the indicator function of the set $\{\tau_T^m \geq \Delta\}$.

Our central claim in this section is that instead of searching over all time allocation policies for an optimal one, it is enough to search among the simple ones. This is the content of the following proposition.

Proposition 3.3. *For any time allocation policy (τ^m, τ^s), there exists a simple time allocation policy $(\overline{\tau}^m, \overline{\tau}^s)$ with the same total expected reward.*

Proof: Let (τ^m, τ^s) be any time allocation policy and define

$$\overline{\tau}_t^m = \min\{t, \tau_T\}, \;\; \overline{\tau}_t^s = t - \overline{\tau}_t^m \tag{26}$$

It is easy to verify that $(\overline{\tau}^m, \overline{\tau}^s)$ is also a time allocation policy. Moreover, by definition, τ_T^m is a stopping time adapted to the filtration generated by W^m. Therefore, $(\overline{\tau}^m, \overline{\tau}^s)$ is simple. Finally, note that $\overline{\tau}_T^m = \tau_T^m$ and $\overline{\tau}_T^s = \tau_T^s$, so (τ^m, τ^s) and $(\overline{\tau}^m, \overline{\tau}^s)$ give the same total expected reward, by Equation 25. □

Proposition 3.3 is crucial because it reduces the dynamic time allocation problem to an optimal stopping problem, a class of stochastic optimization problems that has been extensively studied (Peskir & Shiryaev, 2006). This allows us to employ the broadly used algorithmic solution described in the next section.

4 Algorithmic solution for the dynamic allocation problem

The goal of this section is twofold. First, we want to describe how to numerically calculate a time allocation policy for the dynamic time allocation task that is arbitrarily close to being optimal. Second, we want to highlight the fact that calculating such a policy involves a backwards induction mechanism, which in our opinion is more readily appreciated in the discretized version of the problem, rather than in the continuous one.

The dynamic time allocation problem, in the form described in Section 3.2.1, is a stochastic optimal control problem, whose solution is described by a partial differential equation known as the Hamilton-Jacobi-Bellman equation (Bertsekas, 1995). In solving it, one has to work backwards in time, at least implicitly, since the "initial" conditions refer to the final time T. A standard method to approximate an optimal solution is to discretize the problem (Kushner & Dupuis, 2013), which leads to the discrete version of the Hamilton-Jacobi-Bellman equation, referred to as the Bellman equation (Bellman, 2013). This has the advantage of making much clearer the need for backwards induction in order to calculate the optimal policy. Here we describe the discrete version of our problem and derive the Bellman equation associated with it.

Recall that the agent has total time T to allocate between two tasks, one of which provides a reward proportional to the time invested, and the other provides a large reward B only if the performance in this task exceeds some threshold Δ. In the dynamic allocation problem, the agent at each time knows their

performance so far and can use this information to adapt their strategy. The agent's performance q_m in the make-or-break task and q_s in the safe-reward task are given by

$$q_m(t_m) = \lambda_m t_m + \sigma_m W^m_{t_m} \quad \text{and} \quad q_s(t_s) = \lambda_s t_s + \sigma_s W^s_{t_s}, \tag{27}$$

respectively, where t_m is the time allocated (so far) to the make-or-break task, λ_m is the skill level parameter for this task, W^m is a Wiener process, σ_m measures the uncertainty of the performance, and similarly for the safe-reward task.

The agent then has to decide on how to distribute time between the two tasks, with the goal of maximizing the total expected reward. We consider the following discrete version of the problem: the agent may only switch tasks at times that are multiples of T/n, for some natural number n. In other words, the time T is divided into n intervals, and the agent commits to a single task during each interval. Moreover, the agent receives the reward for the make-or-break task only if their performance exceeds the reward threshold Δ at one of these discrete time points. As the number of allowed switching points increases, the difference between the discrete and continuous version of the problem becomes negligible and the expected reward of the optimal policy for the discrete version converges to the optimal reward of the continuous version (see Chapter 10 in Kushner & Dupuis, 2013).

In specifying a time allocation policy for the make-or-break task, the agent has to make n choices, at times $t_0 = 0, t_1 = \frac{T}{n}, \ldots, t_{n-1} = \frac{(n-1)T}{n}$, based on the performance for the make-or-break task at that time. As in the continuous time version of the problem, there is no benefit from starting with the safe-reward task earlier before switching to the make-or-break task; the optimal strategy involves beginning with the make-or-break task and at some point switching to the safe-reward task (Proposition 3.3). The switch may only happen at times $t_0 = 0, t_1, \ldots, t_{n-1}, t_n = T$, with t_0 corresponding to starting off with the safe-reward task and t_n corresponding to never switching. Accordingly, for any $k = 0, \ldots, n-1$, at time t_k the agent has two possible strategies: perform the safe-reward task for the rest of the time remaining; or perform the make-or-break task for one time interval and re-evaluate whether to continue or to switch tasks at time t_{k+1} (when there will be new information). The optimal choice is the one that gives a higher expected payoff.

The expected payoff on investing the remaining time in the safe-reward task can be calculated immediately. However, the payoff for the make-or-break task in the interval $[t_k, t_{k+1}]$ depends not only on the performance outcome, but also on the choices made at later times. If we know the optimal strategy to follow from time t_{k+1} onwards, we may assume that the agent will follow it. In other words, if we have already found the optimal policy in the interval $[t_{k+1}, T]$, then we may use it to calculate the payoff from choosing to perform the make-or-break task in the interval $[t_k, t_{k+1}]$. This suggests that we solve the problem with backwards induction.

The solution is illustrated in Fig. 2. We start by considering the decision at time t_{n-1}. Since there cannot be any task switching after that time, we only have to compare two strategies: perform the make-or-break task or the safe-reward task for the interval $[t_{n-1}, T]$. Recall that we are assuming that the reward threshold has not been reached, so in particular $q_m(t_{n-1}) < \Delta$. The expected reward from performing the safe-reward task is $v \cdot \lambda_s \cdot (T - t_{n-1}) = v \cdot \lambda_s \cdot \frac{T}{n}$. The reward from performing the make-or-break task depends on the performance at time T, $q_m(T)$. If the performance is $y = q_m(T)$, the reward will be $R = g_m(y)$ (see

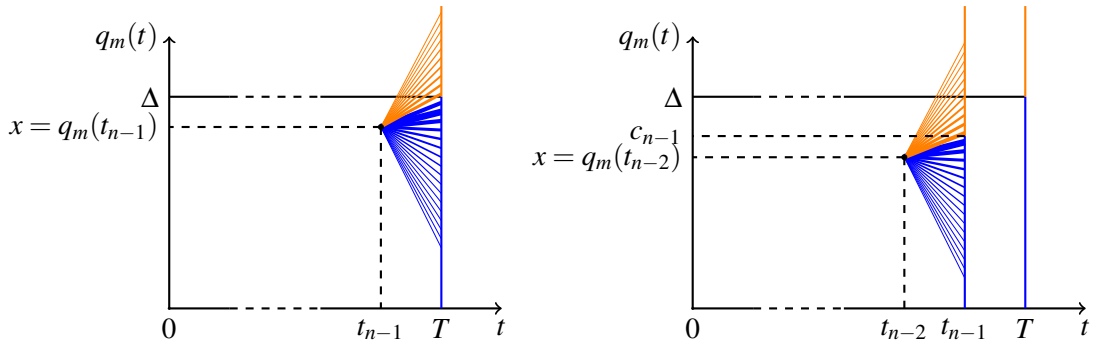

Figure 2: The process of backward induction that an agent has to follow to numerically calculate the returns from the optimal policy. **Left:** For any performance level at time t_{n-1}, the agent has to calculate the expected reward from performing the make-or-break task in the interval $[t_{n-1}, T]$, by looking at all possible terminal performance values at time T. For terminal performance values above the reward threshold Δ (denoted by orange color), the reward from the make-or-break task will be B, while for terminal performance values below Δ (blue), the reward will be 0. The expected reward of the task will be a weighted average of these two numbers. The optimal policy can be found by comparing the expected reward of the make-or-break task with that of the safe-reward task for the same interval. For large performance values at t_{n-1}, it will be optimal to invest in the make-or-break task; for smaller performance values, it will be better to invest in the safe-reward task. The optimal giving-up threshold c_{n-1} can be located by finding the break-even point where the two tasks give the same expected reward. **Right:** For any performance level at time t_{n-2}, we calculate the expected reward from performing the make-or-break task in the interval $[t_{n-2}, t_{n-1}]$ and the optimal policy in $[t_{n-1}, T]$, which is known from the previous step (orange for make-or-break task and blue for safe-reward task). We compare this expected reward with that of the safe-reward task for the whole interval $[t_{n-2}, T]$. The larger of the two will be the expected reward of the optimal policy for $[t_{n-2}, T]$. As for c_{n-1}, the optimal giving-up threshold c_{n-2} can be located by finding the point at which the agent is indifferent between the two courses of action. We continue like this for t_{n-3}, t_{n-4},...,t_0.

Equation 2). But at time t_{n-1}, the agent does not have this information; their decision has to be based on their performance up to time t_{n-1}, that is $q_m(t_{n-1})$. Given a performance $x = q_m(t_{n-1})$ at time t_{n-1}, there is a probability distribution for the performance $y = q_m(T)$ at time T. Therefore, to find the expected reward at time t_{n-1}, one has to integrate over all possible performances at time T, weighted by their likelihood. This is shown in Figure 2a. In symbols, we write

$$\begin{aligned} R^m_{n-1}(x) &= \mathbb{E}^n\left[R \,|\, q_m(t_{n-1}) = x\right] \\ &= \int_{-\infty}^{\infty} \mathbb{E}^n\left[R \,|\, q_m(t_{n-1}) = x, q_m(T) = y\right] \cdot \phi_x(y) dy \\ &= \int_{-\infty}^{\infty} \mathbb{E}^n\left[R \,|\, q_m(T) = y\right] \cdot \phi_x(y) dy, \end{aligned} \tag{28}$$

where R is the total reward, $\phi_x(y)$ is the probability density function for the performance at time T, given that at time t_{n-1} the performance was x, and $\mathbb{E}^n[\cdot]$ denotes expectation given that the make-or-break task was performed up to time $t_n = T$. Therefore, $R^m_{n-1}(x)$ is the expected reward from performing the make-or-break task on the last interval (t_{n-1}, t_n), given that this task has been performed up to time t_{n-1}, and the performance at time t_{n-1} was x. The expected value in the first integral of Equation 28 is conditioned on the performance at time t_{n-1} being x, and the performance at time $t_n = T$ being y. But if we know the performance at the last step, the performance at earlier steps is irrelevant for calculating the reward, so this justifies the last equality.

Now, the quantity $\mathbb{E}^n[R|q_m(T)=y]$ is straightforward to calculate; it is the reward, given that only the make-or-break task has been performed and at time T the performance in this task is y. Recalling the definition of g_m in Equation 2, we have that $\mathbb{E}^n[R|q_m(T)=y]=g_m(y)$, so that Equation 28 can be rewritten as

$$R^m_{n-1}(x)=\int_{-\infty}^{\infty} g_m(y)\cdot\phi_x(y)dy. \tag{29}$$

To find $\phi_x(y)$, note that the performance change for an interval of length $t_{n-1}-t_{n-2}=\frac{T}{n}$ is normally distributed, with mean $\lambda_m\cdot\frac{T}{n}$ and variance $\frac{\sigma_m^2T^2}{n^2}$. That is,

$$\phi_x(y)=\frac{n}{\sqrt{2\pi}\sigma_m T}\cdot e^{-\frac{n^2\left(y-x-\lambda_m\frac{T}{n}\right)^2}{2\sigma_m^2T^2}} \tag{30}$$

Equation 29 gives us the expected reward for performing the make-or-break task in the last time interval, for any observed performance x at time t_{n-1}. In order to decide which task to perform in the last interval, we compare this to the expected total reward of performing the safe-reward task in the last interval (assuming that the make-or-break task has been performed up to time t_{n-1}), that is

$$R^s_{n-1}(x)=g_m(x)+v\cdot\lambda_s\cdot\frac{T}{n}, \tag{31}$$

where the first term is the reward from the make-or-break task, which is equal to either 0 or B, depending on whether x exceeds the reward threshold Δ or not, and the second term is the expected reward from the safe-reward task. Here $x=q_m(t_{n-1})$ again denotes the performance in the make-or-break task at time t_{n-1}.

The expected reward of the *optimal policy* for the last step will be

$$R_{n-1}(x)=\max\left\{R^m_{n-1}(x),R^s_{n-1}(x)\right\}. \tag{32}$$

Clearly, if the reward threshold has already been reached by time t_{n-1}, so that $x>\Delta$, then $R^s_{n-1}(x)$ will always exceed $R^m_{n-1}(x)$. If we focus on the more interesting case of $x<\Delta$, then Equation 32 reduces to

$$R_{n-1}(x)=\max\left\{R^m_{n-1}(x),v\cdot\lambda_s\cdot\frac{T}{n}\right\}. \tag{33}$$

For small values of x, the safe-reward task will give a higher expected reward, so that $R_{n-1}(x)$ will equal $v\cdot\lambda_s\cdot\frac{T}{n}$. Note that this term does not depend on x. For larger x (close to the reward threshold Δ), performing the make-or-break task will yield a higher expected reward, so that $R_{n-1}(x)$ will equal $R^m_{n-1}(x)$, which does depend on x. We denote by c_{n-1} the break-even point, for which the expected rewards of the two tasks are equal. That is, c_{n-1} solves the equation

$$R^m_{n-1}(c_{n-1})=v\cdot\lambda_s\cdot\frac{T}{n}. \tag{34}$$

Once c_{n-1} has been calculated, the optimal policy for the interval $[t_{n-1},T]$ can be simply described as follows: If $q_m(t_{n-1})>c_{n-1}$, continue performing the make-or-break task; otherwise, switch to the safe-

reward task.

Now that we know the expected reward of the optimal policy for the last step for any performance value x, we can go one step back, to calculate the expected reward assuming only that the make-or-break task has been performed up to time t_{n-2}. Again, the expected reward of performing the safe-reward task for the rest of the time is straightforward to find:

$$R^s_{n-2}(x) = g_m(x) + v \cdot \lambda_s \cdot (T - t_{n-2}) = g_m(x) + v \cdot \lambda_s \cdot \frac{2T}{n}, \tag{35}$$

where $x = q_m(t_{n-2})$ is the performance in the make-or-break task up to time t_{n-1}. We now consider the expected reward for performing the make-or-break task for the interval $[t_{n-2}, t_{n-1}]$ and *assuming that the optimal policy will be followed for the interval* $[t_{n-1}, T]$, which is the case for a fully rational agent.

Suppose that at time t_{n-2} the performance is $x = q_m(t_{n-2})$. Then, there is a probability distribution for the performance $y = q_m(t_{n-1})$ at time t_{n-1}. We take this into account in calculating the expected reward for performing the make-or-break task. This is illustrated in the right part of Figure 2b, and can be expressed as

$$\begin{aligned} R^m_{n-2}(x) &= \mathbb{E}^{n-1}\left[R \,|\, q_m(t_{n-2}) = x\right] \\ &= \int_{-\infty}^{\infty} \mathbb{E}^{n-1}\left[R \,|\, q_m(t_{n-2}) = x, q_m(t_{n-1}) = y\right] \cdot \phi_x(y) dy \\ &= \int_{-\infty}^{\infty} \mathbb{E}^{n-1}\left[R \,|\, q_m(t_{n-1}) = y\right] \cdot \phi_x(y) dy \\ &= \int_{-\infty}^{\infty} R_{n-1}(y) \cdot \phi_x(y) dy, \end{aligned} \tag{36}$$

where $\mathbb{E}^{n-1}$ denotes expectation, given that the make-or-break task is performed up to time t_{n-1} and the optimal policy is followed afterwards. Thus, $R^m_{n-2}(x)$ denotes the expected reward, given that the make-or-break task has been performed up to time t_{n-2}, the performance at the make-or-break task at time t_{n-2} is x, the make-or-break task is performed in the interval (t_{n-2}, t_{n-1}), and the optimal policy is followed afterwards. Note that the last equality in Equation 36 follows from the definition of $R^m_{n-2}(x)$ in Equation 28.

The expected reward of the optimal policy is

$$\begin{aligned} R_{n-2}(x) &= \max\left\{R^m_{n-2}(x), R^s_{n-2}(x)\right\} \\ &= \max\left\{\int_{-\infty}^{\infty} R_{n-1}(y) \cdot \phi_x(y) dy,\ g_m(x) + v \cdot \lambda_s \cdot \frac{2T}{n}\right\}. \end{aligned} \tag{37}$$

The above procedure can be continued inductively, to get

$$R^m_k(x) = \int_{-\infty}^{\infty} R_{k+1}(y) \cdot \phi_x(y) dy, \tag{38}$$

$$R^s_k(x) = g_m(x) + v \cdot \lambda_s \cdot \frac{(n-k)T}{n}. \tag{39}$$

and

$$\begin{aligned} R_k(x) &= \max\{R_k^m(x), R_k^s(x)\} \\ &= \max\left\{\int_{-\infty}^{\infty} R_{k+1}(y)\cdot\phi_x(y)dy,\ g_m(x)+v\cdot\lambda_s\cdot\frac{(n-k)T}{n}\right\}, \end{aligned} \tag{40}$$

for any $k = 0,\ldots,n-1$, where, according to Equation 29, $R_n(y) = g_m(y)$.

Equation 40 is an instance of the Wald-Bellman equations. A more rigorous proof that the above equations give the reward of the optimal policy can be found in Section 1.2 of Peskir and Shiryaev (2006).

For any $k = 0,1,\ldots,n-1$, the performance break-even point c_k can be calculated as in Equation 34: it is the unique solution of the equation

$$R_k^m(c_k) = v\cdot\lambda_s\cdot\frac{(n-k)T}{n}. \tag{41}$$

For $q_m(t_k) > c_k$, the agent should continue performing the make-or-break task in the interval $[t_k, t_{k+1}]$. Otherwise, they should switch to the safe-reward task. For consistency, we set $c_N = \Delta$.

To summarize, the algorithm for the optimal policy is the following:

- For each k and each x, find $R_k(x)$ inductively from Equation 40, and c_k from Equation 41. Initialize with $R_N(x) = g_m(x)$, where g_m is given by Equation 2, and $c_N = \Delta$.
- If $c_0 \geq 0$, then perform the safe-reward task only.
- If $c_0 < 0$, start with the make-or-break task and switch to the safe-reward task at time

$$\tau = \min\{k : q_m(t_k) \leq c_k \text{ or } q_m(t_k) \geq \Delta\}, \tag{42}$$

  with the understanding that $\tau = t_N$ implies that there is no switching.

5 Time-discounting

The discussion above applies if we assume a risk-neutral agent and no time-discounting. In this section we generalize the results to allow for time-discounting and in the next section we deal with non-risk-neutral agents.

If we want to take into account time-discounting, then the time that a reward is obtained matters. For the make-or-break task we assume that the reward is obtained upon crossing the reward threshold, while for the safe-reward task we assume that the reward is obtained continuously. We will convert everything to discounted value at time $t = 0$. The discounted value of the expected safe reward if the safe task is performed from time $t_k = \frac{kT}{n}$ onwards will be

$$\int_{\frac{kT}{n}}^{T} v\cdot\lambda_s\cdot e^{-\beta t}dt, \tag{43}$$

where β is the discount factor.

If the make-or-break reward threshold was reached at time t_k, then we have to add the discounted value of the make-or-break reward, $B\cdot e^{-\frac{\beta kT}{n}}$. Otherwise, given that the agent performs the safe-reward task for the

rest of the time, they will not earn the make-or-break reward [3]. Therefore, the total expected reward from performing the safe-reward task starting at time t_k is

$$R_k^s(x) = g_m(x) \cdot e^{\frac{-\beta kT}{n}} + \int_{\frac{kT}{n}}^{T} v \cdot \lambda_s \cdot e^{-\beta t} dt \tag{39'}$$

Equation 38 for the expected reward of the make-or-break task, $R_k^m(x)$, remains unchanged, so Equation 40 becomes

$$\begin{aligned} R_k(x) &= \max\{R_k^m(x), R_k^s(x)\} \\ &= \max\left\{ \int_{-\infty}^{\infty} R_{k+1}(y) \cdot \phi_x(y) dy,\ g_m(x) \cdot e^{\frac{-\beta kT}{n}} + \int_{\frac{kT}{n}}^{T} v \cdot \lambda_s \cdot e^{-\beta t} dt \right\}, \end{aligned} \tag{40'}$$

with initial conditions $R_n(x) = g_m(x) \cdot e^{-\beta T}$.

The performance break-even point c_k, for any $k = 0, 1, \ldots, n-1$, will be the unique solution of the equation

$$R_k^m(c_k) = \int_{\frac{kT}{n}}^{T} v \cdot \lambda_s \cdot e^{-\beta t} dt. \tag{41'}$$

Since the make-or-break reward will be obtained in the future, consistently and optimally behaving time-discounting agents should become more conservative and shift their giving-up threshold upwards.

6 Risk-aversion

We now consider risk who are not risk-neutral, and who seek to maximize a utility function u, rather than the expected value from the two activities. More specifically, the agent seeks to maximize the expected utility of the total reward obtained by time T. Our main results about the single transition from the make-or-break to the safe activity does not always hold in this cases, and the algorithm we described in section 3.2.1 in not guaranteed to find the optimal solution. The problem arises from the fact that now the (stochastic) performance in the safe-reward task can affect the additional utility from rewards in the future, thus possibly changing the balance between the make-or-break and the safe-reward task. As a result, we may no longer assume that the optimum policy consists of performing the make-or-break task first, safe-reward task second, with only one switch from the first to the second, which was the result of Section 3. For example, a risk-averse agent may initially decide that the expected utility of the safe-reward task is higher than that of the make-or-break reward, but poor performance in the safe-reward task can make a high reward more attractive, justifying a switch to the make-or-break task.

To circumvent this problem, we are going to assume that the performance in the safe-reward task is not random, but deterministic. That is, we will assume that $\sigma_s = 0$. Assuming that the opportunity cost is safe is a common assumption in work on optimal investment problems in economics, finance and operations research (e.g., Dixit & Pindyck, 1994). In this case performing the safe-reward task early indeed provides no benefit (no extra information) over performing it later, and the results of Section 3 still hold, so that an

[3] We ignore the case that the reward threshold was reached at an earlier step, before t_k, because then the agent would have already switched to the safe-task earlier and there is nothing to compute.

optimal policy will involve at most one switch from the make-or-break task to the safe-reward task.

It is now straightforward to modify Equations 38-41 to account for a utility-based maximization problem. First, we modify the definition of $R_k(x)$ to denote the expected *utility*, assuming that the make-or-break task was performed up to time t_k, and the optimal policy was followed afterwards. We also modify the definitions of $R_k^m(x)$ and $R_k^s(x)$ accordingly, to refer to utility, rather than reward. In particular, we have

$$R_k^s(x) = u\left(g_m(x) + v \cdot \lambda_s \cdot \frac{(n-k)T}{n}\right), \tag{39''}$$

Equation 38 for $R_k^m(x)$ remains unchanged, so that Equation 40 becomes

$$\begin{aligned} R_k(x) &= \max\left\{R_k^m(x), R_k^s(x)\right\} \\ &= \max\left\{\int_{-\infty}^{\infty} R_{k+1}(y)\cdot\phi_x(y)dy,\ u\left(g_m(x) + v\cdot\lambda_s\cdot\frac{(n-k)T}{n}\right)\right\}, \end{aligned} \tag{40''}$$

with initial conditions $R_n(x) = u(g_m(x))$.

The performance break-even point c_k, for any $k = 0, 1, \ldots, n-1$, will be the unique solution of the equation

$$R_k^m(c_k) = u\left(v \cdot \lambda_s \cdot \frac{(n-k)T}{n}\right). \tag{41''}$$

The above discussion applies to any utility function u. For our results in Figure 6 we use a power utility function, $u(x) = \frac{x^{1-\rho}}{1-\rho}$, which is one of the most commonly implemented expected utility functions in the literature (see Holt & Laury, 2002; O'Donoghue & Somerville, 2018). For $\rho > 0$ agents are risk-averse, while for $\rho < 0$ they are risk-seeking. An optimally and consistently behaving risk-averse agent would be more conservative compared to a risk-neutral agent, so that their giving-up threshold would be shifted upwards. By contrast, optimally and consistently behaving risk-seeking individuals will become relatively more perseverant and will shift their giving-up threshold downwards.

7 Analytic expressions of hitting times and expected returns for the play-to-win strategy

In this section we provide a formula for calculating the expected reward for the play-to-win strategy. Recall from Section 3.2.2 that the time τ at which an agent using the play-to-win strategy will switch to the make-or-break task is given by

$$\tau = \min\{t : q_m(t) \geq \Delta\} = \min\left\{t : W_t^m + \frac{\lambda_m}{\sigma_m}\cdot t \geq \frac{\Delta}{\sigma_m}\right\}, \tag{44}$$

as long as $\tau < T$. The expected reward from the make-or-break task is then

$$\mathbb{E}[r_m(t_m)] = B \cdot \mathbb{P}(\tau \leq T) \tag{45}$$

and the expected reward from the safe-reward task is

$$\mathbb{E}[r_s(t_s)] = \mathbb{E}[v \cdot \lambda_s \cdot (T - \tau) \cdot \mathbf{1}_{\tau \leq T}] \tag{46}$$

$$= \mathbb{E}[v \cdot \lambda_s \cdot T \cdot \mathbf{1}_{\tau \leq T}] - \mathbb{E}[v \cdot \lambda_s \cdot \tau \cdot \mathbf{1}_{\tau \leq T}] \tag{47}$$

$$= v \cdot \lambda_s \cdot T \cdot \mathbb{P}(\tau \leq T) - v \cdot \lambda_s \cdot \mathbb{E}[\tau \cdot \mathbf{1}_{\tau \leq T}], \tag{48}$$

where $\mathbf{1}_{\tau \leq T}$ denotes the indicator function of the set $\{\tau \leq T\}$; it equals 1 if $\tau \leq T$ and 0 otherwise.

Therefore, the total expected reward for the play-to-win strategy is

$$\mathbb{E}[r_m(t_m) + r_s(t_s)] = (v \cdot \lambda_s \cdot T + B) \cdot \mathbb{P}(\tau \leq T) - v \cdot \lambda_s \cdot \mathbb{E}[\tau \cdot \mathbf{1}_{\tau \leq T}]. \tag{49}$$

To continue, we need an expression for the probability density function of τ. From Equation 44 we see that τ is the hitting time at level $\frac{\Delta}{\sigma_m}$ of a Brownian motion with drift $\frac{\lambda_m}{\sigma_m}$ (Bass, 2011; Karatzas & Shreve, 2012). Its probability density is an inverse Gaussian distribution (see Section 3.2 in Chhikara, 1989), given for any $t > 0$ by

$$f_\tau(t) = \frac{\Delta}{\sqrt{2\pi t^3} \cdot \sigma_m} \cdot e^{-\frac{(\Delta - \lambda_m \cdot t)^2}{2t{\sigma_m}^2}}. \tag{50}$$

Using this, we can write the total expected reward as

$$\mathbb{E}[r_m(t_m) + r_s(t_s)] = \frac{\Delta}{\sqrt{2\pi} \cdot \sigma_m} \cdot \int_0^T \left[(v \cdot \lambda_s \cdot T + B) t^{-\frac{3}{2}} - v \cdot \lambda_s \cdot t^{-\frac{1}{2}} \right] \cdot e^{-\frac{(\Delta - \lambda_m \cdot t)^2}{2t{\sigma_m}^2}} dt. \tag{51}$$

References

Bass, R. F. (2011). *Stochastic processes* (Vol. 33). Cambridge, United Kingdom: Cambridge University Press.

Bellman, R. (2013). *Dynamic programming*. Princeton, NJ: Courier Corporation.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Belmont, MA: Athena Scientific.

Chhikara, R. S. (1989). *The inverse gaussian distribution: theory, methodology, and applications*. New York, NY: M. Dekker.

Dixit, A. K., & Pindyck, R. S. (1994). *Investment under uncertainty*. Princeton, NJ: Princeton University Press.

Holt, C. A., & Laury, S. K. (2002). Risk aversion and incentive effects. *American Economic Review*, *92*(5), 1644–1655.

Karatzas, I., & Shreve, S. (2012). *Brownian motion and stochastic calculus*. New York, NY: Springer Science & Business Media.

Kushner, H., & Dupuis, P. G. (2013). *Numerical methods for stochastic control problems in continuous time*. New York, NY: Springer Science & Business Media.

O'Donoghue, T., & Somerville, J. (2018). Modeling risk aversion in economics. *Journal of Economic Perspectives*, *32*(2), 91–114.

Peskir, G., & Shiryaev, A. (2006). *Optimal stopping and free-boundary problems*. Basle, Switzerland: Birkhäuser.