An approxiate solution to a locked box game using failed recursions

Henry Bottomley
5 Leydon Close, London SE16 5PF
se16@btinternet.com

March 10, 2006

Abstract

The solution to a particular two player game can be derived by using a reverse recursion, using approximate solutions and adjusting them when the results of the recursion are implausibile.

1 Introduction and game rules

In a usenet discussion on *sci.math* and *alt.math.recreational* called "A Recursive Game" in December 2005, Frank J. Lhota proposed a game involving two individuals, one of whom we will call the house and the other the player, and two transparent but invisibly lockable boxes. In each round, the house would lock one of the boxes and unlock the other. The player could see the contents of the boxes but not which was locked, and would then try to open one of the boxes; if it was unlocked, the player would obtain its contents and the game would end, while if it was locked, the house would add one dollar to the unlocked box and then move to the next round by choosing which box to lock next. Initially both boxes start empty. What is the value of the game to the player (i.e. the expected winnings, or the fair entry fee to be charged by the house) if both play optimally? How long should the game last?

2 Payoff tables

Broadly following the notation of the usenet discussion set out by the user "quasi" (who also did some of the analysis), if in a particular position one box has n dollars and the other m dollars, we can call the value of the position $v[n, m]$. The payoff table for the player looks like

	House locks first box	House locks second box
Player tries to open first box	$v[n, m+1]$	n
Player tries to open second box	m	$v[n+1, m]$

We are trying to calculate $v[0, 0]$ and that depends on the first round where the payoff matrix is

	House locks first box	House locks second box
Player tries to open first box	$v[0, 1]$	0
Player tries to open second box	0	$v[1, 0]$

Clearly $v[1, 0] = v[0, 1]$ since we simply reverse the payoff matrix between the first and second boxes; this value is positive since the player has some chance of winning something. The symmetry of this first round means that the optimal tactic is for each player is to choose each box with equal probability, so there is half a chance of the game ending in the first round with no win for the player and half a chance of it continuing. So we can conclude that $v[0, 0] = v[1, 0]/2$.

In general, $v[n, m] = v[m, n]$ since we simply reverse the payoff matrix between the first and second boxes. We can go further and note that the player can be certain of winning at least the smaller amount in the two boxes and that both the house and the player need only make their decisions based on the difference between the two boxes. So

$$v[n+m, m] = v[m, n+m] = m + v[n, 0]$$

and since only the difference matters we can simplify the notation further by writing $v[n]$ for $v[n, 0]$ or $v[0, n]$. So basing the payoff table on the assumption one of the boxes is empty, it becomes

	House locks valuble box	House locks empty box
Player tries to open valuble box	$1 + v[n-1]$	n
Player tries to open empty box	0	$v[n+1]$

This is enough to put some useful constraints on $v[n]$. For example, we have

$$v[n] \leq \max\{1 + v[n-1], 0\}$$

since the house can decide to lock the more valuable box. Similarly

$$v[n] \geq \min\{1 + v[n-1], n\}$$

since the player can try to open the more valuable box. Combining these gives

$$n \leq v[n] \leq 1 + v[n-1].$$

3 Minimax solutions

These constraints mean we are not going to find a simple strategy for each round of the game and so we must look for a mixed strategy: the player and the house must randomise their choices to play optimally. Intuitively this is not a surprise, since if the house has clearly decided to lock a particular box the player will generally try to open that box, adding to the value of the game and seeing more money being put in the boxes. Similarly if the player has clearly decided to try to open a particular box, the house will try to unlock that box, reducing the value of the game and avoiding having to add more money overall to the boxes.

Writing $p[n]$ as the probability that the house locks the emptier box when the difference between them is n, and $q[n]$ the probability that the player tries to open the emptier box, we find the payoff matrix above gives the value (for $n > 0$)

$$v[n] = p[n]q[n]v[n+1] + p[n](1-q[n])n + (1-p[n])(1-q[n])(1+v[n-1])$$

and we need to find the minimax solutions. Fortunately this value is linear in both $p[n]$ and $q[n]$, implying that the following are true simultaneously for optimal $p[n]$ and $q[n]$:

$$\begin{aligned}
v[n] &= p[n]v[n+1] \\
v[n] &= q[n]v[n+1] + (1-q[n])n \\
v[n] &= p[n]n + (1-p[n])(1+v[n-1]) \\
v[n] &= (1-q[n])(1+v[n-1])
\end{aligned}$$

Eliminating $p[n]$ and $q[n]$ gives

$$v[n] + v[n-1]v[n] + v[n]v[n+1] = v[n+1] + v[n-1]v[n+1] + v[n]n$$

which on the face of it is not particularly helpful. There is little point in trying to solve this for $v[n]$ in terms $v[n-1]$ and $v[n+1]$, so we should instead look at one of

$$\begin{aligned} v[n-1] &= \frac{v[n](v[n+1]+v[n]-n)-v[n+1]}{v[n+1]-v[n]} \\ v[n+1] &= \frac{v[n](1+v[n-1]-n)}{1+v[n-1]-v[n]} \end{aligned}$$

At first glance both of these fail to give us a starting point for a recursion, indeed doubly so since we need two values to calculate each new value.

Fortunately, if we choose the second recursion we can halve our problem since we know from the first round of the game that

$$v[1] = 2v[0].$$

This still is not enough since the calculations soon get complicated

$$v[2] = \frac{2v[0]^2}{1-v[0]}$$

$$v[3] = \frac{2v[0]^2(2v[0]-1)}{1+v[0]-4v[0]^2}$$

etc.

But it may point the way to go.

4 Testing approximate solutions and finding the value of the game

Perhaps if we guessed a value for $v[0]$, doubled the guess for $v[1]$, and then applied the recursion, we could see what values were produced.

Suppose for example, we tried $v[0] = 0.5$. We would get

$v[0]$	0.5
$v[1]$	1
$v[2]$	1

We can stop there, because we know that we should have $n \leq v[n]$ from earlier. So instead suppose we start with $v[0] = 0.75$. This time we would get

$v[0]$	0.75
$v[1]$	1.5
$v[2]$	4.5

Again we can stop there because we should have $v[n] \leq 1 + v[n-1]$.

But we are now on the right approach. 0.625 gets us one step further with

$v[0]$	0.625
$v[1]$	1.25
$v[2]$	2.0833333333
$v[3]$	3.125

which is unfortunately too high, but suggests an algorithm: whenever we have to stop the recursion because $v[n]$ seems to be too low compared with n, we should restart with a higher guess for $v[0]$; and whenever we have to stop because $v[n]$ seems to be too high compared with $1+v[n-1]$, we should restart with a lower guess for $v[0]$.

This algorithm and recursion indeed works to produce more accurate estimates of $v[0]$, though we still soon get unnaceptable figures for some $v[n]$. Trying 0.6245357205 gives (with decimals truncated down if necessary from now on)

$v[0]$	0.6245357205
$v[1]$	1.249071441
$v[2]$	2.0776669711
$v[3]$	3.0191015838
$v[4]$	4.0038064511
$v[5]$	5.0002210521
$v[6]$	5.3085018300

while trying 0.6245357206 gives

$v[0]$	0.6245357206
$v[1]$	1.2490714412
$v[2]$	2.0776669723
$v[3]$	3.0191016059
$v[4]$	4.0038079734
$v[5]$	5.0007192467
$v[6]$	6.1651961331

showing that with even ten decimal places of accuracy for $v[0]$ fails to give a credible figure for $v[6]$. But we can reach any degree of precsion we want, providing we start with an precise enough value for $v[0]$ and our calculations are accurate enough.

Here are the first 2000 digits of $v[0]$:
0.62453 57205 82943 32938 46829 38284 32119 58793 87410 61271
18900 21598 60686 28558 85492 65561 10883 05132 56736 67572
04253 36594 40259 76637 19538 02045 40240 32455 39981 81163
83759 03475 76459 45776 22499 93720 28404 76635 10561 76153
71558 30685 66948 59734 98859 05566 26222 10903 88587 05734
46176 34262 88698 35117 65609 54784 43629 40815 31105 88772
70526 46601 66196 01565 06338 54681 43123 24510 57392 66916
07789 21430 22626 96101 69744 37273 04614 84888 99698 35311
27217 87016 43759 58387 68004 83165 92446 68277 54918 21521
03627 84866 66579 10524 75388 51449 49569 48084 01182 52827
86261 35863 25240 32903 41106 18983 79587 12720 48828 72942
87648 02388 40268 84505 45443 32978 79667 72092 31954 31145
14330 09332 47954 79505 81609 27497 37793 11671 57438 62644
02263 08906 97711 62830 43159 38877 17503 68691 88010 61473
41635 94511 02995 26847 84236 72165 41344 00852 36182 39212
95401 07906 59758 01377 79115 29771 66628 24128 81128 32528
66381 53184 50537 35445 03953 48589 67934 97827 42892 80837
94954 24338 06443 44988 70076 67107 79544 72662 58011 74640
97850 11253 74730 66587 60120 69738 96715 07782 98803 01246
33408 49831 67023 55099 90828 11447 91776 88165 60010 76063
85980 75564 82929 74039 42493 81335 01410 88796 47101 85306
50849 13201 22942 41144 73699 04392 04855 10001 05775 53353
74032 38322 23347 28243 40190 28178 08055 36679 17783 70532
39048 70730 50566 34974 11743 66549 79863 86616 58447 54387
39928 00354 27900 42927 24065 46616 02916 90819 68064 22532
46153 08111 05163 67128 06216 37655 96921 11957 81319 52978
83414 85287 33519 80643 12355 93091 88785 26848 16663 42762
73203 58495 39914 19431 57115 98186 96726 05198 63597 43687
26366 61885 05773 91301 13945 77986 36934 89741 18429 78908
96856 05305 29705 05493 27753 06253 41079 76702 09483 87535
91267 44289 80699 41628 15008 98211 22656 43299 69480 92264
33599 04901 77622 63856 47461 34643 17983 53047 05460 55705
09579 38264 97131 67150 51440 17162 89857 84593 32582 14763
78222 99581 64400 66357 71839 68487 44236 60215 27487 55533
23860 20970 56799 71014 61202 39578 13607 96807 79863 17417
86091 22109 47925 69171 29925 58508 86894 49579 35867 06416
88391 15094 28366 76219 94641 63145 27899 83783 45226 93056
26646 83121 54513 75380 84960 18543 88729 02409 53587 93295
02179 25961 73326 97575 94327 81492 84781 67984 93800 15075
94656 09521 66643 78964 91108 19045 01155 51281 74675 01833
and in a sense we now have a solution for the value of the game.

5 Optimal strategies

So far this has not yet told us much about the optimal strategies for the house and player. Going back to the minimax solution, we can find an easy way of calculating $p[n]$ the probability that the house locks the emptier box when the difference between them is n, and $q[n]$ the probability that the player tries to open the emptier box. We can use

$$\begin{aligned} p[n] &= \frac{v[n]}{v[n+1]} \\ q[n] &= \frac{v[n]-n}{v[n+1]-n} \end{aligned}$$

to produce the following table

n	$v[n]$	$p[n]$	$q[n]$
0	0.6245357205	0.5	0.5
1	1.2490714411	0.6011894388	0.2311209748
2	2.0776669721	0.6881739159	0.0762112158
3	3.0191016021	0.7540575916	0.0190291446
4	4.0038077138	0.8006599767	0.0038053002
5	5.0006342643	0.8334264590	0.0006342069
6	6.0000906018	0.8571544134	0.0000906008
7	7.0000113251	0.8750012780	0.0000113251
8	8.0000012583	0.8888890162	0.0000012583
9	9.0000001258	0.9000000115	0.0000001258
10	10.0000000114	0.9090909100	0.0000000114

For large n, we find $v[n]$ is close to but above $n + 0.4566281571/(n+1)!$. So $p[n]$ gets close to $n/(n+1)$ and $q[n]$ gets close to $0.4566281571/(n+1)!$.

This tells us that when the difference between the boxes is great, the player usually tries to open the more valuable box, and that the house tends to leave that box unlocked, though not with quite such a high probability. There is an intutitive explanation for that behaviour: the player is only slightly disadvantaged when playing a strategy of always picking the more valuable box when the difference is large, but would be facing a substantial opportunity cost when successfully opening the emptier box, while the house has to balance the small extra cost of leaving the more valuable box unlocked but unopened against the less likely but more profitable opportunity of having the emptier box opened.

6 Expected length of the game

Given an accurate enough estimate of the value of the game, we can also calculate some other information about the game when played optimally, such as the expected number of rounds, the distribution of the number of rounds, and the distribution of how much the player takes from the boxes.

For example, if $t[n]$ is the expected number of future rounds when the difference between the boxes is n then we have

$$\begin{aligned} t[0] &= 1 + t[1]/2 \\ t[n] &= 1 + p[n]q[n]t[n+1] + (1-p[n])(1-q[n])(t[n-1]) \end{aligned}$$

and since we now know $p[n]$ and $q[n]$ we can use the same techniques of recursive approximation with

$$\begin{aligned} t[1] &= 2t[0] - 2 \\ t[n+1] &= \frac{t[n] - (1-p[n])(1-q[n])t[n-1] - 1}{p[n]q[n]} \end{aligned}$$

with the requirement that

$$1 \leq t[n]$$

since we know that if the game has not finished then we must play at least one more round; in this game we also have

$$t[n+1] \leq t[n]$$

since as n increases the game is likely to end more quickly since the house and the player both increase the likelihood of the more valuable box being opened. This method will give a value of $t[0]$ of just over 1.9023718998 (slightly less expected time than flipping a coin and waiting until it comes up heads). Even using high precision values of $p[n]$ and $q[n]$ we would still get an implausible value for $t[6]$ starting from $t[0]$ with ten decimal places of accuracy.

Here are the first 200 decimal places of the true value for $t[0]$:

1.90237 18998 37624 04708 14619 39689 78343 00723 57222 99043
52580 57263 24893 39292 80652 19959 58614 51160 99647 56773
42930 58758 81813 44989 31395 82703 20343 62351 08945 20891
82465 09104 41578 34861 67959 86409 84083 59844 40650 19046

giving the following truncated values:

$t[0]$	1.9023718998
$t[1]$	1.8047437996
$t[2]$	1.5934512036
$t[3]$	1.4028392395
$t[4]$	1.2822768795
$t[5]$	1.2140782081
$t[6]$	1.1734990523
$t[7]$	1.1466953926
$t[8]$	1.1274115374
$t[9]$	1.1127412512
$t[10]$	1.1011583046

7 Checking results and producing further information

We can also get good estimates of $t[n]$ by an alternative more direct method, simply by calculating the probabilities of reaching each possible position since we know $p[n]$ and $q[n]$.

Clearly reaching the position $[0,0]$ has a probability of 1 since the game starts in that position, while $[1,0]$ and $[0,1]$ each have a probability of 0.25 of being reached, and there is a probability of 0.5 of finishing after just one round and the player leaving empty-handed.

Similarly, the positions $[2,0]$ and $[0,2]$ each have a probability 0.0347368722 of being reached, while $[1,1]$ has a probability of 0.1533185377 of being reached; there is a probability of 0.2772077176 of finishing after exactly two rounds, with a probability of 0.2311209748 of the player taking one dollar in the second round and a probability of 0.0460867428 of the player being left empty-handed in the second round.

Continuing this calculation further allows the possibility of finding that the probability that the game ends when the player chooses the more valuable box which the house has unlocked is 0.345640505868, when the player chooses the emptier box which the house has unlocked is 0.0592654469, and when the boxes are equal and the player happens to have chosen the unlocked box is 0.5950940471 (most likely in the first round).

The probability of the game ends after a particular number of rounds is

Number of rounds in game	Probability
1	0.5
2	0.2772077176
3	0.1224766469
4	0.0563087341
5	0.0243016050
6	0.0110695279
7	0.0047702535
8	0.0021715786
9	0.0009357196
10	0.0004259541

allowing the value of $t[0]$ to be checked, while the probability of winning different amounts in the game is

Amount won	Probability
0	0.5477548539
1	0.3170345118
2	0.1054875782
3	0.0238104294
4	0.0047516720
5	0.0009332202
6	0.0001830643
7	0.0000359078
8	0.0000070432
9	0.0000013815
10	0.0000002709

enabling us to check the value of $v[0]$ calculated earlier.