The Traveler's Dilemma:
"An airline loses two suitcases belonging to two different travelers. Both suitcases happen to be identical and contain identical antiques. An airline manager tasked to settle the claims of both travelers explains that the airline is liable for a maximum of $100 per suitcase, and in order to determine an honest appraised value of the antiques the manager separates both travelers so they can't confer, and asks them to write down the amount of their value at no less than $2 and no larger than $100. He also tells them that if both write down the same number, he will treat that number as the true dollar value of both suitcases and reimburse both travelers that amount. However, if one writes down a smaller number than the other, this smaller number will be taken as the true dollar value, and both travelers will receive that amount along with a bonus/malus: $2 extra will be paid to the traveler who wrote down the lower value and a $2 deduction will be taken from the person who wrote down the higher amount. The challenge is: what strategy should both travelers follow to decide the value they should write down?"
--Wikipedia [2]

Tournament:
Round robin: Each agent plays 1 match against each other agent
A match consists of T rounds.
$T = 1000$

Action space:
$A = \{2, \ldots, 100\}$

Valid strategy:
A valid strategy is some function $S\colon \cdot \to A$ that maps some set of inputs to an action.

Reward per round for action α against action β:
$R : A \times A \to \mathbb{Z} \in [0, 101]$
$\alpha, \beta \in A$
$R(\alpha, \beta) = min(\alpha, \beta) + 2 \cdot sgn(\beta - \alpha)$

Competitors:
$C = \{c \mid c \in S \wedge c \text{ is in the tournament}\}$
$x, y \in C$
$x_t = \text{the bid traveler } x \text{ makes on round } t.$

Total reward received by agent x in a match against y:
$M : S \times S \to \mathbb{R}$

$$M(x,y) = \sum_{t=1}^{T} R(x_t, y_t)$$

Utility metrics

Total reward:

$$U_1(x) = \frac{1}{101|C|T} \sum_{j \in C} M(x,j)$$

Pairwise Victory Count:

$$U_2(x) = \sum_{j \in C} sgn(M(x,j) - M(j,x))$$

Perfect Score Proportion:

$$U_3 = \frac{1}{|C|} \sum_{j \in C} \left[\frac{1}{T} \sum_{i=1}^{T} \frac{R(x_t, j_i)}{R(max(2,(j_i - 1)), j_i)} \right]$$

Perfect Bid Proportion:

$$U_{3'} = \frac{1}{|C|} \sum_{j \in C} \left[\frac{1}{T} \sum_{i=1}^{T} \left(1 - \frac{|x_t - j_t|}{98} \right) \right]$$

Strategies

Randoms:

Each of these strategies plays a random value in a given range.
Ranges are as follows:

1. [2, 100]
2. [99, 100]

Simpletons:

Each of these strategies plays the exact same value every round.
Values are as follows:

1. 2
2. 51
3. 99
4. 100

Tit-for-Tat, simple:

Each strategy assumes that the opponent will stick to a simple strategy, and thus play accordingly. The strategies bid as follows:

1. $x_t = y_{t-1} - 1$
2. $x_t = x_{t-1} - 2$

Tit-for-Tat, predictors:
Each strategy assumes the opponent is attempting to outbid this strategy using strategies similar to the Tit-for-Tat strategies. These strategies bid as follows:

1. $x_t = \begin{cases} x_{t-1}, & x_{t-1} < y_{t-1} \\ x_{t-1}, & x_{t-1} = y_{t-1} \\ y_{t-1} - 1, & x_{t-1} > y_{t-1} \end{cases}$

2. $x_t = \begin{cases} x_{t-1}, & x_{t-1} < y_{t-1} \\ x_{t-1} - 2, & x_{t-1} = y_{t-1} \\ y_{t-1} - 1, & x_{t-1} > y_{t-1} \end{cases}$

3. $x_t = \begin{cases} x_{t-1} - 2, & x_{t-1} < y_{t-1} \\ x_{t-1}, & x_{t-1} = y_{t-1} \\ y_{t-1} - 1, & x_{t-1} > y_{t-1} \end{cases}$

4. $x_t = \begin{cases} x_{t-1} - 2, & x_{t-1} < y_{t-1} \\ x_{t-1} - 2, & x_{t-1} = y_{t-1} \\ y_{t-1} - 1, & x_{t-1} > y_{t-1} \end{cases}$

5. $x_t = \begin{cases} y_{t-1} - 1, & x_{t-1} < y_{t-1} \\ x_{t-1} - 1, & x_{t-1} = y_{t-1} \\ x_{t-1} - 1, & x_{t-1} > y_{t-1} \end{cases}$

Mixed:
These strategies mix the previous strategies based on a random value $R \in [1,100]$ in the following manner:

1. $x_t = \begin{cases} \text{TFT Predictor 1}, & R \in [1,80] \\ 100, & R \in [81,100] \end{cases}$

2. $x_t = \begin{cases} \text{TFT Predictor 1}, & R \in [1,80] \\ 2, & R \in [81,100] \end{cases}$

3. $x_t = \begin{cases} \text{TFT Predictor 1}, & R \in [1,80] \\ 100, & R \in [81,90] \\ 2, & R \in [91,100] \end{cases}$

4. $x_t = \begin{cases} \text{TFT Predictor 5}, & R \in [1,80] \\ 100, & R \in [81,100] \end{cases}$

5. $x_t = \begin{cases} TFTpredictor5, & R \in [1,80] \\ 2, & R \in [81,100] \end{cases}$

6. $x_t = \begin{cases} TFTpredictor5, & R \in [1,80] \\ 100, & R \in [81,90] \\ 2, & R \in [91,100] \end{cases}$

7. $x_t = \begin{cases} TFTsimple1, & R \in [1,80] \\ Rand[99,100], & R \in [81,100] \end{cases}$

Buckets - Probability Distribution:
These strategies keep track over time how often each action is taking by the opposing agent. This is used as a probability distribution to predict the opposing agent's next move by pulling a random value from it. The agent then plays one under its prediction, i.e. x(t) = pred(y(0), y(1), . . ., y(t-1)) - 1. This strategy has a built in rate of retention, γ. The values held in each bucket is multiplied by this rate at each time step. Then, the bucket representing the opponents current bid is incremented.

$$\forall\, i\colon\ i \in (2{,}100)\ \wedge\ i \neq y_t \rightarrow bucket_{t+1}[i] = \gamma * bucket_t[i]$$
$$i = y_t \rightarrow bucket_{t+1}[i] = 1 + \gamma * bucket_t[i]$$

The values used for rate of retention are as follows:

1. 1.0
2. 0.8
3. 0.5
4. 0.2

Note: Predrag, I have flipped this scheme around from the one we discussed last Thursday. It seemed to make more sense to do it this way as I wouldn't have to subtract the rate of forgetfulness from 1 in order to use it. Before, the value 0.0 meant there was no time bias, now 1.0 means this.

Buckets - Deterministic:
These strategies keep track over time how often each action is taking by the opposing agent. This is used to predict the opposing agent's next move by selecting the bucket that is most full (i.e. selecting the value that the opponent has bid most often). The agent then plays one under its prediction, i.e. x(t) = pred(y(0), y(1), . . ., y(t-1)) - 1.

$$\forall\, i\colon\ i \in (2{,}100)\ \rightarrow bucket_{t+1}[i] = bucket_t[i] + 1$$

We are implementing several different tie-breaking paradigms as follows:

1. Highest valued bucket wins
2. Lowest valued bucket wins
3. Random bucket wins
4. Newest tied bucket wins

Q-learning:
This strategy uses the Q-learning method to attempt to predict the opponent's next move. Again, the agent will play 1 under the predicted value.

Often, Q-learning represents the various state/action pairs across the entire problem space in a table. This, however, does not scale well. A single time-step t contains 98 different states (representing the opponents last action) and 98 different actions (representing the strategies next move) leading to 9604 possible configurations in the problem space for that step alone. However, this becomes a particularly daunting concern when we wish to capture the opponent's moves over a length of time greater than just the previous time step. In order to alleviate this scaling issue, we will be breaking up the state/action spaces into a few clusters. Currently, these clusters are defined as follows:

State:
1. The opponent played higher than our last bid
2. The opponent played the same bid we did
3. The opponent played lower than our last bid

Action:
1. Play higher than our last bid
2. Played the same bid as last time
3. Play lower than our last bid

This is for just one time-step. The actual implementation will keep treat the state as a collection of moves by the opponent over the last k rounds. We have not discussed values for k yet, but I am inclined to pick 5 as an arbitrary value for it. This leads to a problem space with 729 possible configurations (3^6).

Each state/action pair will be updated as follows:
$$Q(s_t,a_t) \leftarrow Q(s_t,a_t)(1-\alpha)+\alpha[r_{t+1}+\gamma \max_a Q(s_{t+1},a)]$$

where:
- $Q(s_t,a_t) =$ the expected reward if action α_t is taken in state s_t
- α is the learning rate such that $0<\alpha\leq 1$
- r_{t+1} is the reward given at time $t+1$
- The discount factor γ is such that $0\leq\gamma<1$
- $\max_a Q(s_{t+1},a) =$ the best expected award received from the state that follows from taking action a at time t

We are implementing this with the learning rates as follows:
1. 0.8
2. 0.5
3. 0.2
4. 0.0

Note: We have not discussed values for the discount rate. My understanding is that when closer to 1.0, the learning rate places greater emphasis on future rewards.

Zeuthen Strategy:
This strategy bases its bid on the level of one's willingness to risk. Here, risk is the ratio of the difference in utility of the agent's proposal and of two other deals: if one accepts your opponents proposal and if the conflict deal must be accepted. For this agent, the conflict deal would be the Nash Equilibrium of (2, 2). While this isn't a strict negotiation, I plan on treating individual bids (i.e. x(t) and y(t)) as the proposals. The assumption is made that if y(t) = i, then the proposal being made by Y is (i+1,i), because this would be the best outcome for Y, given its bid. The same applies for X's bid. Thus, at each time step t the agent looks at the two bids made at (t-1) and determines the amount of risk of each agent.

As defined in the [1], the risk of an agent is defined as:

$$R_x(t) = \frac{\text{utility agent } x \text{ loses by conceding and accepting agent } y\text{'s offer}}{\text{utility agent } x \text{ loses by not conceding and causing a conflict}}$$

More specifically:

$$R_x(t) = \begin{cases} 1 \\ \dfrac{U_x(x_{t-1}, x_{t-1}+1) - U_x(y_{t-1}+1, y_{t-1})}{U_x(x_{t-1}, x_{t-1}+1) - U_y(2,2)} \end{cases}$$

$$U_x(x_{t-1}, x_{t-1}+1) - U_y(2,2) = 0$$

$$otherwise$$

where:
$R_x(t)$ is the Risk of agent x at time step t
$U_x(a_x, a_y)$ is the Utility gained by agent x when the actions a_x and a_y are taken by agents x and y.

If the agent's risk is higher than its opponents, it continues to make the same bid. If it is lower, then the agent makes the minimal sufficient concession. The concession is sufficient if it either reverses the relationship of the agents' risks (or possibly makes them equal). The minimal concession is the one that increases the opponents utility the least. Due to an oddity in the payout matrix, it is possible for the "concession" to actually lead to a loss of utility for the opponent. This, however, seems to go against the notion of what a concession is. Thus, we will be implementing the following two strategies:

1. Only positive concessions are allowed. The strategy will find the minimal sufficient concession that does not decrease the opponent's utility.
2. Negative and positive concessions are allowed. The strategy will find the truly minimal "concession" that is enough to reverse the relationship in the agents' risk values.

Simple Trend:
The agent will look at the last k rewards and establish a line of best fit. The slope of this line will determine the strategy taken. If the slope is near zero (by near I mean within some arbitrary threshold, in this case 0.5) than the system of bidding and payouts is relatively stable and the agent will play the 1 under strategy. If the slope is negative and outside of the dead band (i.e. not greater than -0.5), then the system is trending towards the Nash equilibrium and, thus, the smaller rewards. In this case, the agent will attempt to maximize social welfare and play 100. If the trend is positive and above the threshold, than the agent will continue to place the same bid it has been. (Also, we could actually track the agent's bid, opponent's bid, and the payout over time and find a line of best fit to try and determine the correlation between all three).

For the line of best fit we are using the method of least squares. For linear regression we can find the slope via the following formula for all points (x, y) in the last k rewards:

$$m = \frac{n\sum(xy) - \sum x \sum y}{n\sum(x^2) - \left(\sum x\right)^2}$$

Again, we are using an arbitrary slope threshold ϵ:
$\epsilon = 0.5$

The strategies bid is then determined as follows:

$$x_t = \begin{cases} x_{t-1} & m > \epsilon \\ y_{t-1} - 1 & otherwise \\ 100 & m < \epsilon \end{cases}$$

We are implementing this strategy with the following values of k:
1. 3
2. 10
3. 25

Bibliography:

1. Jeffrey S. Rosenschein and Gilad Zlotkin, *Rules of encounter: designing conventions for automated negotiation among computers* (MIT Press, 1994).
2. "Traveler's dilemma - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Traveler%27s_dilemma.