

# Supply Chain Management League (SCML)

## Challenge

**Design and build an autonomous agent that negotiates on behalf of a factory manager situated in a supply chain management simulation.**

The goal of a factory manager in SCML is to maximize its profit given its private production capabilities by negotiating trades with other agents. A factory manager can engage in several negotiations simultaneously, for which its utility functions are in general interdependent. These negotiations, and any ensuing contracts, are bilateral. Moreover, they are private to the agents involved.

## Negotiation Protocol

Participating agents are factory managers that control factories with predefined manufacturing profiles which are revealed privately to each agent at the start of each simulation. Factory manager agents need to negotiate bilaterally with other agents to buy the necessary inputs to their manufacturing process, and to sell the outputs.

All negotiations are carried out via the alternating offers protocol. This protocol specifies that two negotiators take turns making offers. One agent starts the negotiation with an opening bid, after which the other party can take the following actions:

1. Accept the offer
2. Make a counter offer, thus rejecting and overriding the previous offer
3. Walk away, thus declaring an end to the negotiation without having reached an agreement

This process is repeated until either an agreement is reached, or the deadline arrives. To reach an agreement, both parties must accept the offer. If no agreement has been reached by the deadline, the negotiation fails.

A single simulation runs for a predefined number of steps with an overall time limit of two hours. All negotiations are conducted for a predefined number of steps/turns of the alternating offers protocol (with a predefined time limit on each).

Factory manager agents are reset after each simulation. This means that they cannot learn from previous simulations. They can, however, accumulate information about agents during a simulation, as they know their negotiating partners' names.

## Platform

Entrants to the competition will develop and submit an autonomous agent that runs on [NegMAS](#). NegMAS is a Python-based negotiation platform in which you can create simulated worlds, like the SCM world, populated with agents capable of engaging in multiple negotiations. The platform allows you to run the SCM world with the same settings as the ones employed in ANAC 2019, or with personalized settings.

In NegMAS, the alternating offers protocol is implemented as a special bilateral case of the multilateral [Stacked Alternating Offers Protocol](#) (slightly modified to allow for a limited number of rounds).

NegMAS also includes an implementation of a [greedy factory manager agent](#), which can serve as a sample implementation of an agent to guide participating teams.

The greedy factory manager agent is written in Python. Java can be also used to write factory managers, using [JNegMAS](#), a Java interface to NegMAS, but a Python installation will also be necessary for testing purposes. Python is the recommended development language for SCML.

The official version of NegMAS (and JNegMAS) was released on April 1st 2019.

## Evaluation

There will be three separate competitions in the 2019 SCM league.

In the first, the basic competition, at most one instantiation of each team's agent will run in each simulation. In some of these simulations, all the other agents will be greedy factory manager agents. In others, agents submitted by other teams will also participate, but again at most one instantiation of each.

In the second, the collusion competition, multiple instantiations of the same team's agent may run during a single simulation (with multiple greedy factory manager agents as well). The exact number of instantiations of each will vary across simulations, and will not be announced in advance. In this competition, it is possible for instances of the same agent to try to collude with one another to corner the market, or exhibit other collusive behaviors.

The final, the sabotage competition, is intended to uncover fragile aspects of the SCML design. Teams who enter this competition should try to sabotage the market, for example, by preventing trades, or by negatively impacting the profits accrued by others. Sabotaging agents will not compete against one another directly; they will be evaluated independently in the presence of non-sabotaging agents only. Furthermore, they will be excluded from the other two competitions.

An agent's performance will be measured by its **score**. In the basic and collusion competitions, an agent's score will be the average profits accrued by all its factories in all its instantiations in all simulations. In the sabotage competition, agents' profits will not factor into their score; only their ability to sabotage the market/game will matter.

The three competitions will be conducted in two rounds, a qualifying round and a final round. All entrants that are not judged to break any of the SCML and ANAC submission rules will be entered into the qualifying rounds. Top-scoring agents in the qualifying round will then be entered in the final round.

The teams that built the top-scoring agents will be notified in June, with the final results and awards announced at IJCAI 2019. It is expected that finalists will send a representative to the ANAC workshop at IJCAI 2019, where they will be given the

opportunity to give a brief presentation describing their agent. Three awards will be announced at IJCAI 2019 (with associated monetary rewards) corresponding to the three competitions (basic, collusion, and sabotage).

The organizing committee will determine the number of simulations needed in each round to ensure a fair comparison among all submitted agents. All participating agents that achieve scores that are not statistically different from the winners' will be inducted into the SCM league's hall of fame.

**Submission** ~~(Deadline: 20 May, 2019 AoE [UTC-12])~~ **Deadline: 31 May, 2019 AoE [UTC-12]**

Participants must submit the following on or before the deadline:

1. Team member names, affiliations, and contact information.
2. Agent source code (in a zip file) for python agents, and both source code and compiled class files for Java agents (in a zip or jar file).
3. Academic report describing their agent.

**Submission website:** <https://forms.gle/VsQm4z9493fgrc9L7>

Submitted code and the academic report should be considered to be in the public domain, and may be incorporated in full or part in future releases of [NegMAS](#) and/or [JNegmas](#), or any other media. In such cases, all team members will receive proper attribution.

**Academic Report:** Each participating team must prepare a 2 page report describing the novel aspects of their agent according to academic standards (consider it an extended abstract). This report will be evaluated by the organizers of this league, but will not affect the possibility of winning as long as the agent is not judged to be a simple repetition of an existing agent. Submission of this report is an integral part of the submission process, and agents without an associated report will be disqualified.

At a minimum, the academic report should address the following points:

- **Negotiation choices:** how your agent chooses what to negotiate about, when, and with whom
- **Utility function(s):** how your agent calculates its utility for the various outcomes of its negotiations
- **Simultaneous negotiations coordination:** how your agent coordinates its behavior among multiple simultaneous negotiations (if it does)
- **Risk management:** any measures your agent takes to alleviate the risks involved in trading with other agents
- **Evaluation:** evaluate the performance of your agent against the built-in greedy factory managers.
- **Collusion (If employed):** Did your agent employ a collusive strategy? If so, how, and was it effective in your evaluations?
- **Sabotage (If employed) / Design loophole exploitation (Optional):** Did your agent try to sabotage other agents or exploit loopholes in the SCML design? If so, how, and was it effective in your evaluations? For the benefit of the agent negotiation research community, please provide some suggestions that would render your agent's nonconforming behavior less effective in future renditions of SCML.

## Rules of Encounter

Agents will be disqualified for violating the spirit of fair play. In particular, the following behaviors are strictly prohibited:

1. Accessing any information about the simulation or other agents that is not describe in Section 2.1 "information revelation" of the [detailed game description](#).
2. Accessing/modifying any "private" method/member in python (these variable names should all be indicated by an initial underscore).
3. Hacking or exploiting bugs in the software.
4. Communicating with the agent during the competition.
5. Altering the agent during the competition.

Moreover, agents that wreak havoc on the simulator (intentionally or otherwise) may be disqualified. For more information, see <http://ii.tudelft.nl/anac/>.

Organizers of the ANAC 2019 SCML competition will enforce these rules. We also reserve the right to disqualify agents under ANAC's usual circumstances.

Finally, note that ANAC's competition rules allow multiple entries from a single institution, but require that each agent be developed independently.

## Resources

For more information about SCML, please refer to the following links:

1. [An overview of the SCM world/competition](#): Here, you can find an overview of the SCM world, the agents that inhabit it, and competition rules.
2. [Project skeleton for starting up your development](#): You are not required to use this skeleton, but they should be helpful as you get started.
3. [A detailed description of the SCM world](#): Here, you can find a more detailed (and formal) description of the SCM world, including behavior of the built-in agents. This document also describes the specific settings of the simulator for the ANAC 2019 SCML competition.
4. [Participation Tutorial](#): Here, you can find a step-by-step description of the process of developing, testing and submitting an agent for the ANAC 2019 SCML competition using python. (Available also as a downloadable [Jupyter Notebook](#).)
5. [NegMAS Documentation](#): Here, you can find complete documentation of the NegMAS library.
6. [NegMAS Source Code on GitHub](#): Here, you can (but are by no means required to) read the source code of the NegMAS library, which is released under the GPL 2.0 license.
7. [NegMAS PyPi Page](#): Here, you can find the official release of the NegMAS library, and download the latest version (or just use `pip install` as described [here](#)). Version 0.2.0 (available April 1st, 2019) will be compatible with the distributed version.

8. [JNegMAS Source Code](#): Here, you can find the source code for the Java interface. Version 0.2.0 (available April 1st, 2019) will be compatible with the official league platform. Please note that you still need to install NegMAS to run simulations, even if you are developing in JNegMAS.
9. The JNegMAS jar file is available at the [Maven Central Repository](#).
10. [JNegMAS Documentation](#): Simple documentation of the Java interface.

## Questions and Answers

Please check our [FAQ](#). You can post your questions there (preferable), or address any concerns you prefer remain private to [Yasser Mohammad](#).

## Organizing Committee

- Yasser Mohammad, NEC-AIST & Assiut University (main contact)
- Katsuhide Fujita, Tokyo University of Agriculture and Technology & NEC-AIST
- Amy Greenwald, Brown University & NEC-AIST
- Mark Klein, MIT & NEC-AIST
- Satoshi Morinaga, NEC-AIST
- Shinji Nakadai, NEC-AIST

## Important Dates

<b>Official release of the league platform (python and Java)</b>	April 1, 2019
<b>Website opens for submission</b> <i>A confirmation email will be sent upon successful submission (~2 business days)</i>	April 10, 2019
<b>Submission deadline</b> <i>Servers may be busy on the last day. Please do not wait until the last minute</i>	May 20, 2019
<b>Notification to finalists</b> <i>A representative of each finalist team is expected to attend ANAC 2019 @ IJCAI</i>	June 15, 2019 (tentative)
<b>Event</b> <i>Announcement of winners and league report during ANAC 2019</i>	mid August, 2019 (during IJCAI)

## Sponsors

