Game Theoretical Approaches to Modelling and Simulation II

Michael H. F. Wilkinson

Institute for Mathematics and Computing Science
University of Groningen
The Netherlands

January 2003

This Week

- Asymmetric games: Bimatrix models
- Adaptive dynamics
- Evolution of cooperation
- Iterated Prisoner's Dilemma.
  - Theory,
  - Different strategies
  - Experimental results
  - Links to real world?
  - Implementation in C
  - The practical assignment.

Asymmetric games: Bimatrix models

- In the previous cases we have looked at symmetric games:
  - If moves are interchanged from player to player, so are the payoffs
  - Modelling is done using a single payoff matrix
- In practice, games between players may be asymmetric
- The goals may be different to players
- The values of resources may be different to different players: why is a hare faster than a fox? A hare runs for his life, a fox for his meal!
- The roles may be different (e.g. parent – child)
- To model such games we use *two* payoff matrices, or a *bimatrix*.

Bimatrix models: Battle of the Sexes I

- A classic example for a bimatrix game is the battle of the sexes, which concerns parental investment in the offspring
- For males who abandon the females after mating can go on to mate with other females.
- Females could prevent this by being “coy”, demanding an investment E from the male before mating, during a so-called “engagement” period.
- After such an investment, it would pay more for a male to help raise his young (because he is now relatively sure they are his), rather than find another mate (by which time the mating season may be over).
- However, once all males have been selected for faithfulness, a “fast” female, who will mate without engagement cost will gain
- This in turn leads to the appearance of “philandering” males, who will mate and desert the females to mate with another

Bimatrix models: Battle of the Sexes II

- We can formalize this:
  - If a coy female mates with a faithful male, they both win gain G and share the cost of upbringing C, and both pay E, so each wins $G - C/2 - E$.
  - If a fast female mates with a faithful male, they both win gain G and share the cost of upbringing C, without the cost E, so each wins $G - C/2$.
  - If a fast female meets a philandering male, she gets $G - C$, whereas he gets G.
  - If a coy female encounters a philandering male, she refuses to mate, so both receive 0
- In terms of payoff matrices we have

$$\mathbf{A} = \begin{bmatrix} 0 & G \\ G - \frac{C}{2} - E & G - \frac{C}{2} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & G - \frac{C}{2} - E \\ G - C & G - \frac{C}{2} \end{bmatrix} \tag{1}$$

  with $\mathbf{A}$ the matrix for males and $\mathbf{B}$ the matrix for females

- It turns out there is no stable equilibrium for this case.

Adaptive dynamics

- In the previous models we only look at single encounters
- Strategies change in a population through competition or copying
- Each strategy consists of a fixed set of probabilities for each move
- Unless mutations are allowed, no new strategies are developed
- In reality, players may change their strategy, depending on previous experience.
- Adaptive strategies are formulated differently;
  - The set of probabilities for moves is a function of the state of the player
  - The state of a player depends on previous games, either of the player himself, or those of others.
- The resulting adaptive dynamics can be highly complex

Evolution of Cooperation

- One field in which adaptive dynamics are important is that of the emergence of cooperation
- This emergence might concerns the evolution of cooperative behaviour in animals (not just the lack of aggression as in the Hawk-Dove game)
- It might also be the cooperation in economics and sociology: formation of coalitions, companies, etc.
- The core question is always: Why, when faced with an easy quick win at the expense of another, do many people or animals take a lower profit which does not harm the other.
- Another way of looking at the problem might be: why do we have such a strong feeling of fairness? Why do we get angry seeing someone cheat another when he should have shared?
- It turns out that single encounter games cannot solve this problem

Iterated Prisoner's Dilemma (IPD) I

- Iterated prisoner's dilemma is the classic example for adaptive strategies *and* the evolution of cooperation
- Prisoner's dilemma is a simple two player game in which there are two possible moves: cooperate (C) or defect (D)
- If both players cooperate, they receive a reward R
- If both players defect, they receive a punishment P
- If a player defects, but the other cooperates, he receives a temptation T
- If a player cooperates and the opponent defects, he receives the sucker's reward S
- In all cases we assume

$$T > R > P > S \quad \text{and} \quad 2R > T + S \tag{2}$$

IPD II

- The payoff matrix is

	C	D
C	R	S
D	T	P

- If the game is played once: the best strategy is always defect (*AllD*):
  - If the other cooperates, cooperating gets you R.
  - If the other cooperates, defecting gets you $T > R$.
  - If the other defects, cooperating gets you S.
  - If the other defects, defecting yourself gets you $P > S$.

  Therefore, you are always better off defecting

- This basically formalizes the selfishness problem

IPD III

- Now consider the case when you will encounter the same player again, with a probability w.
- Let A_n denote the payoff in the n-th round, the total expected payoff is given by

$$A = \sum_{n=0}^{\infty} A_n w^n \tag{3}$$

- In the limiting case of $w = 1$, this diverges, so instead we take the limit of the mean

$$A = \lim_{n \leftarrow \infty} \frac{\sum_{n=0}^{N} A_n w^n}{N+1} \tag{4}$$

- Obviously, if w is very small, each player should still just defect, since the possibilities for revenge are small.

IPD IV: Classifications of Strategies

- Strategies in IPD are programs which tell you which move to make in each round.
- Strategies are sometimes classified as:

  Nice: Does not defect first
  Retaliatory: Punishes defection
  Forgiving: Returns to cooperation following cooperation of opponent
  Suspicious: Does not cooperate until the other cooperates
  Generous: Does not always retaliate at a first defection

- No best strategy exist, it all depends on the opponent

IPD V

- If the opponent is an *AllC* player, *AllD* is best, because its payoff will be

$$A = \sum_{n=0}^{\infty} T w^n = \frac{T}{1-w} \tag{5}$$

- However, if the opponent is *Grim*, who is nice, retaliatory and totally unforgiving, the payoff after your first defection will be

$$A = \sum_{n=0}^{\infty} P w^n = \frac{P}{1-w} \tag{6}$$

  at best!

- This means *AllC* would perform better ($A = R/(1-w)$), provided

$$w > \frac{T-R}{T-P} \tag{7}$$

IPD VI: Tit-For-Tat

- A simple strategy which does very well in round-robin tournaments (each player competes in turn with each other player) is *Tit-For-Tat* (TFT).
- Curiously, TFT never gets *more* points per game than its opponent.
- It starts of with C, so it is nice
- It then copies the opponents last move
- This behaviour makes it retaliatory, because a defection will be repayed by a defection
- It is also forgiving, because it will return to playing C if the opponent returns to C
- It can outcompete a population of *AllD* and gain dominance if

$$w \geq \max\left(\frac{T-R}{T-P}, \frac{T-R}{R-S}\right) \tag{8}$$

IPD VII: Pavlov

- TFT has two weaknesses:
  1. It is sensitive to noise: if there is a small probability of a message (C or D) being misinterpreted, two TFT players enter into a round of mutual retaliations
  2. It is sensitive to invasion by other strategies such as *Pavlov*, or any nice strategy.
- *Pavlov* takes both his own and the opponents last move into account to compute the next
- This can be formalized as a function of the last *reward*
  - If the last reward is R, play C
  - If the last reward is P, play C
  - If the last reward is S, play D
  - If the last reward is T, play D
- In effect, *Pavlov* retains his strategy after high payoff (T or R) and changes strategy after low payoff (S or P).
- It can correct for occasional mistakes
- Strict cooperators cannot invade

IPD VIII: TFT variants

- The success of TFT resulted in the development of some variants
- *Tit-For-Two-Tats* (TF2T), which retaliates only after two Ds.
  - More generous
  - More tolerant for errors
  - Co-exists with TFT
- *Suspicious-Tit-For-Tat* (STFT) which starts with D instead of C, so it is not nice (gets on with TFT like a house on fire).
- *Observer Tit-For-Tat* Uses observations of potential opponents in other games to decide whether to start with D or C.
  - Requires the possibility of observations
  - Suppresses "Roving" strategies (*AllD* strategies which try to reduce w by selecting new opponents)

IPD IX: Stochastic Strategies

- Rather than using strict strategies, we can define probabilities with which strategies are used.
- This models:
  - Noise in the communications process
  - Faulty memory
- It also has the advantage that the importance of the initial move is lost after a sufficient number of moves.
- One way to define stochastic strategies is by defining 2-tuples (p, q) which denote the probabilities of a C after an opponent's C or D (respectively).
- Nowak and Sigmund (1992) found that *generous* TFT with $p = 1$ and

$$q = \min\left(1 - \frac{T-R}{R-S}, \frac{R-P}{T-P}\right) \tag{9}$$

  was the optimum in the case of $w = 1$.

IPD X: Stochastic Strategies

- Note that *Grim* cannot be modelled using the 2-tuple approach
- The above stochastic model can be extend to include more strategies.
- By setting the probability of cooperation after receiving a reward $R, S, T,$ or$P$ we can devise a large space of possible strategies, *including*
  - TFT: $(1, 0, 1, 0)$
  - *Pavlov*: $(1, 0, 0, 1)$
  - *Grim*: $(1, 0, 0, 0)$
  - *AllD*: $(0, 0, 0, 0)$
  - *AllC*: $(1, 1, 1, 1)$
- Strictly speaking, we should also add a fifth probability, i.e. the probability for C on the first move.

IPD XI: Experiments

- Using 100 random starting points in the 2-tuples-models, and a genetic algorithm using payoff as fitness function was implemented.
- Initially *AllD*-like strategies increased rapidly and *AllC*-like “suckers” were removed.
- Then, if sufficient TFT-like strategies were in the initial population, they eradicated the *AllD*-like strategies
- After this GTFT appeared and started to dominate.
- Similar results were obtained using the 4-tuple approach, but here Pavlov could appear, and did so (it was discovered this way).

Implementation issues in C I

- The C-implementation of the 5-tuple approach is simple, in particular if we choose

$$R = 3, \quad S = 1, \quad T = 4, \quad P = 2 \tag{10}$$

- These values are stored in a payoff matrix, which is a an array of integers:

```
typedef int payOffMatrix[2][2]
```

  we define `DEFECT` as 0 and `COOPERATE` as 1.

- We can then store the strategy in an array `strat` of floating point numbers of length 5, which has indexes running from 0 to 4.
- We store the probability for a C in the first move in `strat[0]`
- We store the probability for a C after a previous payoff of `lastPayOff` in `strat[lastPayOff]`

Implementation issues in C II

- A player can be defined as

```
typedef struct{
  int totalPayOff,
      lastPayOff;
  float strat[5];
} player;
```

- The fields `totalPayOff` and `lastPayOff` must be initialized at 0.
- The function `genMove` is simply:

```
int genMove(player p)
{ return randomdbl() < p.strat[p.lastPayOff];
}
```

with `randomdbl` the random number generator used previously.

Implementation issues in C III

- One round of the game is implemented as

```
void playRound(player *p1, player *p2,
               payOffMatrix mat        )
{
  int move1 = genMove(*p1),   /* compute moves */
      move2 = genMove(*p2);

  p1->lastPayOff = mat[move1][move2]; /* compute new payoffs */
  p2->lastPayOff = mat[move2][move1];

  p1->totalPayOff +=  p1->lastPayOff; /* add last payoffs */
  p2->totalPayOff +=  p2->lastPayOff; /*  to totals       */

}
```

- At the end of this, the scores are updated and the players are ready for the next round.

Assignment

- A file ipd.c is available on the web-site
- Work out at which w you should use in order for TFT to beat *AllD*.
- What is then the expected number of rounds N two player would meet, given w?
- Implement players for the following strategies: TFT, STFT, *Grim*, *AllD*, *AllC*, *Pavlov*.
- Adapt the program to run a round-robin tournament
- Put the results for each match in a table and compute the winner.
- Discuss your results
- Change the deterministic TFT into GTFT, and *Pavlov* into a stochastic version, exchanging 1 for 0.99 and 0 for 0.01
- Rerun against each other, and against *AllD* and the original TFT.
- Again, discuss your results
- If time allows, try to devise other strategies, to see if you can do better.