Perspectives on Perfect and Practical Play of Pig

Todd W. Neller
Gettysburg College

tinyurl.com/piggame
Sow What’s This All About?

- Introduction to the Dice Game “Pig”
- Odds and Ends: Playing to Score
- Perfect Play: Playing to Win
  - “Piglet” example
  - Value Iteration
- Practical Play: How we can play well
The first player to reach 100 points wins.

On a turn, a player rolls a die repeatedly until:

- the player holds, scoring the sum of the rolls ("turn total"), or
- a 1 ("pig") is rolled, and there is no score change.

Example turns:

- roll 4, roll 5, roll 2, hold $\rightarrow$ add $4 + 5 + 2 = 11$ to score
- roll 3, roll 6, roll 6, roll 1 $\rightarrow$ score remains the same
Pig Preliminaries

- Player’s decision is always to *roll/hold*
  - *Roll* – possibly increase turn total, or lose it
  - *Hold* – definitely score current turn total
  - Pig is the simplest of a class of *jeopardy dice games*; ancestor of *Pass the Pigs*
- Hold at 20 - a simple policy that maximizes expected points per turn
Simple odds argument
- Roll until you risk more than you stand to gain.
- “Hold at 20”
  - 1/6 of time: -20 → -20/6
  - 5/6 of time: +4 (avg. of 2,3,4,5,6) → +20/6
Hold at 20?

- Is there a situation in which you wouldn’t want to hold at 20?
  - Your score: 99; you roll 2
  - Case scenario
    - you: 79  opponent: 99
    - Your turn total stands at 20
What’s Wrong With Playing to Score?

- It’s mathematically optimal!
- But what are we optimizing?
- Playing to score ≠ Playing to win
- Optimizing score gain per turn ≠ Optimizing probability of a win
Piglet

- Simpler version of Pig with a coin
- Object: First to score 10 points
- On your turn, flip until:
  - You flip tails, and score NOTHING.
  - You hold, and KEEP the # of heads.
- Even simpler: play to 2 points
What is the information I need to make a fully informed decision?

- My score
- The opponent’s score
- My “turn total”
A Little Notation

- $P_{i,j,k}$ – probability of a win if
  * $i$ = my score
  * $j$ = the opponent’s score
  * $k$ = my turn total
- Hold: $P_{i,j,k} = 1 - P_{j,i+k,0}$
- Flip: $P_{i,j,k} = \frac{1}{2}(1 - P_{j,i,0}) + \frac{1}{2} P_{i,j,k+1}$
Assume Rationality

- To make a smart player, assume a smart opponent.
- (To make a smarter player, know your opponent.)
- \( P_{i,j,k} = \max(1 - P_{j,i+k,0}, \frac{1}{2}(1 - P_{j,i,0} + P_{i,j,k+1})) \)
- Probability of win based on best decisions in any state
\[ P_{0,0,0} = \max(1 - P_{0,0,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1})) \]

\[ P_{0,0,1} = \max(1 - P_{0,1,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,2})) \]

\[ P_{0,1,0} = \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1})) \]

\[ P_{0,1,1} = \max(1 - P_{1,1,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,2})) \]

\[ P_{1,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(1 - P_{0,1,0} + P_{1,0,1})) \]

\[ P_{1,1,0} = \max(1 - P_{1,1,0}, \frac{1}{2}(1 - P_{1,1,0} + P_{1,1,1})) \]
The Whole Story

\[ P_{0,0,0} = \max(1 - P_{0,0,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1})) \]
\[ P_{0,0,1} = \max(1 - P_{0,1,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,2})) \]
\[ P_{0,1,0} = \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1})) \]
\[ P_{0,1,1} = \max(1 - P_{1,1,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,2})) \]
\[ P_{1,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(1 - P_{0,1,0} + P_{1,0,1})) \]
\[ P_{1,1,0} = \max(1 - P_{1,1,0}, \frac{1}{2}(1 - P_{1,1,0} + P_{1,1,1})) \]

These are winning states!
$P_{0,0,0} = \max(1 - P_{0,0,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1}))$

$P_{0,0,1} = \max(1 - P_{0,1,0}, \frac{1}{2}(1 - P_{0,0,0} + 1))$

$P_{0,1,0} = \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1}))$

$P_{0,1,1} = \max(1 - P_{1,1,0}, \frac{1}{2}(1 - P_{1,0,0} + 1))$

$P_{1,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(1 - P_{0,1,0} + 1))$

$P_{1,1,0} = \max(1 - P_{1,1,0}, \frac{1}{2}(1 - P_{1,1,0} + 1))$

Simplified...
The Whole Story

\[
\begin{align*}
P_{0,0,0} &= \max(1 - P_{0,0,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1})) \\
P_{0,0,1} &= \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,0,0})) \\
P_{0,1,0} &= \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1})) \\
P_{0,1,1} &= \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,0,0})) \\
P_{1,0,0} &= \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,1,0})) \\
P_{1,1,0} &= \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,1,0}))
\end{align*}
\]

And simplified more into a hamsome set of equations...
\[ P_{0,0,0} = \max(1 - P_{0,0,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1})) \]
\[ P_{0,0,1} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,0,0})) \]
\[ P_{0,1,0} = \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1})) \]
\[ P_{0,1,1} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,0,0})) \]
\[ P_{1,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,1,0})) \]
\[ P_{1,1,0} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,1,0})) \]

\[ P_{0,0,0} \text{ depends on } P_{0,0,1} \text{ depends on } P_{0,1,0} \text{ depends on } P_{0,1,1} \text{ depends on } P_{1,0,0} \text{ depends on } P_{0,1,0} \text{ depends on ...} \]
A System of Pigquations

Dependencies between non-winning states
How Bad Is It?

- The intersection of a set of bent hyperplanes in a hypercube
- In the general case, no known method (read: PhD research)
- Is there a method that works (without being guaranteed to work in general)?
  - Yes! Value Iteration!
Value Iteration

- Start out with some values (0’s, 1’s, random #’s)
- Do the following until the values converge (stop changing):
  - Plug the values into the RHS’s
  - Recompute the LHS values
- That’s easy. Let’s do it!
Value Iteration

\[ P_{0,0,0} = \max(1 - P_{0,0,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1})) \]
\[ P_{0,0,1} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,0,0})) \]
\[ P_{0,1,0} = \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1})) \]
\[ P_{0,1,1} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,0,0})) \]
\[ P_{1,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,1,0})) \]
\[ P_{1,1,0} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,1,0})) \]

- Assume \( P_{i,j,k} \) is 0 unless it’s a win
- Repeat: Compute RHS’s, assign to LHS’s
Initially, $P_{i,j,k} = 0$

$P_{0,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1}))$

$P_{0,0,1} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,0,0}))$

$P_{0,1,0} = \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1}))$

$P_{0,1,1} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,0,0}))$

$P_{1,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,1,0}))$

$P_{1,1,0} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,1,0}))$
Initially, $P_{i,j,k} = 0$

$P_{0,0,0} = \max(1 - 0, \frac{1}{2}(1 - 0 + 0))$

$P_{0,0,1} = \max(1 - 0, \frac{1}{2}(2 - 0))$

$P_{0,1,0} = \max(1 - 0, \frac{1}{2}(1 - 0 + 0))$

$P_{0,1,1} = \max(1 - 0, \frac{1}{2}(2 - 0))$

$P_{1,0,0} = \max(1 - 0, \frac{1}{2}(2 - 0))$

$P_{1,1,0} = \max(1 - 0, \frac{1}{2}(2 - 0))$
Initially, $P_{i,j,k} = 0$

$P_{0,0,0} = \max(1, \frac{1}{2}) = 1$

$P_{0,0,1} = \max(1, 1) = 1$

$P_{0,1,0} = \max(1, \frac{1}{2}) = 1$

$P_{0,1,1} = \max(1, 1) = 1$

$P_{1,0,0} = \max(1, 1) = 1$

$P_{1,1,0} = \max(1, 1) = 1$
Next, $P_{i,j,k} = 1$

\[
P_{0,0,0} = \max(1 - P_{0,0,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1}))
\]

\[
P_{0,0,1} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,0,0}))
\]

\[
P_{0,1,0} = \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1}))
\]

\[
P_{0,1,1} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,0,0}))
\]

\[
P_{1,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,1,0}))
\]

\[
P_{1,1,0} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,1,0}))
\]
Iteration 2

Next, \( P_{i,j,k} = 1 \)

\[
P_{0,0,0} = \max(1 - 1, \frac{1}{2}(1 - 1 + 1))
\]

\[
P_{0,0,1} = \max(1 - 1, \frac{1}{2}(2 - 1))
\]

\[
P_{0,1,0} = \max(1 - 1, \frac{1}{2}(1 - 1 + 1))
\]

\[
P_{0,1,1} = \max(1 - 1, \frac{1}{2}(2 - 1))
\]

\[
P_{1,0,0} = \max(1 - 1, \frac{1}{2}(2 - 1))
\]

\[
P_{1,1,0} = \max(1 - 1, \frac{1}{2}(2 - 1))
\]
Next, $P_{i,j,k} = 1$

$P_{0,0,0} = \max(0, \frac{1}{2}) = \frac{1}{2}$

$P_{0,0,1} = \max(0, \frac{1}{2}) = \frac{1}{2}$

$P_{0,1,0} = \max(0, \frac{1}{2}) = \frac{1}{2}$

$P_{0,1,1} = \max(0, \frac{1}{2}) = \frac{1}{2}$

$P_{1,0,0} = \max(0, \frac{1}{2}) = \frac{1}{2}$

$P_{1,1,0} = \max(0, \frac{1}{2}) = \frac{1}{2}$
Iteration 3

Next, $P_{i,j,k} = \frac{1}{2}$

$P_{0,0,0} = \max(1 - P_{0,0,0}, \frac{1}{2}(1 - P_{0,0,0} + P_{0,0,1}))$

$P_{0,0,1} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,0,0}))$

$P_{0,1,0} = \max(1 - P_{1,0,0}, \frac{1}{2}(1 - P_{1,0,0} + P_{0,1,1}))$

$P_{0,1,1} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,0,0}))$

$P_{1,0,0} = \max(1 - P_{0,1,0}, \frac{1}{2}(2 - P_{0,1,0}))$

$P_{1,1,0} = \max(1 - P_{1,1,0}, \frac{1}{2}(2 - P_{1,1,0}))$
Next, $P_{i,j,k} = \frac{1}{2}$

$P_{0,0,0} = \max(1 - \frac{1}{2}, \frac{1}{2}(1 - \frac{1}{2} + \frac{1}{2}))$

$P_{0,0,1} = \max(1 - \frac{1}{2}, \frac{1}{2}(2 - \frac{1}{2}))$

$P_{0,1,0} = \max(1 - \frac{1}{2}, \frac{1}{2}(1 - \frac{1}{2} + \frac{1}{2}))$

$P_{0,1,1} = \max(1 - \frac{1}{2}, \frac{1}{2}(2 - \frac{1}{2}))$

$P_{1,0,0} = \max(1 - \frac{1}{2}, \frac{1}{2}(2 - \frac{1}{2}))$

$P_{1,1,0} = \max(1 - \frac{1}{2}, \frac{1}{2}(2 - \frac{1}{2}))$
Next, $P_{i,j,k} = \frac{1}{2}$

$P_{0,0,0} = \max(\frac{1}{2}, \frac{1}{2}) = \frac{1}{2}$

$P_{0,0,1} = \max(\frac{1}{2}, \frac{3}{4}) = \frac{3}{4}$

$P_{0,1,0} = \max(\frac{1}{2}, \frac{1}{2}) = \frac{1}{2}$

$P_{0,1,1} = \max(\frac{1}{2}, \frac{3}{4}) = \frac{3}{4}$

$P_{1,0,0} = \max(\frac{1}{2}, \frac{3}{4}) = \frac{3}{4}$

$P_{1,1,0} = \max(\frac{1}{2}, \frac{3}{4}) = \frac{3}{4}$

This continues until values converge...
But That’s GRUNT Work!

- So have a computer do it, slacker!
- Not difficult – end of CS1 level
- Fast! Don’t blink – you’ll miss it
- Optimal play:
  - Compute the probabilities
  - Determine flip/hold from RHS max’s
  - (For our equations, always FLIP)
- Game to 10
- Play to Score: “Hold at 1”
- Play to Win:
Just like Piglet, but more possible outcomes

\[ P_{i,j,k} = \max(1 - P_{j,i+k,0}, \frac{1}{6}(1 - P_{j,i,0} + P_{i,j,k+2} + P_{i,j,k+3} + P_{i,j,k+4} + P_{i,j,k+5} + P_{i,j,k+6})) \]
Solving Pig

- 505,000 such equations
- Same simple solution method (value iteration)
- Potential Speedup: Solve groups of interdependent probabilities from game end backward
- So what does optimal play look like?
Pig Sow-lution
Pig Sow-lution
Probability Contours
Practical Play of Pig

- Whoa! That’s some funky alien landscape on that optimal policy!
- (scratches head) So I’m supposed to memorize that?
- Computing optimal play of Pig didn’t make me play optimally.
- How does one come up with practical policies for unaided human play?
Approximating Optimality

- KISS Principle: “Keep It Simple and Stupid.”
- Often, much learning benefit comes from attention to few, simple features.
- Observe the optimal policy and look for significant features of the roll/hold boundary.
- Try, try again.
- How does one evaluate simple policy ideas?
Given two policies (yours and optimal):

- set up a system of equations describing play,
- compute the probability of your winning going first/second, and
- average the win probabilities
**Algorithm 1** Policy Comparison

For each \((i, j, k) \in S\), initialize \(P_{i,j,k}^A\) and \(P_{i,j,k}^B\) arbitrarily.

Repeat

\[
\Delta \leftarrow 0
\]

For each \((i, j, k) \in S\),

\[
p_1 \leftarrow \begin{cases} 
\frac{1}{6} \left[ (1 - P_{j,i,0}^B) + \sum_{r \in [2, 6]} P_{i,j,k+r}^A \right], & \text{if Roll}_{i,j,k}^A; \\
1 - P_{j,i,k,0}^B, & \text{otherwise.}
\end{cases}
\]

\[
p_2 \leftarrow \begin{cases} 
\frac{1}{6} \left[ (1 - P_{j,i,0}^A) + \sum_{r \in [2, 6]} P_{i,j,k+r}^B \right], & \text{if Roll}_{i,j,k}^B; \\
1 - P_{j,i,k,0}^A, & \text{otherwise.}
\end{cases}
\]

\[
\Delta \leftarrow \max \left\{ \Delta, \ |p_1 - P_{i,j,k}^A|, \ |p_2 - P_{i,j,k}^B| \right\}
\]

\[
P_{i,j,k}^A \leftarrow p_1
\]

\[
P_{i,j,k}^B \leftarrow p_2
\]

until \(\Delta < \epsilon\)

return \(\left[ P_{0,0,0}^A + (1 - P_{0,0,0}^B) \right] / 2\)
Hold at $n$ (or goal)

Figure 2. Probability of an optimal player winning against a player using the “hold at $n$” policy

$n = 25 \Rightarrow$ optimal advantage = 4.2%; $n = 20 \Rightarrow$ optimal advantage = 8.0%
**Hold value:**

\[ h(i, t_s) = \left\lfloor \frac{100 - i}{t - t_s} \right\rfloor \]

---

**Figure 4.** The probability of an optimal player winning against a player using the "t scoring turns" policy for different values of t.

\[ t = 4 \rightarrow \text{optimal advantage} = 3.3\% \]
Roll if:
- \( k < b \) (you must score at least base value \( b \)),
- \( i + k < j - p \) (you must get within \( p \) of \( j \)), or
- either \( i \geq 100 - e \) or else \( j \geq 100 - e \) (you roll to win when someone is within \( e \) of the goal).

Optimizing parameters, \( b = 19 \), \( p = 14 \), and \( e = 31 \) \( \rightarrow \) optimal advantage = 1.9%
Roll if:
- either \( i \geq 100 - e \) or else \( j \geq 100 - e \), or
- \( k < c + (j - i)/d \).

Optimizing \( c = 21, d = 8, e = 29 \), and rounding division for hold value \( \rightarrow \) optimal advantage = 0.922%

So, if either player’s score is 71 or higher, roll for the goal. Otherwise, subtract your score from your opponent’s and let \( m \) be the closest multiple of 8. (Choose the greater multiple if halfway between multiples.) Then hold at 21 + \( m/8 \).
What we’ve learned:

- Playing to score is not necessarily playing to win.
- Simple rules do not imply simple perfect play.
- Making a guess at a solution and iteratively improving that guess can be a useful method.
- Similar iterative techniques can help us capture the simple essence of good play.
- The computer is an exciting power tool for the mind!
The Game of Pig page:
http://cs.gettysburg.edu/projects/pig

Pig CS teaching resources:
http://cs.gettysburg.edu/~tneller/resources/pig