Internship Report on Multi-Agent Simulation of Climate Negotiations

CHANATTAN SOK, ENS Rennes, France

In an urgent call for policies towards climate change, several meetings such as COP are organized every year. These allow statespersons to agree on decisions ratified by their respective parties. However, negotiations made to reach agreements are no easy processes, hence the study of interactions between such actors became relevant. The model studied here tries to reproduce a negotiation meeting between negotiators who must decide on a social choice ratified by their respective constituencies and agreed upon by every negotiator, a cooperation problem. The approach uses a probabilistic multi-agent simulation built on previous work by Earnest on a 3-cooperation problem, based on two-level game theory from Putnam. The decided social choice must be accepted by every agent with respect to their preferences. Such a model allows us to study if any international consensus is possible and may give intuitions on what we should focus on to improve negotiations. Earnest's base model shows in practice that fewer negotiators with poor transnational influence between constituencies may lead to discord, as opposed to a larger number of negotiators cooperating more and allowing consensus more easily. Several extensions are implemented as a means to generalize and enlighten intuitions in a developed tweakable model. However, the proposed tweakable model hinted at some counter-intuitions with different modelings of parameters. A larger number of negotiators may lead to greater discord overall, reminding us to carefully approach the study of agent-based stochastic models.

1 INTRODUCTION

This internship was carried out at LIP6 in Sorbonne University in the context of the 2-year NEGOCLIM[1] project funded by MITI (Mission for Transversal and Interdisciplinary Initiatives, CNRS) in 2023 with the purpose of modeling and simulating climate negotiations. This internship aims to study a starting point for this purpose and to give insights into future works. As such, a workshop and numerous meetings were held; joining economists, humanitarians, climate scientists, mathematicians, and computer scientists sharing their state-of-the-art knowledge to improve climate negotiations; in which I participated.

Considering the climate context, several climate meetings are organized with negotiations between state representatives in order to reach a consensus for future adopted policies. Nonetheless, such negotiations involve complex interactions, as a negotiator may represent an immense diversity of opinions and is subject to domestic constraints in his party. Our study of interest focuses on climate negotiations between international negotiators as state representatives who must find a consensus on a certain choice to make among starter choices. As a matter of fact, we mainly studied generic negotiations with the hope of a later connection to climate consequences detailed afterward. The main object of study is the work of Earnest, both his article on Coordination in Large Numbers [6] and associated code-base developed on NetLogo.

In order to approach this problem, we make use of an Agent-Based Model (ABM) [2]: a simulation model to study the interactions between autonomous agents (individuals or groups) aiming, here, to replicate human behaviors [8]. The very first goal is to make a close copy of Earnest's model in the Python programming language, with respect to both Earnest's article and his code base. This model is itself based on the two-level game theory introduced by Putnam on diplomacy and

I have been supervised by **Nicolas Maudet** and **Aurélie Beynier** during my internship, from May 15 to July 28, whom I would like to thank for their clear viewpoints and directions. I also thank the rest of the members of the NEGOCLIM project Arianna Novaro, Antoine Mandel, Agnieszka Rusinowska and Stéphane Zuber, as well as the participants of the Workshop on Modeling and Simulating Climate Negotiations, Sorbonne University, June 19, 2023. They all gave prospects to this internship's evolution and provided valuable pieces of information on modern climate negotiations.

[1]https://sites.google.com/view/negoclim/home

domestic politics [11]. We may after that focus on the model's improvement and extensions. This would allow us to confirm or not the intuitions stated by Earnest in his model. The developed model in Python would serve as a code base to constantly extend, making a usable and tweakable model that would be as close as possible to real-world situations.

ABMs have shown great efficiency and are widely used in the modeling of complex macro-behaviors, more than that, given some parameters those models may reveal non-linear interactions between parameters that classic equation-based models, such as World3[2], lack to reveal. The idea is that ABMs are stochastic models which give non-deterministic results that would be closer to real-world seen phenomena.

Based on Earnest's work, we are interested in international negotiations. We want to know whether the negotiators, associated with respective parties, under domestic constraints can reach a global consensus or not, i.e., every agent must agree on the final voted choice by the negotiators.

The use of an ABM is useful to model and detail simple behaviors at an agent level: the view of agents of the world, either partial or whole, their possible set of actions towards it, and their own attributes. As such, agents are different and independent units interacting between them and giving large-scale behavior trends in the global world. The micro-behaviors we are particularly interested in are the influences between agents (both negotiators and constituencies) as they are given preferences over starter choices and the negotiations between negotiators. The replication of Earnest's model was trickier than expected since there were some differences between his code and his article. However, the most interesting and hardest part was to extend the model while ensuring it was as close as possible to real-world negotiation situations.

The second goal was indeed to extend Earnest's model, first extensions were only natural because of initial simplistic modelings made by Earnest to give some basic insights into his model. The rest of the extensions have been worked on both theoretically and in code. Most extensions came from the willingness to improve the model to make it resemble real-world negotiations, as such the main challenges were to try to implement efficiently and realistically those elements into code. Several approaches have been made by scholars in an attempt to model human preferences and negotiations. We followed the direction of Earnest's core model in the hope of making it more generic with more *realistic* options, making this a true tweakable model.

If Earnest's base model allowed to confirm some intuitions on some political turning points or gave interpretations on a global state's behavior under its domestic constraints, the study in detail of his model showed specific modeling choices and simplifications which, considering the stochastic nature of ABMs, revealed to be needing a careful approach on the yielded results. Our study with different modelings, explained in detail, across our tweakable model raised different intuitions notably on the importance of transnational networks and the compliance degree of constituencies over transnational influences. Particular attention has been paid to overall parameters, but some of them may be neglected in relation to other dominant parameter effects. As opposed to Earnest's results, a larger number of negotiators with low negotiation power but quite compliant towards their constituency, and constituencies which slowly adopt the other states' influences with relatively dense transnational connections tend to give more overall discord. This doesn't necessarily cancel out any intuition over generic international negotiations, but this mainly reminds us that meticulous attention is required, especially when working with real-world simplified stochastic models.

[2]Introduced in the book "The Limits to Growth" by Jay W. Forrester (1972)

2 EARNEST'S MODEL

We introduce the terminology and the basic notions from Earnest's model, as well as its study. It must be kept in mind that some of the following definitions refer to this specific model.

Earnest's model is an ABM with agents being negotiators and constituencies. This is a stochastic model built from bottom-up: agents are given certain attributes and a program (i.e., detailed micro-behavior) as pieces of sub-systems that may be linked to other sub-systems by interactions in order for more complex and higher-level systems (i.e., macro-behaviors) to emerge.

Definition 1 (Environment)**.** An environment is a world populating agents where interactions can happen, and agents can have spatial coordinates and therefore a spatial representation. The word *world* will sometimes be confused with the word *environment* here.

The used simulation is the study of a living environment evolving by iteration (i.e., step by step) allowing a clear study with eventually yielded data for each iteration. A *world iteration* is the simulation's state at a given step. The simulation here assumes to execute the program of all agents as a quasi-parallel processing[3] in a certain order, presumed synchronous, the eventual scheduling problems are ignored to focus on the study.

Definition 2 (Coordination problem)**.** A coordination problem is a problem in which actors aim to coordinate (i.e., find a logical order of steps to take to reach a collective agreement).

Based on two-level game theory, we define both negotiators and constituencies as agents, with negotiators being agents of level I and constituencies agents of level II. Each negotiator has a respective constituency. The illustration in Figure 1 from Earnest's article allows a clear vision of the model.

Earnest's model focuses on a 3-cooperation problem, which means it considers only 3 possible choices to choose from. A *choice* is an arbitrary object, e.g., a letter A, B, or C. Each agent has a *vote* for a given iteration which is a choice among the proposed ones, it may change over time.

Definition 3 (Social choice)**.** The social choice is the choice voted among all negotiators at the end of a world iteration.

Definition 4 (Global choice)**.** The global choice is the choice voted among all agents (negotiators and constituencies) for a given world iteration.

Definition 5 (Cooperation)**.** A world iteration is in a cooperation state if the social choice made is the same as the global choice (i.e., the social choice satisfies the global preference).

Definition 6 (Discord)**.** A world iteration is in a discord state if the social choice made is different from the global choice.

Definition 7 (Consensus)**.** The considered consensus here is the state reached once all agents in the world have the same vote for a given iteration, i.e., unanimity.

[3]The microprocessor of a computer executes the program of each agent before the next iteration of the world is taken.

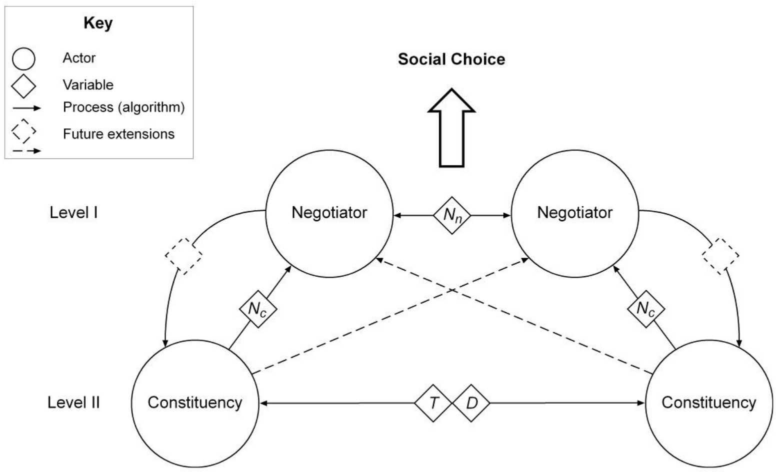

Fig. 1. Earnest's model - Figure from Earnest's article [6] with 2 negotiators - agents (circles) interact between them based on the model's parameters (arrows with labeled variables)

2.1 Objective

The goal of the model is to simulate a 3-cooperation problem with $\boldsymbol{N}$ negotiators in order to find whether they can reach a consensus under which parameters of the model, as the study focuses on a cooperation problem, the simulation ends once a consensus is reached. The simulation may also end prematurely, with the number of iterations empirically capped at 200. This prevents simulations from running too long in the case of an impossible reachable consensus. In this model, any taken vote follows the plurality rule (or relative majority) among voters (in particular the social and sub-social choices).

2.2 Preferences

Given such an objective, each agent has its own preferences over the set of proposed choices that may change over interactions. With this modeling, all agents have access to the possible world choices plus the global choice involves every agent, this approach gives agents a certain viewpoint of the world to agents.

A negotiator's preferences are a descending total order of the possible choices, i.e., A > B > C. Choice A is preferred over B itself preferred over C.

In the modeling of one constituency per negotiator, a constituency represents a population or a state. The population may have divergent opinions on the possible choices, as such, we aggregate such preferences in proportion to model them as weighted preferences on the possible choices, i.e., A: 58%, B: 37%, C: 5%. The population globally prefers the most the choice A over B which itself is preferred over C.

For any agent, the *top choice* is the most preferred choice according to its preferences. It is important to note that a negotiator's vote is its top choice, whereas a constituency's vote is a distinct attribute from its weighted preferences, but obviously based upon.

Agents may change their preferences or vote over interactions during a world iteration. There are two types of interactions between agents :

- Influence only between negotiators, those happen during the bargaining step preceding the decision of the social choice. It involves exactly two negotiators per negotiation.

- Influence between negotiators and constituencies.

Interactions are parameterized in order for the modeler to control and develop intuitions on the model.

2.3 Negotiations

Given two negotiators $\mathscr{N}_1$ and $\mathscr{N}_2$, a negotiation happens, for example when $\mathscr{N}_1$ negotiates with $\mathscr{N}_2$: $\mathscr{N}_1$ proposes its[4] top choice to $\mathscr{N}_2$ which may adopt it with a certain probability parameter $\boldsymbol{N_n}$. It is interpreted as the bargaining power of the negotiator $\mathscr{N}_1$. The adoption of the proposed choice by $\mathscr{N}_1$ is done by swapping the current top choice of $\mathscr{N}_2$ with the proposed choice in $\mathscr{N}_2$'s preferences' order unless both choices were the same.

Example 1. We consider the same negotiators $\mathscr{N}_1$ and $\mathscr{N}_2$ and the starter choices A, B and C. $\mathscr{N}_1$ and $\mathscr{N}_2$ respectively order their preferences as $A > B > C$ and $C > B > A$. $\mathscr{N}_1$ negotiates with $\mathscr{N}_2$ by proposing its top choice A, suppose the negotiation succeeded, A is different from $\mathscr{N}_2$'s top choice C: $\mathscr{N}_2$ swaps its top choice C with the proposed choice A in its order, $\mathscr{N}_2$ now orders its preferences as $A > B > C$.

2.4 Influences

A constituency may influence its negotiator with a certain probability parameter $\boldsymbol{N_c}$, as the pair negotiator constituency suggests, a certain cohesion between preferences of both agents must remain. This way, the negotiator may represent its constituency's opinion. The negotiator adopts its constituency's vote as its new vote, and the negotiator's top choice becomes the constituency's vote.

A negotiator may also influence its constituency with a certain probability $1 - N_c$ considered as the persuading power of the negotiator's preferences. The $1 - N_c$ probability lets to know that such influence is complementary to the influence of the constituency towards its negotiator. Indeed, either one of the influences effectively happens at each world iteration, leading to a *complementary influence*.

Constituencies may be influenced by some neighboring constituencies to model transnational networks around the world. Therefore, for a given constituency $\mathscr{C}$, a subset of constituencies in a radius of $\boldsymbol{T}$ including $\mathscr{C}$ may realize a sub-vote to decide which choice is the most preferred among themselves, or to say, a sub-social choice. If the top choice (and not the vote) of $\mathscr{C}$ differs from this sub-social choice, then $\mathscr{C}$ re-weights its preferences progressively at a certain speed given by the parameter $\boldsymbol{D}$, adding more weight to the decided sub-social choice. $\mathscr{C}$ keeps re-weighting until the next (next iteration) sub-social choice is different from the previous one, focusing on this new one instead. The re-weighting is maxed out at 100% preference.

2.5 Model parameters

As N negotiators negotiate in the simulation, each having a unique constituency, the world populates $2N$ agents. Table 1 shows probabilistic parameters bound between 20% to 80%. From an intuitive viewpoint, extreme values may be translated in the model as (i) denial of bargaining (N_n = 0%) and constituency's impotence (N_c = 0%) or (ii) negotiator's dictatorship (N_n = 100%) and constituency's absolute democratic power (N_n = 100%). In both (i) and (ii) cases, parameters were bound arbitrarily to avoid far-fetched situations differing from real-world relevant negotiation situations.

[4]We consider negotiators and constituencies as agents and thus as objects.

The world is modeled as a 35×35 matrix in which agents are placed. Negotiators are placed as objects onto a circle with equal space distribution in between them, respective constituencies are placed on another circle with the same center as the previous one but with a larger radius equal to 15. The agents do not move. In order to simulate transnational relationships, constituencies have a radius of T in the matrix, which captures a certain number of neighbor constituencies.

If an influence between constituencies happens for a given constituency $\mathscr{C}$ for a sub-social choice A (this means A is different from the top choice of $\mathscr{C}$), then the re-weighting of A by $\mathscr{C}$ is done iteratively (i.e, at each world iteration) in the following way :

(1) Pick a number uniformly between 1 and $D\%$
(2) Add that number to the weight of the choice A in $\mathscr{C}$'s preferences
(3) Re-weight all preferences[5] of $\mathscr{C}$ except A

The model focuses on a 3-coordination problem, let us consider 3 choices A, B, C with the same A as stated above. To re-weight all preferences of $\mathscr{C}$ except A as in step (3), first pick arbitrarily one of the two remaining choices, for example, B, then pick a random number uniformly between 0 and (100 - the weight of A) thus the remaining available weight in proportion becomes the weight of C. Such distribution favors the increment of the sub-social choice while keeping the total of weights to 100%, it also allows generating hardly uniform preferences of a constituency as one population's preferences generally rarely are.

Parameters	N	N_n	N_c	T	D
Range of values	3 - 40	.2 - .8	.2 - .8	10 - 30	1 - 5

Table 1. Parameters bounds

2.6 World initialization

At iteration 0 the negotiators' preferences are generated uniformly randomly from the proposed choices, giving random total orders. The vote of a negotiator is its top choice, and its constituency's vote is initialized to be the same.

Given the proposed starter choices A, B, and C, the constituencies' preferences are generated using the same re-weighting defined above except that it is for all preferences: first, generate a weight $w1$ uniformly from 0 to 100, then a second weight $w2$ from 0 to (100 - $w1$), the remaining available weight to complete to 100% gives $w3$. The weights are sorted in descending order and the first item refers to the weight of the constituency's vote, the second item to the weight of the second preferred choice of its negotiator's preferences, and so on. Constituencies' preferences are initially based on their negotiator's preferences which may be interpreted as coherent cohesion in a state, it is worth noting that it remains an arbitrary initialization.

2.7 Simulation dynamic

The simulation respects a specific order of execution[6] of agents' behaviors:

(1) Negotiators decide a social choice among themselves, if there is no consensus then continue.
(2) Negotiators negotiate/bargain with each other.

[5]The re-weighting is different from the re-weighting of only one choice.
[6]As noted previously, the order of execution is random for all agents. The specific order of execution here must be understood as an order of execution of behaviors. (i) First negotiator agents are executed (still randomly) then (ii) constituency agents are executed (still randomly). Negotiator agents in (i) first negotiate then may influence, similarly in (ii) constituency agents first may influence and then may get influenced.

(3) Negotiators may influence their constituency.
(4) Constituencies may influence their negotiator.
(5) Constituencies may get influenced by their neighbor constituencies.

Steps (2) to (4) involve the probabilistic parameters N_n and N_c. Steps (3) and (4) are directly linked, a constituency may influence its negotiator back if the negotiator failed to influence the constituency. Step (5) involves both parameters D and T, however, the influence by neighbor constituencies for a given constituency is always happening. The constituency may change its preferences only if the neighboring constituencies sub-voted a different choice than the constituency's vote.

2.8 Model study

The model can be studied in many aspects, the use of such simulation allows iterative-yielded data which may get collected as statistical data to see how a simulation progresses. As such, we can study the preferences variations as curves over time, but we mainly study the discord metric, which represents the number of world discord iterations. It should be noted that we can easily deduce the complementary metric of cooperation.

To study the consensus objective, we focus on two sub-objectives: the study of simulations with maximized discord and the study of simulations with minimized discord. The intuitive idea is that cooperation would tend to ease negotiations and thus allow the reach of a consensus earlier, whereas discord would constantly promote the emergence of new propositions, making negotiators never reach an agreement.

Another interesting metric is the discord (or cooperation) metric normalized by the number of past iterations for a given simulation, this gives hints about a world simulation's discord in proportion to its existing time (i.e., before reaching consensus). This metric is only shown for intuition purposes on NetLogo's interface.

In order to study such a model, Earnest uses Active Non-Linear Tests (ANTs) from Miller [10] exploiting genetic algorithms to explore the parameter space. The current model uses 5 parameters (N, N_n, N_c, T, and D) with different value intervals, leading to an infinite amount of possible initialization sets. If we consider the probabilistic parameters only as real numbers bounded to their hundredth, the parameter space would be restrained to 14,846,790 possibilities. Genetic algorithms have proven their efficiency in the exploration of such enormous parameter spaces, allowing a quick selection of parameters optimized for a given objective function or a fitness criterion. In this study, a set of parameters is considered as a solution among others in the parameter space, each solution has a fitness value evaluated according to a fitness criterion, the higher, the better. To remain within the study of our two sub-objectives, the fitness criterion refers to the discord metric. Any given solution computes its fitness value by simulating the model, with the solution's parameters as inputs, to estimate the discord at the end of the simulation.

ANTs run 40 generations of 40 solutions. First 40 random solutions are generated, then the solutions are then tested to evaluate their fitness value. Thereafter, they go through a tournament selection by pairs[7] according to their fitness value (greater fitness solutions are kept for maximized discord and lesser ones for minimized). Finally, the solutions undergo genetic operators with a certain probability, i.e., the solutions may change. These operators guide the algorithm to converge towards optimum sets of solutions for the given fitness. There are two genetic operators[8]: cross-over and mutation. The cross-over operator takes in a pair of two randomly chosen solutions and creates

[7]In genetic algorithms, tournament selection is a type of solution selection where: 1. choose k solutions among the solutions at random and 2. keep in the next generation the solution with the best fitness with probability p, keep the ith (starting with i=1) next best solutions with probability p(1-p)^{i}. Since we consider pairs, we use this selection with k=2.

[8]Actually, the selection may already be considered as a genetic operator itself. However, the algorithm makes a clear distinction between selection and changes in solutions, this detail is irrelevant.

a new solution based on the chosen ones as an intertwined version of the two, it can be considered as a child of the two parent solutions with parameter values taken partially from both parents. Cross-over allows new mixed solutions to emerge, potentially creating better ones. The mutation operator takes in a randomly chosen solution and changes the parameter values with a certain probability decreasing over time. This last operator ensures solutions diversity to avoid reaching prematurely a local minimum in the solutions space, otherwise, the generations may stagnate giving suboptimal solutions. The probability of mutation decreases over generations for the algorithm to converge while preserving optimal solutions. In the used ANTs, the mutation is not used, and the cross-over has a decaying probability over generations. The algorithm loops for 40 generations in these three steps: solutions test, selection, and genetic operators before returning the final solutions. Such tests may take time to run, as we execute 40×40 = 1600 simulations with one simulation per solution test.

It should be emphasized that the elements of the model, by its design, are stochastic, and therefore it is susceptible to yield variable results.

Fig. 2. Solutions generated after running ANTs, table from Earnest's article [6]

Fitness criterion	*Number of Negotiators*	*Extent of Transnational Networks*	*Swings*	*Sensitivity of Negotiators to Constituents*	*Sensitivity of Negotiators to Other Negotiators*
Maximize discord	7	14	1	0.47	0.67
Minimize discord	38	26	3	0.54	0.39

2.9 Empirical results

Figure 2 shows 2 solutions after 40 generations of ANTs have run, the results were produced on NetLogo[9]. The tests have run two times (3200 simulations) to study the two sub-objectives. The model exploration teaches us that fewer negotiators with sparse transnational networks and poor transnational influence may lead to high discord (maximize discord sub-objective). On the other hand, negotiators in large numbers with constituencies with denser and stronger (i.e., higher influence) transnational communications for constituencies lead to less discord.

3 CONTRIBUTION

My contributions are of three natures, (i) the study and implementation of Earnest's model in another programming language and (ii) its extension, and finally (iii) a study of the extended model as more realistic negotiations. The study of Earnest's model is in part already written above, the true work was to understand it alongside Earnest's implementation and documentation. Hence, the study contribution is mainly considered as clarifying and formally defining Earnest's ideas. As (i) and (ii) have taken most of the time during the internship, the (iii) part is still testable in many aspects, especially with ANTs. The tweakable model is developed in Python[10] with a version greater than 3.10, it is available as a notebook. Specific details of the implementation can be found in the code in the comments. More global and theoretical details are available in the appendix.

[9]Model developed by Earnest on NetLogo version 3.1.4 (Wilensky 1999).
[10]The notebook is available here.

3.1 Model implementation

While working through Earnest's work, certain differences were noted. Earnest's work is separated in both his article and his code base (itself documented) on NetLogo. Differences between both works, either from misapprehensions or oversights, resulted in different model perceptions. We can emphasize two types of differences from Earnest: (a) the difference between the work in the article and what is done in his code, and (b) the difference between the work of Earnest (both in the article and his code) and what supposedly should have been made instead to strictly respect the cited work. Therefore, Earnest's model described above is as described in his code while being as close as possible to his article. With that being said, the implemented model was made in Python based on the above description of the initial model.

The first goal was to reproduce a working version of Earnest's model, it can be set up in the notebook by putting Earnest's model parameters in the tweakable model.

3.1.1 Differences of type (a). Following Earnest's article, negotiators' preferences are supposed to prefer a choice over another one which itself is preferred over another, especially in the case of a 3-coordination problem. That is, negotiators carry on orders on choices. However, the code of the initial model only implements such preferences as one choice for each negotiator, i.e., their vote. As a matter of fact, in the model, the negotiators only propose their vote to another negotiator during a negotiation, plus the influences made with constituencies are only about vote adoption.

Two important highlights are about the negotiator-constituency influences. The article states that a negotiator may adopt its constituency's *preferences*: it must be emphasized that the negotiator only adopts the constituency's *vote* as written in the code. The constituency's vote is itself redefined at the end of each world iteration based on its preferences. This is a less refined preference adoption.

Regarding the influence of a negotiator over its constituency, it is the same story of vote adoption. If the influence is effective, the constituency adopts the vote of its negotiator for the given world iteration. But, according to the simulation dynamic of Earnest, the vote of the constituency may change according to its preferences, themselves also potentially changed by the neighboring constituencies. The constituency's preferences are not changed when the negotiator influences it and the constituency's vote may change at the end of the same iteration, taking the choice with the greatest weight in its preferences. As such, the constituency may simply change back its vote to its top choice, even after being influenced by its negotiator. With that said, a negotiator's influence over its constituency only stays in effect if the negotiator's vote corresponds to the most preferred choice of the constituency, i.e., to its vote, which in fact, does not change the vote of the constituency anyway. Hence, a negotiator's influence is ineffective.[11] If a negotiator's influence were to be effective, it would be considered as a *dictatorial* influence, overlooking the constituency's preferences for the current world iteration. Since at the end of the iteration, the consensus is evaluated by considering the vote of all agents and not their preferences (for the default unanimity). The parameter $negotiator\_constituency\_influence$ proposed in this model allows effective negotiator influences to take place as described above. Though, it remains very little effective since a vote influence from a negotiator only stays in place for the current vote round (i.e., iteration) and the preferences of the constituency are not affected. With the base preferences modeling, constituencies, and neighbor constituencies may make their sub-social vote considering their actual vote and not their top choice. This can be done using the $constituency\_neighbors\_sub\_vote$ parameter. Otherwise, by default, constituencies make a sub-vote according to their top choice in their preferences.

[11]It must be emphasized that the influence of a negotiator over its constituency is considered as an extension in Earnest's article, however, later on in the same paper he still mentions this bargaining power in his simulation runs. Supposedly, this functionality is added this way in his NetLogo's code.

Changing the sub-vote to consider constituencies' votes may radically change the global discord of the simulations.

Such differences happen mainly because of the modeling of a separate vote attribute apart from the constituency's preferences, probably made with the hope to compensate for the lack of translation of a negotiator's preferences to a constituency's preferences. As the two agents' preferences are different by nature, it is complicated to translate a total order into weights.

The defined global preference as a cooperation method was actually a vote to decide a social choice but considering *all* agents. Each agent votes and has the same weight of 1, however, the constituencies vote according to their *vote* and not their top choice. In the case of effective negotiator's influence, it would respect the constituencies' preferences by ignoring their true preferences. The code implemented separately two cooperation methods: global preference and global vote. The two take the vote of all negotiators the same (by top choice) but the first takes the top choice of constituencies while the second their vote. For the next studies, the global vote is now considered to be the initial parameter of Earnest's model.

3.1.2 Differences of type (b). We review the ANTs conducted in Earnest's work compared to the ANTs described in Miller's article.

First, Earnest's code implementation does not include the mutation genetic operator generally used in ANTs. The mutation is supposed to have a probability of happening which decays over generations, instead, the cross-over holds that probability. This genetic operator allows us to avoid reaching local minima prematurely in the algorithm.

In addition, the implemented cross-over only happens for the first 20 solutions of the 40 solutions set since cross-over involves pairs of solutions to reproduce, making 40/2 pairs available for reproduction. But the solutions are chosen randomly in pairs, the limit of 20 remains arbitrary. The new solutions from cross-over also replace at most the 20 first solutions, and only those, from the precedent generation arbitrarily.

The implemented ANTs do not use a specific objective function for fitness, as mentioned in Miller's article [10]. ANTs were originally designed, as the name states, to test the structure and the robustness of models by revealing any existing non-linear interactions between parameters. Those tests should be conducted to try to *break* the model by maximizing deviation between the original model's predictions and the one obtained under perturbations[12], it is not the case here as the objective is to maximize/minimize the discord to find optimal solutions and not perturbing the model.

Although, the steps of implementation of ANTs may follow Miller's article, it would be more reasonable to qualify the implemented algorithm as a generic genetic algorithm, with the goal of finding optimal solutions with the fitness discord metric. However, to avoid confusion, we will still use the ANTs term in this report.

The solutions test step in ANTs allows us to define the fitness value of the solutions, the initial model was computing such value as the discord resulting from only one simulation. However, such a metric can greatly vary because of the stochastic nature of the model. Hence, we computed a fitness value as the mean discord over 100 simulations for a given solution. The mean technique to evaluate stable fitness values is often used in genetic algorithms, especially for simulated stochastic models, as shown in Stonedahl's study of ABMs [13] itself inspired by Miller's work.

[12]The initial purpose of ANTs, according to Miller's article, is to test the robustness of a model since the quality of many models is difficult to assess. Although ANTs may give some insights on non-linear interactions which may happen between a model's parameters, it is not sufficient alone to determine the resilience of a model.

3.2 Model extensions

Several extensions have been implemented in this model. The very first objective was to generalize as much as possible the initial model. Afterward, any improvements to the model for more complex and realistic negotiations were implemented within the available time. A short introduction to the developed model and any detail on how an extension/change was made should be available in the appendix. Details on the implementation are written in the notebook's code.

List of extensions (most relevant extensions in bold, not fully implemented yet are starred):

3.2.1 Generalization of the model to n-choice coordination problems. The model has been generalized to n-choice coordination problems. The choices are proposed as abstract objects in a list.

3.2.2 Enlarged parameter bounds. The model bounded parameter values: $10 \leq T \leq 30$ to fit in the 35×35 matrix space representation, probabilistic parameters bounds were explained above and D had arbitrary, probably empirically, bounds. All the parameter bounds were enlarged to allow modelers to explore other situations (N, N_n, and N_c too). However, the run ANTs were still considering initial parameter bounds.

3.2.3 Generalization to 4N+k variables. Earnest's model focused on the study of 5 parameters *fixed* for all agents, it allows a great exploration of the parameter space with ANTs. A natural extension was to consider $4N$+k parameters: agents have their own respective parameters (either probabilistic or not) and would symbolize true distinct states. The rest of k parameters refer to the other fixed model parameters, such as N or the consensus method. The tweakable model allows a study for either fixed parameters, i.e., considering the 5 parameters for all agents, or as individual parameters making $4N$+k parameters.

3.2.4 ***New parameter modelings.*** The parameter T was changed to the number of neighbor constituencies. The values of T[13] are in [0, N-1]. It allows a constituency's neighborhood to be non-uniform. The parameter D was explained above to do a re-weighting of a constituency's preferences once influenced by its neighbor constituencies. Instead, a balanced distribution of weight penalty was applied to the other preferences (the sub-social choice's weight remains incremented between 1 and D%). This helps the preferences of a constituency to remain consistent over time.

3.2.5 ***New preferences modeling.*** Constituencies having different preferences modeling than negotiators have caused some difficulty in practice, all the more that weighted preferences have a limited representation of a group's preferences. Given, constituencies' preferences can be represented as a sample of aggregated same total orders (factorial number of choices), e.g., 15 people support the order A > B > C, 7 others B > A > C, and 10 others B > C > A, note that, in fact, preferences are here not necessarily exhaustive, we then set to 0 supporter the other orders[14].

3.2.6 Generalization of k-constituencies for a given negotiator..* We may want to consider several political parties of a state influencing a negotiator, or a negotiator for several smaller states. Though, the influences are not well established to fit all specific cases.

3.2.7 ***Different voting systems.*** In addition to the plurality vote, an iterative preferential/single transferability vote has been implemented, the approbation vote would have to be added later on. These new ones enable greater use of negotiators' *whole* preferences while getting closer to real-world negotiations.

[13]Or T_i with $0 <= i < N$ if we consider individual parameters.
[14]An example with an influence is given in the appendix.

3.2.8 ***Different influence methods among negotiators.*** As the initial model only focused on top-choice proposals during negotiations, a few other influence methods among negotiators have been implemented. A negotiator may copy the whole preferences of another negotiator, it may also adopt top-k[15] proposed choices with the rest of its choices shifted to the right.

Example 2. Consider a negotiator $\mathscr{N}_1$ proposing its top-2 choices to another negotiator $\mathscr{N}_2$, given the starter choices A, B, C and D, their respective preferences are $A > B > C > D$ and $B > D > A > C$. $\mathscr{N}_2$ copies the top-2 proposed choices $A > B$ in their order as its new top-2 choices and shifts to the right the rest of its available choices, $\mathscr{N}_2$ now prefers $A > B > D > C$.

3.2.9 ***Different consensuses.*** The initial consensus focused on the similarity of the social choice with the global choice (same vote but for all agents), i.e., to satisfy every agent with unanimity. More consensuses have been added. The constituencies could each ratify the decided social choice, meaning that it should match every constituency's top choice. They may also vote for another social choice only among themselves and compare it to the social choice made by negotiators as two separate votes. A revised Fallback Bargaining [3] for the two-level agents has also been implemented. The Fallback Bargaining is customizable to define a $(k1, k2)$-consensus, negotiators must find a consensus on a choice on their top-$k1$ alternatives in their preferences, and the decided choice must itself be ratified by constituencies in their top-$k2$ choices. Other consensuses can easily be added to the model.

Example 3. Let us consider Fallback Bargaining to find a (1, 2)-consensus. Consider two negotiators $\mathscr{N}_1$ and $\mathscr{N}_2$ with their respective orders $['B','A','C']$ and $['B','C','A']$ with their respective constituencies $\mathscr{C}_1$ and $\mathscr{C}_2$. The two negotiators found a consensus on their top choice B which respects the first level imposed by $k1 = 1$.

In the case of the weighted total order modeling for constituencies, a total order can be drawn from the weights with a descending sort compared to the reference order given by the starter choices, by default we consider $['A','B','C']$, e.g., weighted preferences $[A : 30.7, B : 19.3, C : 50]$ gives the total order $['C','A','B']$. In the case of order distribution modeling the we implemented the Borda's counting method [9] to generate a representative order of a constituency's diversity of opinions, e.g., for an order $['B','C','A']$ from the top choice to the bottom the scores are 2, 1 and 0 and we multiply the score of each choice by the number of supporters, i.e., the weight of this order. Once we have a list of scores with their associated choice, we can generate afterward the representative total order similarly to the weighted preferences.

Back to $\mathscr{C}_1$ and $\mathscr{C}_2$, we know they can be represented as total orders. Let us consider their respective total orders $['A','B','C']$ and $['C','B','A']$, they must find a consensus for the choice B at $k2 = 2$ levels. We see that for both orders their top-$k2$ choices are $A > B$ and $C > B$ respectively, in both of these sub-orders the choice B appears, hence they found a consensus at 2 levels. The agents found a (1, 2)-consensus.

3.2.10 Negotiators may take part in transnational communications with constituencies..* Negotiators may influence other negotiators' constituencies to emulate an international bargaining power, still with a certain probability.

3.2.11 Spontaneous changes of preferences by constituencies. With a low probability, constituencies may change spontaneously their preferences. It enables a new self-influence which may help avoid local solutions by adding some noise.

3.2.12 Non-complementary influence between constituency and negotiator. The initial model forces a complementary influence, i.e., either one is guaranteed to happen, between a negotiator and

[15]The k value can be set as a parameter.

its constituency with a complementary probability. This binding can be removed with a new probabilistic parameter N_c for a negotiator to influence its constituency has been added.

3.2.13 ***Agents' memory, strategic behavior.*** **.* Agents have no evolving behaviors, as they obey fixed programs. Genetic algorithms and more generally artificial intelligence can be added to their programs to allow the emergence of strategic behaviors (alliances, temporary votes) by negotiators.

3.2.14 ***Different influence methods among constituencies and negotiators.*** For the weighted total order modeling, once a negotiator is influenced by its constituency, it can adopt the preferences either by copying its constituency's vote (initial model) or generating a new total order based on the probability distribution of its constituency's preferences using a weighted sample without replacement [7].

The order distribution modeling implied redefining all influences. A negotiator could get influenced by its constituency by PPD [4], random adjacent pairing, or voting situation. On the other hand, a constituency could take the influence of its negotiator by spontaneous support, Mallows's distribution [5] [1], or total support[16]. Finally, the influence between constituencies could happen by either summing up and normalizing all their preferences' distribution or by weighted random matching with PPD or other pair influence techniques.

3.2.15 ***Climate consequences, the weight of choices.*** **.* Earnest's model focuses on a cooperation problem in which agents are not interested in the final decided social choice. As choices do not have an impact on the agents, the current model is only a negotiation model and does not imply the climate aspect. Weight of choices related to climate consequences have been implemented to create a climate negotiations dimension.

Some of these extensions have been proposed by Earnest[17] while some others were thought-out extensions to make generic and more realistic modelings.

3.3 Model study

A brief study of the tweakable with Earnest's parameters and other parameters was made. Truthfully, taking the number of parameters into account, a rigorous and deep study of the model would have needed more time and room in this report. The provided results only serve as first intuitions on the tweakable model compared to the initial model, as such, we restrain our study to fixed parameters.

Three main studies were conducted for this model: (i) the comparison of Earnest's model in NetLogo and its close copy on the tweakable model. (ii) The refining of Earnest's model, and a comparison with the more realistic parameter modeling. (iii) Finally, the new preferences modeling with the new associated methods are tested and compared in the model to see if this would reveal a relevant approach for "better" negotiations.

We remain in the study of 3-cooperation problems. We emphasize that we use the initial cooperation method $global\_vote$ as explained in *3.1.1*, an important note is that it doesn't seem to make great change compared to the $global\_preference$ method in overall generated discord. However, using $global\_preference$ often generates more discord (about 10 on average) when considering effective negotiator influence, and may even more (about 20 on average) when we consider, in addition, the sub-vote by vote among constituencies.

Although, some yielded parameter sets can show "exceptions" in the parameter values due to the stochastic nature of both the model and study, the following results were generated and selected through a number of run ANTs as the best representative solutions found. However, if one would

[16] The details and some examples are available in the appendix.

[17] Extensions 10 and 13.

like to have a more precise study, the ANTs can be tweaked and run showing all the starter solutions and the final solutions either with mutation or not in the notebook[18].

3.3.1 Comparison on NetLogo and on the tweakable model. The results of conducted ANTs realized by Earnest were strongly affected by the stochastic nature of the model, ANTs do take some time to run on NetLogo, as such we hope to compare Earnest's article's results to the ones generated through the tweakable model, copying Earnest's model. Note that parameter values written on a line as, for example, $a_1;b_1$, a_2, $a_3;b_3$, a_4, … describe two respective solutions a and b which had relatively close generated mean discord with different parameters 1 and 3 values, the rest of the values are essentially the same if not exactly the same, i.e., $a_2 \approx b_2$, $a_4 \approx b_4$, and so on. A hypothesis is that those parameters' effects are often dominated, i.e., overlooked, by other parameters' effects. This reveals some non-linear interactions in specific configurations.

Table 2. Parameter sets generated by ANTs on the tweakable model compared to Earnest's parameters

Fitness criterion	*Number of Negotiators*	*Extent of Transnational Networks*	*Swings*	*Sensitivity of Negotiators to Constituents*	*Sensitivity of Negotiators to Negotiators*	*Mean Discord*	*Standard Deviation*
			Tweakable model's parameters				
Max. disc.	25;20	24;28	1	0.647	0.227	70.1	27.8
Min. disc.	14	11	5	0.516	0.496	8.43	9.55
			Earnest's parameters				
Max. disc.	7	14	1	0.47	0.67	39.6	29.6
Min. disc.	38	26	3	0.54	0.39	52.9	35.4

Table 2 shows on two sub-tables the generated ANTs results on Earnest's copy above and from Earnest's article below. The mean values are obtained from 1000 simulations. A quick look at the results shows that both minimizing discord and maximizing discord objectives gave rather opposed results. The tweakable model states that discord occurs when (a) the number of negotiators is large; (b) the transnational network is dense; (c) the constituencies slowly reweight their preferences; (d) negotiators are relatively insensitive to other negotiators; and (e) the negotiators are relatively dependent to their constituents. It can be noted that the *Swings* parameter displayed rather consistency with Earnest's findings. The empirical studies have shown that both the parameters T and D seem to prevail over other parameters. The probabilistic parameters hold a wide range of values, but they remain quite similar. We should keep in mind that the tweakable model, as much as it tries to mimic, remains a close copy of Earnest's model since some details might have been omitted, let alone the details on NetLogo's core execution and the stochastic study.

3.3.2 Refining the model. In this study, we progressively refine Earnest's copy. We first toggle on negotiators' influence over their constituency, in addition, the constituencies may sub-vote by preferences by now.

Then, the constituencies may sub-vote according to their vote, it should be reminded that it is directly influenced by both the preferences and the negotiator's influence.

[18]Section VII.

We may, at last, consider the parameters modeling proposed with the tweakable model, all agents have their own probabilistic parameters, and the space is freed from the 35×35 matrix. Most importantly, the constituencies' preferences are generated with the preservation of previous preferences, the neighbor constituencies are selected randomly uniformly. Table 3 shows the results on the three described models.

Table 3. Parameter sets generated by ANTs after refining Earnest's model on the tweakable model

Fitness criterion	*Number of Negotiators*	*Extent of Transnational Networks*	*Swings*	*Sensitivity of Negotiators to Constituents*	*Sensitivity of Negotiators to Negotiators*	*Mean Discord*	*Standard Deviation*
			Effective negotiator's influence				
Max. disc.	26	28	1	0.769	0.433	52.7	23.73
Min. disc.	3	13	1;5	0.201	0.768	0.73	1.05
			Effective negotiator's influence + constituencies sub-vote by vote				
Max. disc.	9;31	28	1	0.798	0.26	65.8	19.25
Min. disc.	3	14;30	3;1	0.204	0.732	0.75	1.03
			Effective negotiator's influence + constituencies sub-vote by vote + new parameters modeling				
Max. disc.	12	0	5;4	0.772	0.604;0.233	33.2	12.17;17.67
Min. disc.	3	2;0	4;1	0.23	0.742	0.77	1.11

The results tend to follow the same pattern on the discord minimization, whereas discord maximization gave rather sparse results highlighting specific cases leading to great discord. It is still open to interpretations, however, note that both the parameters T and D seem again to be key points in overall negotiations.

3.3.3 The new preferences. The new preferences mainly refer to the new modeling of constituencies' preferences, plus, the consensuses and influence methods associated. Unfortunately, there was not enough time to deeply study all the parameters. More than that, the tests for this modeling revealed to generate a lot of discord and to find consensuses in way more iterations, giving an average beyond the max 200 set. A few short conducted tests showed that 100 runs of the model with those parameters may take up to over 40 minutes, thus for 40 solutions over 40 generations would need over 1000 hours on a regular computer. Given that, in order to get a glimpse of intuition, we conducted the tests without using a mean fitness, i.e., the solutions were tested only in one simulation in ANTs. Subsequently, they would be selected after being tested on 100 simulations.

The results displayed in Table 4 are generated through 40 solutions under 40 generations, they were run using custom parameters[19] to respect as much as possible agents' preferences, notably, Fallback Bargaining (1,2), preferential iterative vote and new modelings methods.

The choice of these parameters is arbitrary, further studies can be made in the notebook. The interpretations of these results are open to interpretation, we can note that it generates overall more discord and hardly find a consensus on average in the given time as displayed in table 5.

[19] Global preference, preferential iterative, Fallback Bargaining (1,2), top-1 shift, new parameters modeling, order distribution on all total orders, random matching, Mallows's support, PPD, negotiators then constituencies, sub-vote by vote, no complementary influence, effective negotiator influence.

Table 4. Parameter sets generated by ANTs with the new preferences modeling on the tweakable model

Fitness criterion	*Number of Negotiators*	*Extent of Transnational Networks*	*Swings*	*Sensitivity of Negotiators to Constituents*	*Sensitivity of Negotiators to Negotiators*	*Sensitivity of Constituents to Negotiators*	*Mean Discord*	*Standard Deviation*
Max. disc.	12;31	4;7	1;2	0.238	0.436;0.767	0.614;0.702	97.5	12.4
Min. disc.	3	2;1	2;1	0.698	0.715;0.203	0.622	2.74	2.86

3.3.4 A little perspective. To get a better grasp on these results, we computed the mean discords and mean numbers of iterations before consensus (out of 200 max) on random fixed parameters for all the previous configurations displayed in Table 5[20]. The mean values are obtained from 1000 simulations compared to the 100 ones used in the run ANTs.

Table 5. Mean discords and iterations generated over 1000 simulations with different model configurations

Fitness criterion	*Earnest's Copy*	*Vote + Earnest's Copy*	*Neg.inf. + Earnest's Copy*	*Neg. inf. + Vote + Earnest's Copy*	*Neg. inf. + Vote + New param.*	*Neg. inf. + No Comp. + New pref.*	*Neg. inf. + New pref.*	*Neg. inf. + No Comp. + New pref. v.2*
Mean discord	45.7	64.7	25.9	35.4	4.8	71.2	81.1	17.6
Mean iteration	148	188	136	156	50	186	193	111
Variance	941	436.1	493.3	539.4	45	682.5	736.6	548.9
St. Dev.	30.68	20.88	22.21	23.22	6.71	26.12	27.14	23.43

It is important to remember that solutions are often not *linearly* interpretable, since many non-linear interactions can happen between parameters, two solutions can greatly generate similar discord while being *relatively* opposed. It can be linked to real-world negotiations when opposed extreme situations could finally have some similar results, e.g., 3 persons could never be finding an agreement the same way 100 persons could never too.

Let us briefly focus on the new parameters modeling in the 5^{th} column. This modeling differs from Earnest's copy and showed to reach earlier consensuses, certainly, this seems to radically ease negotiations, however, we may also note that the discord *percentage* metric shows about 9% of relative discord compared to the 19% of Earnest's copy with negotiator influence in the 3^{rd} column. We may use a harder-to-reach consensus to see how overall discord evolves. At last, a quick look at the models in the last three columns, which used the custom parameters from Table 4 with the new parameters modeling, shows that the first two either with complementary influence or without generate great overall discord. Empirically, they would find a consensus way beyond 200

[20]Neg. inf. : Effective Negotiator's Influence; Vote : Constituencies sub-vote by vote; New param. : New parameters modeling; No Comp. : Complementary influence off; New pref. : New preferences modeling; St. Dev. : Standard Deviation.

iterations and sometimes past 1000 or never. The last column's model is similar to the 6^{th}'s, but it uses spontaneous support and voting situation instead of, respectively, Mallows's support and PPD methods. The simulation showed less overall discord with earlier consensuses, these may be consequences of simpler modelings of influences, reminding us to cautiously handle the modeling of interactions especially when interpreting results.

4 CONCLUSION

In this work, Earnest's model has been studied both in Earnest's article and in its NetLogo implementation from theoretical and practical points of view. While the model gave some insights on simply modeled two-level international negotiations, it was revealed to have some limitations in its parameter space study and in some of its core modelings. In order to refine negotiations and studies, the tweakable model has been developed in Python to give access to all parameter modelings and details the model has to offer. It can be extended as much as the modeler wants and runs globally faster with Python than in NetLogo, even though Earnest's code can surely be optimized [12]. The model can also produce animated simulations using Matplotlib, either as a video format or directly watchable on the notebook.

The proposed tweakable model is customizable to define a close copy of Earnest's model or go beyond, it was tested using more stable ANTs and shown different results from Earnest's article. Considering the initial core modeling of the model, more negotiators which are less inclined for negotiations with strong influence (i.e., negotiators are relatively dependent of constituent pressure) from their constituencies and relatively dense but slowly influencing transnational connections may generate a lot of discord. It is opposed to Earnest's results in all points except from the slowly influencing transnational network for constituencies, the opposition is quite similar regarding the study on discord minimization. A few tweaks and more realistic options were made to make an upgrade to Earnest's model, which also gave similar opposite interpretations after its study. The differences can be explained in initial less stable conducted tests, all the more that we have to remember that we work with strongly stochastic models, it is likely that Earnest's results are local minima in the solutions space. Nonetheless, it reminds us to cautiously handle the interpretation of the results, noting that in addition, the conducted tests to obtain these results are itself stochastic, the presented results here may themselves be local minima. Stonedahl suggested improving the search of solutions by making it more robust against high variations of parameters [13]. The exploration of enormous parameter spaces is a problem that has many approaches, many of them were not yet tested.

As for the model's deeper improvement, several already-implemented extensions were proposed, on the basis of Earnest's model. What was interesting is that some of Earnest's interpretations could be found back in the study of the tweakable model, with certain refinements on the modelings. However, we highlighted that negotiations' results strongly rely on the notions of consensus and cooperation, since it may strongly vary in real-world negotiations too. Given the different results from the modelings, it is not easy to tell which configuration models the best international negotiations, knowing that all models can't perfectly replicate the complexity of these situations in any case. We can nonetheless see that the tendency in the results favors numerous negotiators, with relatively dense transnational connections and constituencies slow to change their preferences for greater discord. Although, some specific configurations from the parameter space exploration remind us of the underlying non-linear interactions and that exceptions may arise. The proposed extensions are still open to study and allow a stronger apprehension of negotiation interpretations, though, they remain simple approaches. Particular attention should be given to the modeling of preferences, numerous other modelings could be tested such as Euclidean preferences for constituencies or a cryptographic approach using $SHA256$ to generate random weights to model an order distribution. Further details are given in the appendix.

The climate dimension was not much worked on given the remaining time, effectively Earnest's model did not focus on climate negotiations but rather on a generic cooperation problem. Some thoughts about this part were to make use of existing Integrated Assessment Models (IAMs) [14] as black boxes giving feedback on the environmental impact. The simple idea was to preserve the current model's complexity only for generic negotiations and to include the climate part in the nature of the choices the agents made, which would create another layer of complexity. An independent plug-in black box, i.e., climate model would ideally fit with the tweakable model, supposing refined feedback on negotiations to output from the tweakable model.

At last, simulated ABMs do give an interesting grasp on international negotiations, but their study and the quality of such stochastic models may be delicate. Other models and approaches than Putnam's two-level could complete certain interpretations of these cooperation problems as already made by many scholars, the game theory may also be involved. This first study served as a starter point for various lines of thinking and hopes for future improvements.

REFERENCES

[1] Manel Ayadi. 2019. *Winner Determination under Common Voting Rules using Truncated Ballots.* Ph. D. Dissertation.

[2] Steven C. Bankes. 2002. Agent-based modeling: A revolution? *Proceedings of the National Academy of Sciences* 99, suppl_3 (2002), 7199–7200. https://doi.org/10.1073/pnas.072081299 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.072081299

[3] Steven J. Brams and D. Marc Kilgour. 2001. Fallback Bargaining. *Group Decision and Negotiation* 10, 4 (2001), 287–316. https://doi.org/10.1023/A:1011252808608

[4] Markus Brill, Edith Elkind, Ulle Endriss, and Umberto Grandi. 2016. Pairwise Diffusion of Preference Rankings in Social Networks. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-2016).*

[5] Fabien Collas and Ekhine Irurozki. 2021. Concentric mixtures of Mallows models for top-k rankings: sampling and identifiability. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 2079–2088. https://proceedings.mlr.press/v139/collas21a.html

[6] David C. Earnest. 2008. Coordination in Large Numbers: An Agent-Based Model of International Negotiations. *International Studies Quarterly* 52, 2 (2008), 363–382. http://www.jstor.org/stable/29734239

[7] Pavlos Efraimidis. 2010. Weighted Random Sampling over Data Streams. (12 2010). https://doi.org/10.1007/978-3-319-24024-4_12

[8] Euel Elliott and Lowell Kiel. 2004. Agent-based modeling in the social and behavioral sciences. *Nonlinear dynamics, psychology, and life sciences* 8 (05 2004), 121–30.

[9] Peter Emerson. 2013. The Original Borda Count and Partial Voting. *Social Choice and Welfare* 40 (02 2013). https://doi.org/10.1007/s00355-011-0603-9

[10] John H. Miller. 1998. Active Nonlinear Tests (ANTs) of Complex Simulation Models. *Management Science* 44, 6 (June 1998), 820–830. https://doi.org/10.1287/mnsc.44.6.820

[11] Robert D. Putnam. 1988. Diplomacy and domestic politics: the logic of two-level games. *International Organization* 42 (1988), 427 – 460.

[12] Steven Railsback, Daniel Ayllón, Uta Berger, Volker Grimm, Steven Lytinen, Colin Sheppard, and Jan Thiele. 2017. Improving Execution Speed of Models Implemented in NetLogo. *Journal of Artificial Societies and Social Simulation, The* 20 (01 2017), 3. https://doi.org/10.18564/jasss.3282

[13] Forrest Stonedahl and Uri Wilensky. 2010. Finding Forms of Flocking: Evolutionary Search in ABM Parameter-Spaces. In *mAbs.*

[14] John Weyant. 2017. Some Contributions of Integrated Assessment Models of Global Climate Change. *Review of Environmental Economics and Policy* 11, 1 (2017), 115–137. https://doi.org/10.1093/reep/rew018 arXiv:https://doi.org/10.1093/reep/rew018

A THE TWEAKABLE MODEL

The model can be tested in the Python's notebook in the test section (section V).

The developed model in Python allows an instance of a customizable model, one could pass in a set of parameters as the model parameters. We make use of a Python's dictionary to represent the parameters. The parameters are of different types, some are toggles and some are methods represented as strings.

The initial model of Earnest can be tested in this version with the Earnest_params parameters set already defined in the code. The following describes each parameter, with eventually some custom notes. They should be completed with the ones given in the code's documentation.

A.1 Agents and parameters

The number of negotiators can be set with the N parameter since by default each negotiator has one and only one constituency, it populates the world with $2N$ agents. There must be at least 2 negotiators. Theoretically, the parameter has no upper bound, except if we toggle on the *Earnest_parameters_modeling* parameter. The parameters can be set as a list with the $fixed\_params$ parameter. We emphasize that setting the parameters as fixed values fixes these probabilistic parameters for all the agents, as in Earnest's model. It is particularly useful when we consider a later study with ANTs to explore only these parameters (with N, T, and D), otherwise, we would explore too many parameters. If we want to randomize parameters for *each* agent, then it must be passed in a list of parameters with the ones to be randomly generated set to $None$, e.g., [14, None, None, None, 4] for $N = 14$ and $D = 4$ with complementary influence we don't need to set N_c. If we want to randomize parameters for fixed for *all* agents, then set the parameter to $None$.

The probabilistic parameters are the base probabilistic parameters, their bounds have also been extended to min 0 and max 1. The parameter N_c has been renamed to C_n to stand for Constituency influences Negotiator, N_c is still available but is the probabilistic parameter for the influence of a negotiator over its constituency. Either from considering minimal values or maximum values for N_c or C_n, this may change the model from two-level to almost one-level agent negotiations.

Neighbor influences always happen, we could introduce a new probabilistic parameter for this influence too. As for the new modeling, most of the proposed techniques make an influence faster if not instantaneous to the influenced constituency compared to the 1-D% increment.

A.2 Choices

The starter choices are defined as abstract objects to choose from a list, the default values are abstract characters such as letters, e.g., ['A', 'B', 'C']. It is defined in the *choices* parameter. The code mostly deals with those abstract choices based on their index in the given starter list. The use of a list allows considering n-coordination problems easily.

A.3 Cooperation method

Numerous cooperation methods are defined, it can be chosen in the $cooperation\_method$ parameter. Referring to the defined cooperation above, we can change the notion of a world iteration being in a cooperation state thanks to this parameter. The intuitive idea behind two-leveled negotiations would be to consider the ratification of each constituency of the negotiators' decided social choice, however, the initial coded cooperation method was the global choice. The two are available as parameter values.

To change this parameter means to change the discord evaluation, and as such, to change the discord metric. The future run simulations will have their tendencies and global discord mainly changed, the study can greatly be impacted notably with ANTs. If the cooperation method is set to at_least_half of the constituencies should accept the social choice, then it would generate more discord globally than the $global\_preference$ used by Earnest's model. Hence, they would have different average discords over different simulations even with the same maximized discord parameters. This should be kept in mind.

A.4 Influence method among negotiators

We can tell how negotiators will negotiate between themselves with the parameter $negotiator\_influence\_method$. It can be noted, as late as it is, that the influences among negotiators here are referred to as *negotiations*. However, a negotiation should focus on a bargaining system with eventually some compromises between two negotiators, at last, they may conclude on an agreement to both adopt a new position. In this model, the *negotiation* was modeled with the probabilistic parameter N_n which led to an adoption of choice(s) in case of success. We refer to those *negotiations* as influences among negotiators more than actual negotiations. Note that Fallback Bargaining can be considered as negotiations, but at the consensus level.

Also, there can be multiple influences for one negotiator, the last one currently prevails since an implicit order of execution of agents still remains. But, we may deal with them all at the same time if necessary, taking the choice which may be the best or not for the negotiator's initial preferences, it is left as an extension.

A.5 The vote system

The vote system used by negotiators can be set using the $vote\_system$ parameter.

The initial vote system considered was the plurality one. Although this is the first intuitive vote system used, several votes are made using different vote systems in states. The $preferential\_iterative$ defined in the tweakable model actually refers to the $single\_transferability\_vote$[21] which is widely adopted around the world, this allows to respect the preferences of the negotiators more to define a more refined social choice. If we consider a cooperation method involving the social choice, like $global\_preference$, it will likely impact the global discord of the simulations.

A.6 Consensus method

The consensus method allows defining when a simulation needs to stop if it has not reached yet the maximum number of iterations, it can be set with the *consensus_method* parameter. By default, it respects Earnest's consensus, defined as unanimity.

A few other consensuses have been added, one could easily add his own consensus, though. It is only natural that this parameter greatly impacts the simulations. Harder-to-reach consensuses trying to satisfy both unanimity and preferences can be considered, notably when taking into account the negotiator's preferences' order, which may be compared between them with an order comparison metric. Minimization of discordance between negotiators' preferences is part of another problem category, we could simply consider a minimization threshold percentage to reach to define another consensus. We partially explored some extensions on this, considering Kendall's-τ or Spearman's rank coefficient we could define the dissimilarity between two given orders, we also considered the least squares technique considering indices of choices. Such metric enables deeper consensuses by emphasizing the need of cooperation in global preference as a supplementary condition, the metric could also be used in cooperation methods with a pre-set threshold. Using the least squares method with 80% of similarity between negotiators' preferences it was seen on average to be indeed harder to reach consensus, also the global agents' preferences were more homogeneous at the end.

Earnest defined the "win set" as the *set of all possible agreements among negotiators that the constituencies would ratify*, but in the implementation, he only made use of the global vote. Fallback

[21] I only realized late that I should have named it this way.

Bargaining is particularly useful here, as we think it perfectly matches the definition of the win set defined here.

A.7 Constituencies' preferences modeling

The constituencies preferences modeling can be set with the *constituencies_preferences* parameter.

The weighted total order modeling is the initial modeling, it represents one approach among all the modelings of a diversity of opinions. The big question with the constituencies' preferences modeling is to know how well our modeling will plug in the influences with other agents. The weighted total order allowed, with the modeling of the parameter D, to model an interesting constituency-neighbors influence, while the negotiator-constituency influences only cared about the vote.

Approaching this modeling using a distribution over (total) orders seemed intuitively very rich to recreate the diversity of opinions inside a population. However, this later followed complicated interactions between agents, new influences can still be proposed.

It is only natural that this parameter is a key point in the model, this can drastically change the behavior of the simulations.

A.8 Influences among negotiators and constituencies

The influences among negotiators and constituencies mainly refer to the interactions between them. For the constituency's adoption of neighbor's preferences, set the parameter *constituency_neighbors_adoption*, similarly *constituency_negotiator_adoption* for constituency's adoption of its negotiator's influence. Finally, the *negotiator_constituency_adoption* parameter defines the negotiator's adoption of its constituency's preferences. The default values are all set according to Earnest's base model. Particular attention to *swap_top_choice* for the negotiator's adoption of preferences, it is the modeling of adoption as a top choice while keeping the total order modeling. In Earnest's code base, only a vote attribute representing a choice was coded.

For the order distribution modeling, details are given in the extensions approach. It should be noted that this was one of the biggest works in the model's study and extension.

A.9 Simulation dynamic

This parameter can be set with the *dynamic_order* parameter in the model, the base Earnest's order (negotiators then constituencies), and a total random dynamic are already coded. The total random parameter has not been tested much, since Earnest's order seems to be rather appropriate for real-world negotiations. Further tests may be useful.

A.10 Complementary influence

The complementary influence is set with the *complementary_influence* parameter, it concerns the negotiator-constituency influences. In the initial simulation dynamic, step (4) where constituencies may influence their negotiator, in case of non-complementary influence, is swapped with step (5). This is an arbitrary choice, it can be changed in the code.

The complementary influence set to true showed greater overall discord over simulations, it can naturally be interpreted as forcing an interaction between the negotiator and its constituency, even though they may have opposite preferences. The simulations were hardly reaching a consensus, whereas, with this parameter set to false, most of the simulations would terminate early.

A.11 Negotiator's influence

The negotiator's influence over its constituency can be set with the parameter *negotiator_constituency_influence* we can choose whether we want to consider this influence or not. With the complementary influence set to true, the automatic probabilistic parameter considered is 1 - C_n. Otherwise, this parameter must be set to false to define the probabilistic parameter N_c. Note that this last probability N_c concerns the negotiator's influence over its constituency and is an independent action from the constituency's influence over its negotiator.

A.12 Influences between constituencies

This can be set using the parameter *constituency_neighbors_adoption*. It is already noted for many parameters in the code documentation, but especially for this one, some parameter values can only be used with the appropriate preferences modeling.

A.13 Parameters modeling

The parameters modeling refers to the initial Earnest's modeling of the parameters T and D. This can be used with the *Earnest_parameter_modeling* parameter in the model. Note that if set to true, this will force a 3-cooperation problem, so with 3 choices, and will restrain the parameter bounds of T and D as in Earnest's model.

B EXTENSIONS APPROACH

B.1 Generalization to n constituencies

This extension was implemented quickly and would need to have deeper thoughts on its implementation. As a matter of fact, the negotiators have only been given lists of constituencies, with only one constituency in the studied model. The negotiator-constituency influences are done for each constituency but as successive influences only. Each constituency can influence the negotiator's preferences successively which favors the last influencing constituency. The negotiator influences all of its constituencies in the same way as influencing disjoint constituencies. As such, there is no real cohesion between constituencies. For some instances this may work, e.g., different smaller states are represented by one international negotiator. However, to model different political parties sharing the state's case under the same domestic constraints, cohesion may somewhat be present. The current model lacks this part.

B.2 Parameters modeling

Earnest's parameters modeling is fully replicated in this model, however, we can focus on the modeling of the parameter T and, indirectly, the parameter D. The parameter T initially represents the radius of a constituency within which neighbor constituencies are taken. This modeling emphasizes the uniformity of the neighborhood of a constituency, as they are taken according to space proximity (in the context of a 35x35 matrix, constituencies are placed on a circle of radius 15). The changes made in a constituency's preferences once it has been influenced by its neighbors are done with the increment of 1-D%, those changes were made considering only 3 choices: the proposed choice gets its weight incremented and the arbitrarily selected second choice has its weight randomly generated among the available remaining weight, the third one gets the remaining. Such modeling creates **noise** in the model, as it mostly doesn't preserve previous preferences. With both these modelings, it has been shown in results that discord tends to be higher. The proposed tweakable model offers the possibility to change these modelings (parameter *Earnest_parameters_modeling*): the parameter T becomes a random selection of T neighbor constituencies among all constituencies and, indirectly, the parameter D allows making use of balanced distribution of weight penalty among other choices than the proposed one with respect to previous preferences. These modelings have been shown to simulate less discord on average than Earnest's modeling, moreover, such a choice made a rather different model from the initial one, as presented above.

B.3 The new preferences modeling

B.3.1 Distribution of weights over total orders. Constituencies had weighted preferences orders which were only partially exploited by the initial model's influences, moreover, the influences were defined in specific ways because of the negotiator's preferences in different modeling. The intuitive idea is to redefine constituencies' preferences into total orders in the same way as the negotiator's. On top of that, a constituency should represent diverse opinions as a whole unit, so it was only natural to extend the preferences to a distribution of weights over total orders. The weights represent a certain number/percentage of the population adopting a total order. After that, all influences have to be redefined, starting with the influence between a negotiator and its constituency: the total orders of the constituency would represent voting situations for the negotiator to adopt in case of effective influence, with favor to the heaviest (i.e., most supported) orders. The influence among constituencies had a few propositions.

We consider a distribution over total orders, to emulate the diversity of opinions across a population. However, the distribution would have to be defined upon n! orders with n the number of choices. A first approach is to consider small n values (<= 5) to be able to represent all orders, though it was not possible to store in memory those elements for each constituency for larger n values (making a space complexity of O(n*n!)). Another approach is to only consider a sample for each distribution: either by considering top k choices, restraining to k! partial-order elements, or by picking random total orders among the distribution. This technique allows representing the population with more reasonable time and space complexities, nonetheless, it remains a partial viewpoint over the whole constituency's preferences. In another hand, one could model a constituency's preferences as a majority graph or a weighted tournament tree over choices, which has been proven to be a data structure associated with a unique total order. Such data structure has bigger time complexity than the previous ones and O(n) space complexity, it also lacks representation of a population in the way of diverse opinions but provides a solid base for influences. As the idea of not storing n! total orders was primordial we still needed to put weights over total orders, as such those numbers needed to be defined somewhere, which would lead to an n! memory storage. With that being said, a last approach would be to consider a function f such that for a given total order O, $f\colon O \mapsto w$, $w \in N$ an associated weight. This function should have optimal computation costs for both space and time complexities to get the weight of a given total order. We may also consider a surjective function g such that $g\colon i \mapsto O$ with $i \in N$ and O a unique total order to represent n! total orders and their weights with these two functions. The latter function can be made with different approaches and f can be modeled from a cryptography approach, e.g., using SHA256 we can generate a weight giving an order as concatenated choices in a string plus a given seed and the constituency's ID as parameters. Given the remaining time, the used approach was a k! memory storage for weights (k <= n!) with the use of a function g in O(n) time complexity. The function g must associate a unique total order to a given integer i < n!. The used function uses a simple algorithm in O(n) time complexity: Given i and n

(1) Set p to n
(2) Compute q and r such that i = q*(p-1)! + r (Euclidean division of i by (p-1)!)
(3) Take the $q + 1^{th}$ *available* choice in the currently generated total order
(4) If p > 1 decrement p by 1 and restart the algorithm from step 2) with i set to r

Steps 1) and 2) are in fact simply a for loop with a change of p value, initially it was designed to be recursive, but iterative would b better in Python. The proof of termination is trivial, and the correctness has been proven but is not given here as it is not the purpose of the report. The motive is to associate a unique total order to a given i regardless of the look of the order, i.e., as long as it is a unique order associated to i. The idea of the algorithm is to try to decompose i into a product such that i = $a_1 a_2 a_3 ... a_n$ with a_1 < n, a_2 <= n - 1, a_3 <= n - 2, ... each a_i gives information on which choice to take at a given step of the total order generation. a_1 gives the first choice to take, a_2 the second choice to take considering the first choice has already been taken and is no longer available for choosing, and so on. The factorial decomposition of i gives the direction on which choices should be taken to build the only possible path (considering a complete graph of choices) giving a total order.

Considering a reasonable number of choices, the complexity in both space and time for this algorithm is quite efficient. Nonetheless, there remains $k!$ (k <= n) stored weights in an array, this extension is left as a possible path to explore. A thought to deepen would be to consider the said g function above, which may use a cryptographic approach, but add to it a richer process for weight generation: a parametric weight. We may pass in as a second parameter an order (the negotiator's order), such that the returned weight is bigger if it is closer to the order and lesser if not using an order comparison metric. By default, we may allow no given order for initial weights, afterward it may be useful as a negotiator could influence its constituency by changing its weights since orders are constant with this method.

The opposite function can easily be computed by running the above steps of the algorithm backwards. Given an order and the reference order (the initial order as the list of choices) the function returns the associated unique integer.

One possible extension would be to change the algorithm in the choice of generated orders. Considering a given metric to measure the similarity between orders, if we were to plot the weight distribution of a constituency's preferences the ideal result would be similar to successive normal/Gaussian functions. This gives the power to play with the indices the same as playing with closer/further orders in the sense of the similarity measure.

B.3.2 Influences. The new preferences modeling implied necessary changes in the influences between agents.

Negotiations between negotiators remain the same as they are not classified as influences, however, attention was particular to top-k-choice shift right influence or similar refined negotiations. The goal of this new modeling was to define a more refined model with respect to agents' preferences, we consider votes by preferences with negotiations respecting as much as possible initial preferences.

If a constituency influences its negotiator, we would like to consider all weights of the constituency's preferences (i.e., all orders). Since $nb\_choices!$ weights are stored this could be computationally intensive, some extensions about optimized computations are available though, but we will stick with a simple approach for now. We proposed the following influences:

(1) PPD [4]
(2) Random adjacent pairing
(3) Voting situation

In the PPD influence, we select each pair of adjacent alternatives in the negotiator's order (either starting from the least preferred to the top preferred) and swap or not the two alternatives according to the supported absolute majority in the constituency's preferences for this sub-order of preferences. The next pair is considered with a step of one alternative to the right (or to the left if we start from the bottom) until the algorithm reaches the end of the order, we call this a PPD pass.

Example 4. Consider a 3-cooperation problem and two negotiators $\mathscr{N}$ its constituency $\mathscr{C}$, $\mathscr{N}$'s preferences' order is $['C', 'B', 'A']$ and $\mathscr{C}$'s preferences' distribution is $[18, 12, 0, 30, 24, 16]$ with $\mathscr{C}$ influencing $\mathscr{N}$. Each index i of the weight list has its associated total order, given g, returned by $g(i)$ as an ordered list, e.g., $g(0)$ = $['A','B','C']$. Let us start from the top choices to the bottom choices of the negotiator. First, consider the pair (C, B) translating the sub-order $C > B$ from $\mathscr{N}$'s top-2 choices. $\mathscr{C}$'s preferences' distribution can be seen with the associated orders as $[18 \times g(0), 12 \times g(1), ..., 24 \times g(4), 16 \times g(5)]$, for example, 24 *people* support the order $g(4)$ = $['C','A','B']$ and, more particularly, the sub-order $C > B$ which refers to the pair (C, B). g's algorithm explores all the possible swaps from the bottom to the top starting with the reference order following the alphabet order, in particular, the 3! orders are generated, from 0 to 5, this way: $['A','B','C']$, $['A','C','B']$, $['B','A','C']$, $['B','C','A']$, $['C','A','B']$, $['C','B','A']$. We count all the supporters for this pair in $\mathscr{C}$'s distribution, giving a total of 0 + 12 + 0 + 0 + 24 + 16 = 52 supporters, which is an absolute majority out of 100 supporters: this means that $\mathscr{N}$ doesn't swap its pair of choices (C, B). The next pair to consider is (B, A) which has 46 supporters, it is not an absolute majority, then $\mathscr{N}$ swaps the two choices to change its preferences' order to $['C','A','B']$. A complete PPD pass was made since we reviewed all adjacent pairs.

The random adjacent pairing is the same principle as PPD's but with a random pair selection of adjacent alternatives. Last but not least, the voting situation influence was an intuitive starter influence for this new modeling. Considering an iteration in the simulation, the negotiator may simply adopt the whole order of the greatest weight in its constituency's preferences. A possible extension, which may need improvement, is for the negotiator to choose an order (i.e., an index) among the constituency's preferences based on this later as a weight distribution.

Let us now consider the case where a negotiator influences its constituency. This influence required careful modeling, since the negotiator's sole order needed to impact a whole diversity of opinions with different weights. Obviously, partial influences over all the orders in the constituency's preferences are also considered, though. Here are the starter points proposed:

(1) Reverse PPD
(2) Spontaneous support
(3) Mallows's support
(4) Total support

Reverse PPD influence, as the name states, is the proceeding of the PPD algorithm described above but applied to the constituency's orders each. For each order, for a given pair of this order, if it matches (in the sense of PPD) the pair of the negotiator's order we may increment the weight for this order, and so on. Although normalization is required to preserve the population number. This implementation is not yet coded, as it requires deeper work. Spontaneous support influence is a simple weight to design an increment for the negotiator's preferences in its constituency's. A random (customizable) number d is generated as the new number of supporters (i.e., weight) for the negotiator's order. In the constituency's preferences, some random orders are chosen, and if their weight is greater than d then the algorithm transfers d weight units to the order adopted by the negotiator. If there is not enough weight for a given order, we transfer all the available weight, subtract the quantity from d and we keep seeking other orders with enough weight until there is no remaining. The Mallows's support is an increment that focuses on a global change. We sum up the influenced constituency's preferences with a Mallows's distribution centered on the negotiator's order with the constituency's preferences, followed by normalization. Since we consider integer weights, the rest of the weight to add up to 100 is added to the negotiator's order in the resulted distribution.

Example 5. We consider the current implementation of the new preferences modeling with all total orders considered. Given a constituency $\mathscr{C}$ and a 3-cooperation problem, $\mathscr{C}$'s weight distribution is the list $P = [12, 0, 39, 27, 21, 1]$. Let $\mathscr{M}$[22] be a Mallows's distribution centered on the order $['A', 'C', 'B'] = g^{-1}(1)$ such that $\mathscr{M} = [19, 28, 12, 19, 10, 12]$. The sum of $\mathscr{C}$'s distribution with $\mathscr{M}$ is followed by normalization, i.e., integer division by 2 and increment: the sum gives $[31, 28, 51, 46, 31, 13]$ and the integer division gives $[15, 14, 25, 23, 15, 6]$ which gives a total of 98, the remaining 2 goes to $\mathscr{C}$'s initial most preferred order which was $g(2)$ since $max(P) = 39 = P[2]$. $\mathscr{C}$'s final distribution of preferences is $[15, 14, 27, 23, 15, 6]$.

We make use of Ahmed Boujaada, Fabien Collas and Ekhine Irurozki's python library on Mallows model[23]. The Mallows's distribution is generated through sampling, using the factorization of the Kendall's-τ distance, which can be adapted to top-k orders instead of the usual Repeated Insertion Model (RIM) [5] [1]. Finally, total support sets the maximum weight to the negotiator's order in the constituency's preferences, and sets the rest to 0.

At last, the influences between constituencies involve a radical change from the initial model. Given a set L of neighbor constituencies, preferences (i.e., 2D array) influencing a constituency's preferences C.

(1) Sum up and normalize
(2) Weighted random matching

Sum up and normalize influence is a rather instantaneous influence compared to an increment, the preferences of the neighbor constituencies and of the influenced constituency are summed up and normalized to give a new weight distribution. Since we consider integer weights, the normalization gives a sum of weights less than 100, we arbitrarily add the rest of the weight to add up to 100 to the constituency's initial most supported order. The weighted random matching influence focuses on partial sub-influences, a random constituency is picked among the neighbor constituencies (it could depend on the weight of each constituency but, it has not been implemented yet) and a random order O_1 is picked itself in this constituency's preferences based on this later's eight distribution. This order is then *matched* with another random order O_2 (picked uniformly) in the influenced constituency's preferences. O_1 influences O_2 in the influenced constituency's preferences, a random positive weight (customizable) is drawn which represents the number of supporters who will move out from O_2 to support O_1 instead. If there was not enough weight for O_2 then the influence is non-effective and ignored. This is an arbitrary choice.

B.4 Climate dimension

The climate consequences have not yet been added to the current model. Several approaches can be considered, we considered Integrated Assessment Models (IAMs) as they may plug well with the current model. IAMs allow us to get climate feedback with a computed function given data per iteration in our simulation, the purpose would be according to the feedback to adapt agents' behaviors to prefer more climate-responsible choices. Although this solution may be quite simple, as the climate feedback is simply seen as a black-box oracle for objective climate consequences the model may simply get biased towards climate-friendly decisions. Such a turn could make the model step into the game's theory as a probabilistic path to get to the "best" result. Further thoughts need to be given.

[22] $\mathscr{M}$ should be parametrized by either θ or ϕ, we use $\phi = 0.97$ in the implementation. A value of ϕ closer to 1 would generate a distribution less $strongly$ centered at the given order, whereas a value closer to 0 would concentrate the weights at the given order.
[23] https://github.com/ekhiru/top-k-mallows

B.5 Agents' strategic behavior

Negotiators are currently very simple-minded, following Earnest's work, Earnest starts his work on the premise that negotiators do not lie (as they would have no interest in it) during negotiations.

However, an interesting approach is to implement strategic behavior for agents to make choices, i.e., votes and influences such that it would benefit some iterations after. Some strategies have been shown considering different voting systems. More than that, a negotiator could let itself get influenced to change the choice of its constituency to decrease preference over another non-preferred choice. This takes into account agents' view of the world and knowledge for a given iteration.