The Inquisitiveness Model (`INQ 1.0`)

Supplementary Materials

Davide Secchi

Research Centre for Computational & Organisational Cognition

University of Southern Denmark

June 11, 2020

The document here presents the Inquisitiveness Model (Section 1) by introducing information structured according to the ODD Protocol (Grimm et al., 2020). Some of the information are also reported on the `Info` tab in the `NetLogo` Model (uploaded separately). It then follows a detailed description of the full range of tests used to perform the *sensitivity analysis* (Section 2) to calibrate the model. The document finally shows results that are connected to but are not in the published paper (Section 3).

Text of the CoMSES-OpenABM reviewer's report (received 18 September 2018):
The reviewer for release Inquisitiveness in ad hoc teams v1.0.0 has requested changes.

Thanks for submitting your model for peer review! Our reviewer would like for you to improve the narrative documentation. There is a short narrative but since it does not provide any equations or flow diagrams it is impossible to grasp what the model aims to do. The authors are advised to use a protocol like the ODD protocol of Volker Grimm et al, and use equations and flow diagrams to explain to the reader what the model aims to do.

I have to apologize to CoMSES reviewers, in an attempt to find the reviewer's report, I hit the "ready to resubmit" button. All I wanted was to look into the report and revise the model. Well, I am afraid you had to go over the original model while I was working on this document and on the revised INQ1.0 Model. Real sorry about the waste of time!

CONTENTS

1 THE ODD PROTOCOL

The ODD Protocol is a standard introduced to help modelers describe their ABM by following a threefold structure: Overview, Design, and Details (Grimm et al., 2017). First introduced to suit ecology models, it has been updated several times in the last decade (Polhill, 2010; Grimm et al., 2010) until a last update appeared in the *Journal of Artificial Societies and Social Simulation* (JASSS) in 2020 (Grimm et al., 2020).

This version of the model is slightly different from the one used in the paper Bardone and Secchi (2017). The original model caused `NetLogo` to occasionally produce a number that was too large to compute. This problem did not affect the data but it was just annoying to deal with when running the model 'qualitatively' so to speak (i.e., performing random runs to check how parameters work). The modified lines of code are identified in the new version of file uploaded on OpenABM.

In the following, the protocol is outlined and descriptions presented, divided in its seven sections.

1.1 Overview

This Subsection is dedicated to providing readers with a general understanding of the model, outlining its purpose, describing the agents, and sketching the process.

1.1.1 Purpose and pattern

The `INQ 1.0` Model is structured to study how team dynamics supports problem solving activities depending on whether team members operate within or outside the boundaries of the team. Two aspects should be mentioned. One is that behavior of individual members depends on their personal attitudes towards others, and on the availability of information (other individuals) outside of their team. The other is that is that teams are intended as part of an organization, so that it makes sense that team members "reach out" to other teams and/or organization members.

The model aims at tackling with two separate albeit interconnected purposes (Edmonds et al., 2019). One is to *illustrate* what happens when the team is "broken", so to speak, and it becomes an open system where members lean on information coming from different sources, including those different than the team. Team cohesiveness (Colquitt et al., 2002) is based on a self-referential argument that may be useful in terms of motivation and satisfaction of members, but not necessarily the most effective when it comes to problem solving. The other is to *explore* the theoretical arguments around the concept of *docility* (Simon, 1993; Secchi, 2011; Secchi and Bardone, 2009), the attitude to lean on information coming from social channels to make decisions. *Inquisitiveness*, as described in the related paper (Bardone and Secchi, 2017), is an extension of *docility*. In the model, this is parametrized as *socially-oriented decision making*, or `sodm`. While the latter has been show to work only within well-defined communities, groups, and teams (Secchi and Gullekson, 2016; Secchi, 2016), the former breaks this requirement. Hence, the model serves as an exploration of the theory of docile behavior, and is a bridge towards a possible theory of inquisitiveness.

1.1.2 Entities, state variables, and scales

There are two different agent types in this simulation: *employees* and *problems*. The number of employees $N_e[0, 500]$, and the number of problems $N_p[0, 500]$ can be set through a slider in the Interface tab of the software.

Employee characteristics. Employees are those who populate the organization, deal with tasks by using their own characteristics and exploiting resources. They are distributed randomly in the system and their total number (circle-shaped, yellow) in the system is set by the slider *num_employees*.

Each agent-employee is assigned the following:

- `competence` — Distributed random-normally with $\sim \mathcal{N}(1, 0.5)$, it is the level of professionalism that is relevant to the task/job. Both mean and standard deviation are parametrized and controllable from `Netlogo`'s Interface tab.
- `socially-oriented decision making` (docility) — Distributed random-normally with $\mathcal{N} \sim (0, 1)$, it is the attitude with which one is willing to cooperate with and use information from others. Both mean and standard deviation are parametrized and controllable from `Netlogo`'s Interface tab.
- `enquiry` — Distributed random-normally with $\sim \mathcal{N}(0, 1)$, it is the attitude with which one is willing to accept information from others outside of one's own team. Both mean and standard deviation are parametrized and controllable from `Netlogo`'s Interface tab.

Problem characteristics. They are distributed randomly in the simulation space and their total number (box-shaped, red) is set by the input box *proportion-tsk/prt*, and it should be read as the number of tasks available per participant. So, for example, the number 2 in the box means that tasks are twice the number of employees.

Each agent-problem is assigned the following:

- `difficulty` — Distributed with a random-normal distribution with $\sim \mathcal{N}(3, 1)$, it represents how hard a problem is in relation to its performance and completion. Both mean and standard deviation are parametrized and controllable from `Netlogo`'s Interface tab.

Other setup conditions. There are a series of other inputs that the model requires before it can start. Here they are briefly outlined:

- `looking_for_problems` — When ON, the switch allows agent-employees to move towards one problem, selected at random from those available in the organization.

- `inquisitiveness` — The ON/OFF switch lets inquisitive individuals use a non-linear function to integrate team members' competencies. The switch makes the function work together with the level of enquiry, so that only decision makers with high enquiry levels perform non-linear aggregation.
- `proximity` — This standard feature for ABM sets the visibility radius for each agent, and it can take values according to the range $[0, 20]$.
- `tolerance` — This is a threshold that triggers a decrease of `competence`, if non-cooperation conditions are met (see below under 'process overview').
- `cooperation` — The ON/OFF switch enables agent-employees to work with others on a problem.
- `probs-evolve` — This ON/OFF switch is used to allow problems to multiply, according to a rate specified as:
  - `problem spin-off` — This parameter takes values between $[0, 10]$ and it is the top number through which problems can multiply at any step of the simulation. For example, if $PSO = 4$ it means that at every step (tick) of the simulation, one among the five most difficult problems multiplies and produces a random $[0, 4]$ number of new problems. The `difficulty` of these problems is set to be chosen at the low end of the difficulty distribution of problems in the system.
- `increase_comp_rate` — This parameter range is $[0, 1]$, and it sets the extent to which `competence` increases after an agent-employee solves a problem.
- `decrease_comp_rate` — This parameter range is $[0, 1]$, and it sets the extent to which `competence` decreases after an agent-employee is not able to solve a problem.

1.1.3 Process overview and scheduling

The stop is set at `step`$= 500$. Alternatively, when 90% of the initial number of `problems` are solved or when they become 3 times more than their initial level, the simulation stops. Every 'tick' in the software is what I call an opportunity for the agent-employee to establish interactions and solve problems.

At the beginning of the simulation, a number of `problems` and `employees` are selected and defined according to the parameters above. They appear at random in the environment (an organization). Given that most agents are created with a set of characteristics assigned at random and that we are interested in the study of how teams (of various shape and characteristics) perform, random location would allow for a wide variety of teams.

The general rule for movement in the environment is that they get closer to a problem when a link is established and when `looking_for_problems` is turned OFF. When this switch is ON instead, it is possible to move forward 1 pixel at the time to reach a problem — this is, of course, if the agent-employee is not involved with any problem at that moment. Problems do not move, but their size increases as their difficulty evolves. This feature is not reflected in the flow chart below (Figure 1) but it is important. Problem `difficulty` is not a static quality. In fact, a problem's difficulty increases as time passes by and as it keeps remaining unsolved. However, this is not done for easy problems. At every round, a random selection of 2 the most difficult problems are set to increase their difficulty by 2%.

Each agent scrutinizes the area around `proximity` looking for problems. When it finds one, a link with a problem is established while links with other agent-employees are only established when `cooperation` is ON. Once connections with problems are established, their thickness grows 0.1 at every step (tick), as a reference for multiplying efforts on it. After 20 steps, the link is severed. This is to indicate that an agent cannot keep working on a problem forever; in an organization, there are always new tasks and new problems to solve. One needs to move on.

The solution of a problem is a rather simple matter, when the `competence` of the agent-employee is higher than the `difficulty` of the agent-problem, then it is solved. When solved, a problem disappears from the system.

A flow chart of the main processes in the model is presented in Figure 1. This takes some of the options described above for granted, and part of a general setup, while other procedures that affect the end result more promptly are shown. A few disclaimers to be able to read the chart are necessary. As a starter, all basic functions are performed given a certain value for `proximity` that does not change during a simulation's run. Design and details are covered in the following pages

Shared competences (shared comp in Figure 1) is calculated by taking the mean of `sodm` $\times$ `competence` c for all the agent-employees in range of proximity, hence connected by a link to the decision maker. A decision maker is an agent-employee connected to an agent-problem.

$$S_i = \frac{\sum_{i=1}^{n}(c_i * d_i)}{n} \tag{1}$$

where S_i is the shared competences, n is the number of agent-employees in range, i is an agent-employee, c_i and d_i are competence and docility (socially-oriented decision making) of the agent-employee.

Under *cooperation* there are two possibilities for agent-employees. One is to use the sharing option just described that works for the average individual, with $d_i < \bar{d} - 0.75 \cdot s(d)$, where d_i is the docility level for the agent-employee, $\bar{d}$ is the mean and $s(d)$ is the standard deviation of d in the system.

For highly docile individuals, with $d_i < \bar{d} + 0.75 \cdot s(d)$, the system works differently, because these individuals are supposed (according to the theory) to deal more efficiently with information coming from others. Agents also use `enquiry` e if the switch `inquisitiveness` is ON, such that those agent-employees with $e > \bar{e}$ can use competence from outside of their teams. This implies an increase in the `proximity` parameter that, for these highly inquisitive agents, increases by 50%.

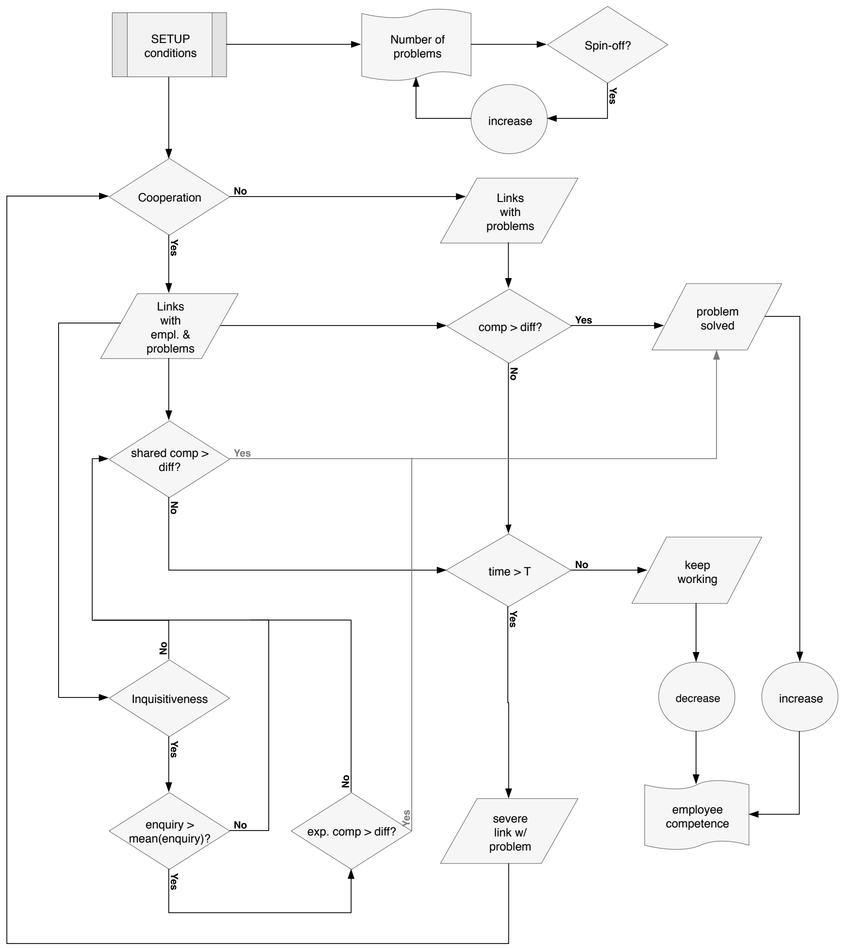


Figure 1: Flow chart of the INQ1.0 Model

For highly inquisitive agents, the shared information is re-coded as:

$$S_e = S_i + S_i \cdot \left(1 + \frac{e_i}{max(e)}\right) \tag{2}$$

where S_e, the shared competence for highly inquisitive agents, the shared competence is S_i (Eq1), e_i is the level of the parameter `enquiry` for the agent, and $max(e)$ is the maximum level of that same parameter in the system. Then, the competence of the inquisitive agent becomes

$$E_i = c_i + \frac{1}{|c_i|^{ln(S_i+1)}} \tag{3}$$

where E_i is benchmarked with the difficulty of a problem d_p and, if

$$E_i > d_p \tag{4}$$

then a problem is solved. Equations 3 and 4 are also the same used for highly docile individuals, with S_i defined as explained above (Equation 1).

At the end of every procedure, when a problem is solved or abandoned, each agent-employee resets its shared competencies but keeps some trace in its competence levels, that increase or decrease according to the two parameters described above.

1.2 Design concepts

The idea of this model is particularly unique in that it takes ad hoc teams and uses them as areas of collective decision making, channeled through one team member (the decision maker). More details on this aspect are in the paper written with Emanuele Bardone (Bardone and Secchi, 2017).

In the following, I will try to cover the various aspects that are indicated in the ODD under this section. Not all of them are relevant or particularly meaningful in the `INQ1.0` Model. I have decided to write a few words even in the case of relatively minor relevance.

1.2.1 Emergence

By taking a classic definition of emergence (e.g., Cunningham, 2001, *epistemic emergence*$_2$), the behavior of the system cannot be fully tracked down to the characteristics of its parts. In this simulation, this is particularly apparent as the number of agent-problems grows disproportionately in relation to the number of agent-employees. In that case, the R^2 of the regressions that use the number of problems solved as a dependent variables decreases dramatically (see Table 2 in our paper, Bardone and Secchi, 2017). This indicates that the amount of variance in the dependent variables is not determined solely by either the parameters or the characteristics of the agents.

It is interesting to notice that these emergent properties are also repeated at the intermediate (*meso*) levels of the model. Put differently, it is difficult to predict — given the initial setup conditions — how ad hoc teams are going to address the problems and when a team with given characteristics will be able to solve a problem. Both the variability in the number of agents associated with the team, slight movements (mimicking ambiguity), and their adjusting characteristics make for some emergent properties of these team working environments.

1.2.2 Adaptation

Both agent types have some characteristics that adapt, and they have been already reviewed above. In short, the agent-employee adapts its competences depending on the experience it gains with solving or abandoning problems. The agent-problem sees its difficulty increase as time goes by, when difficulty is already high.

In addition to these adaptation mechanisms, some agent-employees with $d \sim \bar{d}$ may switch to a more docile behavior if they observe that it is successful. Vice versa, they switch to non-docile if that strategy is successful.

1.2.3 Objectives

The objective of the simulation is simply to explore what are the conditions under which problems are solved. Hence, the adaptation mechanisms above are not necessarily designed to create a "winning" situation, i.e. a situation where the most docile or competent individual solves more problems. On the contrary, the simulation is set to represent seemingly realistic situations and understand what the outcome is. The criterion is problem solved, but the simulation does not attempt at designing the most successful team. In other words, the simulation is a tool to understand what can be achieved when individuals and teams have certain features.

1.2.4 Learning

One could say that the adaptation of competence due to experience in dealing with problems is a learning trait. However, in management and organizational research learning would bring us to an almost completely different area of the discipline. This is why the paper does not refer to learning even though there is some, although it is captured in a rudimental way.

1.2.5 Prediction

The only predictive feature of the model is a "history bias" that some agent-employee make when they attempt to imitate their most successful peers (see above). As far as predictive qualities of this model, there is no data associated to it and there has not been any validation/verification (as in Boero and Squazzoni, 2005), only calibration.

1.2.6 Sensing

This is explained in details in the section above. The model takes a very simple approach, and that is agent-employees sense the difficulty of a problem, as well as filter (through docility and enquiry) competences of others around them.

1.2.7 Interaction

The environment represents an organization where employees behave. The space is not physical, but a mental (psycho-cognitive) area where agents share competences and attempt at solving some of the problems that appear in their work life. The interaction is therefore apparent when these occurrences happen in shared locations of the environment.

The interaction is numerical. Individual agents share values according to the mechanisms above, and this is the way in which they interact.

1.2.8 Stochasticity

Random components in this simulation are:

- initial location of all agents
- distribution of competences, docility, difficulty, enquiry
- number and characteristics of new problems
- agent-employees movement (when allowed; see above)

1.2.9 Collectives

Teams can be considered collectives in this simulation. This means that they are the unit of analysis and of “action” as far as problem solving activities are concerned.

1.2.10 Observation

No empirical data has been collected on this topic so far. This could be an interesting plan for future research.

1.3 Details

This Subsection is necessarily slim, because of the purpose of the `INQ1.0` Model. In fact, the model serves *illustration* and *theoretical exploration* purposes, and there is no input data.

1.3.1 Initialization

At setup, the model needs most parameters to be specified at a given level. When the start button is clicked, agents appear in random position on the environment, some close to each other, some distant. Again, I believe the information here has been specified above and it is, in part, also repeated in the next section. I won’t repeat it here.

1.3.2 Input data

The model does not uses any input data.

1.3.3 Submodels

The `INQ1.0` Model is relatively simple, although it has a few sections of its code that can be considered as *blocks* rather than submodels. I am writing this because none of these portions of code does not work in isolation, but needs the others to work properly. These modeling *blocks* are:

- setup — defines the initial conditions;
- move — makes agents relocate smoothly on the environment;
- activate — defines the state of agent-employees depending on their docility levels;
- solve — general procedure to match the difficulty of a problem with the competence of an employee;
- evolve — establishes adaptation patterns for all agents;
- cooperate — establishes ways in which agents share their competencies.

2 PRELIMINARY ANALYSES

A series of preliminary computational experiments have been performed on the supercomputer Abacus 2.0, eScience infrastructure, available at the University of Southern Denmark. Results were used to launch a sensitivity analysis on the parameters ranges.

Most of the information related to these preliminary runs is available on the article Bardone and Secchi (2017) and will not be repeated here. The result of these runs led to a factorial design of $2 \times 2 \times 2 \times 2 \times 3 \times 3 \times 2 = 288$ configuration of parameters, as it can be seen from Table 1 (this is largely what is in the article).

Here comes my standard text in Supplementary Materials files when dealing with power:

> "Given we are in front of a stochastic simulation, there is the possibility that one run is not enough to understand and interpret results. However, by using a technique suggested by Secchi and Seri (2017), I used power analysis to calculate how many times each simulation should have been performed. Assuming one compares the means of the outcome variable with an ANOVA, then power analysis requires the number of groups (experimental runs, in our case), then an estimate of the effect size $f \approx 0.1$, that was derived from preliminary runs, $\alpha = 0.01$ and power $1 - \beta = 0.95$ (as suggested by Secchi and Seri, 2017; Seri and Secchi, 2017). This result was of the reversed F formula was $n \approx 30$ runs. This meant that 30 run are enough to prevent error Type II from happening."

The result of the procedure was $288 \times 30 = 8640$ computational experiments.

Table 1: Parameter Notations and Values

Parameter	Values	Description
steps	**500**	The number of opportunities that agents have to interact with each other when dealing with problems.
initial number of problems, $N_{P,0}$	**100**‡**, 200**‡**, 300**	Initial number of problems in a given environment (organization), i.e. at time zero
problem spin-off, pso	**2**‡**, 4**‡	This is the top number through which problems can multiply at any step of the simulation.
initial number of decision makers, $N_{dm,0}$	**100**‡**, 200**‡**, 300**	Initial number of decision makers in a given environment (organization), i.e. at time zero
difficulty, d	$\sim \mathcal{N}(3,1)^{\ddagger}$	Each problem is associated with a difficulty level, random-normally distributed.
competence, c	$\sim \mathcal{N}(\mathbf{1},\mathbf{1.5})^{\ddagger}$, $\sim \mathcal{N}(3,1.5)^{\ddagger}$	This is the knowledge — associated to each decision maker — that is needed to solve a given problem.
competence increase rate, γ_i	0.15‡, **0.30**‡	The rate at which competence increases if a problem is resolved.
competence decrease rate, γ_d	0‡, **0.05**‡	The rate at which competence decreases if a problem is not resolved.
socially-oriented decision making, $sodm$	$\sim \mathcal{N}(0,1)$	This is the docility of each agent and measures, on average, how much one leans on information coming from others to make decisions.
enquiry, e	$\sim \mathcal{N}(0,1)^{\ddagger}$	This is the enquiry level that would facilitate agents dealing with knowledge (competence) coming from others in the simulated organizational environment.
inquisitiveness	**true**‡ **/ false**‡	This triggers the different ways that agents have to deal with groupwork.
range	**6**‡	This is the value used to explore the environment that surrounds each agent.

Note. ‡ = parameter values included in the first simulation test; **bold font** = parameter values included in the simulation discussed in the analysis.

3 GRAPHICAL ANALYSIS

The following pages present preliminary analyses conducted on the data by producing plots and images using the software R for statistical computations. Some of the plots below have been included in the final article (Bardone and Secchi, 2017), while most of them have been used as a preliminary step towards understanding what was actually in the data.

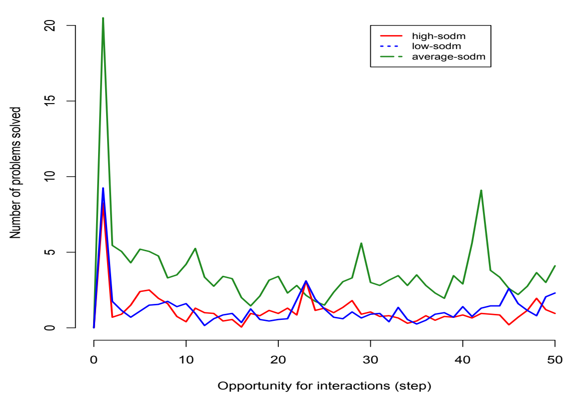
There are sporadic and sparse comments on the figures that follow, just a plot is produced and, whenever possible, figures are classified according to the parameter values used to produce them.

The first batch of figures, from Figure 2 til Figure 11 are part of a preliminary set of analyses with two main aim: (a) ascertain that the simulation is performing as it should and (b) understand wether there are general patterns.

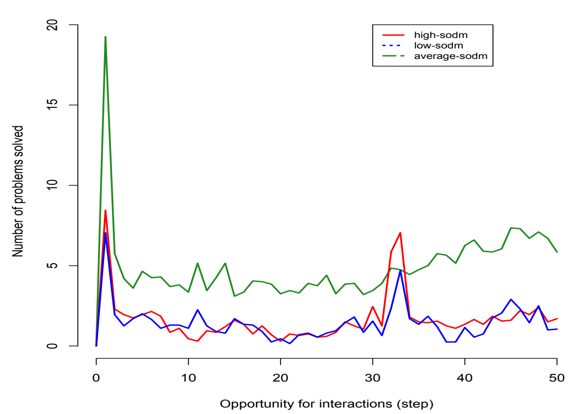
3.1 Number of problems solved

The two panes of Figure 2 focus on the number of problems solved over time and compare `cooperation` OFF (pane (a)) vs ON (pane (b)). Agent-employees with average docility ($\bar{d}$) are those who solve more problems. This is hardly surprising, since they are roughly 50% of the population.

What seems interesting is that the number of problems solved by these "average" employees increases with cooperation, even though they are not the ones to lead cooperation, highly docile employees are. It seems as if they exploit competencies of the highly docile. These two panes indicate that an exploration of the dynamic of competences in the system is probably the key to understand patterns in the data. The next plots are an attempt to do exactly that.



(a) `cooperation`=OFF

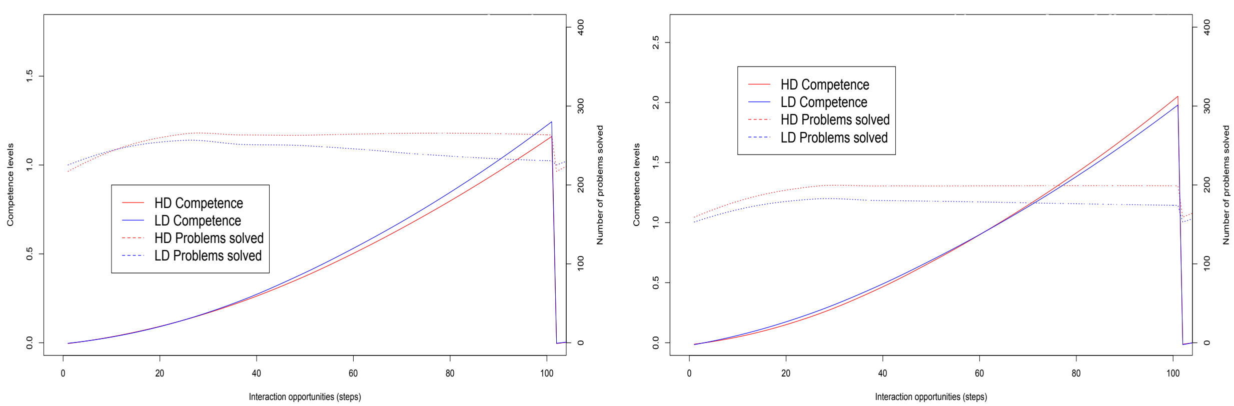


(b) `cooperation`=ON

Figure 2: Number of problems solved over time

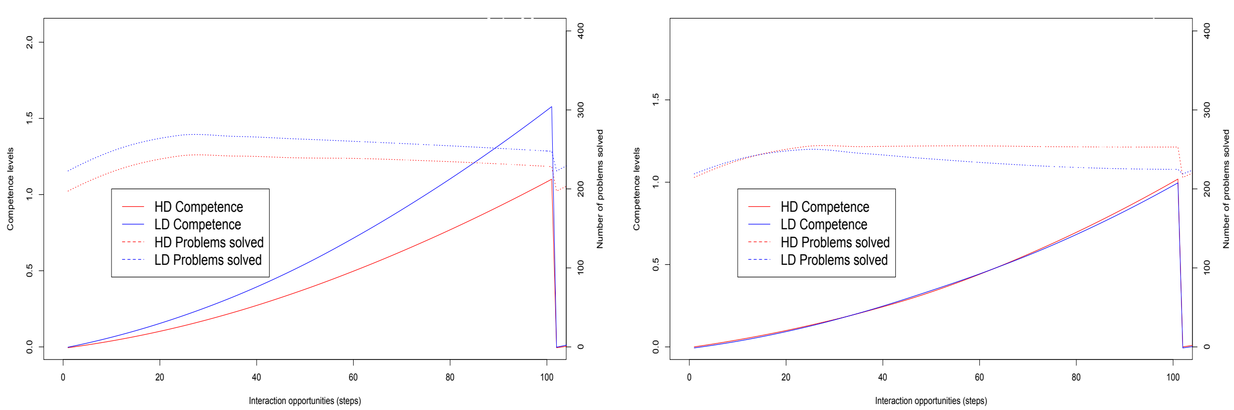
3.2 Exploring competence

The plots under this subsection all follow a very similar pattern. They show the evolution of competence (left y axis) and the number of problems solved (right y axis) for high and low docile agent-employees. Each figure then compares `inquisitiveness` ON vs OFF and contains a variation of the initial number of agent-problems $\{100, 200, 300\}$, of the spin-off $pso = \{2, 4\}$, and of the number of agent-employees $\{100, 200, 300\}$.



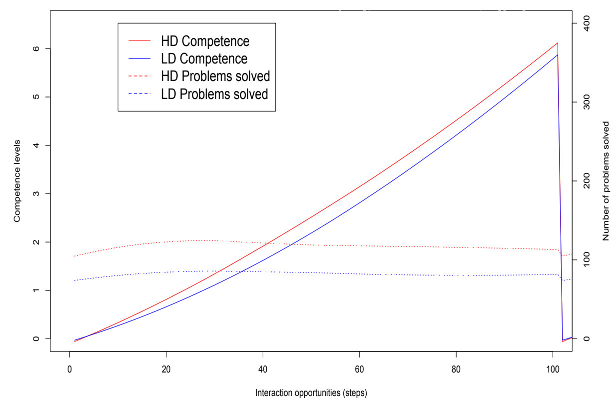
(a) `inquisitiveness`=ON (b) `inquisitiveness`=OFF

Figure 3: Employee-agent competencies (left scale) and number of problems (right scale) over time ($pso = 4$, $N_e = 200$, $N_p = 100$)

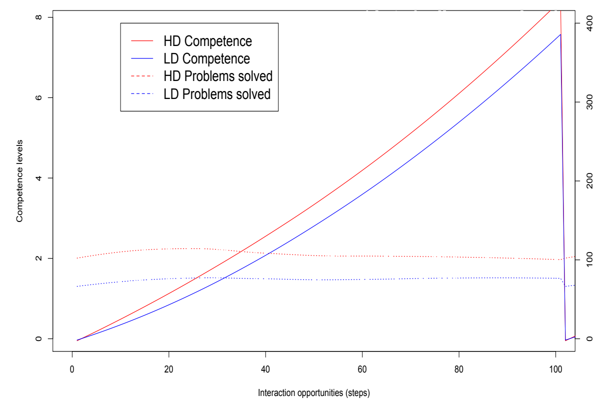


(a) `inquisitiveness`=ON (b) `inquisitiveness`=OFF

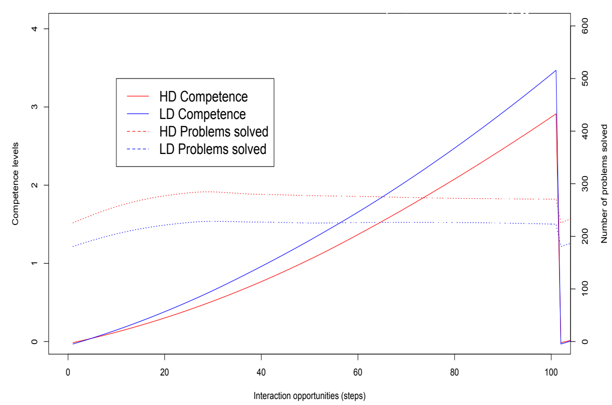
Figure 4: Employee-agent competencies (left scale) and number of problems (right scale) over time ($pso = 2$, $N_e = 100$, $N_p = 100$)



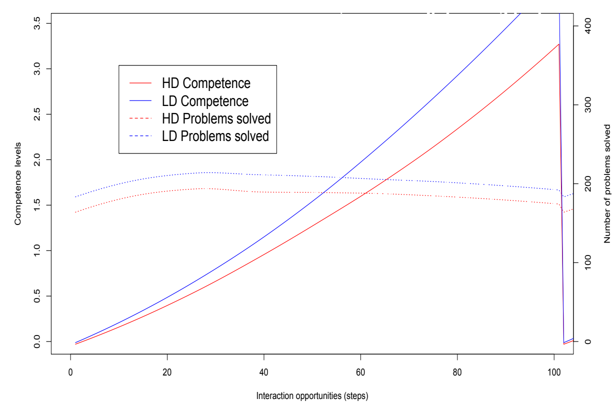
(a) `inquisitiveness`=ON, $pso = 2$



(b) `inquisitiveness`=OFF, $pso = 2$

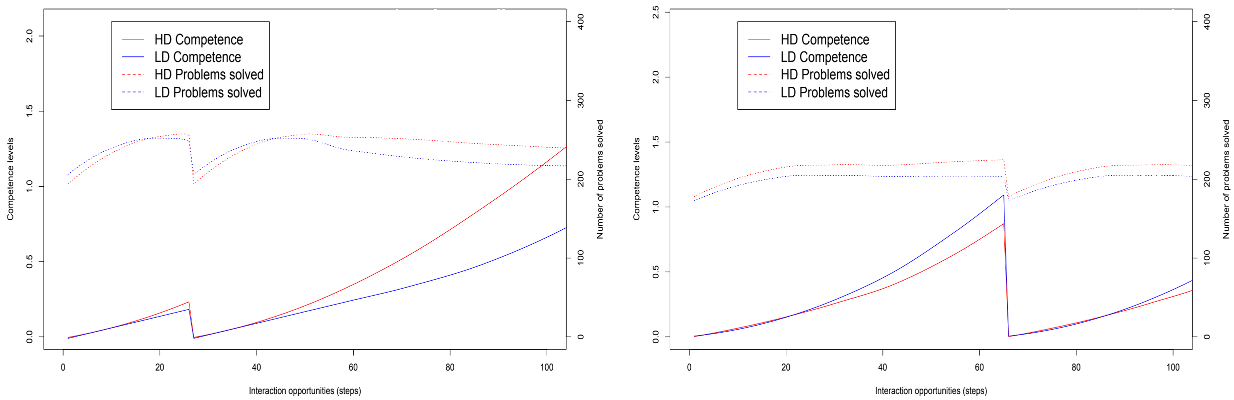


(c) `inquisitiveness`=ON, $pso = 4$



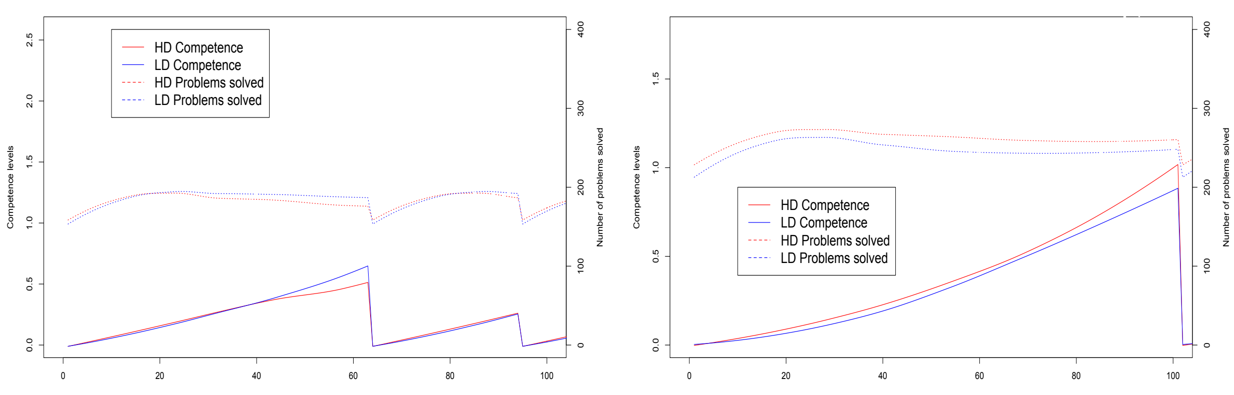
(d) `inquisitiveness`=OFF, $pso = 4$

Figure 5: Employee-agent competencies (left scale) and number of problems (right scale) over time ($N_e = 100, N_p = 300$)



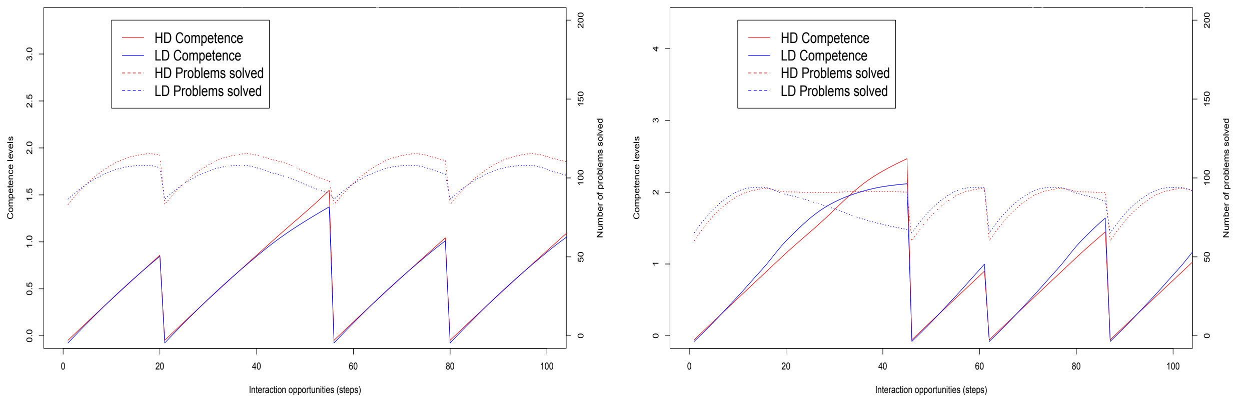
(a) `inquisitiveness`=ON (b) `inquisitiveness`=OFF

Figure 6: Employee-agent competencies (left scale) and number of problems (right scale) over time ($pso = 4, N_e = 300, N_p = 100$)



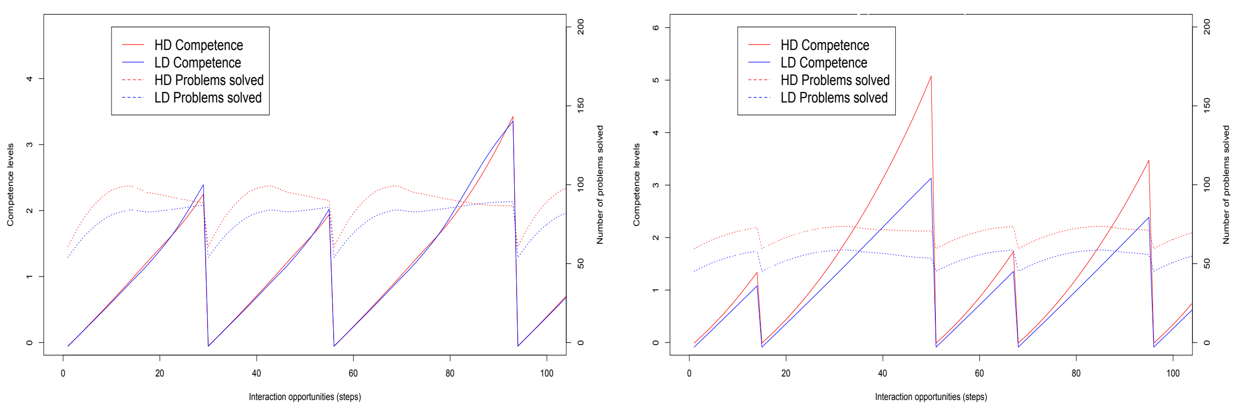
(a) `inquisitiveness`=ON (b) `inquisitiveness`=OFF

Figure 7: Employee-agent competencies (left scale) and number of problems (right scale) over time ($pso = 2, N_e = 300, N_p = 100$)



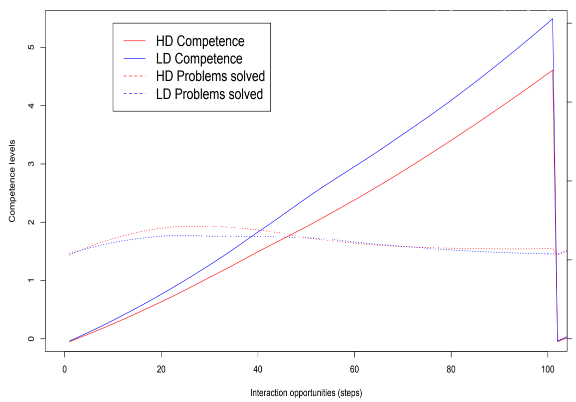
(a) `inquisitiveness`=ON (b) `inquisitiveness`=OFF

Figure 8: Employee-agent competencies (left scale) and number of problems (right scale) over time ($pso = 2, N_e = 300, N_p = 300$)

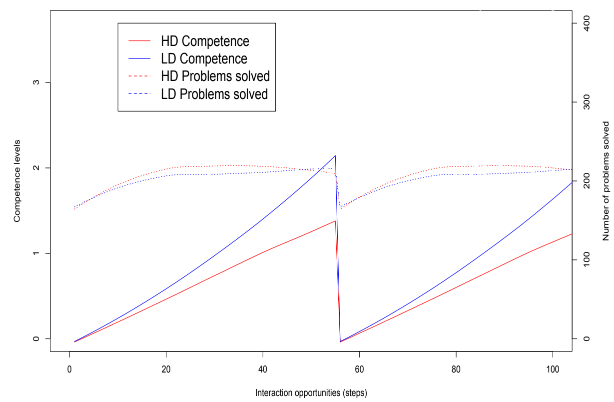


(a) `inquisitiveness`=ON (b) `inquisitiveness`=OFF

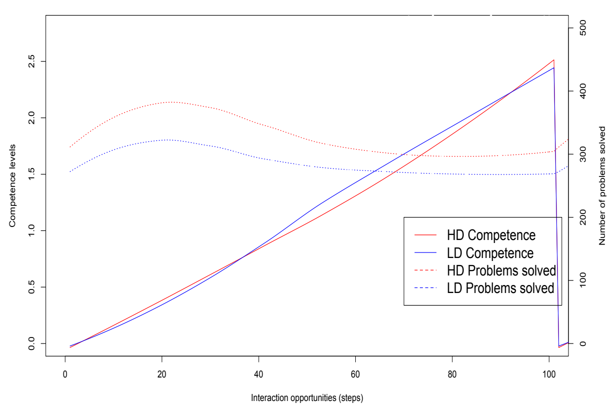
Figure 9: Employee-agent competencies (left scale) and number of problems (right scale) over time ($pso = 4, N_e = 300, N_p = 300$)



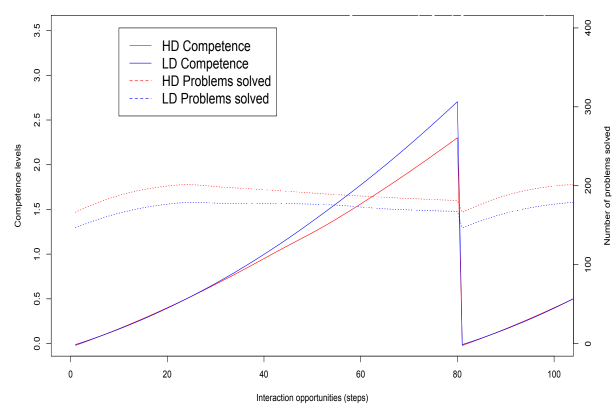
(a) `inquisitiveness`=ON, $pso = 2$



(b) `inquisitiveness`=OFF, $pso = 2$

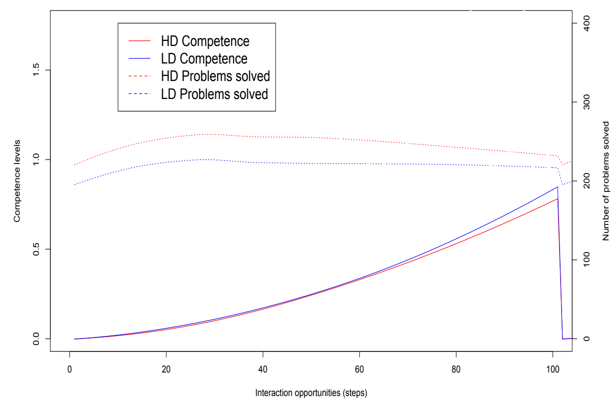


(c) `inquisitiveness`=ON, $pso = 4$

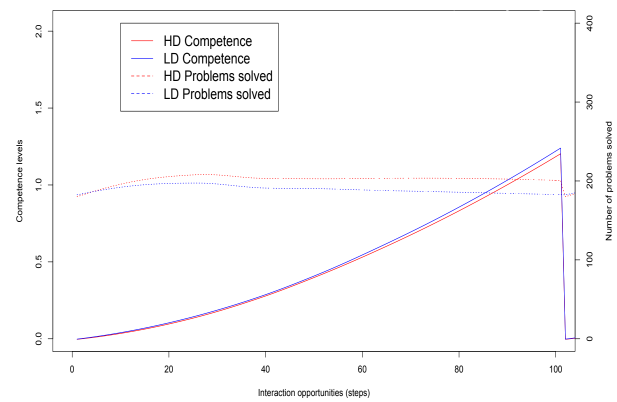


(d) `inquisitiveness`=OFF, $pso = 4$

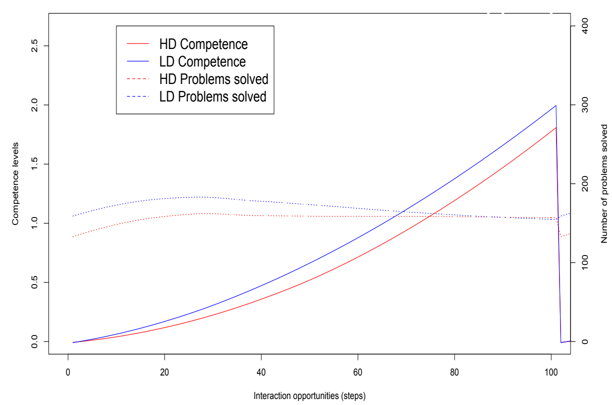
Figure 10: Employee-agent competencies (left scale) and number of problems (right scale) over time ($N_e = 200, N_p = 300$)



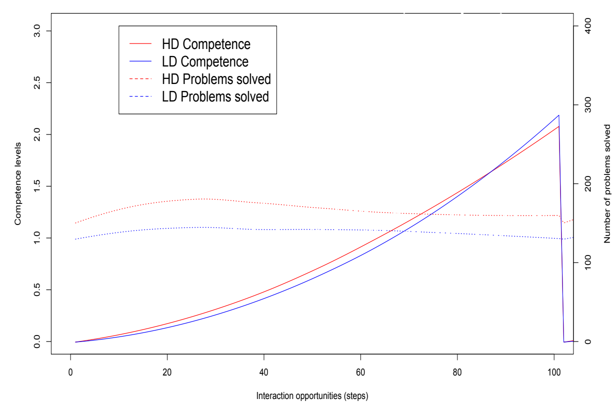
(a) `inquisitiveness`=ON, $pso = 2$



(b) `inquisitiveness`=OFF, $pso = 2$



(c) `inquisitiveness`=ON, $pso = 4$



(d) `inquisitiveness`=OFF, $pso = 4$

Figure 11: Employee-agent competencies (left scale) and number of problems (right scale) over time ($N_e = 100, N_p = 100$)

3.3 Overview figures

The figures in this section (from Figure 12 to Figure 16) can be intended as a summary of the results above. While the figures above have been an attempt to explore the data "space", the ones in this subsection intend to present an overview of the most noticeable trends.

In fact, the figures published in the article (Bardone and Secchi, 2017) have been selected from the ones below.

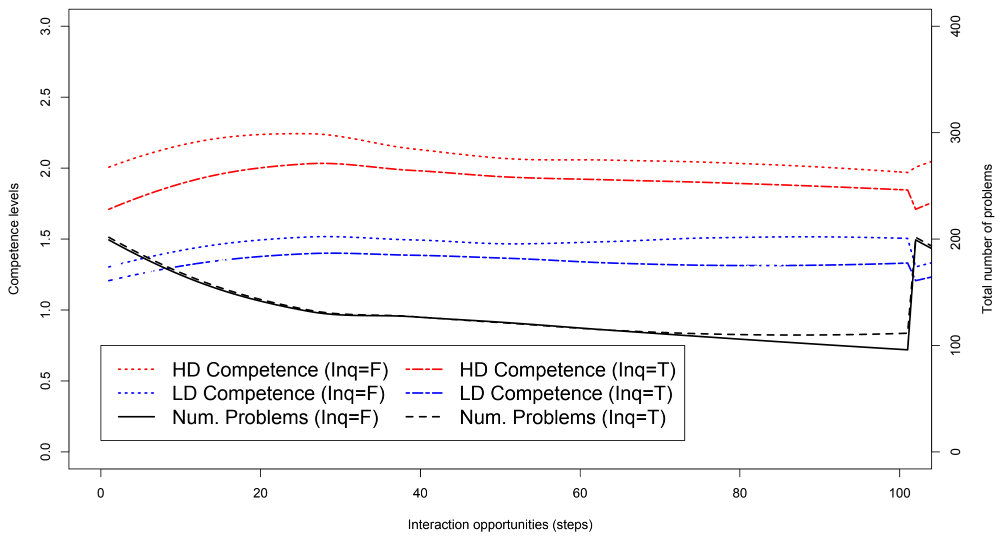
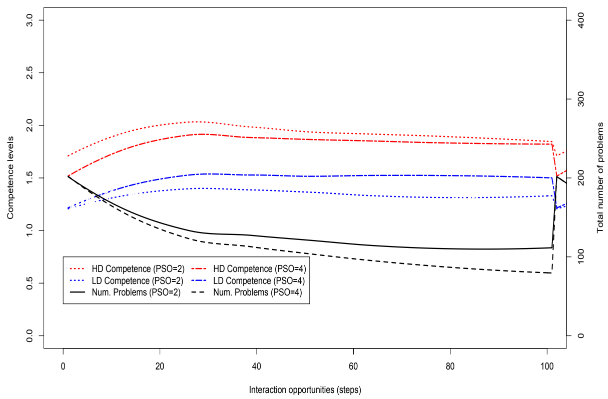
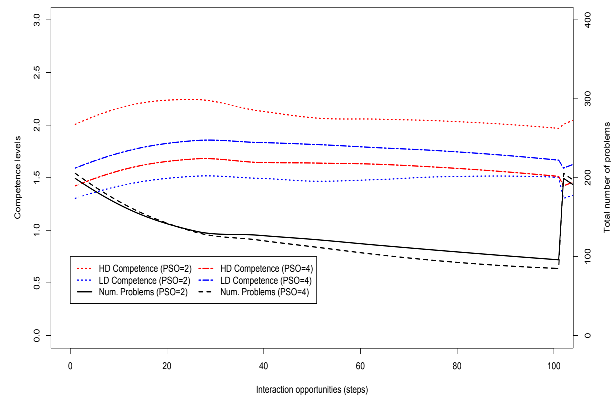


Figure 12: Employee-agent competencies over time (`cooperation`=ON)

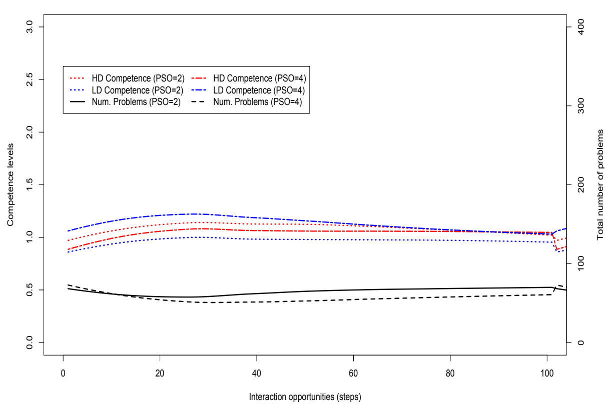


(a) `inquisitiveness`=ON

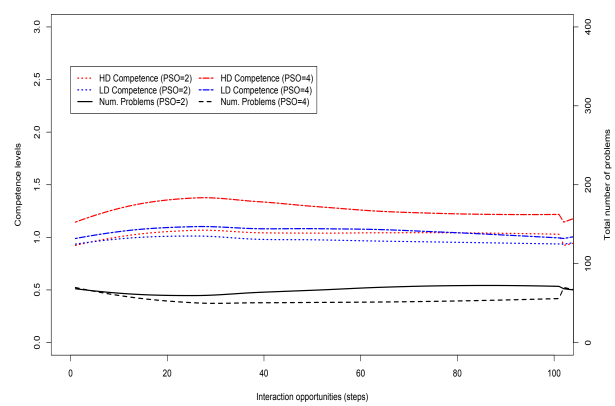


(b) `inquisitiveness`=OFF

Figure 13: Employee-agent competencies (left scale) and number of problems (right scale) over time ($N_e = 100, N_p = 300$)



(a) `inquisitiveness`=ON



(b) `inquisitiveness`=OFF

Figure 14: Employee-agent competencies (left scale) and number of problems (right scale) over time ($N_e = 100, N_p = 100$)

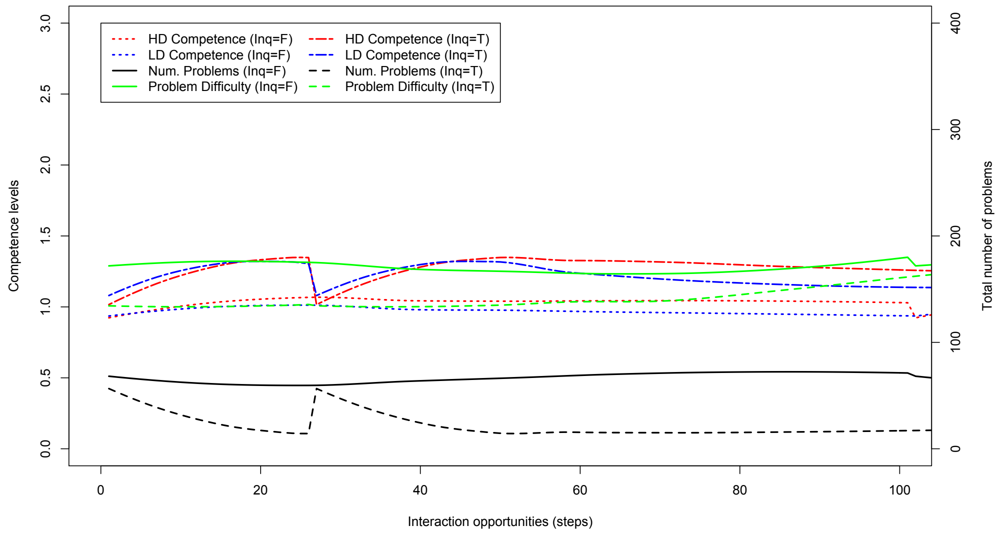


Figure 15: Employee-agent competencies (left scale) and number of problems (right scale) over time (`cooperation`=ON)

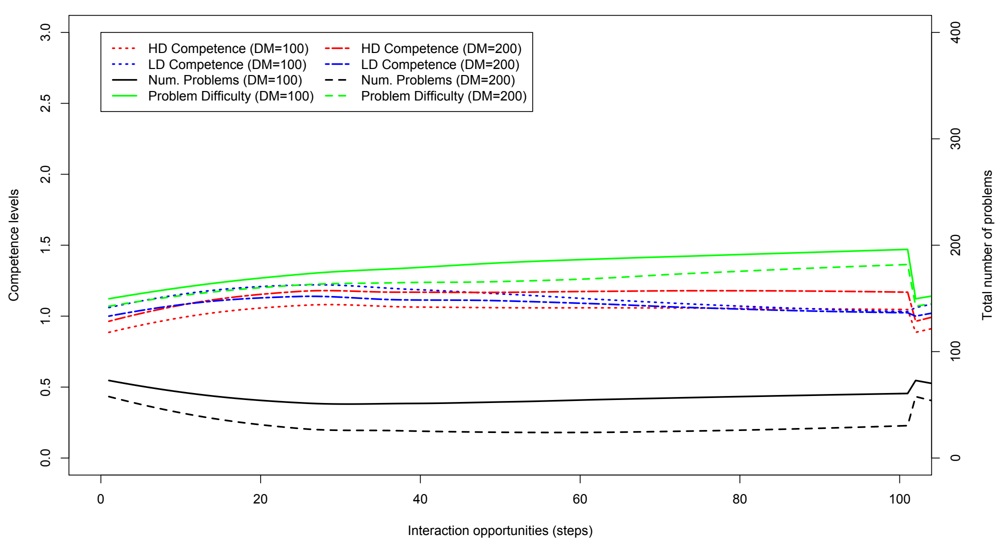


Figure 16: Employee-agent competencies (left scale) and number of problems (right scale) over time (`cooperation`=ON)

4 WHAT'S NEXT

The `INQ1.0` model had been very interesting for me and Emanuele Bardone, it has helped us move forward from the concept of docility. The next steps would include testing the limits of inquisitiveness, after fully evaluating its theoretical strengths. In order, the latter should come before the former.

One model that tests a similar scenario to this is the `IOP2.0` Model, also available on CoMSES-OpenABM. That model uses a diversity of decision making tactics based on distributed cognition (e.g., Hutchins, 1995), including some related to docility and inquisitiveness. There is also a related publication (Secchi, 2020).

REFERENCES

Bardone, E. and Secchi, D. (2017). Inquisitiveness: Distributing rational thinking. *Team Performance Management*, 23(1/2):66–81.

Boero, R. and Squazzoni, F. (2005). Does empirical embeddedness matter? methodological issues on agent-based models for analytical social science. *Journal of artificial societies and social simulation*, 8(4):6.

Colquitt, J. A., Noe, R. A., and Jackson, C. L. (2002). Justice in teams: Antecedents and consequences of procedural justice climate. *Personnel Psychology*, 55(1):83 – 109.

Cunningham, B. (2001). The reemergence of 'emergence'. *Philosophy of Science*, 68:S62–S75.

Edmonds, B., Page, C. L., Bithell, M., Chattoe-Brown, E., Grimm, V., Meyer, R., Montanola-Sales, C., Ormerod, P., Root, H., and Squazzoni, F. (2019). Different modelling purposes. *Journal of artificial societies and social simulation*, 22(3):6.

Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., and Railsback, S. F. (2010). The odd protocol: a review and first update. *Ecological modelling*, 221(23):2760–2768.

Grimm, V., Polhill, G., and Touza, J. (2017). Documenting social simulation models: The ODD protocol as a standard. In Edmonds, B. and Meyer, R., editors, *Simulating Social Complexity. A Handbook*, pages 349–365. Springer, Cham.

Grimm, V., Railsback, S. F., Vincenot, C. E., Berger, U., Gallagher, C., DeAngelis, D. L., Edmonds, B., Ge, J., Giske, J., Groeneveld, J., et al. (2020). The odd protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, 23(2).

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: MIT Press.

Polhill, J. G. (2010). ODD updated. *Journal of Artificial Societies and Social Simulation*, 13(4):9.

Secchi, D. (2011). *Extendable rationality. Understanding decision making in organizations*. New York: Springer.

Secchi, D. (2016). Boundary conditions for the emergence of 'docility:' An agent-based model and simulation. In Secchi, D. and Neumann, M., editors, *Agent-Based Simulation of Organizational Behavior. New Frontiers of Social Science Research*, pages 175–200. New York: Springer.

Secchi, D. (2020). Cognitive attunement in the face of organizational plasticity. *Evidence-Based Human Resource Management*, forthcoming.

Secchi, D. and Bardone, E. (2009). Super-docility in organizations. An evolutionary model. *International Journal of Organization Theory and Behavior*, 12(3):339–379.

Secchi, D. and Gullekson, N. (2016). Individual and organizational conditions for the emergence and evolution of bandwagons. *Computational and Mathematical Organization Theory*, 22(1):88–133.

Secchi, D. and Seri, R. (2017). Controlling for 'false negatives' in agent-based models: A review of power analysis in organizational research. *Computational and Mathematical Organization Theory*, 23(1):94–121.

Seri, R. and Secchi, D. (2017). How many times should one run a computational simulation? In Edmonds, B. and Meyer, R., editors, *Simulating Social Complexity. A Handbook*, pages 229–251. Heidelberg: Springer, 2nd edition.

Simon, H. A. (1993). Altruism and economics. *American Economic Review*, 83(2):156–161.