Stochastic Games and Dynamic Programming

Henk Tijms

1. Introduction

Stochastic games are fun and instructive for teaching purposes on one hand and involve challenging research questions on the other hand. A basic tool for analysing stochastic games that involve a sequence of actions to be taken is the method of dynamic programming. This recursive approach is also known as the method of backward induction and is a computational tool for optimisation problems in which a sequence of interrelated decisions must be made in order to maximise reward or minimise cost. As a simple but illustrative example, consider the game of rolling a fair die at most five times. You may stop whenever you want and receive as a reward the number shown on the die at the time you stop. What is the stopping rule that maximises your expected payoff in this optimal stopping game? To answer this question, the idea is to consider a sequence of nested problems having planning horizons of increasing length. For the one-roll problem in which only one roll is permitted, the solution is trivial. You stop after the first roll and your expected payoff is $1\times\frac{1}{6}+2\times\frac{1}{6}+\cdots+6\times\frac{1}{6} = 3.5$. In the two-roll problem, you stop after the first roll if the outcome of this roll is larger than the expected value 3.5 of the amount you get if you do not stop but continue with what is an one-roll game. Hence, in the two-roll problem, you stop if the first roll gives a 4, 5, or 6; otherwise, you continue. The expected payoff in the two-roll game is $\frac{1}{6}\times 4 + \frac{1}{6}\times 5 + \frac{1}{6}\times 6 + \frac{3}{6}\times 3.5 = 4.25$. Next consider the three-roll problem. If the first roll in the three-roll problem gives an outcome larger than 4.25, then you stop; otherwise, you do not stop and continue with what is a two-roll game. Hence the expected payoff in the three-roll problem is $\frac{1}{6}\times 5+\frac{1}{6}\times 6+\frac{4}{6}\times 4.25 = 4.67$. Knowing this expected payoff, we can solve the four-roll problem. In the four-roll problem you stop after the first roll if this roll gives a 5 or 6; otherwise, you continue. The expected payoff in the four-roll problem is $\frac{1}{6}\times 5+\frac{1}{6}\times 6+\frac{4}{6}\times 4.6667 = 4.944$. Finally, we find the optimal strategy for the original five-roll problem. In this problem you stop after the first roll if this roll gives a 5 or 6; otherwise, you continue. The maximal expected payoff in the original problem is $\frac{1}{6}\times 5+\frac{1}{6}\times 6+\frac{4}{6}\times 4.944 = 5.129$.

The above method of backward induction decomposes the original problem in a series of nested problems having planning horizons of increasing length. Each nested problem is simple to solve and the solutions of the nested problems are linked by a recursion. The above argument can be formalised as follows. For $k = 1, 2, \ldots, 5$, define

$f_k(i)$ = the maximal expected payoff if still k rolls are permitted and the outcome of the last roll is i,

where $i = 0, 1, \ldots, 6$. This function is called the value-function. It enables us to compute the desired maximal expected payoff $f_5(0)$ and the optimal strategy for achieving this expected payoff in the five-roll problem. This is done by applying the recursive equation

$$f_k(i) = \max\left[i, \frac{1}{6}\sum_{j=1}^{6} f_{k-1}(j)\right]$$

for $0 \le i \le 6$, where k runs from 1 to 5. The recursion is initialised with $f_0(j) = j$ for all j.

The method of backward induction is very versatile, and does not require that the outcomes of the successive experiment are independent of each other. As an example, take the following game. You take cards, one at a time, from a thoroughly shuffled deck of 26 red and 26 black cards. You may stop whenever you want and your payoff is the number of red cards drawn minus the number of black cards drawn. What is the maximal expected value of the payoff? The approach is again to decompose the original problem in a sequence of smaller nested problems. Define the value function $E(r, b)$ as the maximal expected payoff you can still achieve if r red cards and b black cards are left in the deck. Using conditional expectations, we can establish the recursive scheme

$$E(r,b) = \max\left[b-r,\ \frac{r}{r+b}E(r-1,b)+\frac{b}{r+b}E(r,b-1)\right].$$

The desired maximal expected $E(26,26)$ is obtained by "backward" calculations starting with

$$E(r,0) = 0 \quad \text{and} \quad E(0,b) = b\,.$$

The maximal expected payoff is $E(26, 26) = 2.6245$. The optimal decisions in the various states can be summarised through threshold values β_k: stop if the number of red cards drawn minus the number of black cards drawn is β_k or more after the kth draw; otherwise, continue. The numerical values of the β_k are $\beta_1 = 2$, $\beta_2 = 3$, $\beta_3 = 4$, $\beta_4 = 5$, $\beta_5 = 6$, $\beta_6 = 5$, $\beta_7 = 6$, $\beta_8 = 7$, $\beta_9 = 6$, $\beta_{2m} = 5$ and $\beta_{2m+1} = 4$ for $5 \le m \le 11$, $\beta_{2m} = 3$ and $\beta_{2m+1} = 4$ for $12 \le m \le 16$, $\beta_{2m} = 3$ and $\beta_{2m+1} = 2$ for $17 \le m \le 21$, $\beta_{44} = 1$, $\beta_{45} = 2$, $\beta_{46} = 1$, $\beta_{47} = 2$, $\beta_{48} = 1$, $\beta_{49} = 0$, $\beta_{50} = 1$, $\beta_{51} = 0$. In the next sections we discuss several other problems that can be tackled by the method of backward induction.

2. The Game of Pig

The game of Pig involves two players who in turn roll a die. The object of the game is to be the first player to reach 100 points. In each turn, a player repeatedly rolls a die until either a 1 is rolled or the player holds (voluntarily stops). If the player rolls a 1, the player gets a score of zero for that turn and it becomes the opponent's turn. If the player holds after having rolled a number other than 1, the total number of points rolled in that turn is added to the player's total score and it becomes the opponent's turn. At any time during a player's turn, the player must choose between the two decisions "roll" or "hold". It is assumed that a toss of a fair coin decides which player begins in the game of Pig. Then, under optimal play of both players, each player has a probability of 50% of being the ultimate winner. But how to calculate the optimal decision rule? The dynamic programming approach proceeds as follows. State s is defined by $s = ((i,k),j)$, where (i,k) indicates that the player whose turn it is has a current score of i and has k points accumulated so far in the current turn and j indicates that the opponent's current score is j. Define the value function $P(s)$ by

$P(s)$ = the probability that the player rolling the die will win the game given that state s is the present state,

where $P(s)$ is taken to be equal to 1 for those $s = ((i,k),j)$ with $i + k \ge 100$ and $j < 100$. To write down the optimality equations, we use the simple observation that the probability of a player winning after rolling a 1 or holding is one minus the probability that the other player beginning with the next turn will win. Thus, for state $s = ((i,k),j)$ with $k = 0$,

$$P((i,0),j) = \frac{1}{6}\{1 - P((j,0),i)\} + \sum_{r=2}^{6}\frac{1}{6}P((i,r),j)\,.$$

For state $s = ((i,k),j)$ with $k \ge 1$ and $i+k,\ j < 100$,

$$P((i,k),j) = \min\Big[1 - P((j,0),i+k)\,,\ \frac{1}{6}\{1 - P((j,0),i)\} + \sum_{r=2}^{6}\frac{1}{6}P((i,k+r),j)\Big],$$

where the first expression in the right side of the last equation corresponds to the decision "hold" and the second expression corresponds to the decision "roll". Using the method of successive substitution, these optimality equations can be numerically solved, yielding the optimal decision to take in any state $s = ((i,k),j)$. Starting with $P_0(s) = 0$ for all s, the functions $P_1(s)$, $P_2(s), \ldots$ are recursively computed from

$$P_n((i,0),j) = \frac{1}{6}\{1 - P_{n-1}((j,0),i)\} + \sum_{r=2}^{6}\frac{1}{6}P_{n-1}((i,r),j)$$

and

$$P_n((i,k),j) = \min\Big[1 - P_n((j,0),i+k)\,,\ \frac{1}{6}\{1 - P_n((j,0),i)\} + \sum_{r=2}^{6}\frac{1}{6}P_n((i,k+r),j)\Big].$$

Then, $\lim_{n\to\infty} P_n(s) = P(s)$ for all s. The computation of an optimal decision rule is a nontrivial job and has been done in Neller and Presser [4]. These authors have a nice website on computational aspects of the game of Pig and its variants. A variant of the game of Pig is as follows. Each turn, the player repeatedly rolls two dice until either the roll shows a 1 or the player holds. In the event of a roll showing a single 1, the player loses only the turn total, but in the event of a roll showing a double 1 both the turn total and the current score are lost.

The game of Hog (fast Pig) is a variation of the game of Pig in which players have only one roll per turn but may roll as many dice as desired. The number of dice a player chooses to roll can vary from turn to turn. The player's score for a turn is zero if one or more of the dice come up with the face value 1. Otherwise, the sum of the face values showing on the dice is added to the player's score. The players alternate in taking turns rolling the dice. The first player to reach 100 points is the winner. The game of Hog can also be analysed by the method of dynamic programming. The modification of the optimality equations are rather straightforward and will not be discussed here.

A challenging variant of the game of Hog arises when the two players have to take *simultaneously* a decision in each round and only partial information is available.[a] Think of the following television game. Two contestants each sit behind a panel with a battery of buttons numbered as $1, 2, \ldots, D$, say $D = 10$. In each stage of the game, both contestants must simultaneously press one of the buttons, where they cannot observe each other's decision. The number pressed on the button is the number of dice the contestant must throw. The score of the contestant's throw is added to his/her total, provided that none of the dice showed the outcome 1; otherwise no points are added to the current total of the contestant. In case both contestants reach the goal of 100 points in the same move, the winner is the contestant who has the largest total. In the event of a tie, the winner is determined by a toss of a fair coin. At each stage of the game both contestants have full information about his/her own current total and the current total of the opponent. What does the optimal strategy look like? The computation and the structure of an optimal strategy is far more complicated than in the problems discussed before. The optimal rules for the decision problems considered before are deterministic, but the optimal strategy will involve randomised actions for the problem of the television game show. In zero-sum games, randomisation is a key ingredient of the optimal strategy. The problem of the television-show game is discussed in detail in Tijms and Van der Wal [7] and still has open questions.

Remark. An interesting heuristic can be given for the single-player version of the game of Pig in which the player's goal is to reach 100 points in a minimal expected number of turns. The heuristic is to stop the turn when the turn total is 20 or more points with the stipulation that you also hold when the turn total is l or more if your current score lacks l points with $1 \leq l \leq 19$. The rationale behind this hold-at-20 rule: if you put 20 points at stake, your expected loss of $\frac{1}{6} \times 20$ points equals your expected gain of $\frac{5}{6} \times 4$ points. Under the hold-at-20 rule the expected value of the number of turns needed to reach 100 points is 12.637, while the minimal expected number of turns is 12.545. The structure of the decision rule leading to the minimal expected number of turns has been studied in Haigh and Roters [2]. The expected value of the number of turns for the heuristic can be computed by using a Markov chain model (see Tijms [6]) and the minimal expected number of turns can be computed by dynamic programming. Let state $(i, 0)$ mean that a turn has just been completed and the player's current score is i, and let state (i, k) mean that the turn total is k with $k \geq 2$ and the player's current score is i. Defining $V(s)$ as the minimal expected number of additional turns to reach 100 points from state s, we have the following optimality equations. For state $s = (i, 0)$ with $i < 100$,

$$V((i,0)) = 1 + \frac{1}{6}V((i,0)) + \sum_{r=2}^{6} \frac{1}{6}V((i,r))$$

and, for state $s = (i, k)$ with $k \geq 2$ and $i + k < 100$,

$$V((i,k)) = \min\Bigg[V((i+k,0))\,, \\ \frac{1}{6}V((i,0)) + \sum_{r=2}^{6} \frac{1}{6}V((i,k+r))\Bigg].$$

For the single-player version of the variant of the game of Pig with two dice, an excellent heuristic is to stop the turn in state (i, k) if

$$\frac{10}{36} \times k + \frac{1}{36}(i + k) \geq \frac{25}{36} \times 8$$

and to continue otherwise. Under this heuristic the expected number of turns to reach 100 points is 17.164, while the minimal expected number of turns is 16.923. For the single-player version of the game of Hog a good heuristic is to use the five-dice rule prescribing to roll five dice in each

[a] Other interesting decision problems with partial information are discussed in a nice paper by Hill [3].

turn with the stipulation that trunc($l/2$) dice rolled when still l points with $1 \leq l \leq 9$ are required (the expected score in a single turn is maximal when rolling five dice). Under the five-dice rule the expected number of turns to reach 100 points is 13.623, while the minimal expected number of turns is 13.039.

3. A Coin-tossing Game

The following game is very simple but still has open questions. Toss a fair coin repeatedly and stop whenever you want. The payoff is the proportion of heads accrued at the time you stop. What is the maximal expected payoff and what is an optimal stopping rule? It is known that an optimal stopping rule exists and is characterized by a sequence of integers $\beta_1, \beta_2, \ldots$. You stop after the nth toss when the number of heads minus the number of tails is larger than or equal to β_n. Obviously, $\beta_1 = 1$. It has also been proved that

$$\lim_{n\to\infty} \beta_n / \sqrt{n} = 0.83992\ldots .$$

However, the computation of the exact values of the maximal expected payoff and the critical numbers β_n is still an open problem. The difficulty is that backward induction will not work for the optimality equation for the coin-tossing problem. Let state (i, n) mean that n tosses have done so far and have resulted in i heads, and define $V(i, n)$ as the maximal expected payoff obtainable from state (i, n). Then, the optimality equation is given by

$$V(i, n) = \max\left[\frac{i}{n}, \frac{1}{2}V(i+1, n+1) + \frac{1}{2}V(i, n+1)\right].$$

Backwards induction will not work here since there is no a priori end to the sequence and, hence, no future time to calculate backwards from. Nevertheless, numerical results can be obtained by putting an upper bound on the number of tosses allowed. Suppose that no more than N tosses can be done. For a fixed value of N, define the value-function $f_k(i)$ as the maximal expected payoff obtainable if still k tosses are allowed and i heads have obtained so far. Then, the following recursive equation can be given

$$f_k(i) = \max\left[\frac{i}{N-k}, \frac{1}{2}f_{k-1}(i+1) + \frac{1}{2}f_{k-1}(i)\right]$$

for $0 \leq i \leq N-k$, where k runs from 1 to N. Starting with $f_0(i) = \frac{i}{N}$, the functions $f_1(i), \ldots, f_N(i)$ can be successively computed. The maximal expected payoff $V(0, 0)$ and the critical numbers β_n can be approximated by doing the recursive computations for a sufficiently large value for the length N of the planning horizon. It is interesting to see the numerical values of $f_N(0)$ for several values of N. The restricted maximal expected payoff $f_N(0)$ has the values 0.7679, 0.7780, 0.7839, 0.7912, 0.79206, 0.79263, 0.79289, and 0.79294 for $N = 25$, 50, 100, 1,000, 2,500, 10,000, 100,000, and 1,000,000. For large N, the value of $f_N(0)$ approximates the desired value of the maximal expected payoff $V(0, 0)$. It is remarkable how slowly $f_N(0)$ converges as N gets larger. Experimental mathematics done by Wiseman [8] provides strong evidence that

$$V(0, 0) = 0.79295350\ldots .$$

In Hägström and Wästlund [1] very sharp upper and lower bounds on $V(0, 0)$ are established and the bounds

$$0.79295301 < V(0, 0) < 0.79295560$$

are in agreement with the conjecture of Wiseman. A remarkable finding is that the heuristic stopping rule prescribing to stop as soon as the proportion of heads exceeds 0.5 has an expected payoff of $\pi/4 = 0.7853982$, being very close the maximal expected payoff $V(0, 0)$. On the basis of extensive numerical computations, Medina and Zeilberger [4] conjecture the true values of β_n for $1 \leq n \leq 185$ (the computer analysis in Hägström and Wästlund [1] confirm the proposed values of the optimal stopping levels β_n except for β_{127}). In addition to $\beta_1 = 1$, we mention the values $\beta_2 = 2$, $\beta_3 = 3$, $\beta_4 = 2$, $\beta_5 = 3$, $\beta_8 = 2$, $\beta_{10} = 4$, $\beta_{15} = 3$, $\beta_{25} = 5$, $\beta_{50} = 6$, $\beta_{75} = 7$, $\beta_{99} = 9$, and $\beta_{100} = 8$. In particular, stopping is not optimal if you have 2 heads and 1 tails after 3 tosses, but it is optimal if you have 5 heads and 3 tails after 8 tosses. Coin-tossing problems are always full of surprises.

References

[1] O. Hägström and J. Wästlund, Rigorous computer analysis of the Chow–Robbins game, Chalmers University of Technology, January 2012, see also *arXiv*:1201.0626v1 [math.PR].

[2] J. Haigh and M. Roters, Optimal strategy in a dice game, *J. Applied Probability* **37** (2000) 1110–1116.

[3] T. P. Hill, Knowing when to stop, How to gamble if you must — the mathematics of optimal stopping, *American Scientist* **97** (2009) 126–133.

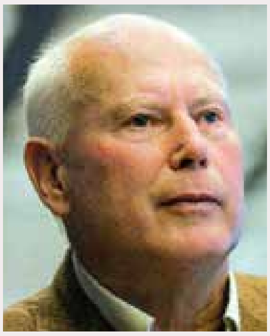
[4] L. A. Medina and D. Zeilberger, An experimental mathematics perspective on the old, and still open, question of when to stop, in *Gems in Experimental Mathematics*, Vol. 517 (AMS), eds. T. Amdeberhan *et al.*, (2010), pp. 265–274, see also *arXiv*:0907.0032v2 [math.PR].

[5] T. W. Neller and C. G. M. Presser, Optimal play of the dice game Pig, *The UMAP Journal* **25** (2004) 25–47, see also the website http://cs.gettysburg.edu/projects/pig/piglinks.html.

[6] H. C. Tijms, *Understanding Probability*, 3rd edn. (Cambridge University Press, 2012).

[7] H. C. Tijms and J. van der Wal, A real-world stochastic two-person game, *Probability in the Engineering and Informational Sciences* **25** (2006) 1–12.

[8] J. D. A. Wiseman, The Chow and Robbins problem: stop at h=5 t=3, www.jdawiseman.com/papers/easymath/coin-stopping.html.



Henk Tijms

Vrije University, The Netherlands
tijms@quicknet.nl

Henk Tijms is emeritus professor of operations research at the Vrije University in Amsterdam. He studied mathematics at the University of Amsterdam and obtained his PhD degree in 1972 at the same university. His research focused on the fields of applied probability and stochastic optimisation. He published several textbooks in these fields, including the introductory probability book *Understanding Probability*. He won with this book and other activities the 2008 INFORMS Expository Writing Award of the American Society of Operations Research. Also, he has put much effort in popularising probability and operations research at Dutch high schools.