Frequently Asked Questions about the negotiation assignment

This document will act as a reference guide for the assignment. You can type your questions in the Open Questions section below; please do not edit any other parts of the document!

Answered Questions

Where can I find the assignment?
See: http://www.cs.uu.nl/docs/vakken/mas/

Install Java on Linux Systems.

1. Check if your system is 32 or 64 bits with lscpu: https://www.howtogeek.com/198615/how-to-check-if-your-linux-system-is-32-bit-or-64-bit/
2. Download JDK from Oracle (you need to make an account): https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html ; or you can consider openjdk
3. Extract the tar.gz ; use update-alternatives to set the java version; change PATH variables: https://www.tecmint.com/install-java-jdk-jre-in-linux/ ; for changing the java version only: https://crunchify.com/how-to-change-default-java-jdk-jvm-classpath-in-linux-using-bash_profile/
4. Update .bash_profile (or .bashrc) file with the exports: https://www.journaldev.com/41479/bashrc-file-in-linux
5. Install Eclipse (or your desired IDE) and configure the java version for your project to 1.8
6. In case of the libawt_xawt.so error see: https://askubuntu.com/questions/674579/libawt-xawt-so-libxext-so-6-cannot-open-shared-object-file-no-such-file-or-di
7. Start by following the assignment and Genius user guide

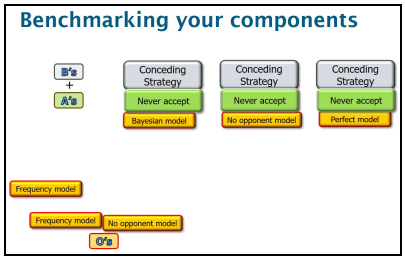
The way that we currently planned our agent is that already in the bidding strategy, the opponent model is used and the set of bids generated is based on that. Therefore our opponent model strategy doesn't do a whole lot except maybe selecting the bid with most utility from that set. Is that fine or do you advise us to change it? Since in the ideal

version the opponent model is only used in the opponent model strategy, from what we read.

That is completely okay: the opponent model can be used in several places and can be invoked by the bidding strategy (and sometimes even acceptance strategy) as well.

With regard to preference uncertainty, we created a new class offering a method which receives a bid and returns the utility of the bid if there is no uncertainty, or an estimate of the utility in the case of uncertainty. We then call this method every time we need to get the utility of a bid. Is this approach ok? Or must we integrate the function bid-->utility in the agent in a different way?
This approach is how it should be! It does not necessarily need to be defined in a separate class though; the estimateUtilitySpace() method of the agent is also a possible choice to provide this functionality.

How can we test components outside of the Genius environment? Is it possible to run "virtual" sessions via code?
You can use the command line runner to test your agent outside of Genius, specifying the setup via an XML file; see section 5.3 of the user manual. If you want to test components of your agent, the best way is usually to keep all other components fixed while varying another; e.g. for opponent models:



How can we use existing components? Should we extract them from a JAR file?
You can find BOA components in the BOA panel of Genius:

Are the limits of the ranked bids (min and max bid utility) the overall limits of the utility space or are they just the min and max of the sampled bids?
They are the min and max utility of the sampled bids.

In the uncertain situation, are we getting the utility for the bid after we placed the bid?
No; to be clear, the information state of your agent with regard to preference uncertainty does not change after the start of the negotiation. You will need to provide an estimate of the user utility using this starting information.

Is the OutcomeComparison class faithful even if the Preference is uncertain? Hence, if I have a bid in the Ranking and another Bid outside of it will I still be able to say whether it is better or not?
Suppose the number of outcomes in the preference uncertainty setting is set to 50. In every negotiation there is a limited number of possible bids, say 3000 bids in that specific domain. Only of the 50 bids of that ranking you have the relative ranking of; the best (utility) bid: bid #1, to the worst (utility) bid: bid # 50. Without further estimation methods, you cannot compare the other 2950 with each other or with the 50 ranked bids; for that you need to provide a user model estimate.

Is it just round based, or time based (or both)?
Your agent should be able to handle both cases

Just to clarify: Are the bid rankings from all the possible bids sampled every round (we get 50 different bids ranked every round)? Does the ranking populate with each bid (when we bid something will our bid become known in the space)? Or we just get some bids, for example 50, in the beginning and we need to construct our whole comparisons around that?
It is the latter: you get 50 fixed (but randomly chosen) comparisons at the start and you need to work with that.

What uncertainty settings does our agent need to be able to handle? Can we always assume that in the uncertainty situation it has the full ranking of possible bids or will a variety of different settings be tested?
In the case of preference uncertainty your agent has the ranking of a number of possible bids depending on the uncertainty setting (see also the screenshot below), but not necessarily all of them. In the uncertainty setting, your agent only knows the utility of two bids: the utility of the best possible bid, and the utility of the worst possible bid.

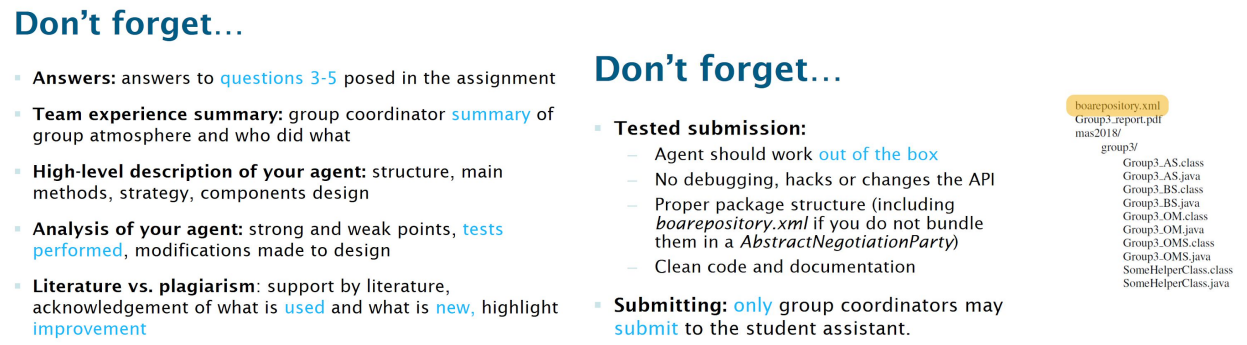
How are we supposed to implement the OpponentModel? Which methods shall we override?
The point of the OpponentModel class is to provide an estimate of the opponent's preferences. For that, any OpponentModel implementation provides:

```
public AbstractUtilitySpace getOpponentUtilitySpace() {
        return opponentUtilitySpace;
}
```

In order to update this utility space, an implementation needs provide (override) the method to update the opponent model with every incoming offer:

```
protected abstract void updateModel(Bid bid, double time);
```

Do we need to listen to some kind of presentation as well now?
No, 'all' the action is here in the document! You are encouraged to work on the assignment with your group and ask any questions that pop up here in the document. Some final pointers:



getBidRanking().getBidOrder() from the usermodel only gives a few possible bids instead of a ranking of all possible bids. Is that how it is supposed to work? And if yes, is there a way to get a complete ranking of all possible bids?
Yes. The amount of bids in the ranking is determined by the uncertainty setting of the preference profile (e.g. you get 47 bids in the ranking by choosing the settings as in the figure above):

In the multilateraltournament.xml file, how can I specify Class Path for my agent?
In all the XML files, you can refer to the classpath of the agent as follows:

```
classPath="negotiator.parties.ConcederNegotiationParty.class"
```

However, if you require other imports for your agent, you need to add them to the classpath of your project in Eclipse by editing the build path (see manual).

We have been having some issues with getting genius to return isEndNegotiation() as true at the end of the negotiation, it seems to always return false. Is there some way to get it to work or some other way of checking for end negotiation?
Normally, your agent gets reset after each interaction and hence checking for the end of the negotiation is not needed. If you need to do so regardless (e.g. because you are using RL techniques), we suggest enabling “data persistence”. In that case, you can load and save data between sessions.

Do we have to deal with issue types other than DISCRETE and INTEGER?
No, in fact you only need to deal with DISCRETE issues.

Can we reuse every class that is already in the genius jar? (For instance, the matrix library under agents.Jama)
Yes you can; in fact, you may reuse any class or jar by others to support your agent. Just be sure to be open about your usage of other people’s work in your report and how you improve upon their work.

How will the final tournament be run?
We are intentionally vague about the exact setting (i.e. about what timing or scenarios will be used) to incentivize you to focus on building a general negotiating agent. However, what we can tell you is that you will be graded based on individual utility over many interactions with the agents from other groups. Your agent will be reset between each encounter, so it cannot learn from one negotiation to the next. The setting will be fair, in the sense that each agent will play both sides of each scenario. There will be no discount factor and no reservation value, but there may very well be preference uncertainty.

When will the reports be graded?
We (the assistant) will strive to have the reports returned before the next report is due.

Final report and agent

- The final report is evaluated as follows:

Criterion	Measure	Weight
Description	Completeness, correctness and clarity of the agent description	10%
Strategy	Motivation and understanding about the strategic aspects of the negotiation game	10%
Creativity	Sophistication and originality of the agent design	20%
Literature	Motivation by, support from and improvement over existing literature	10%
Analysis	Discussion of the performance of the agent, and ways to overcome the deficiencies	20%
Agent performance	Performance in a tournament with other agents including those of other groups	20%
Code quality	Code correctness, readability, and class composition	10%

How should we compute and draw the Pareto efficient frontier?
You can let GENIUS draw it for you after defining the scenario; for Report 1, you can use any two agents (A and B) that result in a clean outcome plot.

What is meant with the negotiation setting?
The negotiation setting describes the entire context of the negotiation: for example, the deadline, the number of bids exchanged, etc. Report 1 asks you to keep the domain and preferences fixed and then discuss the most salient aspects of the negotiation setting that influence the final agreement that is reached.

How is a fair negotiation outcome defined?
Fairness can be defined in many ways, and this is equally true for an outcome of a negotiation (e.g. Nash outcome, Pareto efficient outcome, …). Report 1 asks you to *choose* an outcome you consider most fair for both players and *compare* this choice with other fairness notions. Using these definitions, you are expected to argue why you believe certain outcomes are more fair than others.

What should be in Report 1?
The answers to Question 1 (which begins under subsection 3.2). For more details and what to do for the other reports, see the deliverables table in the Submission section 4.6.

What can be done against the following error: "java.lang.ClassNotFoundException: javax.activation.DataSource"?

Students seem to get this issue resolved by consulting the following StackOverflow.
(If this doesn't work, try to add the dependency by right-clicking the pom.xml file and select dependencies, and then add "activation 1.1.1".)

Are the slides presented at the tutorial/lectures available online somewhere?
We explicitly do not put the tutorial slides online as they are not part of the exam material, which could cause confusion. On top of that, it serves as a bit of incentive to attend the tutorials. However, if you need to see material again or have a question, please approach the instructors during the tutorial.

How does preference uncertainty work, exactly?
For the exact details of how this works in Genius, please refer to the Genius manual (which comes in the same zip file as the jar), at 7.3. The idea here is that instead of getting all the details that you need to calculate the utility of a Bid, as you normally would, you instead just get a list of example Bids, ordered by utility. You don't get the actual utilities. You will have to use these examples to assign a value or order to other Bids you may encounter.

You will not get this ordering in addition to the preference profile. You will either get the preference profile, or (when operating under uncertainty) the ordered list of example Bids.

Where can we find examples of good agents (like ANAC winners)?
Some of the ANAC agents have published papers; Google (Scholar) is your friend. There's also a paper (source [47] in the assignment) which contains a whole paragraph of agents. See: https://link.springer.com/chapter/10.1007%2F978-4-431-54758-7_4