Negotiation Protocols

1. General aspects

Automated negotiation among autonomous agents is needed when agents have conflicting objectives and a desire to cooperate. This typically occurs when agents have competing claims on scarce resources, not all of which can be simultaneously satisfied. Resources can be commodities, services, time, money etc., or anything that is needed to achieve some objective.

A negotiation problem is one where multiple agents try to come to an agreement or deal. Each agent is assumed to have a preference over all possible deals. The agents send messages to each other in the hope of finding a deal that all agents can agree on. These agents face an interesting problem. They want to maximize their own utility but they also face the risk of a break-down in negotiation, or expiration of a deadline for agreement. As such, each agent must negotiate carefully, trading off any utility it gains from a tentative against a possibly better deal or the risk of a breakdown in negotiation. It must decide whether the current deal is good enough or whether it should ask for more and risk agreement failure.

Given a set of agents, a set of resources, an existing resource allocation and a set of other possible allocations, the main goal of negotiation is to find an allocation that is better in some sense, if such allocation exists. In order to achieve this goal, agents need some mechanism. Abstractly, a mechanism specifies the rules of encounter, what agents are allowed to say and when, how allocations are calculated, whether calculations are done using a centralised algorithm or in a distributed fashion, and so on.

Different mechanisms may have different properties. Following is a list of desirable features for negotiation mechanisms.

1. Simplicity: A mechanism that requires less computational processing and communication overhead is preferable to one that does not.

2. Efficiency: A mechanism is efficient if it produces a good outcome. What is meant by "good," however, may differ from one domain to another. One common criteria is that of Pareto Optimality, where no agent could be better off in a different allocation without some other agent being worse of. Another criteria might be Global Optimality, where the sum of agents benefits is maximised.

3. Distribution: It is preferable to have a negotiation mechanism that does not involve a central decision-maker. Centralisation may lead to communication bottlenecks or decreased reliability due to the single-point-of-failure.

4. Symmetry: This property implies that the mechanism should not be biased for or

against some agent based on inappropriate criteria. Again, what constitutes an 'inappropriate' criteria depends on the domain in question.

5. Stability: A mechanism is stable if no agent has incentive to deviate from some agreed-upon strategy or set of strategies. In an auction, for example, we may require that no agent lie by making a false bid, or that no group of agents can form strategic coalitions to create a disadvantage for other agents.

6. Flexibility: The mechanism should lead to agreement even if agents did not have complete and correct private information in relation to their own decisions and preferences. This property requires a complementary mechanism for rational investigation and possible refinements of internal decisions during negotiation itself.

2. Negotiation using game theory

In the bargaining problem, we say that each agent ***i*** has a utility function $\boldsymbol{u_i}$ denned over the set of all possible deals Δ. That is, $u_i : \Delta \rightarrow R$. We also assume that there is a special deal δ^- which is the no-deal deal. Without loss of generality we will assume that for all agents $u_i(\delta^-) = 0$, so that the agents will prefer no deal than accepting any deal with negative utility. The problem then is finding a protocol which will lead the agents to the best deal.

Another way to think about what solution will be arrived at in a bargaining problem is to formalize the bargaining process, assume rational agents, and then determine their equilibrium strategies for their bargaining process. That is, define the solution concept to be the solution that is reached by automated rational agents in a bargaining problem. This method does raise one large obstacle: we need to first formally define a bargaining process that allows all the same "moves" as real-life bargaining. A negotiation protocol is needed that is simple enough to be formally analyzed but still allows the agents to use most of their moves.

One such model is the Rubinstein's alternating offers model. In this model two agents try to reach agreement on a deal. The agents can take actions only at discrete time steps. In each time step one of the agents proposes a deal δ to the other who either accepts it or rejects it. If the offer is rejected then we move to the next time step where the other agent gets to propose a deal. Once a deal has been rejected it is considered void and cannot be accepted at a later time. The agents, however, are always free to propose any deal and to accept or reject any deal as they wish. We further assume that the agents know each other's utility functions.

The alternating offers model does not have a dominant strategy. For example, imagine that the two agents are bargaining over how to divide an amount of money. Each agent wants to keep the entire amount to itself and leave the other agent with nothing. Under the basic alternating offers protocol the agents' best strategy is to keep proposing this deal to the other agent. That is, each agent keeps telling the other one "I propose that I keep the entire amount" and the other agent keeps rejecting this proposal. This scenario is not viable. In order to fix this situation, we further assume that time is valuable to the agents. That is, the agents' utility for all possible deals is reduced as time passes.

3. Negotiation protocols

3.1. Monotonic concession

In this protocol the agents agree to always make a counter offer that is slightly better for the other agent than its previous offer. Specifically, in monotonic concession the agents adhere to the following algorithm:

```
MONOTONIC-CONCESSION
δ_i ← arg max_δ u_i(δ)
Propose δ_i
Receive δ_j proposal
if u_i(δ_j) ≥ u_i(δ_i)
    then Accept δ_j
    else δ_i ← δ'_i such that u_j(δ'_i) ≥ ε + u_j(δ_i) and u_i(δ'_i) ≥ u_i(δ^-)
goto 2
```

Each agent starts by proposing the deal that is best for itself. The agent then receives a similar proposal from the other agent. If the utility the agent receives from the other's proposal is bigger than the utility it gets from its own proposal then it accepts it and negotiation ends. If no agreement was reached then the agent must propose a deal that is at least an increase of *e* in the other agent's utility.

The rules of this protocol are as follows:

- Negotiation proceeds in a series of rounds.
- On the first round, both agents simultaneously propose a deal from the negotiation set.
- An agreement is reached if the two agents propose deals δ_1 and δ_2, respectively, such that either $u_1(\delta_2) \geq u_1(\delta_1)$ or $u_2(\delta_1) \geq u_2(\delta_2)$, i.e. if one of the agents finds that the deal proposed by the other is at least as good or better than the proposal it made.
- If agreement is reached, then the rule for determining the agreement deal is as follows:
    - If both agents' offers match or exceed those of the other agent, then one of the proposals is selected at random.
    - If only one proposal exceeds or matches the other's proposal, then this is the agreement deal.

- If no agreement is reached, then negotiation proceeds to another round of simultaneous proposals. In round *n+1*, no agent is allowed to make a proposal that is less preferred by the other agent than the deal it proposed at time *n*.

- If neither agent makes a concession in some round u>0, then negotiation terminates, with the conflict deal.

It should be clear that this protocol is effectively verifiable: it is easy for both parties to see that the rules of the protocol are being adhered to. Using the monotonic concession protocol, negotiation is guaranteed to end (with or without agreement) after a finite number of rounds. Since the set of possible deals is finite, the agents cannot negotiate indefinitely: either the agents will reach agreement, or a round will occur in which neither agent concedes. However, the protocol does not guarantee that agreement will be reached quickly.

If neither agent makes a new proposal then there are no more messages sent and negotiations fail, so they implicitly agree on the no-deal deal δ^-. The monotonic concession protocol makes it easy for the agents to verify that the other agent is also obeying the protocol. Namely, if agent *i* ever receives a proposal whose utility is less than a previous proposal then it knows that agent *j* is not following the protocol. The protocol can be visualized as in Figure 1. Here we see a simple example with linear utility functions where the agents reach an agreement (δ^4) after four time steps. Notice that the agreement reached is not the point at which the lines intersect.

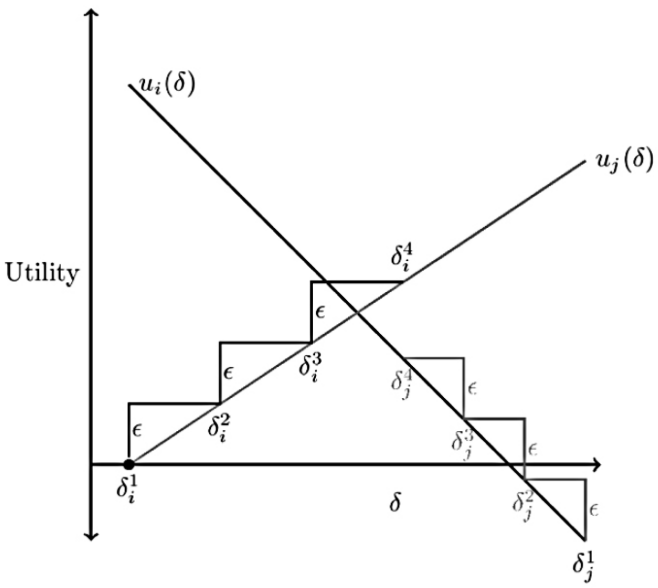

Figure 1. Monotonic concession protocol - example

Monotonic concession has several drawbacks. It can be very slow to converge. Convergence time is dictated by the number of possible deals, which is usually very large, and the value of *e*. It is also impossible to implement this algorithm if the agents do not know the other agents' utility function. In practice, it is rare for an agent to know its opponent's utility function. Finally, the monotonic concession protocol has a tricky last step. Namely, the two agents could make simultaneous offers where each one ends up preferring the other agent's offer to the one it just sent. This is a problem as both of them now want to accept different offers.

This can be solved by forcing the agents to take turns. But, if we did that then neither will want to go first as the agent that goes first will end up with a slightly worse deal.

3.2. The Zeuthen Strategy

So far, we have said nothing about how negotiation participants might or should behave when using the monotonic concession protocol. On examining the protocol, it seems there are three key questions to be answered as follows.

What should an agent's first proposal be?

On any given round, who should concede?

If an agent concedes, then how much should it concede?

The first question is straightforward enough to answer: an agent's first proposal should be its most preferred deal.

With respect to the second question, the idea of the Zeuthen strategy is to measure an agent's willingness to risk conflict. Intuitively, an agent will be more willing to risk conflict if the difference in utility between its current proposal and the conflict deal is low.

In contrast, if the difference between the agent's current proposal and the conflict deal is high, then the agent has more to lose from conflict and is therefore less willing to risk conflict - and thus should be more willing to concede.

Agent *i*'s willingness to risk conflict at round *r*, denoted $risk_i^t$, is measured in the following way:

$risk_i^t$ = (utility lost by *i* by conceding and accepting *j*'s offer) / (utility lost by *i* by not conceding and causing conflict)

Until an agreement is reached, $risk_i^t$ will have a value between 0 and 1. Higher values of $risk_i^t$ (nearer to 1) indicate that *i* has less to lose from conflict, and so is more willing to risk conflict. Conversely, lower values of $risk_i^t$ (nearer to 0) indicate that *i* has more to lose from conflict, and so is less willing to risk conflict. Furthermore, if $u_i(\delta_i^t) = 0$, then $risk_i^t = 1$, because, in this case, the utility to *I* of its current proposal is the same as from the conflict deal, in which case *i* is completely willing to risk conflict by not conceding.

Therefore the Zeuthen strategy proposes that the agent to concede on round *t* of negotiation should be the one with the smaller value of risk.

The next question to answer is how much should be conceded? The simple answer to this question is just enough. If an agent does not concede enough, then on the next round, the balance of risk will indicate that it still has most to lose from conflict, and so should concede again. This is clearly inefficient. On the other hand, if an agent concedes too much, then it 'wastes' some of its utility. Thus, an agent should make the smallest concession necessary to change the balance of risk, so that on the next round, the other agent will concede.

There is one final refinement that must be made to the strategy. Suppose that, on the final round of negotiation, both agents have equal risk. Hence, according to the strategy, both should concede. Bur, knowing this, one agent can 'defect' by not conceding, and so benefit from the other. If both agents behave in this way, then conflict will arise, and no deal will be struck. We extend the strategy by radmonly choosing who should concede if ever an equal risk situation is reached on the last negotiation step.

The Zeuthen algorithm is as follows:

ZEUTHEN-MONOTONIC-CONCESSION

1 $\delta_i \leftarrow \arg\max_\delta u_i(\delta)$
2 Propose δ_i
3 Receive δ_j proposal
4 **if** $u_i(\delta_j) \geq u_i(\delta_i)$
5   **then** Accept δ_j
6 $\text{risk}_i \leftarrow \frac{u_i(\delta_i) - u_i(\delta_j)}{u_i(\delta_i)}$
7 $\text{risk}_j \leftarrow \frac{u_j(\delta_j) - u_j(\delta_i)}{u_j(\delta_j)}$
8 **if** $\text{risk}_i < \text{risk}_j$
9   **then** $\delta_i \leftarrow \delta'_i$ such that $\text{risk}_i(\delta'_i) > \text{risk}_j(\delta_j)$
10   **goto** 2
11 **goto** 3

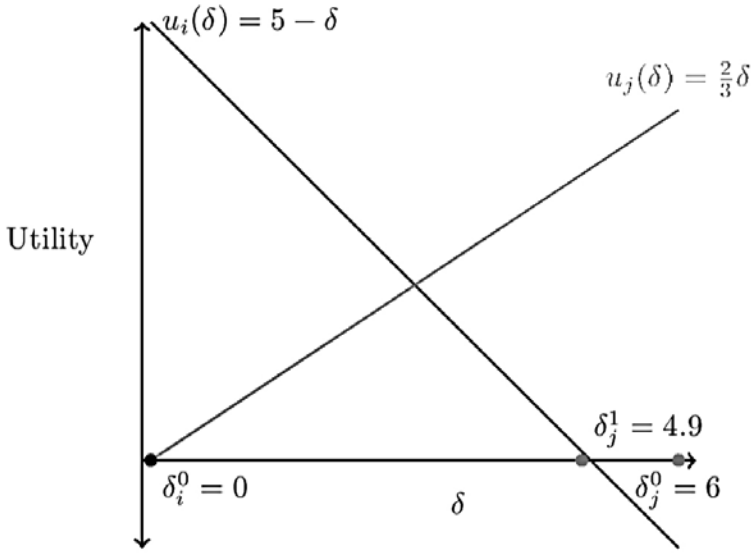

Figure 2. Illustration of the Zeuthen strategy

Figure 2 shows a graphical representation of the first step in a Zeuthen negotiation. After the initial proposal the agents calculate their risks to be:

$$risk_i^0 = \frac{5-(-1)}{5} = \frac{6}{5} \qquad risk_j^0 = \frac{4-0}{4} = 1$$

Since *j* has a lower risk it must concede. The new deal must be such that *j* will not be forced to concede again. That is, it must ensure that:

$$risk_i = \frac{5-(5-\delta_j)}{5} < \frac{\frac{2}{3}\delta_j - 0}{\frac{2}{3}\delta_j} = risk_j$$

Which simplifies to $\delta_j < 5$. As such, *j* can pick any deal $\delta < 5$. In the case presented in Figure 2, the agent chooses $\delta_j^1 = 4.9$.

4. Tasks

1. Study the example application which implements the Zeuthen strategy beween two negotiating agents.
2. Implement the simple monotonic concession protocol.
3. Considering the same utility functions and starting offers, measure which protocol converges faster (Monotonic concession or the Zeuthen strategy), i.e. in which case an agreement is reached in fewer steps.
4. Run both protocols using various combinations of utility functions. Choose from the following types:
   a. Linear functions: $u(x) = a_1 x + a_0$
   b. Quadratic functions: $u(x) = a_0 x^2$
   c. Sqrt functions: $u(x) = sqrt(a_0 x)$

   In which cases does the buyer agent / seller agent obtain a better deal (whose utility is higher when an agreement is reached)?

   Which combination of utility functions causes faster convergence (see Task 3.)?

   Show your findings clearly in the console.

(Optional) Plot the graphs of the two utility functions.