Multi-Agent Systems - Exam Report

Giovanni Billo, Roshanak Berhouz

July 2025

Contents

Professor Padoan's part - Automated Mechanism Design

Project aim

Mechanism Design studies how to define game rules so that self-interested agents collectively reach desirable outcomes, typically formalized via a *social choice function* g.

This project implements Automated Mechanism Design for one-to-one bartering of n goods between two agents: a designer (Agent 1) and a player (Agent 2), based on [1]. The designer's type is fixed; the player has m types. The goal is to find, for each type of player θ, a mechanism that satisfies the following constraints:

- **individual rationality (IR)**: bartering with the designer must be a sensible choice for the player, meaning that, after performing a barter, its utility has to be greater or equal than that of the initial allocation.
- **Truthfulness (i.e IC, Incentive Compatibility)**: the player has to have no incentive to misreport its types θ, i.e a reported type different than the true should result in less utility.

The IR and IC constraints are explicitly checked to ensure that each mechanism is both truthful and individually rational.

Implementation

The Python implementation generates all bartering outcomes for user-defined n and m. The types are randomly sampled from $1, 2, 3, 4, 5$. A depth-first search filters results satifying the IR and IC constraints. For each θ, the mechanism maximizing the designer's objective g is selected among the remaining ones, constituting the optimal mechanism.

Verification

To verify truthfulness, all possible misreports for a certain type were enumerated, and utilities compared. No misreport yielded higher utility than the truthful mechanism selected by the algorithm.

Limitations and Future Work

While conceptually simple, the enumeration approach scales poorly with n and m. Experiments with six goods already showed a significant run-time. The time complexity of the algorithm is considerable: the outcome tree search has a worst-case time complexity $O(2^{|O|}) = O(2^{2^n})$, which is superexponential. This happens if all branches are always fully explored. Instead, the comparison function between states has time complexity $O(|O|^2 \times |\Theta|)$. In practice, even if

```
Chosen outcome for θ[12)]:
  Agent 1 gets: {'g0', 'g1'}
  Agent 2 gets: {'g2', 'g3'}
  Agent 1 utility (g): 0
  Agent 2 utility (u): 0
  Agent 1 had frozenset({'g0', 'g1'}) with utility 0 initially (actual initial goods valuation: 6),
 Agent 2 had frozenset({'g2', 'g3'}) with utility 0 (actual initial goods valuation: 9)
Verifying truthfulness for a true theta with utility 0 for the chosen outcome.
Generated 625 possible misreports.
Truthfulness verified for this theta: All misreports resulted in utility less than or equal to the true utility.

Chosen outcome for θ[13)]:
  Agent 1 gets: {'g2', 'g3', 'g1'}
  Agent 2 gets: {'g0'}
  Agent 1 utility (g): 5
  Agent 2 utility (u): 2
  Agent 1 had frozenset({'g0', 'g1'}) with utility 0 initially (actual initial goods valuation: 6),
 Agent 2 had frozenset({'g2', 'g3'}) with utility 0 (actual initial goods valuation: 3)
Verifying truthfulness for a true theta with utility 2 for the chosen outcome.
Generated 625 possible misreports.
Truthfulness verified for this theta: All misreports resulted in utility less than or equal to the true utility.
```

(a) example verbose output for the optimal mechanism of 2 types.

we don't fully explore all branches during the algorithm, thus having a better expected time complexity at least for the tree search, the two together result in a rather expensive computation overall. A promising direction is to reimplement the algorithm in a compiled language and parallelize the search over types, while also removing some redundance in the verification method (misreports that were already covered in previous checks).

Professor Petrov's Part – Consensus in Robotic Swarms

Project Aim

The goal of this project was to explore consensus dynamics in robotic swarms using the cross-inhibition model. We focus on how specific parameters, such as reaction thresholds and zealot proportions, influence the stability and convergence of the swarm to a shared decision.

Methodology

The interactions between agents were formalized as a *Chemical Reaction Network* (CRN), following Gillespie's stochastic formulation [2]. The system's evolution is governed by a *chemical master equation*:

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{A}\,\mathbf{P}(t) \tag{1}$$

where the rate matrix $\mathbf{A}$ is defined as:

$$A_{ij} = \begin{cases} w_{\vec{n}_i \leftarrow \vec{n}_j} & \text{if } i \neq j \\ -\sum\limits_{k \neq i} w_{\vec{n}_k \leftarrow \vec{n}_i} & \text{if } i = j \end{cases} \tag{2}$$

Here, $w_{\vec{n}_i \leftarrow \vec{n}_j}$ denotes the transition rate from state $\vec{n}_j$ to state $\vec{n}_i$, as determined by the system's stochastic dynamics. Each state $\vec{n}$ encodes the count (or proportion) of agents of types X, Y, and U at a given time.

To simulate the system, we employed the **Gillespie algorithm**, which generates statistically exact sample trajectories of continuous-time Markov chains (CTMCs) governed by the master equation.

Implementation

The simulation was implemented in Python. Each experimental configuration was run multiple times and results were averaged. The core logic followed the Gillespie algorithm, parametrized by interaction rates q_X, q_Y and zealot proportions.

Results

We aimed at exploring **robust consensus**, defined as:

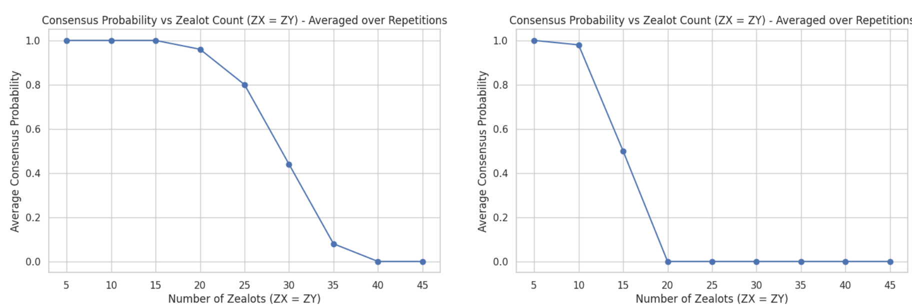
$$F^{\leq t}\left(G^{\leq h}\left(\left[X + Z_X > \min_m \wedge X - Y > d\right] \vee [Y + Z_Y > m \wedge Y - X > d]\right)\right)$$

Key observations from the simulations include:

- **Robust consensus**: A robust consensus (defined as stability over a window $h = 40$) was reached before $t = 35$ in 45% of the simulations.
- **Effect of zealots**: Introducing a moderate zealot fraction ($Z_X = Z_Y = 0.05$) led to robust consensus in 65% of trials. However, increasing zealots beyond 40% reduced consensus rates sharply.
- **Effect of group size**: Larger swarm sizes, with constant zealot proportions, increased the likelihood of reaching consensus.

What follows is the description of our experiments and the results that were gathered. Both experiments were performed keeping both by keeping the zealots "external" to the population and including them in it, as we noticed that this simple difference led to drastically different results. More specifically, this means keeping, in the first case, the population equal to N **excluding the zealots**, while in the second keeping it equal to N including the zealots; to give an example of this, with a proportion of 0.10 zealots for both X and Y, we will have a total of 120 robots in the first case, and 100 in the second.

- **Robust Consensus Probability**: A robust consensus is reached with high probability both with 'internal' (0.995) and 'external' (0.999) zealots.
- **the effect of zealots on Robust Consensus**: scaling the proportion p of zealots leads to drastically different results for internal and external zealots. While in the first case we had no consensus just for a $p \geq 0.4$, in the second no consensus was reachable already at $p \geq 0.2$

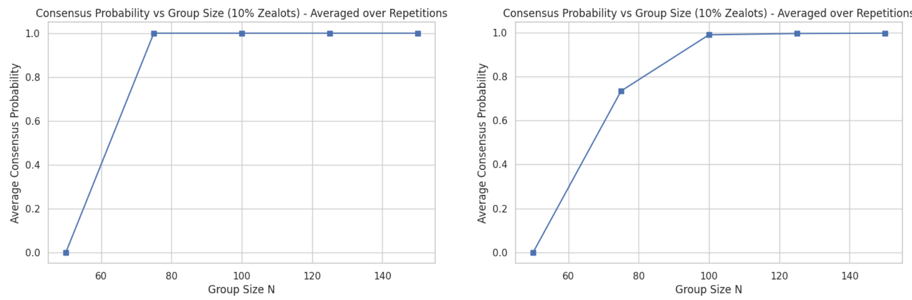


(a) varying "external" zealots

(b) varying "internal" zealots

Figure 1: Effect of varying zealot proportions on consensus probability under different configurations.

- **explore the effect of group size on Robust Consensus**: Larger swarm sizes, with constant zealot proportions, increased the likelihood of reaching consensus. However the latter went up less rapidly with "internal" zealots



(a) varying ”external” zealots **(b)** varying ”internal” zealots

Figure 2: Effect of varying zealot proportions on consensus probability under different configurations.

Limitations and Future Work

Although our study provides insight into how zealot proportions and group sizes influence consensus, it only scratches the surface. Future work should explore the role of network topology, spatial constraints, and noise. A deeper statistical characterization of consensus time distributions and tipping points across parameter regimes is also warranted. Furthermore, the stark contrast in results for ”internal” and ”external” zealots should be further explored and formalized, as it bears great resemblance to many situations in reality.

Code

All the code for both projects can be found here: github.com/GiovanniBillo/MultiAgentSystems_Projects

References

[1] Vincent Conitzer and Tuomas Sandholm. An algorithm for automatically designing deterministic mechanisms without payments. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 128–135, 2004.

[2] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.