# Optimal, Approximately Optimal, and Fair Play of the Fowl Play Card Game

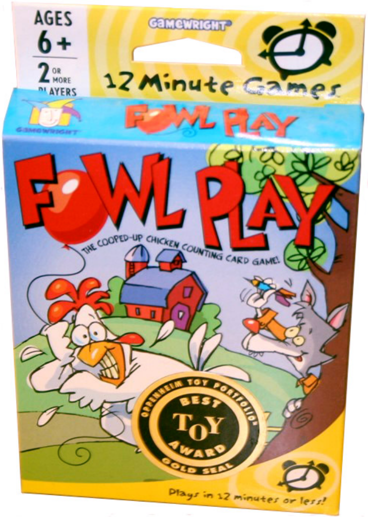
Todd W. Neller Marcin Malec Clifton G. M. Presser
Forrest Jacobs

Gettysburg College
Computer Science

ICGA Conference, Yokohama 2013

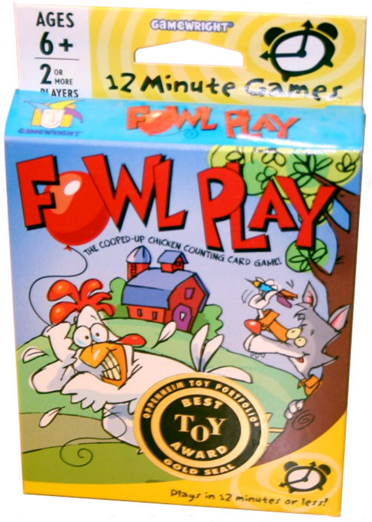
# Outline

1. Introduction to the Fowl Play jeopardy card game



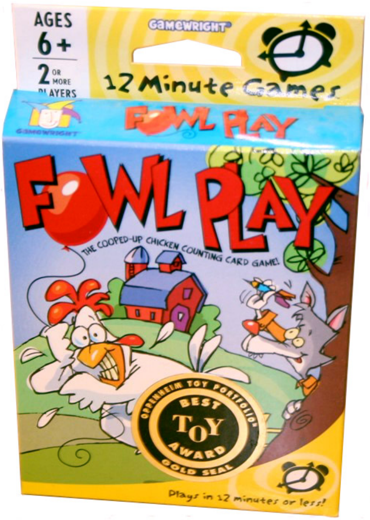
# Outline

1. Introduction to the Fowl Play jeopardy card game
2. Computation of optimal play



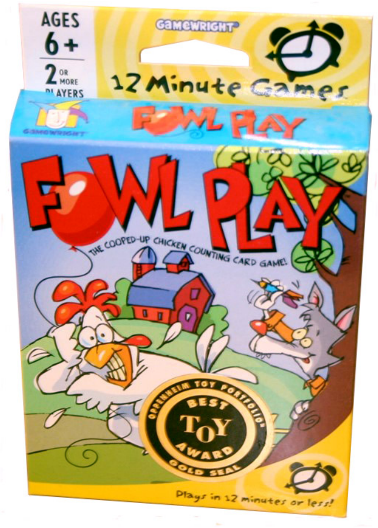
# Outline

1. Introduction to the Fowl Play jeopardy card game
2. Computation of optimal play
3. Neural network approximation of optimal play



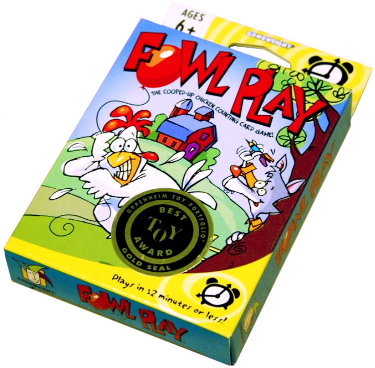
# Outline

1. Introduction to the Fowl Play jeopardy card game
2. Computation of optimal play
3. Neural network approximation of optimal play
4. Red Light: computer-aided redesign of Fowl Play optimizing fairness

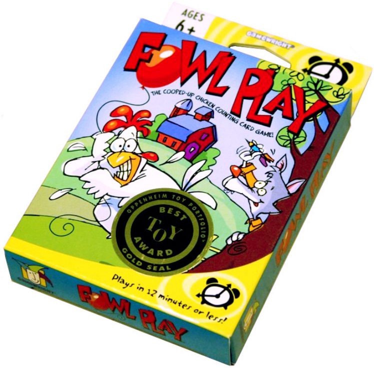


## Fowl Play

- Designed by Robert Bushnell and published by Gamewright in 2002

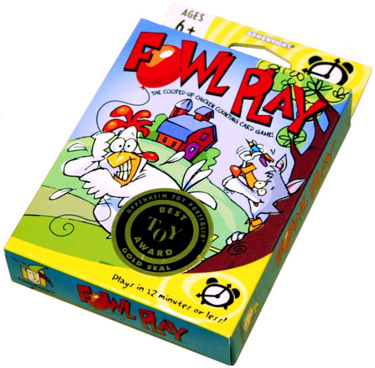


- Designed by Robert Bushnell and published by Gamewright in 2002
- Object: Be the first of 2 or more players to reach a given goal score.



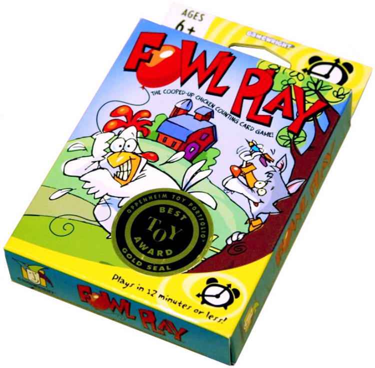
# Fowl Play

- Designed by Robert Bushnell and published by Gamewright in 2002
- Object: Be the first of 2 or more players to reach a given goal score.
    - We consider 2 players, 50 point goal case.



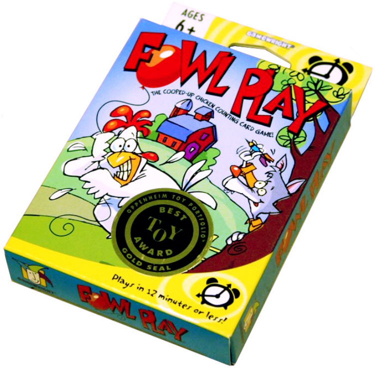
# Fowl Play

- Designed by Robert Bushnell and published by Gamewright in 2002
- Object: Be the first of 2 or more players to reach a given goal score.
  - We consider 2 players, 50 point goal case.
- Materials: Shuffled 48 card deck with



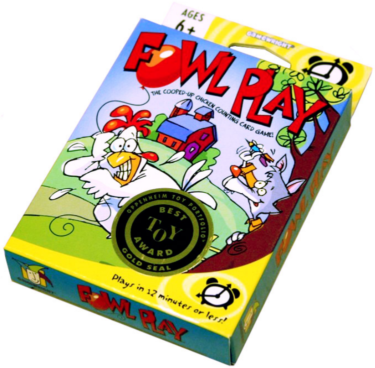
# Fowl Play

- Designed by Robert Bushnell and published by Gamewright in 2002
- Object: Be the first of 2 or more players to reach a given goal score.
  - We consider 2 players, 50 point goal case.
- Materials: Shuffled 48 card deck with
  - 42 chicken cards (each incrementing turn total)



## Fowl Play

- Designed by Robert Bushnell and published by Gamewright in 2002
- Object: Be the first of 2 or more players to reach a given goal score.
  - We consider 2 players, 50 point goal case.
- Materials: Shuffled 48 card deck with
  - 42 chicken cards (each incrementing turn total)
  - 6 wolf cards (each causing loss of turn total)



- On each turn, draw one or more cards until:

- On each turn, draw one or more cards until:
  - you draw a wolf and score no points, or

# Fowl Play (cont.)

- On each turn, draw one or more cards until:
  - you draw a wolf and score no points, or
  - you hold and score the number of chickens you've drawn.

- On each turn, draw one or more cards until:
  - you draw a wolf and score no points, or
  - you hold and score the number of chickens you've drawn.
- After the last ($6^{th}$) wolf is drawn, reshuffle all cards.

- 5 variables: Let $i$ and $j$ be the current player and opponent scores, respectively. Let $k$ be the current turn total. Let $w$ and $c$ be the number of wolves and chickens drawn since the last shuffle, respectively.
- Let $c_{rem}$ and $w_{rem}$ be the number of chickens and wolves remaining, respectively.
- Balancing reward and risk, draw iff:

$$\frac{c_{rem}}{w_{rem} + c_{rem}} > \frac{w_{rem}}{w_{rem} + c_{rem}} \times k$$

- Simplifying:

$$c_{rem} > w_{rem} \times k$$

## Optimizing Win Probability

- Assuming optimal play, non-terminal win probabilities are defined as:

$$P_{i,j,k,w,c} = \begin{cases} P_{i,j,k,w,c,draw} & \text{if } k = 0 \text{ and,} \\ max(P_{i,j,k,w,c,draw}, P_{i,j,k,w,c,hold}) & \text{otherwise} \end{cases}$$

where $P_{i,j,k,w,c,draw}$ and $P_{i,j,k,w,c,hold}$ are the probabilities of winning if one draws and holds respectively. These probabilities are given by:

$$P_{i,j,k,w,c,draw} = \frac{c_{rem}}{w_{rem} + c_{rem}} \times P_{i,j,k+1,w,c+1}$$
$$+ \frac{w_{rem}}{w_{rem} + c_{rem}} \times \begin{cases} 1 - P_{j,i,0,w+1,c} & \text{if } w < w_{total} - 1 \text{ and,} \\ 1 - P_{j,i,0,0,0} & \text{otherwise} \end{cases}$$

$$P_{i,j,k,w,c,hold} = 1 - P_{j,i+k,0,w,c}$$

- Equations are solved using a generalization of value iteration.

## Optimizing Win Probability

- Storing optimal policy would be problematic on mobile and embedded systems.
- There is a number of possible design choices for the function approximation task.
- Multi-layer feed-forward neural networks showed the greatest promise for closely approximating optimal play with minimal memory requirements.

# Neural Network Structure

- **Network Input:** In addition to the input features $i$, $j$, $k$, $w_{rem}$, and $c_{rem}$, the features $\frac{c_{rem}}{w_{rem}+c_{rem}}$ and $\frac{w_{rem}}{w_{rem}+c_{rem}} \times k$, the probability of drawing chicken and wolf cards respectively, scaled to range $[-1, 1]$ aided in the function approximation.
- **Network Output:** The single output unit classifies draw or hold actions according to whether the output is above or below 0.5, respectively. For training purposes, target outputs are 1 and 0 for draw and hold, respectively.
- **Hidden Layer:** Optimizing the number of hidden units empirically so as to boost learning rate and generalization, the single hidden layer worked best with 13 units.
- **The activation function** for all hidden and output units is the logistic function $f(x) = \frac{1}{1+e^{-x}}$

# Neural Network Structure

- Network weights are initialized using the Nguyen-Widrow algorithm.
- Online learning is used with the training cases generated through simulations of games with optimal play.
- Backpropagation learning algorithm is then applied with learning rate $\alpha = 0.1$ to update network weights.
- After $100,000$ games the performance of the network is evaluated through network play against an optimal player for $1,000,000$ games. If the win rate is better than 49.5% and the highest win rate so far, we re-evaluate using policy iteration with $\epsilon = 1 \times 10^{-6}$. We then continue to alternate between training and evaluation for total of 400 iterations.
- The End Result:
  - 10487700 nonterminal states
  - 118 weights (5 orders of magnitude reduction)
  - 49.79% win rate.

- Assuming optimal player play, we have computed the probability of winning with the second player starting with any number of points.

## Optimal Komi

- Assuming optimal player play, we have computed the probability of winning with the second player starting with any number of points.
- We can thus find the komi (compensation points) that optimizes fairness for Fowl Play:

## Optimal Komi

- Assuming optimal player play, we have computed the probability of winning with the second player starting with any number of points.
- We can thus find the komi (compensation points) that optimizes fairness for Fowl Play:
  - current: $2^{nd}$ player starts with 0 points $\rightarrow$ $1^{st}$ player wins 52.42% of games.

# Optimal Komi

- Assuming optimal player play, we have computed the probability of winning with the second player starting with any number of points.
- We can thus find the komi (compensation points) that optimizes fairness for Fowl Play:
  - current: $2^{nd}$ player starts with 0 points $\rightarrow$ $1^{st}$ player wins 52.42% of games.
  - best: $2^{nd}$ player starts with 1 point $\rightarrow$ $1^{st}$ player wins 50.16% of games.

# Optimizing Fowl Play for Fairness

- Further, we can

- Further, we can
  - vary the number of good (chicken) and bad (wolf) outcomes,

# Optimizing Fowl Play for Fairness

- Further, we can
  - vary the number of good (chicken) and bad (wolf) outcomes,
  - compute optimal play for each game,

# Optimizing Fowl Play for Fairness

- Further, we can
  - vary the number of good (chicken) and bad (wolf) outcomes,
  - compute optimal play for each game,
  - compute komi optimizing fairness, and

- Further, we can
  - vary the number of good (chicken) and bad (wolf) outcomes,
  - compute optimal play for each game,
  - compute komi optimizing fairness, and
  - thus optimize the game design for fairness.

- Further, we can
  - vary the number of good (chicken) and bad (wolf) outcomes,
  - compute optimal play for each game,
  - compute komi optimizing fairness, and
  - thus optimize the game design for fairness.
- With our optimized game of Red Light, the first player wins 50.001% assuming optimal play.

# Red Light

- Object: Be the first of 2 players to reach 50 points. The $2^{nd}$ player begins with 1 point.

# Red Light

- Object: Be the first of 2 players to reach 50 points. The $2^{nd}$ player begins with 1 point.
- Materials: 28 poker chips in a bag with

# Red Light

- Object: Be the first of 2 players to reach 50 points. The $2^{nd}$ player begins with 1 point.
- Materials: 28 poker chips in a bag with
    - 24 green “green light” chips (each incrementing turn total)

# Red Light

- Object: Be the first of 2 players to reach 50 points. The $2^{nd}$ player begins with 1 point.
- Materials: 28 poker chips in a bag with
  - 24 green “green light” chips (each incrementing turn total)
  - 4 red “red light” chips (each causing loss of turn total)

- On each turn, draw one or more chips until:

# Red Light (cont.)

- On each turn, draw one or more chips until:
  - you draw a red light and score no points, or

# Red Light (cont.)

- On each turn, draw one or more chips until:
  - you draw a red light and score no points, or
  - you hold and score the number of green lights you've drawn.

# Red Light (cont.)

- On each turn, draw one or more chips until:
  - you draw a red light and score no points, or
  - you hold and score the number of green lights you've drawn.
- After the last ($4^{th}$) red light is drawn, all chips are returned to the bag and reshuffled.

# Red Light (cont.)

- On each turn, draw one or more chips until:
    - you draw a red light and score no points, or
    - you hold and score the number of green lights you've drawn.
- After the last ($4^{th}$) red light is drawn, all chips are returned to the bag and reshuffled.
- Alternate materials: Standard ("French") deck of playing cards using Ace (red light) and $2 - 7$ (green lights) of each suit.

# Conclusion

- Fowl Play is a simple yet non-trivial jeopardy card game. We have:

## Conclusion

- Fowl Play is a simple yet non-trivial jeopardy card game. We have:
  - computed optimal play for 2-players, 50 point goal

# Conclusion

- Fowl Play is a simple yet non-trivial jeopardy card game. We have:
    - computed optimal play for 2-players, 50 point goal
    - computed a good NN approximation of optimal play

## Conclusion

- Fowl Play is a simple yet non-trivial jeopardy card game. We have:
  - computed optimal play for 2-players, 50 point goal
  - computed a good NN approximation of optimal play
  - optimized the game for fairness, creating the variant Red Light

# Conclusion

- Fowl Play is a simple yet non-trivial jeopardy card game. We have:
  - computed optimal play for 2-players, 50 point goal
  - computed a good NN approximation of optimal play
  - optimized the game for fairness, creating the variant Red Light
- Red Light presents

# Conclusion

- Fowl Play is a simple yet non-trivial jeopardy card game. We have:
  - computed optimal play for 2-players, 50 point goal
  - computed a good NN approximation of optimal play
  - optimized the game for fairness, creating the variant Red Light
- Red Light presents
  - a challenge for identifying features for good human play, and

# Conclusion

- Fowl Play is a simple yet non-trivial jeopardy card game. We have:
  - computed optimal play for 2-players, 50 point goal
  - computed a good NN approximation of optimal play
  - optimized the game for fairness, creating the variant Red Light
- Red Light presents
  - a challenge for identifying features for good human play, and
  - an accessible, minimalist game of chance for research and teaching.