

OR/MS Games: 4. The Joy of Egg-Dropping in Braunschweig and Hong Kong

Moshe Sniedovich

Department of Mathematics and Statistics
The University of Melbourne
Parkville, VIC 3052, Australia

m.sniedovich@ms.unimelb.edu.au

Abstract

In this discussion we examine the famous 2-egg puzzle from an OR/MS perspective and explore the structure of the optimal policies for this puzzle. It is shown that the puzzle provides a vivid environment for teaching/learning basic concepts related to decision making under uncertainty, including the differences and similarities between the Worst Case and Expected Value scenarios. The more general problem where N test eggs -rather than 2- are provided is also examined. A number of interactive modules for experimenting with and solving this interesting puzzle are provided. Lecturers teaching dynamic programming (DP) might be interested in using this puzzle to illustrate how DP works and how it can be used to derive closed-form solutions to discrete optimization problems. A reference is provided to a practical real world application of the mathematical model used to analyze and solve the game.

1. Introduction

As we all know only too well, fresh eggs tend to break when they are dropped on a hard surface. Suppose that our mission is to determine, through an experiment, the critical height at which this happens. That is, we assume that there is some critical height, call it C , such that if we drop an egg from height $h \leq C$ the egg does not break, whereas it will always break if dropped from height $h > C$. Our mission is then to determine the value of C . For this purpose we are given 2 fresh eggs and a vertical scale from which the eggs can be dropped from different heights, say $h = 1, \dots, H$, where $H \geq C$. For obvious reasons we want to minimize the number of trials. What then is the optimal egg-dropping policy?

This puzzle appears prominently on a number of distinguished lists. In particular, it features on Stan Wagon's list of dozen favourite problems/puzzles, on William Wu's list of Hard Riddles, and on the Math Forum web site as Problem of the Week 921. Yet surprisingly very little has been written on this little treasure. It would appear that although this puzzle is well known and popular among puzzle aficionados, it does not appear in print other than in the collection Which Way Did the Bicycles Go?

But even in this source the analysis is minimal: there is a graphical recipe for solving a specific instance of the problem (2 test eggs, $H = 36$, Worst Case scenario) and the entire discussion is less than half a page long. A general solution for a simple version of the game (2-egg problem, Worst Case Scenario) is given at the [Math Forum's web site](#).

In short, although our puzzle is well known and highly regarded in some math and computer science circles, it appears that it had never been subjected to an in-depth formal analysis. This is a pity because, as we shall see, the optimal policies of this puzzle exhibit very interesting and attractive patterns. A comparison between the optimal policies of the Worst Case and Expected value versions of the puzzle is also interesting.

However, the important thing is not so much that this puzzle can be solved efficiently and elegantly by an established and well known OR/MS method, but that this method -DP- is also capable of solving many other interesting puzzles within a coherent unified and systematic framework. Indeed, this article is a product of the [author's current effort](#) to promote the idea that the time has come to recognize GAMES and PUZZLES as important and challenging application areas of OR/MS. Furthermore, as indicated in Section 5, the mathematical model used to model, analyze and solve the puzzle apparently has a practical real world application!

In this paper we demonstrate that this puzzle provides a friendly framework for the introduction, illustration and analysis of a number of important OR/MS concepts such as decision making under uncertainty, Worst Case and Laplace Scenarios, optimal policies, sequential decision processes, robustness, and so on. These concepts can be discussed at different technical levels depending on the students' mathematical background, inclination and interests. Lecturers who are reluctant to include games in their courseware are encouraged to read the first article in this series ([Sniedovich, 2002a](#)).

With this in mind let us have a closer look at the puzzle itself.

The first thing to observe about the problem is that while the existence of the critical height C is guaranteed, no information whatsoever is provided about its value, except that $0 \leq C \leq H$. Note that if $C = 0$ then an egg will break if dropped from any height in the range $\{1, \dots, H\}$ and if $C = H$ then an egg will not break if dropped from any height in this range. We shall refer to H as the size of the problem.

As a consequence, strictly speaking, the problem described above is not well defined because the uncertainty regarding the value of C is not quantified. Indeed, it is impossible to simulate the above formulation, let alone find an optimal policy for the problem. This situation is typical of other popular games and puzzles, for instance the counterfeit coin problem ([Sniedovich, 2003](#)).

Decision theory suggests a number of approaches to overcome difficulties of this type ([French, 1988](#)). The most popular ones are the *Worst Case Scenario* and the *Expected Value Scenario* or the *minmax* approach and the *Bayesian approach*, respectively.

Under the *Worst Case Scenario* we assume the worst. That is, we assume that whatever we do in terms of egg-dropping, the value of C will be that which is the least favourable to the experiment. In short, under this scenario we assume that *Mother Nature* is playing against us. This means that to solve the problem we have to determine two interrelated optimal policies: one for Mother Nature and one for the decision maker (DM) responsible for solving the problem (us!).

This is the version of the puzzle that completely dominates the market place. The following is a typical description of the *Worst Case Scenario* version of the problem:

Suppose that we wish to know which windows in a 36-story building are safe to drop eggs from, and which will cause the eggs to break on landing. We make a few assumptions:

- An egg that survives a fall can be used again.
- A broken egg must be discarded.
- The effect of a fall is the same for all eggs.
- If an egg breaks when dropped, then it would

If an egg breaks when dropped, then it will break if dropped from a higher window.

- If an egg survives a fall then it would survive a shorter fall.
- It is not ruled out that the first-floor windows break eggs, nor is it ruled out that the 36th-floor windows do not cause an egg to break.

If only one egg is available and we wish to be sure of obtaining the right result, the experiment can be carried out in only one way. Drop the egg from the first-floor window; if it survives, drop it from the second floor window. Continue upward until it breaks. In the worst case, this method may require 36 droppings. Suppose 2 eggs are available. What is the least number of egg-droppings that is guaranteed to work in all cases?

Which Way Did the Bicycle Go? (page 53)

Although the term "Worst Case" is not used explicitly in this description, this idea is conveyed by the term "*in all cases*". As we shall see, one major advantage of adopting this version of the puzzle is that the problem is then relatively easy.

Under the Expected Value Scenario we assume that C is the realization of a *random variable*, call it C , whose probability distribution function over $\mathbf{H} := \{0, 1, 2, \dots, H\}$ is known. It is assumed that in this case we wish to minimize the *expected value* of the total number of trials required to determine the value of C , assuming that Mother Nature determines the value of C via some given probability distribution over \mathbf{H} . The difficulty with this approach is that we have to postulate a probability distribution over \mathbf{H} . The most famous instance of this scenario is *Laplace's Scenario* under which it is assumed that C is *uniformly distributed* over \mathbf{H} . We assume that the reader is familiar with the distinction we make here between the random variable C and a particular value that it can take, c .

We shall consider in detail the Worst Case Scenario and Laplace's Scenario and show how they can be analyzed and solved via dynamic programming (DP). And for the readers convenience, in the main body of the paper we shall consider only the simple case where 2 test eggs are available. The analysis and solution of problems where more than two test ages are available is provided in the [Appendix](#).

As a prelude you may wish to experiment with the interactive module provided below. It simulates the Worst Case Scenario so when you play with it assume that Mother Nature is indeed playing against you.

Worst Case Scenario: 2 Test Eggs																																																																	
<div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <p>Caption</p> <table style="width: 100%; border-collapse: collapse;"> <tr><td style="color: red; width: 15px;"></td><td>- Egg 1</td></tr> <tr><td style="color: blue; width: 15px;"></td><td>- Egg 2</td></tr> <tr><td style="color: yellow; width: 15px;"></td><td>- Egg survived</td></tr> <tr><td style="color: orange; width: 15px;"></td><td>- Egg broken</td></tr> </table> <hr/> <p>No. of trials <input type="text" value="0"/></p> <p style="text-align: center;"><input type="button" value="Reset"/></p> </div> <div style="flex: 1; text-align: center;"> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15px;"></th> <th style="width: 15px; text-align: center;">h</th> <th style="width: 15px; text-align: center;">1</th> <th style="width: 15px; text-align: center;">2</th> </tr> </thead> <tbody> <tr><td style="color: red;">15</td><td style="background-color: red;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: blue;">14</td><td style="background-color: blue;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: yellow;">13</td><td style="background-color: yellow;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: orange;">12</td><td style="background-color: orange;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: yellow;">11</td><td style="background-color: yellow;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: red;">10</td><td style="background-color: red;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: yellow;">9</td><td style="background-color: yellow;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: blue;">8</td><td style="background-color: blue;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: yellow;">7</td><td style="background-color: yellow;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: orange;">6</td><td style="background-color: orange;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: yellow;">5</td><td style="background-color: yellow;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: red;">4</td><td style="background-color: red;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> <tr><td style="color: yellow;">3</td><td style="background-color: yellow;"></td><td style="background-color: red;"></td><td style="background-color: blue;"></td></tr> </tbody> </table> </div> </div>		- Egg 1		- Egg 2		- Egg survived		- Egg broken		h	1	2	15				14				13				12				11				10				9				8				7				6				5				4				3				<p style="text-align: center;">Notice Board</p> <p>Instructions :</p> <p>-----</p> <p>Click the red (Egg 1) and blue (Egg 2) columns at the positions (h) from which you wish to drop the respective test egg.</p> <p>The result of a trial will be displayed by either a white box (egg survived the fall) or a gold box (egg is broken) at the h position of the trial.</p>
	- Egg 1																																																																
	- Egg 2																																																																
	- Egg survived																																																																
	- Egg broken																																																																
	h	1	2																																																														
15																																																																	
14																																																																	
13																																																																	
12																																																																	
11																																																																	
10																																																																	
9																																																																	
8																																																																	
7																																																																	
6																																																																	
5																																																																	
4																																																																	
3																																																																	

Commentary:

To solve the puzzle using this module, you have to either (i) completely change the red column into a white column or (ii) have a gold box somewhere in the blue column and a white box just below it in the red or blue column or (iii) have a gold box in the first position in either the red or blue column. In case (i) we have $C = H$ and in case (ii) we have $C = G-1$, where G is the position of the lowest gold box in the red column, and in case (iii) we have $C = 0$.

Needless to say, the objective is not merely to determine the value of C , but rather to determine how the trials should be conducted so as to determine the value of C with the least number of trials, regardless of how Mother Nature plays her game. Observe, however, that if we know the value of C , then it is easy to construct an optimal policy to solve the problem: we drop Egg 1 from position $h = C + 1$. Naturally, it will crash. Then we drop Egg 2 from $h = 1$, then from $h=2$, and so on up to $h = C$. The second egg will survive all these trials. Note that this policy requires exactly $C + 1$ trials: one with Egg 1 and C with Egg 2. Furthermore, using this strategy we shall break only one egg. The difficulty is then that the value of C is not known in advance.

Before we explain how DP solves problems of this kind, let us examine in more detail the essential issues faced by Mother Nature and the Decision Maker (DM). This will facilitate the formulation of the DP model for the puzzle.

2. Preliminary Analysis

In order to appreciate the complexity of the puzzle it is instructive to consider first a much simpler puzzle, namely the puzzle where only *one* test egg -rather than two- is provided. Once we obtain a solution to this problem we can use it in the construction of the optimal solution to problems where there are two test eggs. Hopefully, this will then enable us to obtain a solution for the more general case where there are n test eggs. This is a typical DP approach to problem solving.

The other thing we ought to do is agree on suitable notation, terminology and conventions for the formulation of the mathematical model for the puzzle. Here again we use a DP inspired paradigm according to which the problem under consideration is regarded as a *sequential decision process*. The core ingredients of such processes are objects called *state variables*, *decision variables* and *transitions*.

It is convenient to describe these objects using the notion *interval of uncertainty* (IOU). This interval is simply the range of heights that are yet to be tested. Thus, initially IOU = $\{1,2,\dots,H\}$. Now suppose that we drop an egg from height h and the egg breaks. Then the IOU will be equal to $\{1,2,\dots,h - 1\}$. If the egg does not break then the IOU would be equal to $\{h + 1 ,\dots,H\}$.

The *size* of a problem is the size (cardinality) of the IOU associated with the problem. Thus, initially the size of the problem is H . The size of a problem associated with the IOU = $\{1,\dots,h - 1\}$ is $h - 1$ and the size of a problem associated with IOU = $\{h + 1,\dots,H\}$ is equal to $H-h$. So we see that if we drop an egg from height h , the size of the original IOU -hence the size of the original problem- will decrease from H to either $h - 1$ (if the egg breaks) or to $H-h$ (if the egg does not break). The question regarding which one of these two events will actually take place must be dealt with within the assumed behaviour of Mother Nature.

States

Three parameters are required to provide a complete description of the state of

Three parameters are required to provide a complete description of the state of the process:

- n = number of test eggs available, $n = 0, 1, 2, 3, \dots, N$.
- k = size of the problem, $k = 0, 1, 2, \dots, H$.
- m = the lowest position in the IOU, $m = 1, 2, \dots, k$.

For instance, the state $s = (2, 6, 4)$ indicates that 2 test eggs are available, and IOU = $\{4, 5, 6, 7, 8, 9\}$. The initial state of the process is $s = (N, H, 1)$ where N denotes the number of test eggs available at the commencement of the experiment. The process terminates either when there are no more test eggs ($n = 0$) or when the IOU is empty ($k = 0$), whichever occurs first. Let S denotes the set of all feasible states. Observe that for state $s = (n, k, m)$ to be feasible with respect to the original problem we must have $m + k - 1 \leq H$.

Decision variables

There are two kinds of decisions: those made by the decision maker (DM), call them x , and those made by Mother Nature, call them z . Let $DX(s)$ denote the set of feasible decisions available to DM when the process is state s . Then from the description of the problem it follows that $DX(s)$ is the set consisting of the elements of the IOU associated with state $s = (n, k, m)$, namely $DX(n, k, m) = \{m, \dots, m + k - 1\}$. Alternatively, $DX(s)$ could be defined as the set of corresponding *relative positions* $\{1, 2, \dots, k\}$. For the purposes of our analysis the later is more convenient. So from now on we shall let

$$DX(n, k, m) := \{1, \dots, k\} \quad (1)$$

denote the set of feasible decisions available to DM when the process is in state $s = (n, k, m)$. The exception is the case where $n=1$: here we have

$$DX(1, k, m) := \{1\} \quad (2)$$

that is, if only one test egg is available then we must drop the egg from the first (lowest) position in the IOU.

As far as Mother Nature is concerned, the set of feasible solutions pertaining to state s and decision x (on the part of DM) is as follows:

$$DN := \{\text{Break}, \text{!Break}\} \quad (3)$$

where $\text{!Break} := \text{Do not Break}$.

Transitions

Given that the process is in state s , the DM selects decision x and Mother Nature selects decision z , the new state of the process would be $s' = (n', k', m') = t(s, x, z)$ where the values of n' , k' and m' are as follows:

$z =$	Break	!Break
$n' =$	$n-1$	n
$k' =$	$x-1$	$k-x$
$m' =$	m	$m+x$

We refer to t as the *transition function*. The following figure illustrates how the transition function works, highlighting how the two values of z affect the new state and the new IOU of the process.



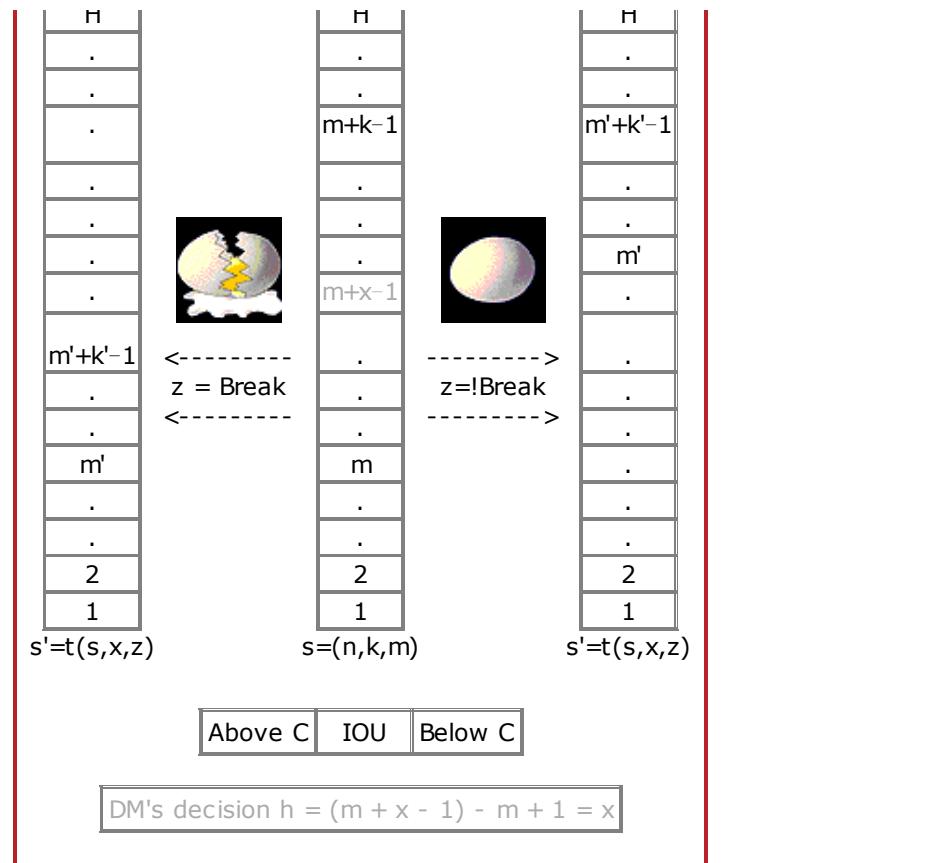


Figure 1: State Transitions

Optimal Scores

To construct DP functional equation for the problem we now define the following (conditional) optimal scores:

$W(s) :=$ minimum number of trials required to identify the value of C under the Worst Case Scenario given that the initial state of the process is $s = (n, k, m)$.

$A(s) :=$ minimum value of the expected value of number of trials required to identify the value of C under the Expected Value Scenario given that the initial state of the process is $s = (n, k, m)$.

$L(s) :=$ minimum value of the expected value of number of trials required to identify the value of C under Laplace's Scenario given that the initial state of the process is $s = (n, k, m)$.

Note that here the initial state of the process is deployed a parameter. We are interested in the values of $W(s)$, $A(s)$ and $L(s)$ for $s = (N, H, 1)$ which is the initial state of the original problem.

1-Egg problem

If $n = 1$, namely if there is only one test egg, then the problem is extremely simple, in fact it is trivial. This is so because in this case there is only one feasible policy: we must drop the egg from the lowest position ($x = 1$) in the IOU. So we first drop the egg from position $h = 1$, then from position $h = 2$, and so on, until it either breaks, or we reach $h = H$. If it breaks, say at height $h = q$, then $C = q-1$. If it does not break then $C = H$. Observe that should we skip a step and move from say height q to height $q + 2$, then we may expose ourselves to a major embarrassment: the egg will break at $h = q + 2$ and we shall not know whether $C = q$ or $C = q + 1$. We do not want to do that!

Since this simple policy is the *only* feasible policy, it must be the optimal policy. Clearly then, if Mother Nature is playing against us (trying to prolong the experiment as much as possible), She will set $C = H$, in which case exactly H trials will be needed to solve the problem.

If the value of C is determined by a probability distribution function, then the number of trials is a random variable, call it \underline{I} , whose probability distribution can be determined from the distribution of \underline{C} . More specifically, it can be shown that

Lemma 1

Under the optimal policy for the case where $n = 1$ we have

$$P[\underline{I} = T|s] = P[\underline{C} = T-1], T = 1, \dots, H-1; s = (1, H, m) \quad (4)$$

$$P[\underline{I} = H|s] = P[\underline{C} \geq H-1|s] = P[\underline{C} = H-1|s] + P[\underline{C} = H|s], s = (1, H, m) \quad (5)$$

$$E[\underline{I}|s] = 1 \times P[\underline{I} = 1|s] + \dots + H \times P[\underline{I} = H|s] = E[\underline{C}|s] + P[\underline{C} \leq H-1|s], s = (1, H, m) \quad (6)$$

where $P[\text{event}]$ denotes the probability of event, $P[\dots | s]$ denotes conditional probability and $E[\dots | s]$ denotes conditional expectation given state s .

Observe that the fact that \underline{C} is uniformly distributed does not entail that so is \underline{I} . For example, consider the case where $H=3$ and \underline{C} is uniformly distributed over $\{0, 1, 2, 3\}$. Then $P[\underline{I} = 1] = 1/4$, $P[\underline{I} = 2] = 1/4$, and $P[\underline{I} = 3] = 1/2$. Hence $E[\underline{C}] = 3/2$ and $E[\underline{I}] = 9/4$.

If \underline{C} is uniformly distributed, then $P[\underline{C} = H] = 1/(H+1)$ and $E[\underline{C}] = H/2$. Hence, (6) implies the following:

Lemma 2

If \underline{C} is uniformly distributed, then

$$E[\underline{I}|s] = H/2 + 1 - 1/(H+1) = H/2 + H/(H+1) = (H^2 + 3H)/2(H + 1). \quad (7)$$

So here are the basic results:

Corollary 1

$$W(1, k, m) = k, k = 1, 2, 3, \dots, H. \quad (8)$$

$$A(1, k, m) = E[\underline{C}|m] + 1 - P[\underline{C} = k|m], k = 1, 2, 3, \dots, H. \quad (9)$$

$$L(1, k, m) = (k/2) + k/(k+1), k = 1, 2, 3, \dots, H. \quad (10)$$

Important Observations:

- Neither $W(1, k, m)$ nor $L(1, k, m)$ depend on m . We shall therefore sometime omit m and write $W(1, k)$ and $L(1, k)$ respectively.
- Recall that we use k as a parameter stipulating the size of the problem (ie. size of the IOU). Initially $k = H$ and at (successful) termination $k = 0$. If upon termination $k > 0$, then the critical value of C has not been identified. The implication is that the policy that we used was not feasible.

As indicated by Table 1, as k increases, $W(1, k)$ can be much - almost twice - larger than $L(1, k)$.

Table 1

k	$W(1, k)$	$L(1, k)$
1	1	1/3

1	1	1
2	2	1.66667
3	3	2.25
4	4	2.8
5	5	3.33333
10	10	5.90909
50	50	25.98039
100	100	50.99009
???		

Now that we know how to handle the 1-test-egg situation, we can examine the more difficult case where two test eggs are available. We shall first consider the Worst Case Scenario and then the Expected Value Scenario. Once we accomplish this, we shall consider the case where more than two test eggs are provided.

3. Worst Case Scenario

To derive a DP functional equation for $W(s)$ it is constructive to ask the following question: what is the best we can do under the Worst Case Scenario given that the initial state of the process is $s=(n,k,m)$ and that in the next trial we drop the egg from position x within the IOU? That is, it is useful to stipulate a given state and a given decision (on the part of DM) and attempt to solve the resulting conditional problem. So let

$W(s|x) := \text{minimum number of trials required to identify } C \text{ under the Worst Case Scenario given that the state of the process is } s \text{ and that } h=x \text{ in the next trial, } (s = (n,k,m), x = 1, \dots, k).$

The following result is an immediate consequence of the fact that the first trial of an optimal policy must place the first egg at some position x in $\{1, \dots, k\}$. In other words, there must be some x in $\{1, \dots, k\}$ such that $W(s) = W(s|x)$ and this value of x minimizes $W(s|x)$ over x in $\{1, \dots, k\}$:

Lemma 3

$$W(s) = \min \{W(s|x) : x \in \{1, \dots, k\}\}, s \in S. \quad (11)$$

This means that once we know how to determine the value of $W(s|x)$ for each x in $\{1, \dots, k\}$ we can easily determine the value of $W(s)$. Observe that because under the Worst Case Scenario Mother Nature is playing against us, it is not too difficult to figure out how She will react to our choice of x . In fact, all we have to do is examine each one of the two possible reactions Mother Nature has in Her arsenal - 'Break it!' or 'Don't break it!' - and check which one is best for Her (worse for us). Thus, let

$W(s|x,z) := \text{minimum number of trials required to identify } C \text{ under the Worst Case Scenario given that the state of the process is } s, \text{ the next trial position is } h=x \text{ and Mother Nature selects decision } z \text{ in } DN := \{\text{Break, !break}\}.$

Given that Mother Nature is playing against us, it is clear that She will select the larger of $W(s|x, \text{Break})$ and $W(s|x, \text{!break})$. Thus,

Lemma 4

$$W(s|x) = \max \{W(s|x, \text{Break}), W(s|x, \text{!Break})\}, s \in S. \quad (12)$$

$$W(s) = \min \{\max \{W(s|x, \text{Break}), W(s|x, \text{!Break})\} : x = 1, \dots, k\}, s \in S. \quad (13)$$

We are almost there! All we have to do now is derive suitable expressions for $W(s|x, \text{Break})$ and $W(s|x, \text{!Break})$ and substitute them in the right hand side of (13) so that we can compute $W(s)$. The following results are direct

implications or the definitions of $W(s | x, z)$.

Lemma 5

$$W(s|x, z) = 1 + W(t(s,x,z)) \quad (14)$$

where t is the transition function. Hence,

$$W(n,k,m|x, !Break) = 1 + W(n,k-x,m+x-1). \quad (15)$$

$$W(n,k,m|x, Break) = 1 + W(n-1,x-1,m). \quad (16)$$

In words, if the egg we drop from position x breaks, then we are left with $n-1$ test eggs and still have to examine the positions below x . Hence, $W(n,k,m|x, Break) = 1 + W(n-1,x-1,m)$. If the first egg does not break, we are left with n test eggs and have to examine the positions above x . Hence, $W(s | x, !Break) = 1 + W(n,k-x,m+x-1)$. The 1 in both cases represents the "cost" of one trial.

We therefore conclude that

Theorem 1

$$W(n,k) = 1 + \min\{\max(W(n-1,x-1), W(n,k-x)): x \in \{1,2,\dots,k\}\}, \quad n=1,\dots,N; k=2,3,4,\dots,H, \quad (17)$$

observing that by definition $W(n,k) = 1$ if $k = 1$.

This is the *DP functional equation* for the puzzle under the Worst Case Scenario. Note that in (17) we write $W(n,k)$ rather than $W(n,k,m)$ because $W(n,k,m)$ does not depend on m .

Since we already know that $W(1,k) = k$ for all $k = 1,\dots,H$, this functional equation can be easily solved for $n = 2,\dots,N$ and $k = 1,\dots,H$. In particular, for $n = 2$ we have:

$$W(2,k) = 1 + \min\{\max(W(1,x-1), W(2,k-x)): x \in \{1,2,\dots,k\}\}, \quad k = 2,3,4,\dots,H, \quad (18)$$

with $W(2,1)=1$

The values of $W(2,k)$ for $k = 1,\dots,15$ are displayed in Table 2. This table also displays the set of optimal values of x pertaining to $W(2,k)$, denoted by $X(2,k)$. That is,

$$X(2,k) := \{x \in \{1,\dots,k\}: W(2,k | x) = W(2,k)\} \quad (19)$$

Needless to say, we shall pay careful attention to the patterns exhibited by the optimal solutions $X(2,k)$ and the optimal values of the objective function $W(2,k)$.

Table 2

k	$W(2,k)$	$X(2,k)$
1	1	{1}
2	2	{1,2}
3	2	{2}
4	3	{1,2,3}
5	3	{2,3}
6	3	{3}
7	4	{1,2,3,4}
8	4	{2,3,4}
9	4	{3,4}

10	4	{4}
11	5	{1,2,3,4,5}
12	5	{2,3,4,5}
13	5	{3,4,5}
14	5	{4,5}
15	5	{5}

You can now use the following interactive module based on (13) to generate optimal solutions to much larger problems and confirm the validity of the patterns suggested by Table 2.

Worst Case Scenario: 2 Test Eggs

for $H =$ (recommended range 20-200)
 computation of $W(2,k)$ and $X(2,k)$ for $k =$ ($1 \leq k < H$)
 output in a printer-friendly format (new window)
 Full Align: Left Center Right

Module 2

It often happens in life that once you solve a problem the hard way (eg. numerically), you detect an obvious structure in the solution and discover a nice, elegant closed-form solution. Fortunately, this is the case here. So our mission now is to quantify the nice and constructive patterns exhibited by $W(2,k)$ and $X(2,k)$.

This is not difficult. Indeed, even a semi-blind cat should be able to detect the

pattern governing $W(2,k)$ as k is increasing. The horizontal version of this pattern is as follows:

$$1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, \dots$$

Formally, it can be described as a sequence of the positive integers, where integer j is repeated exactly j times. Let $\text{First}(j)$ denote the position of the first occurrence of j in this sequence. By inspection $\text{First}(1) = 1$, $\text{First}(2) = 2$, $\text{First}(3) = 4$, $\text{First}(4) = 7$, and more generally,

Lemma 6

$$\text{First}(j) = 1 + j(j-1)/2, j = 1, 2, 3, \dots \quad (20)$$

Perhaps more useful is the following result, which can be easily derived from Lemma 6:

Lemma 7

$$W(2,k) = \text{Integer}((1 + (8k - 7)1/2)/2), k = 1, 2, 3, \dots, H \quad (21)$$

where $\text{Integer}(a)$ denotes the integer part of a , eg. $\text{Integer}(4)=4$, $\text{Integer}(5.6)=5$.

Using this recipe we can now quickly compute the value of $W(2,k)$ for any positive integer k :

$k =$

$W(2,k) =$

Type in the desired value of k and then click the $W(2,k)$ field.

Utilising these results and observing the pattern of the $X(k,2)$ values, it is clear that the following result holds:

Lemma 8

$$X(2,k) = \{(1 + k - \text{First}(W(2,k))), \dots, W(2,k)\}, k = 1, 2, 3, \dots \quad (22)$$

So here is a simple engine to compute the values of $X(2,k)$ for any positive integer k :

$k =$

$X(2,k) =$

Type in the desired value of k and then click the $X(2,k)$ field.

The following result is an immediate consequence of the fact that $W(k,2)$ is an element of $X(2,k)$ for any positive integer k :

Lemma 9

The following is an optimal policy for the 2-egg problem of size k under the Worst Case Scenario:

- Drop Egg 1 from position $h = W(2,k)$ - (Mother Nature will break it).
- Drop Egg 2 from position 1, then from position 2, and so on up to position $W(2,k)-1$.

This is the *unique* optimal policy in cases where $k = W(2,k) + \text{First}(W(2,k)) - 1$.

Last but not least, the question of Mother Nature's reaction to our decisions. That is, what is Mother Nature's optimal policy?

The following is an immediate implication of Lemma 9:

Lemma 10

The following is an optimal policy for Mother Nature in the context of the 2-egg problem of size k under the Worst Case Scenario:

- Break Egg 1 if and only if it is dropped from position $h = W(2,k)$ or higher.
- Break Egg 2 if and only if it is dropped from position $h > 1$ but has not been dropped yet from position $h - 1$. Needless to say, break it if it is dropped from a position that is at least as high as that from which Egg 1 was dropped when it was broken.

It must be stressed that the above results should be interpreted in a *parametric* fashion due to the dynamic nature of the game. In particular, since Mother Nature is playing against us, the critical value of C is changing depending on the positions from which Egg 1 is dropped. Also, the values of k and the optimal decisions (elements of $X(2,k)$) are *relative* rather than absolute in nature. That is, they represent positions within a **given** range, rather than absolute heights related to the original problem. The set of optimal solutions $X(2,k)$ are defined in this manner.

We illustrate this important point by considering the problem where $H = 11$, and where *the optimal policy is constructed by selecting the second largest element of $X(2,k)$* . That is, in the context of a problem of size k we drop Egg 1 from position $p = W(2,k)-1$ if $X(2,k)$ is not a singleton, otherwise ($W(2,k)$ is a singleton) we drop it from position $p = W(2,k)$.

So we start with $k^{(1)} = 11$ for which the second largest element of $X(2,11) = \{1,2,3,4,5\}$ is $p^{(1)} = 4$. So we drop Egg 1 from position 4. Since the critical value of C for $k^{(1)} = 11$ is equal to $W(2,11)-1 = 5-1 = 4$, it follows that the egg will not break. Thus, we have to test the (absolute) heights $\{5,6,7,8,9,10,11\}$. Using $h = 4$, rather than $h = 0$, as the origin, our next problem is then of size $k^{(2)} = k^{(1)}-p^{(1)} = 11-4 = 7$ and consists of the relative positions $\{1,2,3,4,5,6,7\}$.

Next, according to our policy, we now drop Egg 1 from position $p^{(2)} = 3$ which is the second largest element of $W(2,k^{(2)}) = W(2,7) = \{1,2,3,4\}$. Since for $k^{(2)}=7$ the critical value of C is equal to 3, Egg 1 will not break. We are then left with a problem of size $k^{(3)} = k^{(2)}-p^{(2)} = 7-3 = 4$, consisting of the absolute heights $\{8,9,10,11\}$. For $k = 4$ we have $X(2,4) = \{1,2,3\}$ so according to our policy we now drop Egg 1 from position $p^{(3)} = 2$. Since the critical value of C for $k = 4$ is equal to 2, the egg will not break. We are now left with a problem of size $k^{(4)} = k^{(3)}-p^{(3)} = 4-2 = 2$ consisting of the original (absolute) heights $\{10,11\}$ or equivalently the relative positions $\{1,2\}$ in the current interval. So we consider now a problem of size $k = 2$, for which we have $X(2,2) = \{1,2\}$. According to our policy we now drop Egg 1 from position $p^{(4)} = 1$. The critical value of C for $k=2$ is equal to 1, so the egg will not break. We are now left with a simple problem of size 1 whose range consists of one position only, namely $k^{(5)} = 1$. So we now drop Egg 1 from this position, hence $p^{(5)} = 1$. It will either break or survive, but in either case the location of C will be identified after this trial.

Note that, as expected, exactly $W(11,2) = 5$ trials were needed. The relative positions from which Egg 1 was dropped are as follows: $p^{(1)} = 4$, $p^{(2)} = 3$, $p^{(3)} = 2$, $p^{(4)} = 1$, $p^{(5)} = 1$. The actual (absolute) heights from which Egg 1 was dropped are then

$$\begin{aligned} h^{(1)} &= p^{(1)} = 4 \\ h^{(2)} &= h^{(1)} + p^{(2)} = 4 + 3 = 7 \\ h^{(3)} &= h^{(2)} + p^{(3)} = 7 + 2 = 9 \\ h^{(4)} &= h^{(3)} + p^{(4)} = 9 + 1 = 10 \\ h^{(5)} &= h^{(4)} + p^{(5)} = 10 + 1 = 11. \end{aligned}$$

Observe that although using this policy Egg 1 was not broken when dropped from height $h=10$, the critical value of C is only $C = W(2,11) - 1 = 4$. Indeed, if in the first trial you drop Egg 1 from position 5 it will break.

Exercise 1 Show that under the Worst Case Scenario, $W(2,k) = \max \{x \text{ in } X(2,k)\}$, $k = 1,2,3,\dots,H$.

4. Expected Value Scenario

The derivation of the DP functional equation for this version of the problem is

very similar to that presented above except that here Mother Nature is using a probability density function on $\mathbf{H} = \{0,1,2,\dots,H\}$ to decide what is the value of C . As a consequence we try to minimize the *expected value* of the number of trials, rather than the maximum number of trials. It is assumed that the probability distribution function is of C on \mathbf{H} is known.

Consider then the following conditional optimal scores associated with $A(s)$:

$A(s|x) :=$ minimum expected value of number of trials required to identify C under the Expected Value Scenario given that the initial state is s and that the next trial is from position $h=x$ relative to the current IOU.

$A(s|x, Break) :=$ Same as $A(s|x)$ except that it is also given that Mother Nature applies decision $z = \text{Break}$.

$A(s|x, !Break) :=$ Same as $A(s|x, Break)$ except that $z = \text{!Break}$.

Given that Mother Nature is applying a probability density function to determine C , it follows that:

Lemma 11

$$A(s|x) = P[x \leq \underline{C}|s]A(s|x, !Break) + P[x > \underline{C}|s]A(s|x, Break) \quad (23)$$

$$A(s) = \min \{P[x \leq \underline{C}|s]A(s|x, !Break) + P[x > \underline{C}|s]A(s|x, Break) : x \in \{1, \dots, k\}\} \quad (24)$$

where $P[|s]$ denotes the conditional probability of the specified event given that the state of the process is s . For example, if \underline{C} is uniformly distributed, then $P[5 \geq \underline{C}|s] = 6/(k+1)$, $s = (n, k, m)$. Note that the values that \underline{C} can take are relative to the IOU under consideration.

To complete the derivation of the functional equation we only have to derive proper expressions for $A(s | x, Break)$ and $A(s | x, !Break)$. To this end we recall that from the definition of the critical value of \underline{C} it follows that

$$A(n, k, m | x, !Break) = 1 + A(n, k-x, m+x-1) \quad (25)$$

$$A(n, k, m | x, Break) = 1 + A(n-1, x-1, m) \quad (26)$$

Substituting these expressions in the right hand side of (24) yields:

Lemma 12

$$A(n, k, m) = 1 + \min \{P[x \leq \underline{C}|n, k, m]A(n, k-x, m+x-1) + P[x > \underline{C}|n, k, m]A(n-1, x-1, m) : x \in \{1, \dots, k\}\} \quad (27)$$

This is the DP functional equation for the Expected value Scenario version of the problem. It can be easily solved, but it is not very convenient to experiment with it because one has to specify the probability density function of \underline{C} .

We therefore now consider the special case where \underline{C} is *uniformly distributed* over the IOU, namely the *Laplace's Scenario*. The attractive feature of the Laplace's Scenario is that within its framework $A(n, k, m)$ does not depend on m and it is easy to compute the two probabilities need in (27). That is, $P[x \leq \underline{C}|n, k, m] = (1+k-x)/(1+k)$ and $P[x > \underline{C} | n, k, m] = x/(1+k)$. Thus, (27) yields

Theorem 2

$$L(n, k) = 1 + \min \{((1+k-x)/(1+k))L(n, k-x) + (x/(1+k))L(n-1, x-1) : x \in \{1, \dots, k\}\}, \quad (28)$$

where for $n=1$ the value of $L(n,k')$, $k'=x-1$, is given in (10). Note that by definition $L(n,1)=1$ for all $n >=1$.

The functional equation (28) can be easily solved for $k=1,2,3,\dots,H$ utilising the values of $L(1,k)$ stipulated by (10). Here is a sample of $L(2,k)$ values and the respective sets of optimal decisions $Y(2,k)$.

Table 3

k	$L(2,k)$	$Y(2,k)$
1	1	{1}
2	10/6	{1,2}
3	2	{2}
4	12/5	{2,3}
5	8/3	{2,3}
6	20/7	{3}
7	25/8	{3,4}
8	10/3	{3,4}
9	7/2	{3,4}
10	40/11	{4}
11	23/6	{4,5}
12	4	{4,5}
13	29/7	{4,5}
14	64/15	{4,5}
15	35/8	{5}

Obviously the results exhibit a number of patterns. You are encouraged to use the following interactive module based on (28) to generate optimal solutions for the Laplace's Scenario and identify these patterns (both for $L(2,k)$ and $Y(2,k)$).

Laplace's Scenario: 2 Test Eggs

- for $H = \boxed{136}$ (recommended range 20-200)
- computation of $L(k,2)$ for $k = \boxed{11}$ ($1 \leq k \leq H$)
- output in a printer-friendly manner (new window)

As expected, $L(2,k)$ is strictly increasing with k and $L(2,k) \leq W(2,k)$. The pattern for $Y(2,k)$ is obvious and can be detected by inspection. The pattern for $L(2,k)$ is more subtle. Table 4 presents the results of Table 3 in a format that makes it easier to detect these patterns.

Table 4

k	$L(2,k)$	$Y(2,k)$
1	2/2	{1}
2	5/3	{1,2}
3	8/4	{2}
4	12/5	{2,3}
5	16/6	{2,3}
6	20/7	{3}
7	25/8	{3,4}
8	30/9	{3,4}
9	35/10	{3,4}
10	40/11	{4}
11	46/12	{4,5}
12	52/13	{4,5}
13	58/14	{4,5}
14	64/15	{4,5}
15	70/16	{5}

These results inspire a number of exercises:

Exercise 2 Find the pattern for the smallest and largest elements of $Y(2,k)$, hence the pattern for $Y(2,k)$.

Exercise 3 Find the pattern for $L(2,k)$. Hint 1: write $L(2,k) = N(k)/D(k)$. Then by inspection, $D(k) = k+1$. So all you have to do now is find the pattern for $N(k)$. Hint 2: examine the pattern for $N(k+1) - N(k)$.

Definition 1 An egg dropping policy is said to be *robust* if it is optimal for both the Worst Case Scenario and Laplace's Scenario.

Exercise 4 Determine whether there is a *robust* optimal policy for the 2-egg problem.

Note that from Table 2 and Table 3 it follows that $Y(2,k)$ is a subset of $X(2,k)$ for all $k \leq 15$. Consider then:

Exercise 5 Determine whether $Y(2,k)$ is a subset of $X(2,k)$ for all $k \geq 1$.

5. Perspective

There are a lot of famous educationally rich games and puzzles out there. Some have been used for many years in math and computer science education. The obvious question is: why not in OR/MS? As we have already noted elsewhere (Sniedovich, [2002a](#), [2002](#), [2003](#)), the OR/MS community should have vested interest in giving such games an OR/MS perspective. Not only this will make it easier to incorporate such games in OR/MS courseware. it will also facilitate the

...
establishment of *GAMES* and *PUZZLES* an important new OR/MS application area.

In the case of the 2-egg puzzle, we introduced in this paper (to the delight of Bayesian decision analysts) an obvious new (Expected Value) version of the puzzle and developed closed-form solutions for both versions of the problem. What we have not done yet though is examine the most obvious generalization of the puzzle, namely we have not considered yet the case where more than 2 test eggs are available. Needless to say, treating this more general version of the problem will require the deployment of more elaborate math tools. For this reason we do not include this treatment in the main body of this paper, but rather in the [Appendix](#). The good news is that with the aid of DP it is possible to formulate closed form solutions to the two (Worst Case Scenario and Expected Value) versions of the problem. The Appendix also includes a short discussion on Mother Nature's optimal policy in the context of the Worst Case Scenario and on robustness.

The following is dedicated to those who worry about the relevance of this puzzle to real life applications of OR/MS.

Egg-Dropping mini-Project Write a short report on real-life applications that are similar to this puzzle and that can be solved by algorithms that are similar to those discussed in this paper. If possible find a commercial software package relevant to such applications. **Hint:** According to Wu ([2002](#)):

"... Note: Interestingly, the solution is similar to a commercial algorithm used for stress-testing the reliability of TCP/IP networks. Got this from Spring 2002 CS170, taught by Dr. Satish Rao ... "

Perhaps also relevant to this study is [NASA's Parachuting Egg project](#). Its objective is to design, build, and test a parachute for an egg. This is a kids' version of the real project involving real parachutes designed for NASA's solid rocket boosters.

A module dedicated to an OR/MS perspective of this puzzle can be found at the [tutOR's web site](#), where other modules related to other OR/MS puzzles/games can also be found. In particular, the interested reader may wish to compare this puzzle with the [counterfeit coin puzzle](#).



Acknowledgement

The first draft of this article was written during the author's visit to the Department of Business Administration, Information Systems and Information Management, Technical University of Braunschweig, Braunschweig, Germany (July 28 - September 23, 2002). Further work was done during the author's visit to the Department of Systems Engineering and Engineering Management, Chinese University of Hong Kong, Hong Kong (October 15 - December 15, 2002).

References

- French, S. (1988), *Decision Theory*, Ellis Horwood, Chichester, UK.
- Konhauser J.D.E., D. Velleman, and S. Wagon (1996), *Which way did the Bicycle Go?*, Dolciani Mathematical Expositions - No-18, The Mathematical Association of America.
- Math Forum Problem of the Week 921: The Egg Drop, fall 2000. [\[link\]](#)
- Petkovsek M., H. Wilf, and D. Zeilberger (1996), *A=B*, A K Peters, Ltd., Wellesley, Mass. A free copy can be downloaded from [\[link\]](#).
- Stan Wagon's List of dozen favorite problems/puzzles: [\[link\]](#)
- Sniedovich, M. (2002a), "OR/MS Games: 1. A Neglected Educational Resource", *INFORMS Transactions on Education*, Vol. 2, No. 3, [\[link\]](#).
- Sniedovich, M. (2002), "OR/MS Games: 2. Towers of Hanoi", *INFORMS Transactions on Education*, Vol. 3, No. 1, [\[link\]](#).

Sniedovich, M. (2003), "OR/MS Games: 3. The counterfeit coin problem",
INFORMS Transactions on Education, Vol. 3, No. 1, .

Solution to Math Forum Problem of the Week 921: The Egg Drop. .

tutOR web site: .

tutOR's Egg module: .

William Wu's Riddles web site: .

©INFORMS

	To download a printable version (pdf) of this paper, click here . To download the Adobe Acrobat reader for viewing and printing pdf files, click here .
	To reference this paper, please use: Sniedovich, M. (2003), "OR/MS Games: 4. The Joy of Egg-Dropping in Braunschweig and Hong Kong," <i>INFORMS Transactions on Education</i> , Vol. 4, No 1, http://ite.pubs.informs.org/Vol4No1/Sniedovich/

Appendix

1. N-Egg problem

The DP functional equations derived in the preceding sections can be easily solved (numerically) for cases where $n > 2$. However, the detection of the basic patterns exhibited by the optimal solutions is not so easy any more. It will therefore take a bit more effort to formulate analytical solutions to the general N-egg problem.

Table 5 depicts the values of $W(n,k)$ and $X(n,k)$ for $n = 1,2,3$ and $k = 1,\dots,15$, computed according to (17). Try to detect the patterns exhibited by these objects as n increases.

Table 5

k	W(1,k) W(2,k) W(3,k)			X(1,k)	X(2,k)	X(3,k)
	1	2	3	{1}	{1}	{1}
1	1	1	1	{1}	{1}	{1}
2	2	2	2	{1}	{1,2}	{1,2}
3	3	2	2	{1}	{2}	{2}
4	4	3	3	{1}	{1,2,3}	{1,2,3,4}
5	5	3	3	{1}	{2,3}	{2,3,4}
6	6	3	3	{1}	{3}	{3,4}
7	7	4	3	{1}	{1,2,3,4}	{4}
8	8	4	4	{1}	{2,3,4}	{1,2,3,4,5,6,7}
9	9	4	4	{1}	{3,4}	{2,3,4,5,6,7}
10	10	4	4	{1}	{4}	{3,4,5,6,7}
11	11	5	4	{1}	{1,2,3,4,5}	{4,5,6,7}
12	12	5	4	{1}	{2,3,4,5}	{5,6,7}
13	13	5	4	{1}	{3,4,5}	{6,7}
14	14	5	4	{1}	{4,5}	{7}
15	15	5	5	{1}	{5}	{1,2,3,4,5,6,7,8,9,10,11}

Table 6 depicts the $L(n,k)$ and $Y(n,k)$ values for $n = 1,2,3$ and $k = 1,\dots,15$. Try to detect the patterns exhibited by these values as n increases.

Table 6

k	$L(1,k)$	$L(2,k)$	$L(3,k)$	$Y(1,k)$	$Y(2,k)$	$Y(3,k)$
1	2/2	2/2	2/2	{1}	{1}	{1}
2	5/3	5/3	5/3	{1}	{1,2}	{1,2}
3	9/4	8/4	8/4	{1}	{2}	{2}
4	14/5	12/5	12/5	{1}	{2,3}	{2,3}
5	20/6	16/6	16/6	{1}	{2,3}	{2,3,4}
6	27/7	20/7	20/7	{1}	{3}	{3,4}
7	35/8	25/8	24/8	{1}	{3,4}	{4}
8	44/9	30/9	29/9	{1}	{3,4}	{4,5}
9	54/10	35/10	34/10	{1}	{3,4}	{4,5,6}
10	65/11	40/11	39/11	{1}	{4}	{4,5,6,7}
11	77/12	46/12	44/12	{1}	{4,5}	{4,5,6,7}
12	90/13	52/13	49/13	{1}	{4,5}	{5,6,7}
13	104/14	58/14	54/14	{1}	{4,5}	{6,7}
14	119/15	64/15	59/15	{1}	{4,5}	{7}
15	135/16	70/16	65/16	{1}	{5}	{7,8}

You may wish to use the following interactive module to compute the values of $W(n,k)$, $L(n,k)$, $X(n,k)$ and $Y(n,k)$. You are encouraged to attempt to detect patterns exhibited by the optimal solutions when $n>2$. It is recommended that you start with relatively small values (eg. default values) because the reports generated are quite extensive.

N Test Eggs Module

Scenario : Worst Case Laplace

H = (recommended range 20-200)

**** Solve ****

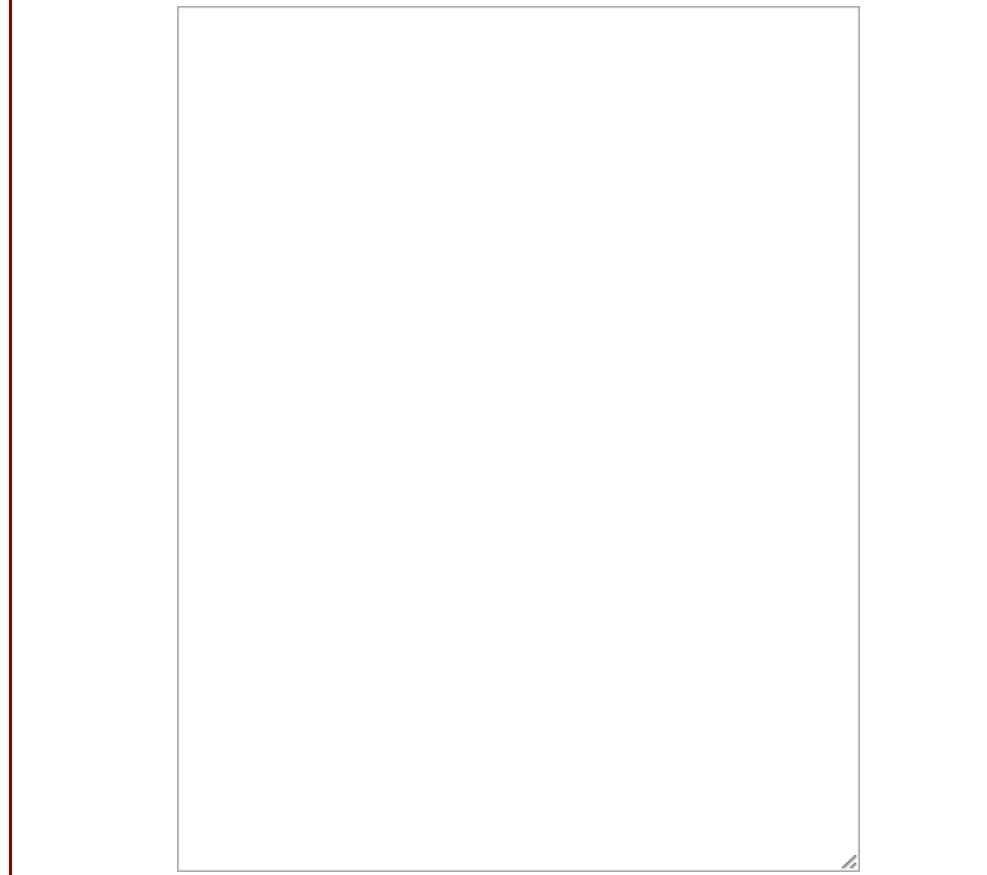
n = (recommended range 3-10)

computation for k = , m = (1<=k<=H, 1<=m<=n)

output in a printer-friendly format (new window)

Full

Align: Left Center Right



Module 4

When you play with the above module you soon discover a treasure: many patterns are hidden in the values of $W(n,k)$, $L(n,k)$, $X(n,k)$ and $Y(n,k)$ when $n > 2$. However, you also soon discover that these patterns are much more subtle than those exhibited by the corresponding values when $n = 2$ which we identified in the preceding sections. A thorough investigation of these new subtle patterns is left to the interested readers. Here we shall just scratch the surface.

2. Basic Patterns

We already detected and analyzed the patterns exhibited by $W(n,k)$, $X(n,k)$, $L(n,k)$ and $Y(n,k)$ when n is fixed ($n = 2$) and k is increasing. In this section we plan to analyze the patterns exhibited by these objects when n increases.

One possible useful outcome of such an analysis is the detection of a simple solution to problems where n is very large.

Worst case Scenario

The pattern hidden in $W(n,k)$ is not so difficult to detect once we observe that for any given value of n , the sequence $Q(n) := (W(n,k) : k = 1, 2, 3, \dots, H)$ is non-decreasing and consists of multiple copies of $j = 1, 2, \dots$. Thus, to completely characterize $W(n,k)$ it is sufficient to determine how many times j appears in $Q(n)$. So let $\text{rep}(j,n)$ denote the number of times j appears in $Q(n)$. We have already established that for $n = 1$ we have $\text{rep}(i,1) = 1$ and for $n = 2$ we have $\text{rep}(i,2) = i$.

In Table 7 we provide values of $\text{rep}(j,n)$ for $n = 1, \dots, 11$ and $j = 1, \dots, 10$. We shall use these values to identify the basic structure of $\text{rep}(j,n)$ so that we can compute it efficiently for any value of n and j . We are primarily interested in the

patterns exhibited by the columns of this table.

Table 7 - rep(i,n)

Table 7 - Rep(j,i))

3	1	3	4	4	4	4	4	4	4	4	4	256	256	256
4	1	4	7	8	8	8	8	8	8	8	8	256	256	256
5	1	5	11	15	16	16	16	16	16	16	16	16	16	16
6	1	6	16	26	31	32	32	32	32	32	32	32	32	32
7	1	7	22	42	57	63	64	64	64	64	64	64	64	64
8	1	8	29	64	99	120	127	128	128	128	128	128	128	128
9	1	9	37	93	163	219	247	255						
												10	10	46
												1	11	176
												511	512	512
												1023	1024	

The following pattern completely characterizes the $\text{rep}(j,n)$ values depicted in Table 7:

Lemma 13

$$\text{rep}(j+1,n+1) = \text{rep}(j,n+1) + \text{rep}(j,n), j = 1,2, \dots ; n = 1,2, \dots \quad (29)$$

$$\text{rep}(j,1) = 1, j = 1,2, \dots \quad (30)$$

$$\text{rep}(1,n) = 1, n = 1,2, \dots \quad (31)$$

Furthermore, the *columns* of Table 7 have the following interesting and constructive features:

Lemma 14

$$\text{rep}(j,n) = 2^{j-1}, n = 1,2, \dots ; j = 1, \dots, n \quad (32)$$

$$\text{rep}(n+1,n) = 2^{n-1} - 1, n = 1, \dots, 10 \quad (33)$$

Upon solving the recurrence relation specified by (29)-(31) we obtain

Theorem 5

$$\text{rep}(j,n) = B(j-1,0) + B(j-1,1) + \dots + B(j-1,n-1) + B(j-1,n), j = n+1, n+2, \dots \quad (34)$$

where $B(p,m) = (p!)/(m!(p-m)!)$, for $p \geq m$, are the beloved *binomial coefficients*. It is convenient in our context to set $B(p,m) = 0$. Here then is the famous Pascal Triangle for $p = m = 11$ displayed in a layout that is convenient for the computation of $\text{rep}(i,n)$.

Table 8 - $B(j,n)$

j \ n	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	0
3	1	2	1	0	0	0	0	0	0	0	0
4	1	3	3	1	0	0	0	0	0	0	0
5	1	4	6	4	1	0	0	0	0	0	0
6	1	5	10	10	5	1	0	0	0	0	0
7	1	6	15	20	15	6	1	0	0	0	0
8	1	7	21	35	35	21	7	1	0	0	0
9	1	8	28	56	70	56	28	8	1	0	0
10	1	9	36	84	126	126	84	36	9	1	0
11	1	10	45	120	210	252	210	120	45	10	1

So it is very easy to compute the values of $\text{rep}(j,n)$ regardless of whether $n \geq j$ or $j \geq n$. Recall that $B(a,b+1) = B(a,b)(a-b)/(b+1)$ so that the right-hand side of (34) can be evaluated incrementally, rather than using the definition of B explicitly. Details regarding numerical evaluation of binomial sums of this type can be found in Petkovsek et al's (1996) free book entitled *A = B*.

It is also not too difficult to show that

Lemma 15

$$\text{rep}(n+1, j+1) = 1 + \text{rep}(1, n) + \text{rep}(2, n) + \dots + \text{rep}(j, n) \quad (35)$$

The implication is that it is very easy to compute the values of $W(k, n)$ using the relevant $\text{rep}(j, n)$ values. In particular,

Lemma 16

$$W(n, k) = \min\{j \in \{1, 2, 3, \dots\} : \text{rep}(1, n) + \text{rep}(2, n) + \dots + \text{rep}(j, n) \geq k\} \quad (36)$$

so (35)-(36) yield:

Theorem 6

$$W(n, k) = \min\{j \in \{1, 2, 3, \dots\} : \text{rep}(j+1, n+1) > k\} \quad (37)$$

For instance, suppose that $n = 7$ and $k = 300$. Then from Table 7 we see that the smallest integer j such that $\text{rep}(j+1, 8) \geq 300$ is $j = 9$. Hence we conclude that $W(300, 7) = 9$.

In view of (32), it follows that if $2^n > k$, then $W(n, k) = 1 + \text{Integer}(\log_2 k)$. This means that for a given n , the value of $W(n, k)$ is increasing very slowly with k , if at all. By the same token, for a small k , the value of $W(n, k)$ is decreasing very slowly with n , if any. In particular, $W(n, k)$ is constant with n for $k \leq 2^n$.

Because $W(n, k)$ is not strictly increasing with k , it does not have a proper inverse with respect to k . So we have to settle for the "first appearance of" $W(n, k)$ in $Q(n)$. Thus, let $\text{First}(j, n)$ denote the smallest value of k such that $j = W(n, k)$, that is let $\text{First}(j, n) := \min\{k : W(n, k) = j\}$. By definition then,

Lemma 17

$$\text{rep}(j, n) = \text{First}(j+1, n) - \text{First}(j, n), j = 1, 2, \dots \quad (38)$$

$$\text{First}(j, n) = \text{rep}(j, n+1), j = 1, 2, \dots \quad (39)$$

$$W(n, i) = W(n, j) = r, \text{ for all } \text{First}(r, n) \leq i, j \leq \text{First}(r+1, n), (r = 1, 2, \dots) \quad (40)$$

Using these basic properties of $\text{rep}(j, n)$ and $\text{First}(j, n)$, it is not too difficult to completely characterize $X(n, k)$ by $\text{First}(j, n)$ and $\text{rep}(j, n)$. For this purpose let $j(k)$ denote the smallest value of j such that $\text{First}(j, n) \geq k$ and let $q(k)$ denote the difference between k and $j(k)$ so that $k = j(k) + q(k)$. More formally, let

$$j(k) := \min\{j = 1, 2, \dots : \text{First}(j, n) \geq k\}, k = 1, 2, \dots \quad (41)$$

$$q(k) := k - j(k), k = 1, 2, \dots \quad (42)$$

It is not difficult to show that $(j(k), q(k))$ is lexicographically strictly monotonic with k , hence the representation $k = j(k) + q(k)$ is unique. Using this representation we can completely characterize $X(n, k)$ as follows:

Lemma 18

$$X(n, k) = \{1, 2, \dots, \text{rep}(j(k), n)\}, \text{ if } q(k)=0 \quad (43)$$

$$X(n, k) = \{\text{rep}(j(k), n)\}, q(x) = \text{rep}(j(k), n)-1 \quad (44)$$

$$X(n, j(k)+q) = \{1+q, \dots, \text{rep}(j(k), n)\}, 1 \leq q < \text{rep}(j(k), n) \quad (45)$$

or in short

$$X(n, j(k) + q) = \{1+q, \dots, \text{rep}(j(k), n)\}, 0 \leq q < \text{rep}(j(k), n) \quad (46)$$

Suppose that n is large so that $k < 2^{n-1}$. Then from (32) and (41)-(43) it follows that one of the optimal policies is "divide by half".

The following module computes the values of $W(n, k)$ and $X(n, k)$ using (38)-(46).

Quick and Elegant

n = , W(n,k) = * Compute * * Explain *

k = , X(n,k) =

Specify the desired value of k and n, then click the "Compute" button.
The modules will compute the values of W(n,k) and X(n,k).
Default values for k and n are provided.

Module 5

Given that we can compute $W(n,k)$ and $X(n,k)$ semi-analytically via (29)-(46), why do we have to bother with the tedious numerical solution outlined by the DP functional equation? The answer is: *indeed, we do not have to bother about the DP functional equation (17) and the numerical algorithm that it outlines.*

It must be stressed, though, that the semi-analytical solution owes its existence to the DP functional equation. It is the numerical algorithm that produced the results that facilitated the detection the crucial patterns. Furthermore, the formal proofs of the key results presented in this section rely on the DP functional equation. In short, the semi-analytical solution outlined in this section is not an alternative to the DP functional equation, but rather a consequence of it.

Laplace's Scenario

To formulate the patterns exhibited by $L(n,k)$ it is useful to write it as $L(n,k)=\text{Nom}(n,k)/\text{Denom}(n,k)$. The pattern for $\text{Denom}(n,k)$ is obvious, namely $\text{Denom}(n,k) = k+1$ so let us focus on $\text{Nom}(n,k)$.

A quick experiment with Module 4 suggests that the clue to the pattern exhibited by $\text{Nom}(n,k)$ lies in the pattern exhibited by $\text{Dif}(j,n) := \text{Nom}(n,j) - \text{Nom}(n,j-1)$. Table 9 depicts the values of $\text{Dif}(j,n)$ for $j = 1, \dots, 16$ and $n = 1, \dots, 11$.

$$\text{Dif}(j,n) := \text{Nom}(n,j) - \text{Nom}(n,j-1)$$

Table 9

j \ n	1	2	3	4	5	6	7	8	9	10	11
1	2	2	2	2	2	2	2	2	2	2	2
2	3	3	3	3	3	3	3	3	3	3	3
3	4	3	3	3	3	3	3	3	3	3	3
4	5	4	4	4	4	4	4	4	4	4	4
5	6	4	4	4	4	4	4	4	4	4	4
6	7	4	4	4	4	4	4	4	4	4	4
7	8	5	4	4	4	4	4	4	4	4	4
8	9	5	5	5	5	5	5	5	5	5	5
9	10	5	5	5	5	5	5	5	5	5	5
10	11	5	5	5	5	5	5	5	5	5	5
11	12	6	5	5	5	5	5	5	5	5	5
12	13	6	5	5	5	5	5	5	5	5	5
13	14	6	5	5	5	5	5	5	5	5	5
14	15	6	5	5	5	5	5	5	5	5	5
15	16	6	6	5	5	5	5	5	5	5	5
16	17	7	6	5	5	5	5	5	5	5	5

Let $G(n)$ denote the list having the pattern exhibited by the n -th column of Table 9 and let $REP(j,n)$ denote the number of times j appears in $G(n)$. Table 10 depicts the values of $REP(j,n)$ for $j = 1, \dots, 12$ and $n = 1, \dots, 11$.

Table 10 - $REP(j,n)$

$j \setminus n$	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	1
3	1	2	2	2	2	2	2	2	2	2	2
4	1	3	4	4	4	4	4	4	4	4	4
5	1	4	7	8	8	8	8	8	8	8	8
6	1	5	11	15	16	16	16	16	16	16	16
7	1	6	16	26	31	32	32	32	32	32	32
8	1	7	22	42	57	63	64	64	64	64	64
9	1	8	29	64	89	120	127	128	128	128	128
10	1	9	37	83	143	209	247	255	256	256	256
11	1	10	46	120	226	352	456	502	511	512	512
12	1	11	56	166	346	578	808	958	1013	1023	1024

Exercise 6

Find the relationship between $rep(j,n)$ and $REP(j,n)$.

Exercise 7

Express $L(n,k)$ in terms of $REP(j,n+1)$ and/or $rep(j,n+1)$ for suitable values of j .

To identify the structure of $Y(n,k)$ we present in Table 10 the values of $Y(n,k)$ for $n = 1, 2, 3, 4$ and $j = 1, \dots, 15$ as well as the respective values of $X(4,k)$.

Table 11

k	$Y(1,k)$	$Y(2,k)$	$Y(3,k)$	$Y(4,k)$	$X(4,k)$
1	{1}	{1}	{1}	{1}	{1}
2	{1}	{1,2}	{1,2}	{1,2}	{1,2}
3	{1}	{2}	{2}	{2}	{2}
4	{1}	{2,3}	{2,3}	{2,3}	{1,2,3,4}
5	{1}	{2,3}	{2,3,4}	{2,3,4}	{2,3,4}
6	{1}	{3}	{3,4}	{3,4}	{3,4}
7	{1}	{3,4}	{4}	{4}	{4}
8	{1}	{3,4}	{4,5}	{4,5}	{1,2,3,4,5,6,7,8}
9	{1}	{3,4}	{4,5,6}	{4,5,6}	{2,3,4,5,6,7,8}
10	{1}	{4}	{4,5,6,7}	{4,5,6,7}	{3,4,5,6,7,8}
11	{1}	{4,5}	{4,5,6,7}	{4,5,6,7,8}	{4,5,6,7,8}
12	{1}	{4,5}	{5,6,7}	{5,6,7,8}	{5,6,7,8}
13	{1}	{4,5}	{6,7}	{6,7,8}	{6,7,8}
14	{1}	{4,5}	{7}	{7,8}	{7,8}
15	{1}	{5}	{7,8}	{8}	{8}

Exercise 8

Express $Y(n,k)$ in terms of $rep(j,n)$ and $First(j,n)$ for suitable values of j . Hint: First find the values of k where $Y(n,k) = X(n,k)$ and both are singletons (See [Lemma 18](#) for inspiration).

The comments made above regarding the semi-analytical solution for the Worst Case version of the problem applies here as well: the semi-analytical solution for the Laplace version of the problem is not an alternative for the DP solution, it is a derivation of it. Here is a module based on the results presented in this sub-section:

Quick and Elegant

Compute

Explain

Decimal

$n =$

, $L(n,k) =$

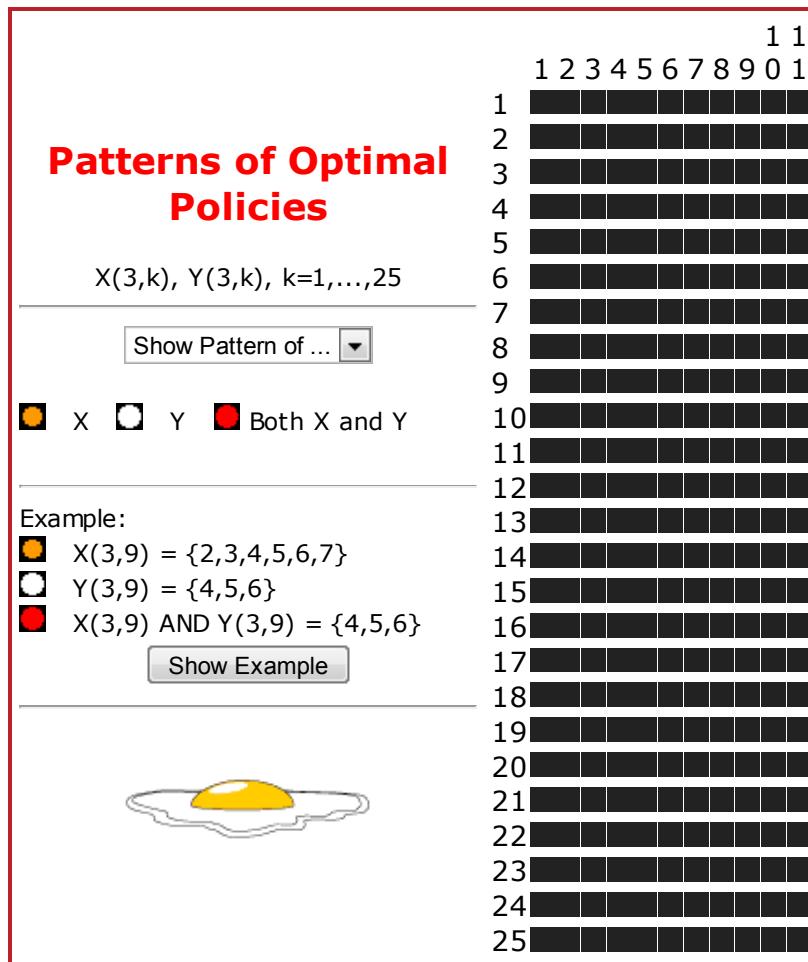
k = , Y(n,k) =

Specify the desired value of k and n, then click the "Compute" button.
The modules will compute the values of L(n,k) and Y(n,k).

Default values for k and n are provided.

Module 6

The following module illustrates the relationship between the essential structures of $Y(n,k)$ and $X(n,k)$. Each possesses a wave -like pattern and they share the same end points of the waves- where both are the same singletons. The shape of the waves are different. The $X(n,k)$ waves are right angle, equilateral triangles whereas the $Y(n,k)$ waves are parallelograms.



Module 7

The following question is unavoidable:

Exercise 9

Determine whether there is a robust optimal solution for any arbitrary n . Hint: determine whether $Y(n,k)$ is a subset of $X(n,k)$ for all k and n , observing that if this is true, then any optimal Laplace policy is robust.

3. What about Mother Nature?

It is instructive, sometimes absolutely essential, to explicitly formulate the optimal solution employed by Mother Nature under the Worst Case Scenario. In particular,

suppose that we wish to construct a simulation module for the problem. For this purpose we have to know how Mother Nature reacts to decision made by the player (us!).

Let then $z(n,k,x)$ denote Mother Nature's optimal reaction to the our decision to drop the egg in the next trial from the relative position x , given that we have a problem of size k with n test eggs. Recall that $z(n,k,x)$ can take either one of two values "break" or "do not break". Which one should Mother Nature select?

The answer is as follows: if Mother Nature selects $z(k,n,x) = \text{"break"}$, then after the trial we will be left with a problem of size $k' = x-1$ with $n' = n-1$ test eggs. If Mother Nature selects $z(n,k,x)$ to be "do not break", then after the next trial we will be left with a problem of size $k'' = k-x$ with $n'' = n$ test eggs. Thus, from Mother Nature's perspective the choice is between $(k' = x-1, n' = n-1)$ and $(k'' = k-x, n'' = n)$. Since Mother Nature plays against us, She will decide on this matter by comparing $W(n-1,x-1)$ with $W(n,k-x)$ and select "break" if the former is larger than the latter. If $W(n,k-x)$ is larger than $W(n-1,x-1)$ then She will select "do not break". And if $W(n-1,x-1) = W(n,k-x)$, then Mother Nature will regard both choices as optimal. She may decide to flip a coin to break this tie for best.

The following then is a complete characterization of Mother Nature's optimal policy for the Worst Case version of the problem:

Lemma 19

Assume that we face a problem of size k , have n test eggs and decide to drop the nest egg from position $x \leq k$. Then Mother Natures optimal response is as follow:

If $W(n-1,x-1) > W(n,k-x)$ then $z(n,k,x) = \text{"Break!"}$
If $W(n-1,x-1) < W(n,k-x)$ then $z(n,k,x) = \text{"Do not break!"}$
If $W(n-1,x-1) = W(n,k-x)$ then $z(n,k,x)$ can be either "Break" or "Do not break!"

Exercise 10

Determine structure of $z(n,k,x)$ under the Worst Case Scenario. In particular, try to identify any patterns in $z(n,k,x(n,k))$, where $x(n,k)$ is an element of $X(n,k)$, namely when DM is using an optimal policy.

Although this can be a difficult task, the recipe given by Lemma 21 is straight forward. It is therefore very easy to simulate Mother Nature's behaviour.

Recall that in the context of the Laplace's version of the problem, Mother Nature determines the value of C according to a uniform distribution on $\{0,1,\dots,H\}$. Thus, given (n,k,x) Mother Nature will decide to break the egg with probability $x/(k+1)$. In particular, the probability that the egg will **not** break when it is dropped from the highest position ($x = k$) is $1/(k+1)$.

