An Agent-Based Model of Corruption: Micro Approach

The ODD Protocol of a Replication of Ross Hammond's Corruption Model, 2009

Valery Dzutsati

September 26, 2015

1 Overview

1.1 Purpose

This is the ODD Protocol (Railsback and Grimm, 2011) of an agent-based model of corruption. The model explores the conditions for the endogenous social transition of society from high-corruption state to low-corruption state (Hammond, 2009). The majority of researchers have focused on the effects on corruption, while this model specifically addresses the issue of dynamics of corruption. In particular, the model shows that what is often seen as a "stable corrupt system" may be quite unstable and transition to lower level of corruption under certain conditions. The change happens endogenously, without any change in the system and its structure.

1.2 Entities, state variables, and scales

Entities of the model include two types of individual agents, "citizens" and "bureaucrats", square grid, sized 33X33 that is not toroidal (does not wrap around), "social networks" that are represented by patches on which agents are stacked. Agents are stacked on patches only with their own "breed," i.e. "citizens" with "citizens" and "bureaucrats" with "bureaucrats." Each group of agents of similar type that are on a patch comprise a "social network." This network is utilized by every agent in it to gather information and make decisions. The model has a "jail" where some corrupt agents end up, based on the rules described below.

State variables include: strategy of the agent that can be either "corrupt" or "non-corrupt", inherent propensity for "honesty."

Each time period (tick) is made up of five steps: the interaction of "citizens" and "bureaucrats", gathering of information from the "social network", reporting of corrupt agents to the central authority, going to "jail" of corrupt agents, release of agents after the expiration of their "jail" term.[1]

List of parameters and switches:

[1] "jail" procedure adapted from Uri Wilensky's 2004 model Rebellion

- The switch *excludeself* regulates whether the agent should include itself, when making calculations about the number of corrupt agents on its patch. When the switch is "on," the agent does not count itself, when it is "off," the agent considers itself too.

- The switch *correction* regulates whether the agent becomes non-corrupt after landing in jail. When the switch is "on," the agent changes its status to non-corrupt after being put in jail, when the switch is "off", the agent does not change its corrupt practices.

- N refers to the number of previous rounds, agents "remember"

- m reflects the number of times an agent should be "turned it" before the authorities put it in "jail"

- k is the length of "jail" term in ticks.

- y is non-corrupt interaction's payoff

- x is corrupt interaction's payoff

- P regulates the number of "social networks," i.e. patches on which the agents are stacked.

- C regulates the number of agents within each "social network."

- *number-of-corrupt-agents* specifies the number of corrupt agents among both "citizens" and "bureaucrats."

1.3 Process Overview and Scheduling

Each time period consists of five stages:

- A "citizen" interacts with a random "bureaucrat"

- Both "citizen" and "bureaucrat" independently choose their expected payoff from corruption, based on the decision rule that is detailed below

- If either "citizen" or "bureaucrat" in each pair is corrupt and the other agent is not, the non-corrupt agent in the pair "reports" the corrupt agent to the central authority

- After the agent accumulates certain fixed number of "reports" of corruption, the agent goes to "jail." Going to "jail" means that the agent cannot receive payoffs and participate in the game for certain number of rounds (ticks)

- Having served the "prison term", the agent is "released" with the strategy "non-corrupt"

2 Design Concepts

2.1 Basic Principles

Unlike previous research that treated corruption as an exogenous factor, this model strives to show the dynamics of corruption in the society. The model attempts to show how and why transition from more corrupt state to less corrupt state might occur. The model assumes that the majority of agents are corrupt and payoffs for the corrupt interactions are higher than payoffs for the non-corrupt interactions. The agents are modeled with limited, bounded rationality whose decisions are based not on universal knowledge of the rules, but rather on the local knowledge of their local network.

2.2 Emergence

The primary result of the simulation is that after "arrests" of "corrupt" agents mount, the released agents change their strategy to "non-corrupt" that spreads the wave of honesty. There are increasingly more mismatches between the "corrupt" and "non-corrupt" agents, which result in ever greater numbers of "corrupt" agents going to "jail", which contributes to spreading "honesty." As a result of these trends an endogenous transition from corruption to honesty happens in the system. The key assumptions are that an agent after serving the "prison term" adopts "non-corrupt" strategy, that corrupt agents are accurately reported to the authorities each time when they interact with non-corrupt agents and that the central authorities act strictly according to the rules.

2.3 Adaptation

Agents decide whether to engage in corrupt activities or not, based on the decision rule that includes probability of encountering a corrupt agent, probability of chances of being caught for a corrupt action, the fixed payoff for the corrupt interaction and the fixed payoff for the non-corrupt interaction.

An agent of type i contemplates receiving the following payoff as the result of corrupt interaction:

$$(1 - B)[Ax_i + (1 - A)y] + B[y - ky] \tag{1}$$

A is the likelihood of encountering a corrupt agent. $A = n/N$, where n is the number of corrupt agents that the agent met in N previous interactions. B is the perceived chances of being caught for a corrupt action in this round. $B = m/M$, where m stands for the number of the agent's friends (agents of its own type that are on the same patch) in jail. M is the number of the agent's friends who were corrupt during the last round. x is the payoff of the corrupt agent and y is the payoff of the non-corrupt agent. Only if both interacting agents are corrupt the maximum payoff y can be achieved, as detailed in Table 1.

The agents own an inherent propensity for "honesty" that stands for a moral quality that adversely affects the payoff of the honest agent that indulges in corrupt activities. For an i of 1 that represents perfect honesty, this moral damage is high, while for an i of 0 that represents perfect corruption, the moral price does not exist. So, every agent

Table 1: 2x2 Payoff Matrix

	Corrupt	Non-Corrupt
Corrupt	x	y
Non-corrupt	y	y

perceives its own x differently, even though the x - the payoff of the corrupt agent - is of a fixed value in the game. Perfectly honest agent gains zero from a corrupt interaction, while only perfectly corrupt agent gains the full amount of x.

$$x_i = (1 - i)x \tag{2}$$

2.4 Objectives

The agents are rational, as they strive to maximize their utility – their payoff. The constraints include, the probability of being caught for corruption, the probability of meeting a non-corrupt agent, the inherent honesty that reduces the payoffs from the acts of corruption.

2.5 Learning

Agents change their strategy from “corrupt” to “non-corrupt”, following the “imprisonment” and subsequent ”release.” Agents learn from their “social network” to adjust their strategy.

2.6 Prediction

Agents have limited memory of how many corrupt agents they met in fixed number of last rounds. The observer has the memory of how many times an agent was “turned in”.

2.7 Sensing

The agent can only see the agents of its own type that are situated on the same patch as the agent itself. This ”social network” of an agent is used to gather dynamically the information about number of “friends” in “jail” and number of “corrupt” agents. The agent uses this information to estimate its highest possible payoff.

2.8 Interaction

Only pairs of different types of agents, “citizens” and “bureaucrats” can interact with each other. The interaction was modeled on the interaction of corrupt officials and citizens. Both are assumed to have equal inclination toward corrupt practices.

Table 2: Base Model

x base payoff to corruption	20
y payoff to honesty	1
i inherent propensity for honesty	uniformly distributed decimals [0, 1]
N size of memory	5 rounds
M size of social network	10 agents
k length of jail term	2 rounds

2.9 Stochasticity

Agents are assigned randomly (uniform) distribution of inherent propensity for honesty, decimal values from 0 to 1. This setup is designed to account for the inherent moral qualities of individuals that are relatively stable. Each round "citizens" interact with "bureaucrats" on random basis. This process shows the real world interactions of citizens with a variety of government offices.

2.10 Collectives

The collectives are fixed. There are two types of agents, "citizens" and "bureaucrats." Each of the agent types is further subdivided into smaller groups. Both divisions are fixed and do not change for the duration of the game, except for corrupt individuals who are reported to the central authority more than certain number of times. Those corrupt individuals are "jailed", but they are "released" into the same group where they were before after their jail time is over.

2.11 Observation

The transition from corrupt behavior to non-corrupt behavior is the primary observable that we should see. So, the output that is needed to see is whether the number of corrupt agents decreases over time.

3 Details

3.1 Initialization

Create empty torus of size LatticeSize x LatticeSize (default 33 x 33) [Von Neumann geometry]

3.2 Input Data

The environment is assumed to be constant, so there is no input data

3.3 Submodels

- *interact.* A "citizen" interacts with a random "bureaucrat", both agents should be out of "jail" at the time of interaction. If the pair of agents has divergent strategies, pertaining to corruption, i.e. one is corrupt and the other one is not, the non-corrupt agent reports the corrupt agent to the central authority. If two corrupt agents interact they both receive highest payoff x, otherwise each agent receives payoff y.

- *decide.* Both the "citizen" and the "bureaucrat" that are out of "jail" independently calculate their expected payoff from corruption, based on the decision rule. If the output of $(1-B)[Ax_i + (1-A)y] + B[y - ky]$ is greater than y (the payoff for non-corrupt practices), each agent rationally makes the decision to become "corrupt" (changes its color from white to red)

- *enforce.* After an agent accumulates certain fixed number of "reports" of corruption, the agent is sent to "jail. The affected agent sets its color to white, its active status to false, its jail term to the specified k, receives a payoff in the amount of y and resets its counter of reports to 0. Being in "jail" means that the agent cannot receive any payoffs and participate in the game for certain number of rounds, governed by k.

- *release.* Having served the "prison term", the agent is "released".

- *reporter "agent-payoff"* reports the weighted payoff of an agent $x_i = (1 - i)x$

- *reporter "encounter-corrupt-agent"* calculates the probability of the agent of encountering a corrupt agent, given the count of corrupt agents it encountered in N previous rounds

- *reporter "perceive-chances-of-jail"* calculates the "subjective" probability of being caught for corrupt practices. This indicator is the number of corrupt agents in jail form the patch of the agent divided by the total number of corrupt agents from the same patch

References

Hammond, Ross A. 2009. "Endogenous transition dynamics in corruption: An agent-based computer model.".

Railsback, Steven F and Volker Grimm. 2011. *Agent-based and individual-based modeling: a practical introduction.* Princeton University Press.